

# Enhancing Science Laboratories with Technology

*Exploring the learning advantages of a tablet application used  
to guide the laboratory experience*

By

*Ennio Claretti*

*Joseph Everett*

*Ben Leone*

*Nikolaos Matthiopoulos*

*Christopher O'Connor*

*Max Saccoccio*

*Michael Yeroshalmi*

An Interactive Qualifying Project:  
Submitted to the Faculty of the  
WORCESTER POLYTECHNIC INSTITUTE  
in partial fulfillment of the requirements for the  
Degree of Bachelor of Science

## ABSTRACT

Technology when used correctly can be a very powerful tool for education. Its multimedia features combined with a good user interface can increase the quality of learning, engage students and provide new mediums to deliver information. We created a multimedia capable user friendly iPad application that we tested in a real learning environment and determined that, although improvements can be made, our beta software showed that using technology like a tablet can make a positive impact in the learning environment.

## Table of Contents

1	Introduction .....	4
	Problem Description .....	4
	Purpose.....	4
2	Background.....	5
	Benefits of Technology in Education .....	5
	Adaptation of Technology in Education .....	5
	Current Products .....	6
	User Demographic .....	7
3	Methodology.....	7
	IRB Application .....	7
	Curriculum Feasibility Study.....	7
	Development of the application .....	8
	Extensible Markup Language .....	8
	Data Collection .....	9
	Information Display.....	9
	Mobile Lab Client.....	10
	Faculty Post-App Builder Survey .....	10
4	Results and Discussion .....	11
	Survey Results .....	11
	Faculty Feasibility Interviews.....	11
	Results of Student Feasibility Survey .....	11
	Faculty Post-App Builder Survey .....	16
	Results From Student Post Lab Survey .....	18
	Mobile Lab Client Demonstration .....	19
	Lab Builder Application Demonstration.....	22
	Intuitiveness .....	29
	Performance .....	29
	Future Projections .....	30
5	Conclusion .....	30
6	Authorship .....	32
7	Bibliography.....	34
8	Appendices .....	35

Appendix A: IRB Exemption Approval .....	35
Appendix B: Faculty and Student Feasibility Surveys .....	36
Appendix D: Faculty Post-App Interview Survey .....	39
Appendix E: Faculty and Student Feasibility Surveys Results.....	40
Appendix F: Student Post-App Survey Results .....	46
Appendix G: Faculty Post-App Interview Results.....	53
Appendix H: XML Standard.....	57
Appendix I: App Source Code.....	59
Appendix J: App-Builder Source Code .....	86
Appendix K: Test App XML code.....	163

## Table of Figures

Figure 1: Student Device Predisposition.....	12
Figure 2:Histogram of responses plotting perceived need of a new application .....	13
Figure 3:Students' preferred method of data collection .....	14
Figure 4: Word cloud of responses to Question 13.....	15
Figure 5: Professors' Willingness to Use Technology in Labs .....	16
Figure 6: Faculty Oppinion on Previewing Labs.....	17
Figure 7: Percieved Benefit of the App .....	18
Figure 8: Student Oppinion on Laptops vs. iPad .....	19
Figure 9: Screen Shot Of Text Section .....	19
Figure 10: Screen Shot of Images .....	20
Figure 11: Screen Shot of Table View.....	21
Figure 12: Screen Shot of Text Input.....	22
Figure 13: Initial lab builder view .....	23
Figure 14: The New/Edit Project view .....	24
Figure 15: Edit tab view.....	25
Figure 16: Text edit view .....	26
Figure 17: Table Edit view .....	26
Figure 18: Question Edit view .....	27
Figure 19: Image Edit view.....	27
Figure 20: The Video Edit view.....	28
Figure 21: Project Previewer view.....	28
Figure 22: Save Project view .....	29

## Table of Tables

Table 1: Project Work Distribution.....	32
Table 2: Report authorship.....	33

# 1 Introduction

When technology is used in the classroom, for presenting new information, collecting homework assignments or administering tests, there is often the fear that the technology will take away from the educational objective. Often teachers find themselves spending more effort overcoming the learning curve or problems with a new educational technology than using it (Clyde, 2004). Technology is a powerful force that is revolutionizing modern teaching methods: the ability to bring any piece of information, pictures, videos, text, etcetera, into the classroom within seconds could be that additional push that teachers need to instill functional knowledge in their students (Bransford, 1999).

## Problem Description

Despite all of the potential advantages of technology in a learning environment, there are three criteria that almost exclusively dictate the success and widespread usage of technological teaching methods:

1. The technology needs to *work* and it must be robust. There is no way any curriculum can be improved with technology if the technology is not trusted by those implementing it to some extent.
2. The technology must provide some potential learning advantage to the student and must be valuable to the curriculum
3. The training time for the technology must not outweigh the learning experience offered by it. (Clyde, 2004)

The relation between these criteria and the way technology is used by the intended user group is often referred to as the Technology Acceptance Model (Davis, 1989). To develop a technology that is used willingly and meets the needs of the end-users, it is necessary to address the aforementioned criteria.

## Purpose

The purpose of this project is to develop an application for mobile devices that can be used by Worcester Polytechnic Institute (WPI) faculty to improve the quality of the laboratory learning experience in introductory WPI biology labs. The final application serves as a test of a functional model that demonstrates the feasibility and benefits of mobile device technology in educational environments.

It is believed that, by moving regular data collection, sharing and interpretation processes of the lab to an interactive media that students are comfortable with and enjoy using, this application will allow students to focus on the learning experience and will engage them. Instead of making students remember to record important data on paper or struggle with the pitfalls of existing technologies, a central solution that is built around a user friendly platform will prompt students for critical data in-line with the procedures in a user friendly interface.

## 2 Background

### Benefits of Technology in Education

Increasing student exposure to technology during the educational process has been shown to be beneficial in many ways. In their book *Effective Teaching With Technology in Higher Education*, A. W. Bates and Gary Poole describe the changing landscape of the business world. The changes they reviewed are parallel with changes in the technology available to commercial enterprises. They described the growing pressure being placed on businesses to keep up with the newest technology available as being driven by the need for business operations to be performed more efficiently and quickly. The drive to move business operations to computer-based management systems quickly infiltrated those business' hiring practices. It was stated that those who have had experience using newer business technologies, such as computerized Enterprise Resource Planning (ERP) systems, will have an advantage over other prospective employees (Bates and Pool, 2003). The authors acknowledge the continually changing technology landscape, but stress that it is important for teachers to start integrating these advances as they occur. This would prevent the teachers, and subsequently the students, from being left behind. The pair stated "It is better to start to use technology for teaching than to wait until everything that is known about how to teach well with technology has been learned" (Bates and Poole, 2003).

In his article *Electronic Roundup: Emerging Technologies in Higher Education*, Byron Anderson seconds this notion. He states that educators should familiarize themselves on a yearly basis with new technologies. He explains that new outlets for education can "provide a standardized interface and platform for advanced applications" as well as "facilitating teamwork and constructing knowledge" (Anderson, 2008). His argument is that the new technologies increase the quality of work, while also promoting characteristics that most business environments value, mainly teamwork, collaboration, and peer review.

Using this supporting information, it is the conclusion of this group that developing an application for a tablet will not only increase the value of the education, but the value of the student as an employee in the continually evolving business world. Experience working with these new technologies has been shown to be valuable to students, as well as promoting a career-oriented environment through advanced collaboration and cooperation.

### Adaptation of Technology in Education

Despite the observed benefits of technology in education, it is also necessary to consider the variables that effect the level to which a technological solution is applied and how it is used. The Technology Acceptance Model is often used to map the needs and attitudes of the user group to the reception of the solution in the community (Gao, 2005). By analyzing the perceived use, perceived usefulness and attitude towards the use of a technological solution, it is possible to determine how and to what extent the user intends to use the technology (Davis, 1989).

Positive results in such a study can serve as validation and allow developers to collect critical information for a project such as the one described later in this report. Assuming the final system

meets the criteria presented to the user at the time of the aforementioned study, the user's actual usage of the technology will in turn validate the results of the preliminary study.

## Current Products

Often teachers try to find ways to use technology to engage students in the classroom in the most enticing ways possible. Sometimes they also want to be able to have the class turn in answers or even compare data as a class in simpler ways. The problems teachers may run into include,

- Hard to use applications that have a steep learning curve
- Cumbersome, expensive hardware
- Bugs
- Rigid, unconfigurable setups

The large number of different uses for technology in education has given rise to a number of different solutions, each with different pros and cons. These solutions vary in their level of interactivity and primary media for delivering information. Some of these solutions are just scanned or processed books that are displayed on a device's screen while other allow students to answer questions and be graded on their responses. Some of representative examples of this technology include custom computer programs, video lectures, and eBooks.

Custom computer programs can be very good for taking data from students, and are often easily customized, but they require a lot of programming to maintain and the reliability is subject to the programming skill of the developer, who is often a student with little programming experience. The program can have unpredictable performance on different platforms, counter intuitive graphical user interfaces, and a lack of portability. Some WPI departments have used Microsoft Office InfoPath-based applications that record data to a centralized server through commonly used Microsoft Office applications like Excel. This system has a significant cost associated with licensing Microsoft's software and it has posed a variety of reliability problems. Users that lack certain plugins or an internet connection can lose important laboratory data (Microsoft User Reviews). So, although the customized system yields great flexibility, the large programming overhead and lack of reliability limits the system's use in other learning environments (Microsoft).

A great example of a video lecture-based technology used in education is Kahn Academy: a computer service for students and teachers used to teach students outside of the classroom using instructional videos (Kahn Academy, 2010). Although it is neither portable nor configurable, and requires a lot of upkeep, Kahn Academy has been featured on TED.com and CNN as well as attracting 2 million dollars from Google for further development and winning the Microsoft Tech award for Education (Ou, 2010).

A final approach, that seems to be increasingly relevant is the eBook. These are similar to textbooks but can be interactive or have hypermedia in them such as videos and interactive questions. In addition to the Kindle line by Amazon, Apple has attempted a similar product based on the iPad. The most prevalent problems are the fact that this technology generally lacks the high level of interactivity needed to command the attention of many students and it suffers from typical problems associated with traditional textbooks: high cost and low customizability (Mullan, 2009).

By creating an application with intuitive functionality, that has a small overhead, simple setup and lab generation process, and requires minimal infrastructure our product will try to address some of these issues.

## User Demographic

The specific user demographic for the developed application includes the instructors and students of entry level (2000 level) biology laboratory classes at WPI. However, it is designed in such a way that it can be expanded so that a large portion of lab instructors at WPI could use it to teach their students. The application could even be expanded for use in other schools. In its grandest scale, the typical user could be any student in any laboratory oriented course at a number of different schools. In the meantime, the following two user groups were identified as possible users of the final design:

- Students, 17-25 years old with extensive prior experience in personal computing and moderate experience with touch-screen enabled devices.
- Teachers, 30-65 years old with varying prior experience in personal computing and little-to-no prior experience with touch-screen enabled devices.

## 3 Methodology

In order to best facilitate timely development of the final solution, the application was developed in parallel to the collection of user data. This allowed the team to develop the app while gathering data from the WPI community then perform two design iterations with the collected feedback from user trials.

After successfully applying for an Institutional Review board (IRB) exemption, a feasibility study was performed to see how the WPI community would receive a touchscreen-based application. After verifying that there was interest in the application and the community would be willing to use it, the remainder of the data collection served as feedback to gain a better understanding of what the community would need and want to see in the final product.

### IRB Application

In order to perform any surveys or interviews with members of the WPI community, IRB exemption was necessary so we submitted an application for exemption to the WPI IRB and were granted the permission to survey students and faculty in and around the WPI community.

Since we did not intend to collect or display any individual data on the survey participants, coerce subjects to participate in the survey, or collect data from infants or otherwise vulnerable participants, the IRB Exemption was relatively straight forward. Our application was approved without amendment in ~2 weeks. The IRB exemption approval is attached in Appendix A.

### Curriculum Feasibility Study

Using the Technology Acceptance Model, we identified two potential problems that could hinder the application's usage due incompatibility with the user demographic:

- Students' lack of familiarity with tablet device interfaces and the use of them: The product will need to have a user interface that is consistent with other touchscreen devices on the market, such as phones and MP3 players (Clyde, 2004).

During the IRB process we researched the development of surveys and found that it was very important to establish a set of goals before designing a survey (Janes, 2001; Iarossi, 2006). These



goals allow the investigator to clarify the survey audience, type of questions to ask, and best means of collecting responses. We decided that the goals for an initial survey would be to determine:

- How easily a tablet-based laboratory application could be assimilated into the WPI curriculum
- What level of experience the WPI community has had with touchscreen-based devices
- Whether a tablet based application could improve the laboratory learning experience.

Following the determination of these goals, two target populations as well as the methods for conducting the surveys were selected. The two user groups were:

1. **2 Faculty & Staff** that manage labs or teach a large number of lab-oriented classes
2. **113 Students** taking a lab-oriented biology class

We chose to use two separate survey methods:

1. Personal interviews for the faculty and staff
2. Web-based MyWPI surveys for the class of students.

While personal interviews tend to be more time consuming and subject to interviewer bias, they also have a more personal feel and tend to solicit more detailed responses. For the small faculty group, we decided that sitting down with the faculty would be more advantageous.

The target population uses MyWPI in their courses, which is a Learning Management System (LMS) where instructors can disseminate material to each class they teach. The student surveys were distributed through MyWPI because it was the fastest, most efficient way to gather survey responses from such a large population. Students were given the option of completing the survey or a written assignment for an extra credit award. In the design of the student survey, we determined that short, simple questions would help encourage better responses (Iarossi, 2006). We made sure to include all possible answers to closed ended questions and ordered the questions in the survey in a manner so as to avoid confusion and encourage participants to complete the survey.

We produced two lists of questions. One for the faculty and one for the students. Both groups were asked questions with the intention of revealing information to achieve the goals set forth at the beginning of the process. The faculty survey also tried to establish whether faculty were willing to use the technology. In short, both surveys assessed what aspects of the lab needed fixing, and the faculty survey also assessed whether or not the professor would want to use the application. The interview sheet for the faculty survey and the student survey questions and options can be found in Appendix B.

## Development of the application

### Extensible Markup Language

One of the first major design decisions in the project was to determine a good method of getting the laboratory assignment from the teacher to the student's devices. This method needed to be able to store the content associated with the laboratory assignment as well as all of the information necessary to provide the student with fields to record data. We decided on Extensible Markup Language (XML) as means of encoding the labs into a format that could be read by a wide range of devices, including the Apple iPad. XML is widely used in many applications to transfer a wide variety of information in a simple encoded package that can be easily utilized by any client application. While there are other options available like LaTeX and HTML, these options have more complicated

structures and require more overhead. Additionally they are not natively supported by the iPad making XML the most simple and reliable possibility.

XML uses tags to contain and organize information that can be parsed by almost any computing device. An opening tag marks the beginning of a set of content and a closing tag marks the end. This allows the app to treat each tagged element like an object, or more specifically for this application a specific item of the the lab. Each opening tag can include properties that store information about the content in that tag, a cascading tag system can be used to store data into subsections which allows for hierarchy encoding.

In order to ensure that all parts of the project are able to recognize the same tags and translate them into the same objects which will be displayed on screen, the team created an XML schema that accommodates all the different functionality in the application and establishes a hierarchy system for the user interface.

### Data Collection

Assessing previous lab material and lab reports, we were able to pinpoint the types of data that students might need to record during a typical lab assignment. We concluded that the final product needed to have the ability to collect data in the form of text and tables. We used this initial, limited feature set to generate the first prototype of the lab application. The XML schema was written to identify each type of input area as a different tag so that the app builder and mobile client could recognize and display them.

A simple structure was used to create the input text box for alphanumeric data. The XML schema allows the user to specify the height of the input text box and the application is able to generate and display the text box based on those specifications; additionally the question is displayed inside the textbox until the user starts to type their response. This allows reasonable flexibility for the text input area to be as large as needed. The ability to resize text input areas was identified as a needed feature due to the variety of expected answer lengths found in labs. While some expected one word, others expected a few sentences. Allowing the adjustment of the size of the text allows the incorporation of short or long answers according to the needs of each question.

Due to the fact that we decided to use an Apple iPad as the final platform for the client, we used the Apple iOS Software Development Kit (SDK) to design the mobile app. Tables were the most difficult element to program due to the fact that there is no predefined feature in the iOS SDK to generate simple tables. To do this each cell of the table is implemented as a separate text input box that is parsed from a standard XML table. The coding structure created allows the professor to define the size of the table (number of rows and columns) with a limited width of 7 cells. From reviewing previous lab reports, we decided that the tables would not need to be larger than 7 columns and placing this limit guarantees that tables will be displayed in a visually pleasing scale. The final product allowed tables up to 7 columns to ensure that each cell was not too small to see. The data from tabular input areas can be converted to any standard format, such as comma separated values (CSV). This allows it to be imported into spreadsheets in Excel or other statistical software using the special paste feature.

### Information Display

In addition to allowing users to record text and tables, we determined that the application should be able to display text, pictures, videos, and tables. The lab construction application is able to

add text for lab instructions, pictures and graphs for visual information, YouTube videos for video content, web-links, display tables, user input tables, and text-input boxes.

These forms of data were deemed adequate to be able to cover most, if not all, of the lab data collection used in WPI biology courses. The application itself was built in such a way that it is able to expand and incorporate more features as the demands for the application change and new technologies find their way into the lab.

### Mobile Lab Client

The iOS SDK breaks down the app development process into different sections: The background tasks are constantly running performing any desired operation the developer needs. While a human interface layer takes human input and displays content in response to user interaction, the XML parser is run in the background to take the XML data created by the app builder and convert it into structures that could be displayed by the interface layer. On the interface layer, a tab system allows the user to always see the different sections of the app on the screen and easily transfer between them. Whenever a user inputs data it is interpreted by the background and stored to prevent lost data.

Due to the user's ability to easily interact with the content, a user interface had to be developed to give the user easy intuitive control over everything. This is why the tab interface was picked. A few options were tested for the human interface of this application. The first one was to create a continuous document resembling the actual print out of the lab where the user needs to scroll through the entire lab. This option was rejected because each lab could be upwards of 20 pages and the user would have to keep track of how far up or down a particular section is. The next option that was tested was a still document resembling the iBook and Kindle application. While this option gave a nice look and feel to the lab that resembled real paper having to flip through the pages was slow and still required the user to remember the layout of the lab. The third option tested was a tab design resembling Apple Mail, where a bar on the left side of the screen displays the title of every section and user can jump between sections easily.

### Faculty Post-App Builder Survey

Our Faculty Post-App Builder Survey (appendix G) was designed to test our App Builder to see if it had been well designed, was user friendly, encouraged the user to use it, and provided the needed functionality for a professor to build an app that they are confident would be useful. Two assumptions that were made were that the professor wanted a tool built for them and that they thought that the iPad or tablet devices were a good implementation method for the App. The survey separated the user friendliness and intuitiveness of the app into several questions. The first was whether the App was user friendly. This question was meant to be an overall judgement of the App as a whole. The question was to be answered on a scale of 1 to 10. The next set of questions asked if each of the portions of the App Builder's design was intuitive. Each menu received its own question with all of the editor windows lumped into one question; again these questions were answered on a scale of 1 to 10 on whether the part in question was intuitive or not. A final question asked if the previewer gave an accurate sense of the App that would be produced to test if the users had a sense of what they were building.

Another set of questions was used to see if the professors thought the App Builder was a good

idea. Two of these questions were, “**would you use this App Builder as it is to make labs for your classes?**” ,and “**If improvements were made to the App Builder would you use it in your classes?**”. These questions were to judge how much more development would be needed for the App Builder so the next development team would have some guidance on how close this first version would be to what was needed for the App Builder to be accepted as a tool. The professors were asked what changes they wanted to see in the App Builder itself and if they had any general comments on the whole system. The professors also said what departments they were from to help organize the responses to allow us to look for any important trends by department.

The actual implementation for the survey was as follows. The professor was presented with an iPad that was loaded with the test lab that was used in the student survey and was given a brief introduction to how the App worked. The professor was encouraged to simply explore the App and their questions were answered to give them a basic understanding of the underlying structure of the lab’s XML code. This was allowed since we are not specifically asking them to judge the XML code. However, a limited understanding of XML was deemed necessary user background for using the App Builder. We also thought seeing a full App would allow them to judge the previewer in the App Builder in a better light. Then they were given the App Builder on their own computer and asked to try to build an App. When they were done, we asked them to answer the survey questions. No further instructions were given.

## 4 Results and Discussion

### Survey Results

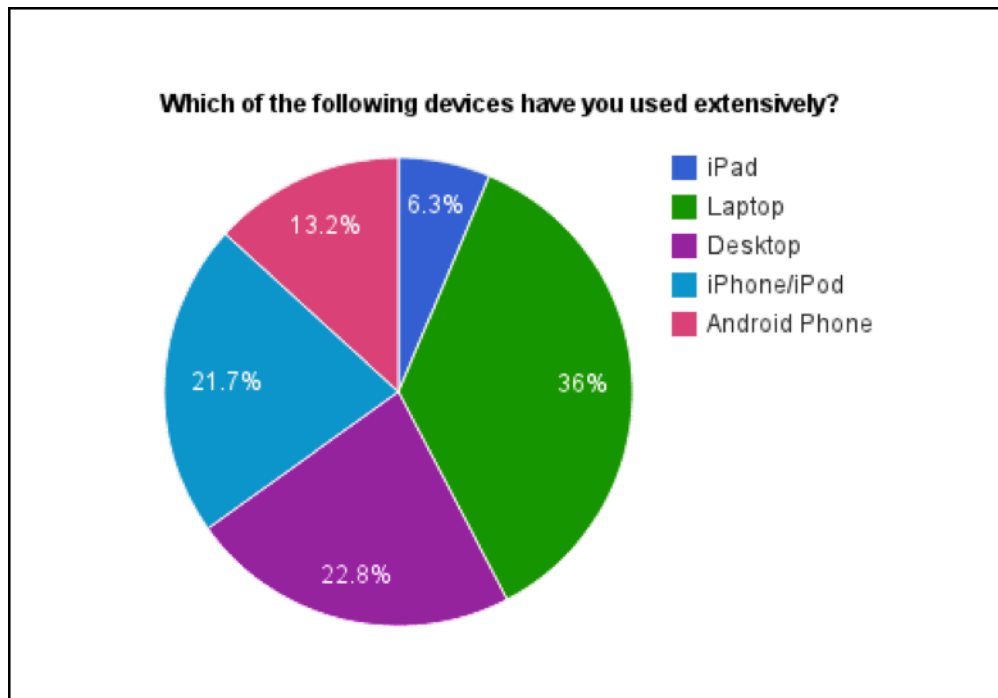
#### Faculty Feasibility Interviews

We were able to meet with two professors to conduct the survey. One of the professors was the project adviser. The professors were from the departments of Biology and Biotechnology and Mechanical Engineering. One interesting fact that was learned through the conducting of this survey was that although both professors used relatively similar means of data collection and distribution, they had different outlooks on the necessity of the application and how it could be implemented in their labs. One professor believed that the application could be a great improvement in his/her current labs while the other believed that based on his/her needs, the product would not be needed or improve the current lab situation. Even with these differences, both professors believed that the mobility of laboratory assignments would be greatly improved with the implementation of new technology.

#### Results of Student Feasibility Survey

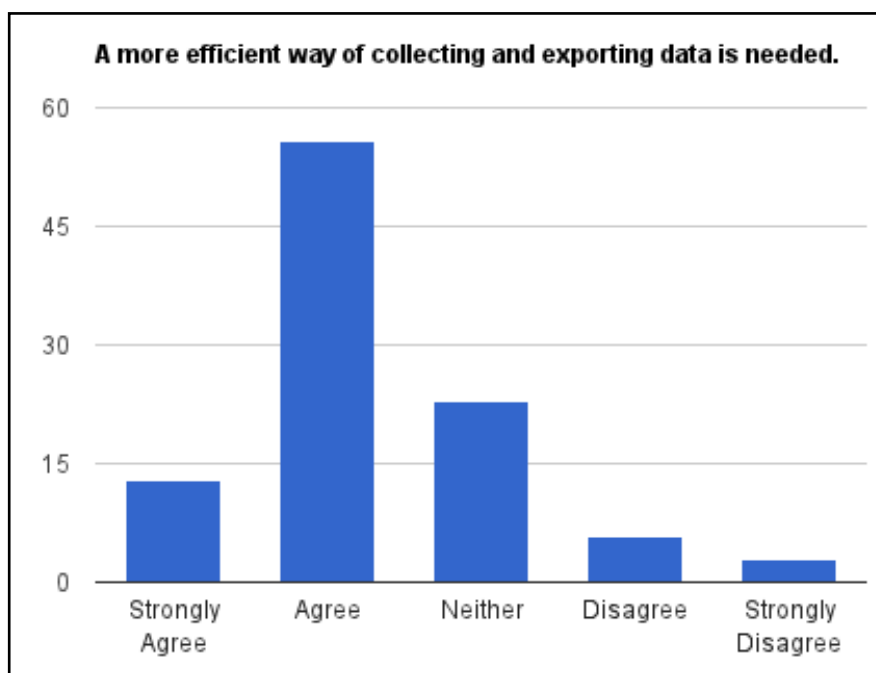
The student feasibility survey revealed that a large portion of the intended user group has had prior experience with touchscreen devices, such as the iPad. It also revealed a large portion would be willing to try the new technology. When asked “which of the following devices have you used extensively?,” 92% of the participants reported having used an Apple product, such as an iPhone or iPod, while close to 31% of the participants had used an Apple iPad. These results are in stark contrast to the results regarding use of the Motorola Xoom or HP TouchSmart, for which fewer than 1% of the survey participants reported previous use. The compounded results for both studies can be seen below

in Figure 1. The community’s prior experience with the Apple iOS touch screen interface (used in Apple iPhone, iPod Touch and iPad) served as good justification to use the Apple iPad as the application platform; the idea being that students will be willing to use a familiar technology in a laboratory setting and there will be a minimal learning curve to the user interface.



**Figure 1: Student Device Predisposition**

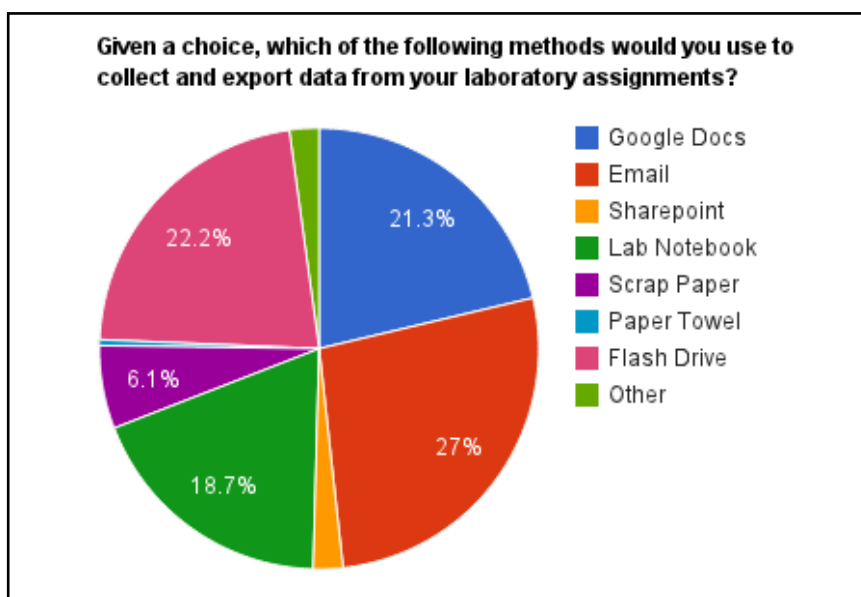
In Questions 3-6, students were asked to rate their level of agreement with 4 statements in order to get a better idea of the perceived need for a new solution to data collection in lab. The first of these asked participants to rate their agreement with the statement “a more efficient way of collecting and exporting data found in labs is needed.” The percentage of students rating their agreement with this statement as either “strongly agree” or “agree” was 72%. Another 21% chose the neutral response. These results are reported below, in Figure 3.



**Figure 2: Histogram of responses plotting perceived need of a new application**

This data suggests not only the willingness of students to experiment with different data collection methods, but also to permanently adopt a more efficient and effective way of collecting data. In question 4, students were asked to rate their level of agreement regarding data collection and organization being a negative aspect of their laboratory experience. Answers for this were spread evenly most likely attributing to data collection and organization being essential for the lab work but still remaining tedious at times. In question 5, students were asked how many times they needed to come back to the lab during unscheduled times to obtain missing or misplaced data due to an error in data collection or sharing methods. The answers here were mixed as well, ranging from “almost never” to “Once or twice a term,” each of these responses with about 20-30 percent. This may show that some number of students excel at current data collection methods, while the same number occasionally forgets or misplaces data. Question 6 asked about the frequency of data forgotten in reports. The answers ranged, with the majority being “not very often” at 43%. However, the data from this question strengthens the notion that not every student is performing in an efficient and organized way in lab, which is something an in-lab application may help improve.

When asked to quantify the amount of text they would be willing to input into a tablet device, a large portion (40%) of students answered “100 words.” The other answers were listed at roughly 20 percent, but the willingness of all students to input a decently large amount of text implies that they would have no problem inputting data into tablet devices from their labs; this is especially due to the fact that the labs often don’t ask for a lot of text. In Question 9, the students selected the following 4 methods of collecting their data: Flash Drive, Google Documents, Email, and Lab Notebook. The results of this are reported below in Figure 3.



**Figure 3: Students' preferred method of data collection**

Tablet was intentionally left off as an option because we were specifically concerned with existing options and did not want the students to be confused with the several definitions of the word. The short answer questions provided the most insight into the needs of the student and the areas of the lab that needed the most improvement. Questions 8 and 10 established the needs for an improvement in the data collection and sharing system. There was a total of 74 responses for every question, with none unanswered. The ones that have been chosen are those that are most similar to the majority. In answering the question regarding what needs improvement, some of the top answers included,

- *“Data could be collected and organized easier and it would make writing the lab report much simpler. Less time could be spent keeping track of data and more time could be spent focusing on the lab,”*
- *“I think that students would be more efficient in documenting their data with the use of a tablet application. When using Google Docs or just writing down data on a piece of paper, it is easy for data to be misread or misplaced. Sometimes students put their data in Google docs incorrectly, but with an App that walked students through where data goes, Lab data would be accurately documented,” and*
- *“A tablet could really improve mobility of the student throughout the lab, collecting data on multiple experiments in the room. The laptop limits you to return to the work bench eventually, which may prove inconvenient depending on the lab environment.”*

Overall, these statements were interpreted to imply the need for easier collection and organization of data, more efficiency in sharing, and an increase in mobility and working space. All of these aspects are ones that the tablet application being developed is hypothesized to improve.

Question 11 gets down to the specifics of the labs, and students were asked what aspects of current collection methods needed improvement. Again, answers were chosen based on their representation of the majority. The most thorough and representative answers included,

- *“I don't like having to record data from a lab or experiment in multiple places because it causes disorganization and usually lost information. Consolidating information recording so that*



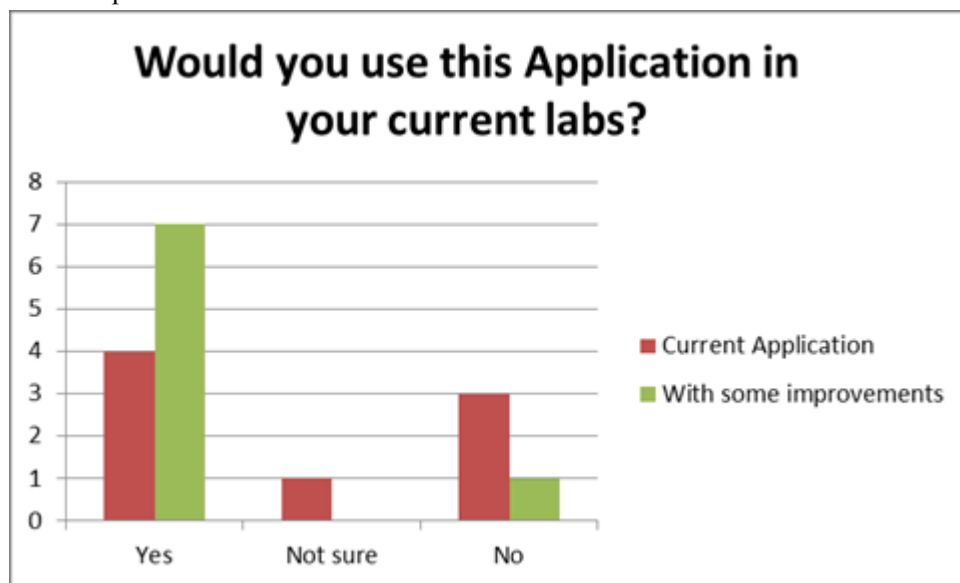


## Faculty Post-App Builder Survey

The application builder was demonstrated to 8 professors after which a survey was used to gather feedback on the intuitiveness of the builder. The goal of this survey was to make sure that professors using this software will be able to learn how to use it.

The survey consisted of a few numerically evaluated questions and a few open ended questions. This helped us make sure that the professors were taking the time to evaluate the application builder fairly and the team could also get their personal opinion as feedback for further improvement.

In general the majority of the professors seemed to like the first implementation of the builder but would like to see it evolve into a more refined product in future versions. Two questions in the survey particularly supported the above conclusion. One asked whether each professor would use the current system and the second one asked if they would use it in the future if and when a few more improvements were implemented.



**Figure 5: Professors' Willingness to Use Technology in Labs**

Figure 5 shows that faculty have mixed opinions about whether they would use the current application as-is. All but one answered that they would use the application after it was improved.

In the further comments section some professors showed excitement about the concept and the approach, but felt that there needs to be more time devoted to improving the application even further. Some of those comments included:

- *“This is a great idea. It is a great first step. I am looking forward to see new improved versions”*
- *“Looks great so far”*
- *“I think this is a great idea and if it works across platforms it would be an essential teaching tool.”*

Other questions were designed for the evaluation of the specific features of the application. These questions established that the professors saw the application builder to be intuitive enough to be used by everyone. Based on the results, it seems that there is room for improvement on the element creation menu.

One of the additional features added to the builder was the application previewer. There was a question included on the survey to receive feedback on the development of this feature. The responses showed that most professors were able to more accurately visualize the final laboratory after they saw the preview of it.

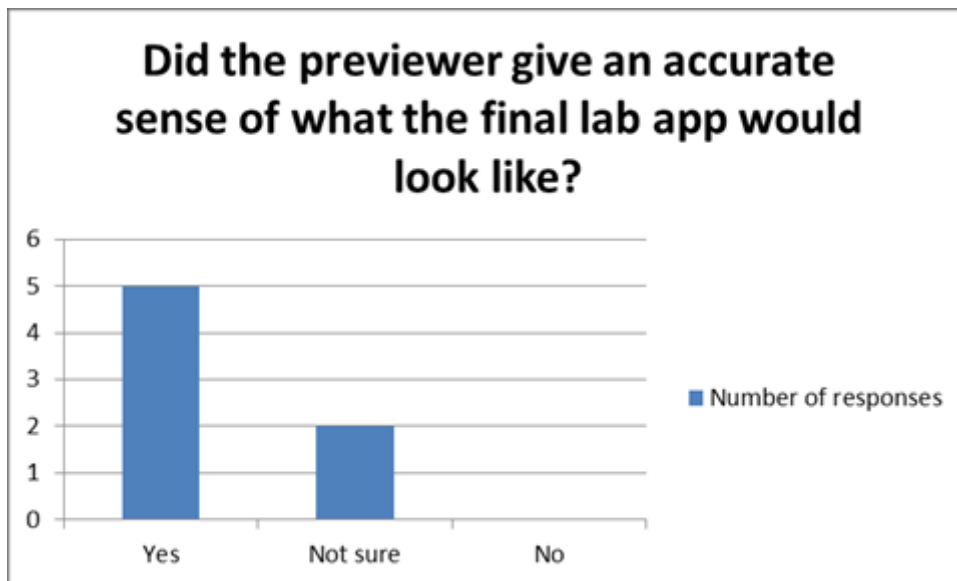


Figure 6: Faculty Opinion on Previewing Labs

A couple of professors took the survey a step further, and decided that on the further comments section they would add their input on the application, the builder and the project as a whole.

“

- *Overall, the concept of a guided lab would be valuable. The lab generation will have to be convenient, the lab should be straightforward for the students to use while being flexible, would be great to be able to see student progress, and should have usable export options.*
- *Need to sell why tablet vs. fixed workstation PC. Is it fixed to station or group?*
- *Do you need to interface with devices? If not it would be much easier to be generic.*
- *Lab suggestions:*
- *Can the lab be stopped, started later, relogged into later, etc*
- *Can the instructor see the status as progressing?*
- *Could the lab output the results and notes in a lab report template? A consistent format would be very helpful for reviewing.”*

Another participant commented:

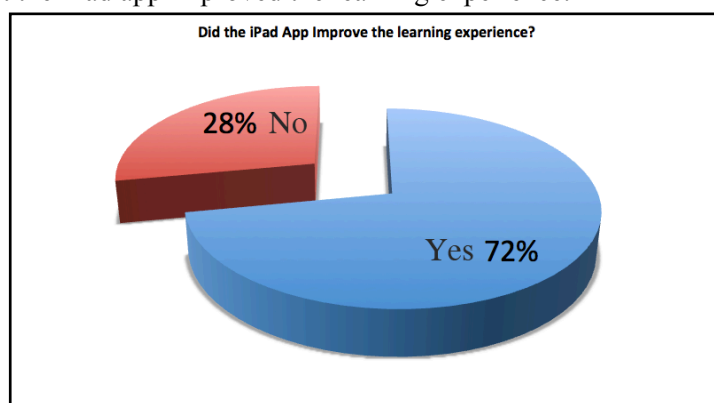
“

- *The interface is intuitive, but not necessarily convenient.*
- *It is very important to have templates available to base the labs off of.*
- *Make WYSIWYG - see what it looks like as you are going along. Model off of surveymonkey or google docs. Or, show something similar to your preview as developing. Would be very helpful to be able to copy and paste elements from other labs and then modify.*
- *How to do formatting in the lab text?*
- *Should be able to export data as excel or other usable format directly*
- *May be worthwhile considering a way to edited all at once (more like Latex/html coding). May want to have multiple ways to generate the common xml format.*
- *can you insert/reorder elements?*
- *Not everyone is going to want the same thing, so there is probably not 1 ideal solution. Having a standardized format and multiple ways to read/write to it may be useful.*
- *Why standalone vs web-based? Both could be OK, but should justify.”*

Their comments were thorough and gave us insight that helped us improve the app and set ideas for further improvement of the application in the future.

#### Results From Student Post Lab Survey

The following data was collected from the Post Lab Test Survey. Overall it shows that people enjoyed using the app and found it beneficial to the lab experience. Figure 7 shows the percentage of students that thought the iPad app improved the learning experience.



**Figure 7: Percieved Benefit of the App**

The next survey question was to gauge whether students would rather use their own laptops instead of the iPad. Overall, the students mostly seemed to want to use the iPad. There is more

information as to why they answered this in the performance section. Figure 8 shows the level of preference to using the laptop computers over the iPad. Overall the students were reluctant to use the laptop if given the choice of an iPad.

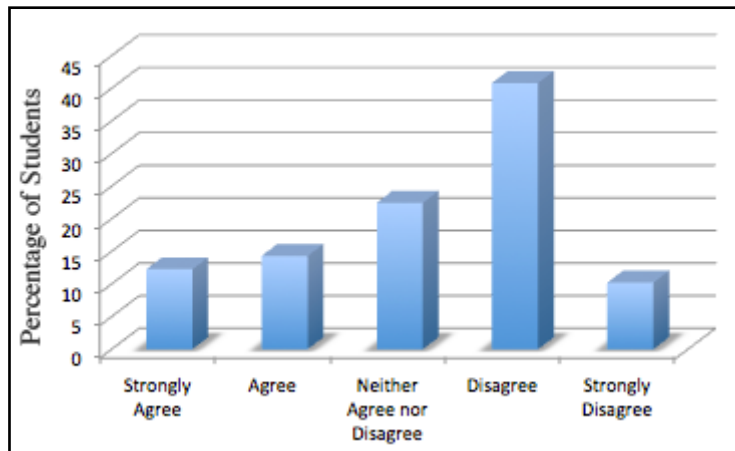


Figure 8: Student Opinion on Laptops vs. iPad

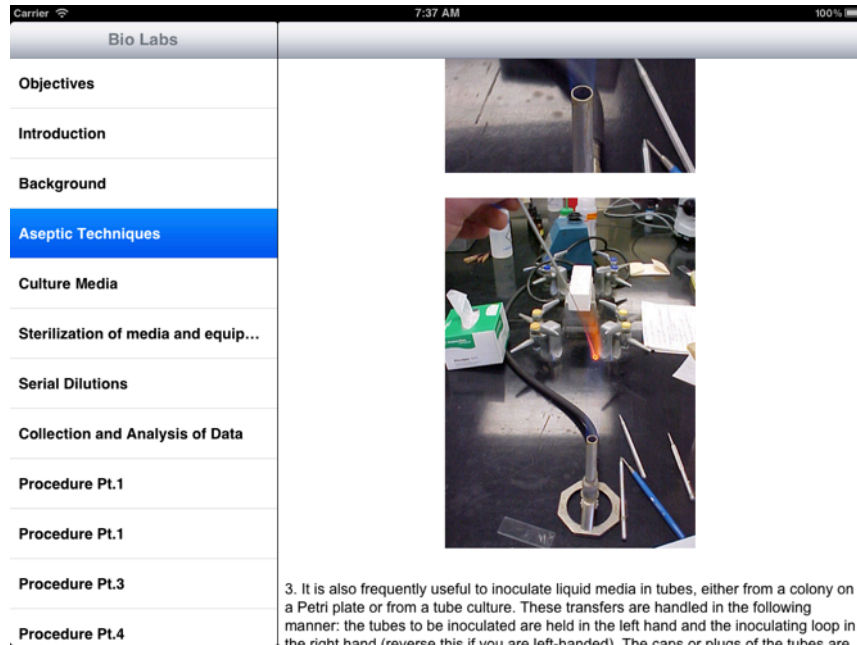
### Mobile Lab Client Demonstration

The app utilizes a tab view framework allowing any user to easily navigate through the sections of the lab. The first tab leads to the home page which consists of the lab title and a short description of the lab. Currently the app is built to handle basic text, images, videos, tables, text input boxes, and table input.

Bio Labs	
Procedure Pt.3	<p>Bioluminescent Bacteria: There are 9 species of bioluminescent bacteria known at this time. Because it is such a unique attribute they were all originally classified as a single genus with the following general characteristics: "Plump, straight rods, 0.8-1.3 um in diameter and 1.8-2.4 um in length. Accumulate poly-beta-hydroxybutyrate under suitable conditions; do not use beta-hydroxybutyrate monomer in the medium. Gram-negative. Motile by 1-3 unsheathed polar flagella, or non-motile. Chemolithotrophs capable of respiratory and fermentative metabolism. Can grow in absence of oxygen when suitable electron-acceptor present. Do not denitrify. Cannot use molecular nitrogen. Fermentation of glucose produces acid. Sodium ions required for growth; K+ will not substitute for Na+. Most strains will grow in synthetic seawater containing glucose and NH4Cl; Some strains also require L-methionine. All strains can use glucose, mannose, fructose, or glycerol as the sole carbon source. All strains grow at 20 deg C. Also slow growth at refrigerator temperature". (reference: Isolation of Pure Culture of Bacteria, Vibrio phosphoreum from Squalid Ocean Fish, <a href="http://www.disknet.com/indiana_biolog/b203.htm">http://www.disknet.com/indiana_biolog/b203.htm</a>, 3/19/05)</p> <p>Because of newer techniques and closer examination the bioluminescent bacteria (BL) have been reassigned to four genera in which they more closely match the other members than they do each other. The 9 species along with some unique attributes are listed below: "Alteromonas hanedai - only obligate aerobic emitting light; grows at 4°C not at 35°C?? (Photobacterium) leiognathi - fish light organs; no growth at 4°C; growth at 35°C Vibrio cholera - a few, rare strains emit faint light; growth at 40°C Vibrio fischeri - yellow-orange growth at 30°C; no growth at 4°C; emits light weakly Vibrio (Photobacterium) phosphoreum - growth at 4°C, no growth 35°C; brightest light emitter of all bacteria; isolated from fish light organs and seawater Vibrio (Lucibacterium) harveyi - many strains emit some light Vibrio logei - growth at 4°C; most strains emit light Vibrio (Beneckea) natriegens - survives years at room temperature; even drying Vibrio splendidus 1 - most strains emit strong light; growth at 40°C Xenorhabdus luminescens - isolated from Heterorhabditis, genus of nematodes" Reference: Isolation of Pure Culture of Bacteria, Vibrio phosphoreum from Squid or Ocean Fish, <a href="http://www.disknet.com/indiana_biolog/b203.htm">http://www.disknet.com/indiana_biolog/b203.htm</a>, 3/19/05)</p>
Procedure Pt.4	
Procedure Pt.5	
Appendix 1: Data tables	
Table 2	
Table 3	
Table 4	
Table 5	
Table 6	
Table 7	
Appendix II: Using google docs	
Export Data	

Figure 9: Screen Shot Of Text Section

Above the text input and video view can be seen the text and other content are evenly spaced to allow for a comfortable user experience, the video can be maximized to full screen mode.



**Figure 10: Screen Shot of Images**

Images are displayed sequentially with a small space in-between them. They are set up to have individual subtitles if the XML designates one. Word wrap around the images was tested but not implemented on the final app because it made the text difficult to read and broke up the sentences.

Table 2: Daphnia Survival after Exposure to (solution name)						
Tube/dilution	Fractional dilution	% Concentration	Actual Dil.	Actual concentr...	# of surviving D...	% survival
F	0-no dilution	100				
10 <sup>-1</sup>	1/10	10				
10 <sup>-2</sup>	1/100	1.0				
10 <sup>-3</sup>	1/1000	0.10				
10 <sup>-4</sup>	1/10,000	0.01				
NC (negative c...	No product added	0				

**Figure 11: Screen Shot of Table View**

The table view is able to display tables with a maximum width of 7 columns on any location in the app. In addition any cell in the table can be set up to be an user input and all the data from the modifiable tables can be exported in an Excel friendly format.

Bio Labs	
Procedure Pt.3	3. Be sure to label your plate with your name, the date, and the organism on the
Procedure Pt.4	There is a reason for labeling the back, what do you think it is?
Procedure Pt.5	
Appendix 1: Data tables	
Table 2	4. Place your plate back up in the 30°C5. Check your plate in 24 and 48 hours to look for growth of bacterial colonies.a)Does your plate resemble the master plate?
Table 3	b)Did you get isolated colonies?c)Are there any spots or colonies that appear odd that might be contamination? The digital camera will be available if you would like to record your results photographically.
Table 4	
Table 5	Does your plate resemble the master plate?
Table 6	
Table 7	Did you get isolated colonies?
Appendix II: Using google docs	
Export Data	Are there any spots or colonies that appear odd that might be contamination?

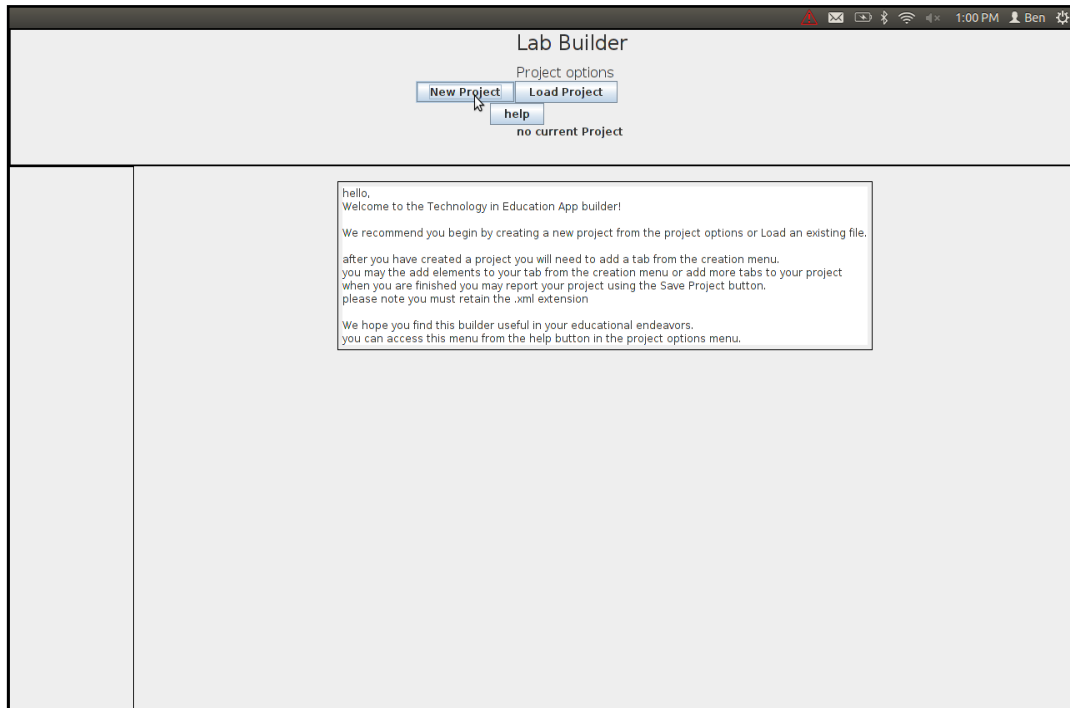
**Figure 12: Screen Shot of Text Input**

The text input (short answer) sections start out with their corresponding question in the box which is overwritten by the user answer. These boxes are set up for text wrap so that when the text reaches the end of the display a new line is created. The question and answer are both exported together allowing the user to not lose track which questions their answers belong to.

### Lab Builder Application Demonstration

The App Builder is made from a series of views. The Project Options menu is the top center view. This menu allows users to load, save, edit, delete, ask for help, or create a new project. The next menu is the Creation Menu, it is located on the left side. The Creation Menu is used for adding tabs to the project or elements to the currently selected tab. The next important section is the drop down selection boxes located below the Project Options, these boxes will load the user workspace in the center. The selection boxes allow the user to navigate the project by selecting the tab or element to edit. The selected tab will always be the tab that new elements are added to.

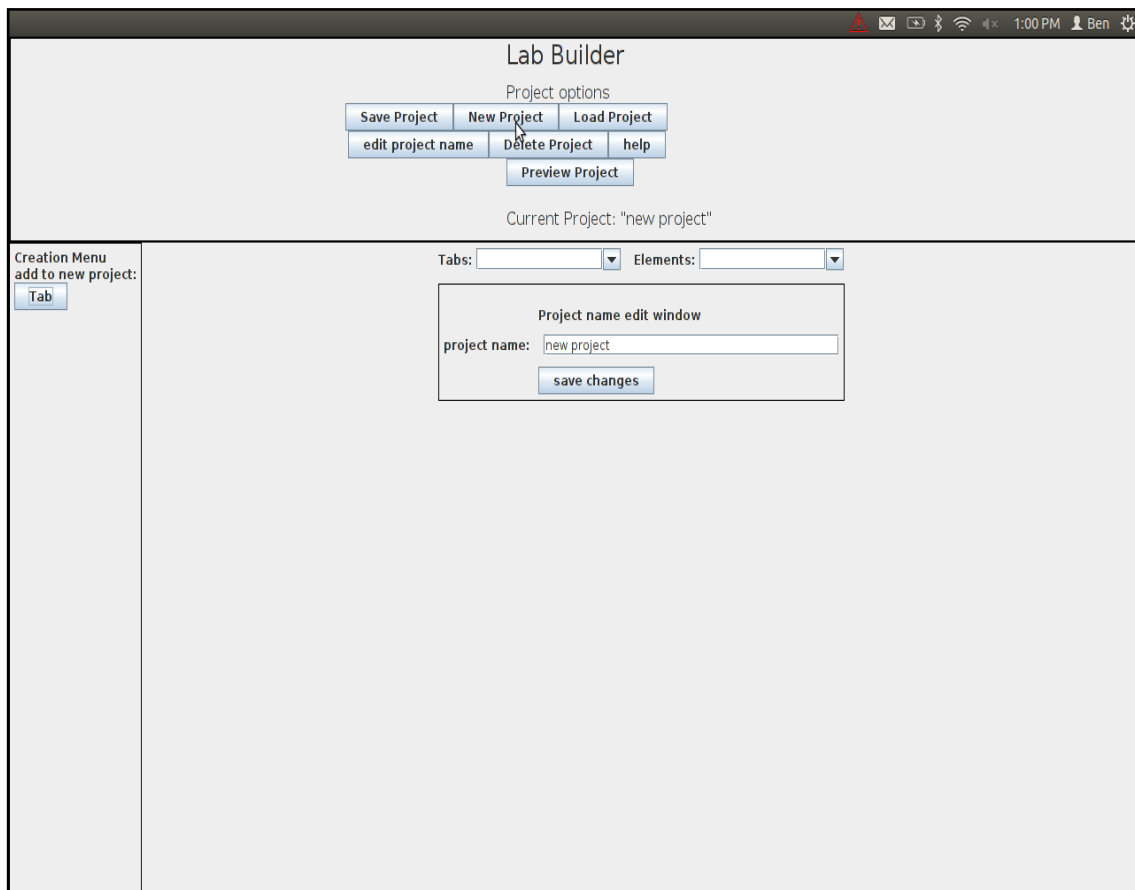
The following figures will show these menus and how the user will build a new project after starting the App Builder.



**Figure 13: Initial lab builder view**

Figure 13 shows the window that greets the user after they start the program. A simple greeting message is followed by general usage instructions for the App Builder. We do assume the user is aware of how the labs function and has actually used one of the labs before. The options available at this point are: New Project, Load Project and Help.





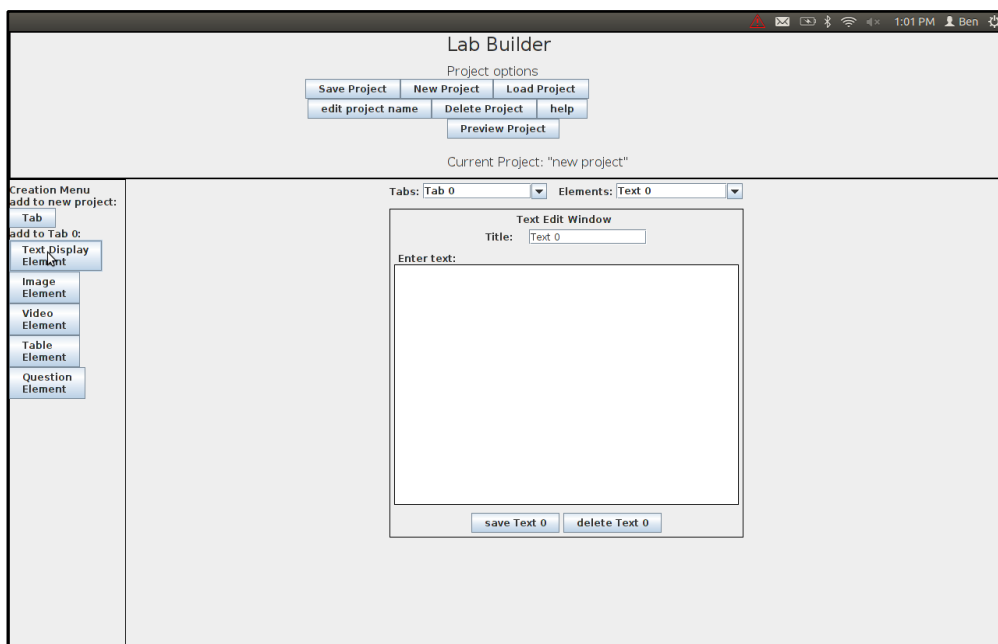
**Figure 14: The New/Edit Project view**

Figure 14 above is the view the user receives after making a new project or choosing to edit the current project. If editing a project the Creation menu on the left side would be present. Here the user can name their project or begin adding tabs or elements depending on where they are in project building. assuming they made a new project their options at this point are: Save Project, New Project, Load Project, Edit Project Name, Delete Project, Help, Preview Project and Add Tab.



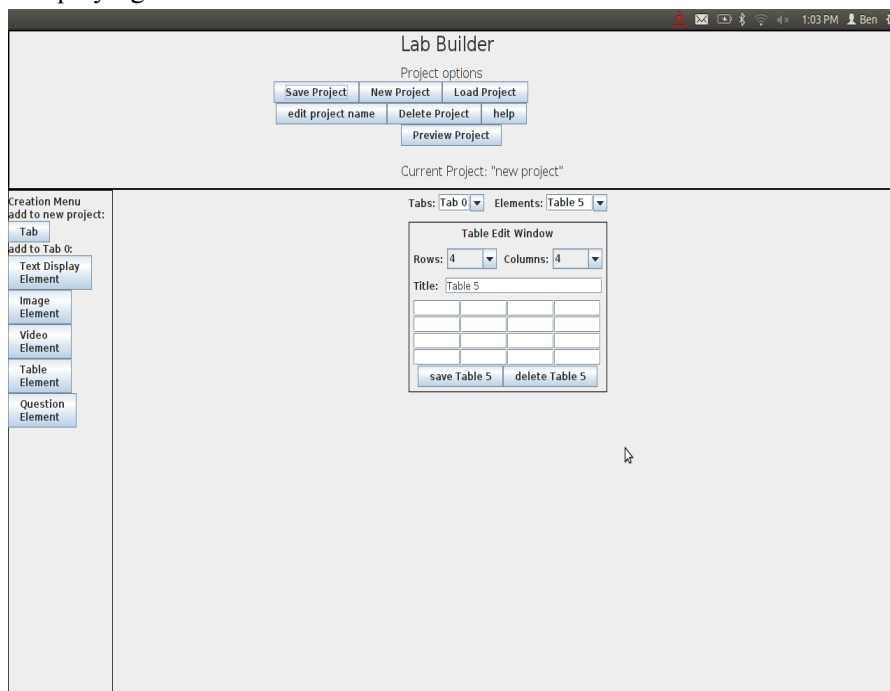
**Figure 15: Edit tab view**

The above view is a view that the user gets after choosing to add/edit a tab. At this point all options are available to the user. the creation menu to the left is fully visible. The tab editor allows a tab to be saved or delete.



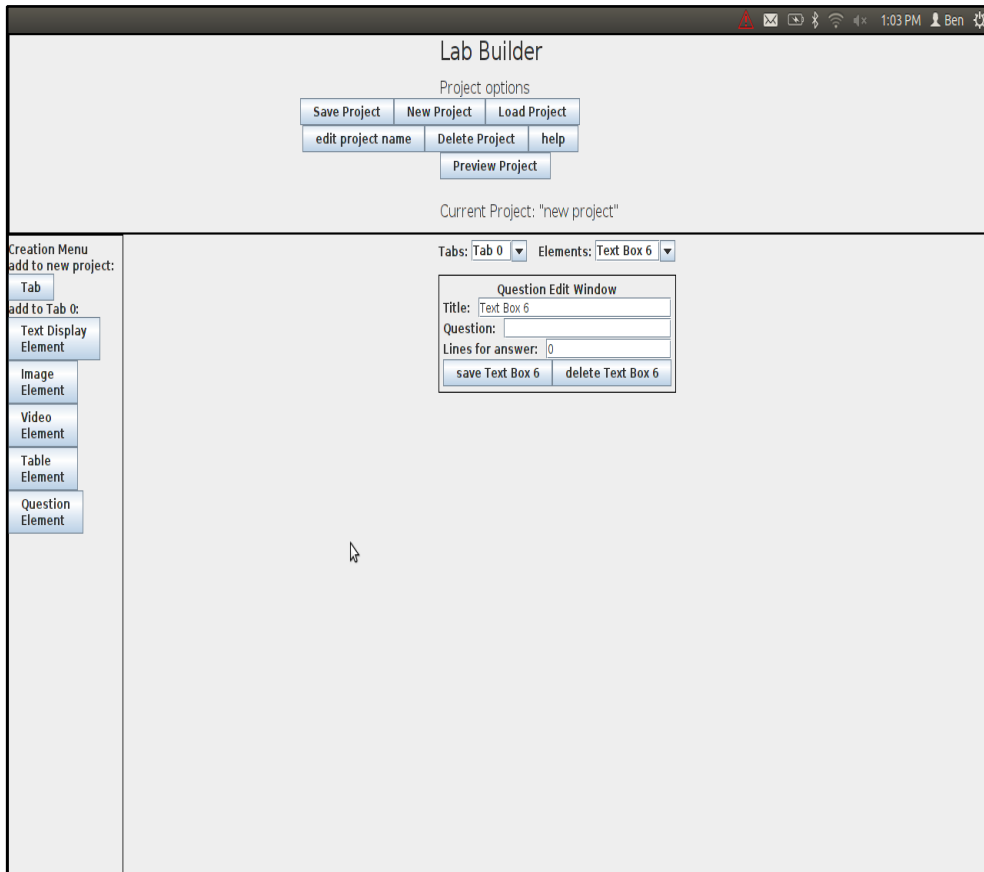
**Figure 16: Text edit view**

The above figure is the view that the user sees when creating a text element in a tab. The text element is for displaying text in the lab.



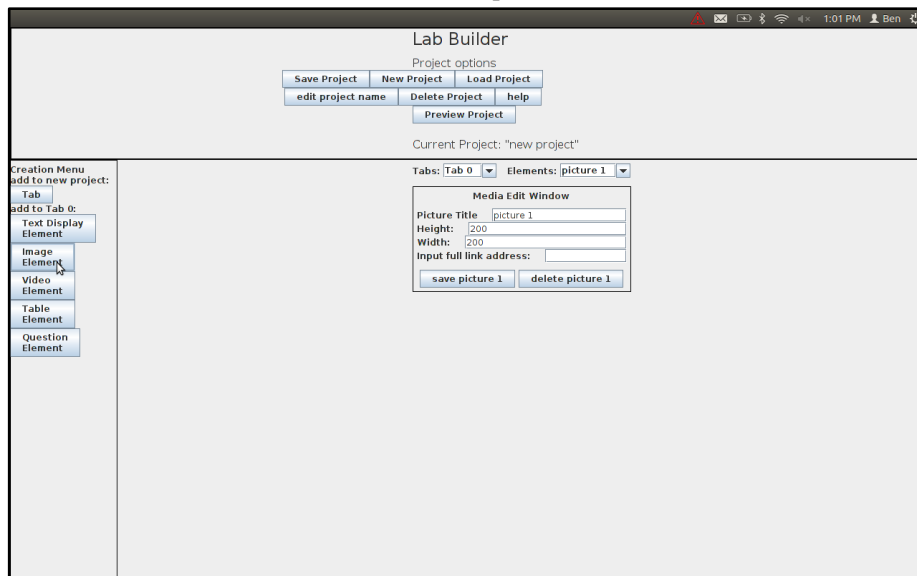
**Figure 17: Table Edit view**

This is the view the user will see for working with tables. The view has two drop downs for selecting the number of rows and columns. The individual boxes can be edited and leaving an empty box will leave it editable by the student in the lab.

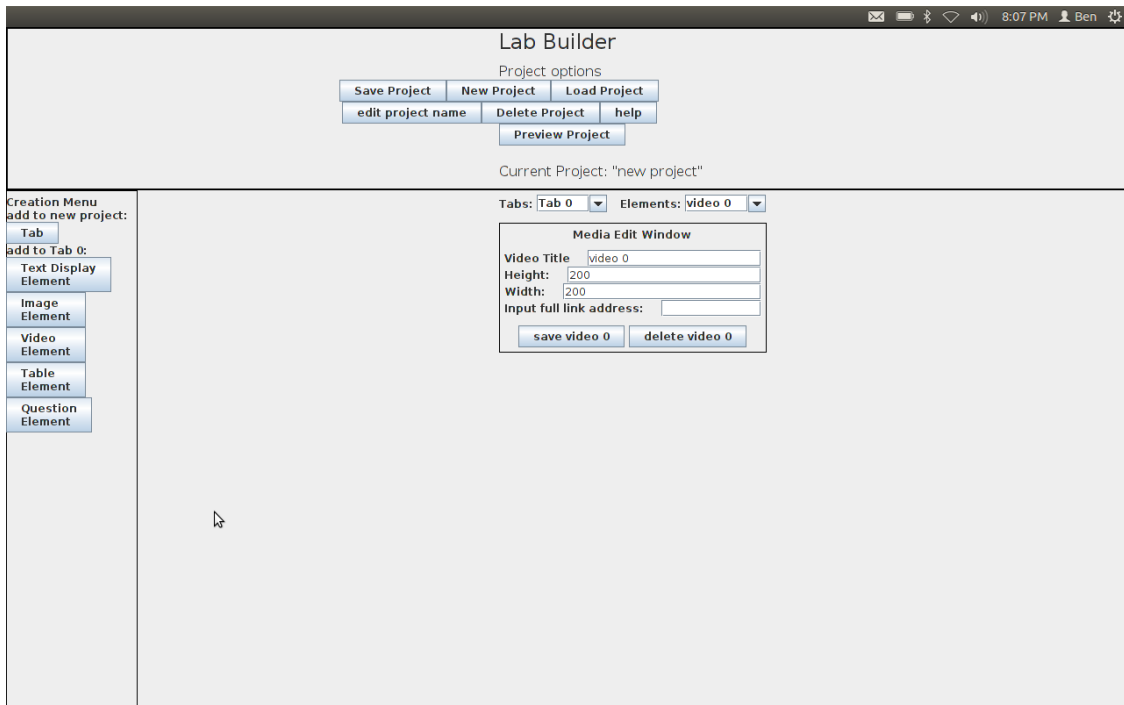


**Figure 18: Question Edit view**

This view is where the user will edit a question element. Here the user can specify the title, number of lines for the student to answer on and the question to be asked.

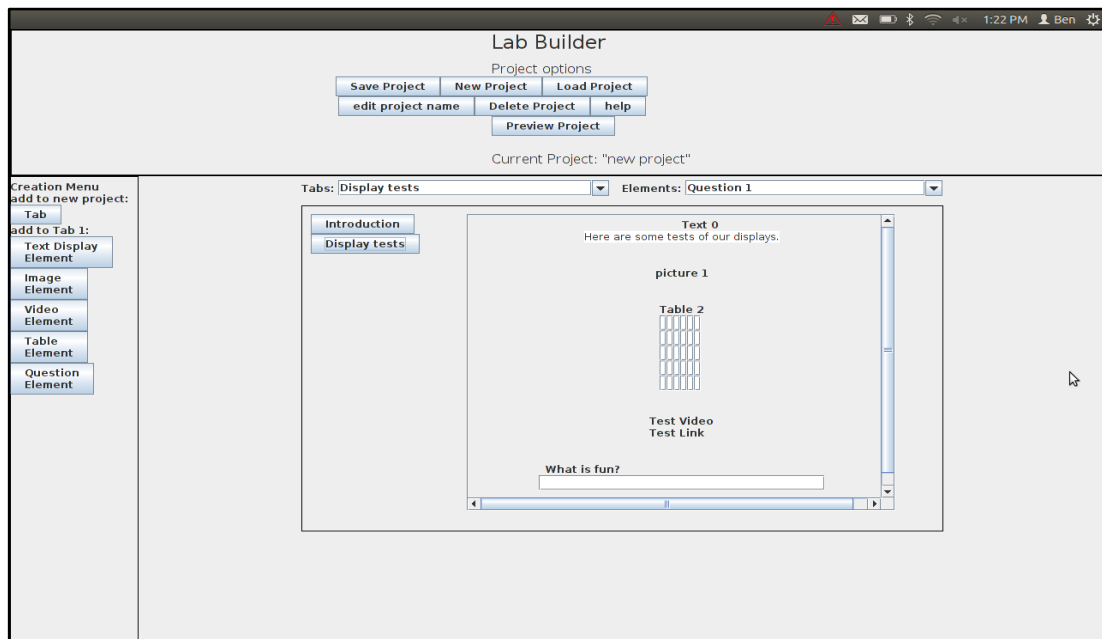


**Figure 19: Image Edit view**



**Figure 20: The Video Edit view**

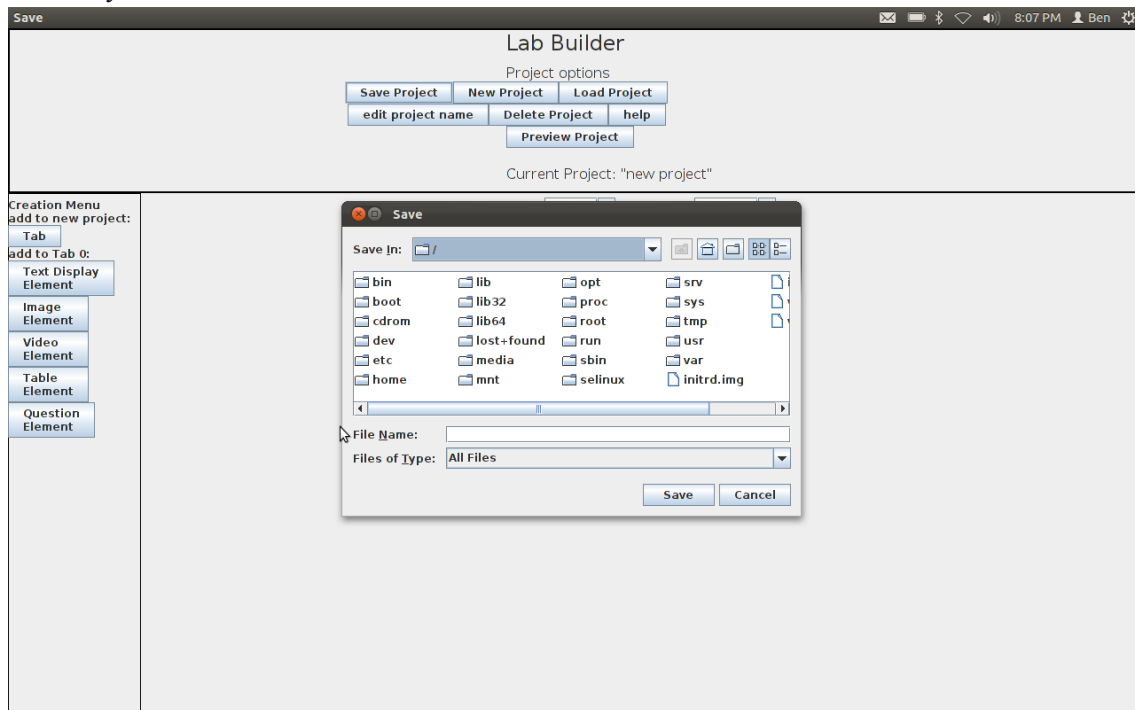
The two above figures are the picture and video edit views. They are for the user to add external links, define the size and add titles to picture and video elements. Here the user can add a picture link or an embedded YouTube link from the internet to add content to the tab.



**Figure 21: Project Previewer view**

This view is the previewer. It is meant to give the user an idea of how the final App will look before they have to save and load it to an iPad. The preview was a relatively high priority piece of the App Builder since without it it is difficult to tell what an App will look like. The previewer also

allows the user to immediately get to any element and edit it or begin adding new elements immediately.



**Figure 22: Save Project view**

After completing their project the App Builder will take the Java project object that the user has produced and convert it to an XML file for the user. The XML file can be placed anywhere on their file system and then read back as a project for editing.

## Intuitiveness

The tablet interface was well received by users, the user interface option selected has proven to be very intuitive and easy to use. A tab interface allows the user to easily navigate the content of the lab and switch between sections with just one tap on the screen.

The XML builder was designed to be intuitive for any professor who might come across it. Intuitive here means that any person could pick this program up and by looking at it know exactly what to do at each step. The XML builder is designed to minimize the number of buttons and sequences that a professor must memorize. Based on where they are in project building, functions are shown or hidden based on what actions they can perform at that time. This is meant to guide the user through application building by hiding functionality from them to keep them from being overwhelmed or confused by extra functionality.

## Performance

The Bio Lab App was made to be as robust as possible. The app was made to not only support the current model of iPads, but it was also made to be usable on future models as well. The software detects which iPad is being used and adjust all of its settings accordingly. The iPad also stores all the data recorded which can then be exported. It can retain the data even if a student leaves the App to go multitask and use the web browser or camera. Currently a shortcoming of the App is its ability to

export data in a user friendly format. Once students clicks on export data they are required to export the data and cannot go back to the lab to change any data. All students have been warned ahead of time of this and it is a currently known bug that can be solved in a future generation of the App. Another bug that has to be fixed for the next generation of the App is a memory leak that occurs when switching tabs. This is where the data is not always getting erased or reused. This causes the app to fill up the iPads ram and also switch between tabs slowly because the media content has to be re-download instead of retrieving it from memory. Currently this problem of robustness was decided to be negligible because the user would have to switch tabs over a thousand times before the application would even begin to show signs of running out of memory and crashing. The biggest bug with the App is that sometimes bullet points are not detected correctly. This can cause different bullet points to be grouped into a single large paragraph that is difficult to read. This issue is caused by the Apple XML parsing library but can be easily overcome by adding HTML bullet point tags in front of every bullet point when the XML is generated.

In the end, the Bio Lab App was successfully created and tested with two different laboratory assignments. The App successfully meets the requirements set up at the beginning of the project. These were displaying data in a similar fashion as the current paper lab and integrating multimedia playback capabilities. The App also surpassed these expectations by adding the capabilities of displaying tables and collecting data in sentence and table form. It also had the ability to export the data collected. The App was regarded as a successful tool by 72% of the survey group. The largest complaints were that the staff wasn't trained on the App and the bullet point bug which was previously discussed. These can be easily fixed in the future. Overall the opinion of the group was that although the App isn't implemented to its full potential yet, it is still a very good platform and will have a positive effect on the lab learning experience.

## Future Projections

A major component of this project is its ability to be transplanted to other departments or universities. Other universities might not have the same availability of servers, or a similar structure which would make a poorly designed product incompatible without expensive hardware or software.

Even though the builder is coded in Java, the information transferred from the builder to the application is stored as XML. The use of XML was a good choice because it uses a schema set by the team. This makes it easy for someone in the future to enhance the existing standard for an updated version of the application or the expansion of the project to Android devices

The Java programming language was chosen for its ubiquity in the WPI community: almost every machine has a Java Runtime Environment pre-installed. Java is generally platform independent and installed on most computers.

## 5 Conclusion

In our biology labs at WPI we felt that a new and better system was needed to increase productivity, organization, and student interest. We believe that this was accomplished with the creation of this application. Although much work can be done to make it better, this application addresses the issues of productivity and organization by allowing the lab to be displayed in one location and also adding mobile technology to the lab process. Students can also use the application to input their data and findings in real time throughout the lab session. We found that many, if not most

students were more interested in using the application for their labs because it was on an iPad, which they preferred over the use of the laptops. The reasons for this preference were not explored.

The App Builder design allows changes to and update of content with relative ease , and requires little technical expertise. This, in turn, permits continual redesign and enhancement of the laboratory content by the instructor. The current form of this application is a great stepping stone to what it could potentially become.



## 6 Authorship

**Table 1: Project Work Distribution**

Ipad Application Desing	Ennio Claretti, Michael Yeroshalmi
Lab Builder Application Desing	Benjamin Leone, Nikolaos Matthiopoulos
Application Standard and Beta run	Benjamin Leone, Nikolaos Matthiopoulos, Ennio Claretti, Michael Yeroshalmi
Development of Faculty Feasibilty Survey	Benjamin Leone, Nikolaos Matthiopoulos, Ennio Claretti, Michael Yeroshalmi, Christopher O'Connor, Joe Everett, Max Saccoccio
Development of Student Feasibility Survey	Benjamin Leone, Nikolaos Matthiopoulos, Ennio Claretti, Michael Yeroshalmi, Christopher O'Connor, Joe Everett, Max Saccoccio
Development of Student Post Lab Survey	Benjamin Leone, Nikolaos Matthiopoulos, Ennio Claretti, Michael Yeroshalmi, Christopher O'Connor, Joe Everett, Max Saccoccio
Development of Faculty Post-App Interviews	Benjamin Leone, Nikolaos Matthiopoulos

**Table 2: Report authorship**

Abstract	
Introduction: Problem Description	Christopher O'Connor, Joe Everett, Max Saccoccio
Introduction: Purpose	Christopher O'Connor, Joe Everett, Max Saccoccio
Background: Benefits of Technology in Education	Christopher O'Connor, Joe Everett, Max Saccoccio
Background: Adaptation of Technology in Education	Christopher O'Connor, Joe Everett, Max Saccoccio
Background: Current Products	Michael Yeroshalmi, Ennio Claretti
Background: User Demographic	Christopher O'Connor
Methodology: IRB Application	Christopher O'Connor, Joe Everett, Max Saccoccio
Methodology: Curriculum Feasibility Study	Christopher O'Connor, Joe Everett, Max Saccoccio
Methodology: Development of the application	Michael Yeroshalmi, Ennio Claretti
Methodology: Extensible Markup Language	Michael Yeroshalmi, Ennio Claretti
Methodology: Data Collection	Michael Yeroshalmi, Ennio Claretti
Methodology: Information Display	Michael Yeroshalmi, Ennio Claretti
Methodology: Mobile Lab Client	Michael Yeroshalmi, Ennio Claretti
Methodology: Lab Builder Application	Benjamin Leone, Nikolaos Matthiopoulos
Results: Faculty Feasibility Interviews	Christopher O'Connor, Joe Everett, Max Saccoccio
Results: Results of Student Feasibility Survey	Christopher O'Connor, Joe Everett
Results: Faculty Post-App Interviews	Benjamin Leone, Nikolaos Matthiopoulos
Results: Results From Post Lab Survey	Michael Yeroshalmi, Ennio Claretti
Discussion: Mobile Lab Client Demonstration	Michael Yeroshalmi, Ennio Claretti
Discussion: Lab Builder Application Demonstration	Benjamin Leone, Nikolaos Matthiopoulos
Discussion: Intuitiveness	Michael Yeroshalmi, Ennio Claretti
Discussion: Performance	Michael Yeroshalmi, Ennio Claretti
Discussion: Future Projections	Michael Yeroshalmi, Ennio Claretti, Benjamin Leone, Nikolaos Matthiopoulos
Conclusion	Christopher O'Connor

## 7 Bibliography

- Bransford, John D., Ann L. Brown, and Rodney R. Cocking. *How People Learn: Brain, Mind, Experience, and School*. Washington, DC: National Academies Press, 1999. Web.
- Brownlow, Mark. "Smartphone Statistics and Market Share." June 2010. *Smartphone Sales and Statistics*. Web. 26 March 2012.
- Clyde, William, and Andrew Delohery. *Using Technology in Teaching*. New Haven, CT: Yale University Press, 2004. Web.
- Davis, Fred D. *A Technology Acceptance Model for Empirically Testing New End-user Information Systems: Theory and Results*. Rep. Boston: MIT, 1989. Print.
- El-Gayar, Omar, Mark Hawkes, and Mark Moran. "Students' Acceptance of Tablet PCs and Implications for Educational Institutions." *Educational Technology & Society* (2011): 58. Web.
- Gao, Yuan. "Applying the Technology Acceptance Model (TAM) to Educational Hypermedia: A Field Study." *Journal of Educational Multimedia and Hypermedia* (2005): 237. Web.
- Iarossi, Giuseppe. *Power of Survey Design : A User's Guide for Managing Surveys, Interpreting Results, and Influencing Respondents*. Herndon, VA: World Bank Publications, 2006. Web.
- Janes, Joseph. "Survey Research Design." *Library Hi Tech* (2001): 419-421. Web.
- Khan Academy. "Salman Khan on CNN : Salman Khan of the Khan Academy on CNN." *CNN News*. CNN. CNN. *Salman Khan on CNN*. Khan Academy, 11 Mar. 2010. Web.
- Karlson, Amy K, et al. "Using Mobile Phones to Present Medical Information to Hospital Patients." 2012. *Microsoft Research*. Web. 1 March 2012.
- Krug, Steve. *Don't Make Me Think!: A Common Sense Approach to Web Usability*. Berkeley, CA: New Riders, 2006. Print.
- Lorraine. "Researching Usability." n.d. *Researching Usability*. Web. 26 March 2012.
- Microsoft. "Powerful and Simple." n.d. *MSDN*. Web. 26 March 2012.
- . "Windows User Experience Interaction Guidelines." n.d. *MSDN*. Web. 26 March 2012.
- Mullan, Eileen. "Ebooks rewrite the rules of education." *EContent* May 2009: 12+. *General OneFile*. Web. 14 May 2012.
- Norman, Donald A. *The Design of Everyday Things*. New York, NY: Basic, 2002. Print.
- Ou, George. "Google Awards \$2M to Khan Academy." *Digital Society*. Digital Society, 25 Sept. 2010. Web. 14 May 2012. <<http://www.digitalsociety.org/2010/09/google-awards-2m-to-khan-academy/>>.
- Phillips, Patricia Pulliam, and Cathy A. Stawarski. *Data Collection : Planning for and Collecting all Types of Data*. Hoboken, NJ: Wiley, 2008. Web.
- Pickering, Craig. "Gestalt Design Laws." n.d. *Squidoo*. Web. 26 March 2012.
- World Wide Web Consortium. "Web Content Accessibility Guidelines 1.0." n.d. *W3C*. Web. 26 March 2012.
- "Microsoft Office 2010 System." - Products. Web. 28 May 2012. <<http://office.microsoft.com/en-us/products/microsoft-office-2010-system-requirements-HA101810407.aspx>>.
- "Microsoft Office 2010 User Reviews." Softonic. Web. 28 May 2012. <<http://microsoft-infopath.en.softonic.com/opinion/a-waste-of-time-347044>>.

## 8 Appendices

### Appendix A: IRB Exemption Approval



100 Institute Road  
Worcester, MA 01609-2280, USA  
508-831-5000, Fax: 508-831-6090  
www.wpi.edu

**Worcester Polytechnic Institute IRB #1  
IRB 00007374**

21 November 2011  
File:11-167

Worcester Polytechnic Institute  
100 Institute Road  
Worcester, MA 01609

**Re: IRB Application for Exemption #11-167 "Enhanced Lab IQP"**

Dear Prof. Buckholt and Rulfs,

The WPI Institutional Review Committee (IRB) has reviewed the materials submitted in regards to the above mentioned study and has determined that this research is exempt from further IRB review and supervision under 45 CFR 46.101(b)(2): "Research involving the use of educational tests (cognitive, diagnostic, aptitude, achievement), survey procedures, interview procedures or observation of public behavior, unless: information obtained is recorded in such a manner that human subjects can be identified, directly or through identifiers linked to the subjects; and any disclosure of the human subjects' responses outside the research could reasonably place the subjects at risk of criminal or civil liability or be damaging to the subjects' financial standing, employability, or reputation."

**This exemption covers any research and data collected under your protocol from 21 November 2011 to 20 November 2012**, unless terminated sooner (in writing) by yourself or the WPI IRB. Amendments or changes to the research that might alter this specific exemption must be submitted to the WPI IRB for review and may require a full IRB application in order for the research to continue.

Please contact the undersigned if you have any questions about the terms of this exemption.

Thank you for your cooperation with the WPI IRB.

Sincerely,

Kent Rissmiller  
WPI IRB Chair

## Appendix B: Faculty and Student Feasibility Surveys

Question 1: What kind of media do you typically include in a laboratory assignment?

What kind of media do you typically include in a laboratory assignment?					
	Professor 1	Professor 2		Professor 1	Professor 2
Video (Youtube)			Excel Table (Eq'ns)		
Video (myWPI)			Data Table		
Video (Other)			Chart		
Photo			Sounds		
Text			Object		

Question 2: What kind of media do you typically collect from your students after the lab?

What kind of media do you typically collect from your students after the lab?					
	Professor 1	Professor 2		Professor 1	Professor 2
Video (Youtube)			Excel Table (Eq'ns)		
Video (myWPI)			Data Table		
Video (Other)			Chart		
Photo			Sounds		
Text			Object		

Question 3: Do you typically collect data from the class?

Do you typically collect data from the class?					
	Professor 1	Professor 2		Professor 1	Professor 2
No			Yes (for class)		
Yes (for Instructor)					

Question 4: If data is shared, how is it collected and redistributed?

If data is shared, how is it collected and redistributed?					
	Professor 1	Professor 2		Professor 1	Professor 2
Google Docs			Emailed Excel		
Infopath			Network		
Other			No Shared		
My WPI			Handouts		

Question 5: What information collection/distribution methods do you think could be improved by a computer application in class?

What information collection/distribution methods do you think could be improved by a computer application in class?					
	Professor 1	Professor 2		Professor 1	Professor 2
Data Collection			Data Analysis		
Assignments			Data Submission		
Pre-Labs			Data Sharing		
Other					

Question 6: What are some of the features you would like in a computer application for in-lab use to include?

Question 7: Have you used an application in labs previously?

Question 8: What types of applications have you used in labs previously?

Question 9: If an application was used previously, what are some problems that have occurred?

Question 10: Do you feel that incorporating a tablet device into the labs would be beneficial to the students, staff, or faculty's laboratory experience?

Student

- 1) Which of the following devices have you used extensively?
- 2) A more efficient way of collecting and exporting data found in labs is needed

- 3) Collecting and organizing data for laboratory assignments detracts from the learning experience.
- 4) How often do you go back to lab because you missed important data or make up data that you missed.
- 5) How much text would you be willing to input on an onscreen keyboard before deciding to use a media other than a tablet device?
- 6) What do you believe could be improved upon in the labs through the addition of a tablet application?
- 7) Would you be willing to use a tablet device, provided by WPI at the start of the lab, as a means of data collection and exportation? Why or why not?

## Appendix D: Faculty Post-App Interview Survey

1. What department are you a part of?
2. Was the App Builder user friendly?
3. Was the Project Options menu intuitive?
4. Was the creation menu intuitive?
5. Was the App Builder intuitive?
6. As a whole were the Element Editor windows intuitive?
7. Did the Previewer give an accurate sense of what the final lab app would look like?
8. would you use this App Builder as it is to make labs for your classes?
9. If improvements were made to the App Builder would you use it in your classes?
10. What improvements would make the App Builder more useful to you?
11. Any final comments?



## Appendix E: Faculty and Student Feasibility Surveys Results

Question 1: What kind of media do you typically include in a laboratory assignment?

What kind of media do you typically include in a laboratory assignment?					
	Professor 1	Professor 2		Professor 1	Professor 2
Video (Youtube)			Excel Table (Eq'ns)		
Video (myWPI)			Data Table		
Video (Other)			Chart		
Photo			Sounds		
Text			Object		

Question 2: What kind of media do you typically collect from your students after the lab?

What kind of media do you typically collect from your students after the lab?					
	Professor 1	Professor 2		Professor 1	Professor 2
Video (Youtube)			Excel Table (Eq'ns)		
Video (myWPI)			Data Table		
Video (Other)			Chart		
Photo			Sounds		
Text			Object		

Question 3: Do you typically collect data from the class?

Do you typically collect data from the class?					
	Professor 1	Professor 2		Professor 1	Professor 2
No			Yes (for class)		
Yes (for Instructor)					

Question 4: If data is shared, how is it collected and redistributed?

If data is shared, how is it collected and redistributed?					
	Professor 1	Professor 2		Professor 1	Professor 2
Google Docs			Emailed Excel		
Infopath			Network		
Other			No Shared		
My WPI			Handouts		

Question 5: What information collection/distribution methods do you think could be improved by a computer application in class?

What information collection/distribution methods do you think could be improved by a computer application in class?					
	Professor 1	Professor 2		Professor 1	Professor 2
Data Collection			Data Analysis		
Assignments			Data Submission		
Pre-Labs			Data Sharing		
Other					

Question 6: What are some of the features you would like in a computer application for in-lab use to include?

Both professors believe that the ability to expand the learning environment would be improved if an application was implemented. (mobility)

Question 7: Have you used an application in labs previously?

One professor had used an application in the lab before while one did not

Question 8: What types of applications have you used in labs previously?

Both professors used desktops in labs before, one also used a laptop.

Question 9: If an application was used previously, what are some problems that have occurred?

N/A for one professor, but mentioned desktops crashing as a problem in the labs. The other professor mentions problems such as space and mobility. The laptops took up too much space, and mobility referred to the fact students were confined to the lab.

Question 10: Do you feel that incorporating a tablet device into the labs would be beneficial to the students, staff, or faculty's laboratory experience?

One professor believed that incorporating a tablet device in the labs would increase the students' willingness to learn, as well as, the ability for them to relate to the material. The other professor believed that it would be beneficial because of the variety of applications that are available already.

#### Student Results

1) Which of the following devices have you used extensively?

		18.
Apple iPad	58%	0.0
Motorola Xoom (or similar)	0%	0.8
HP Touchsmart (or similar)	8%	97.
Laptop	35%	61.
Desktop	06%	61.
Apple iPhone/iPod	95%	
Android Phone/Handheld		30.
Device	97%	

2) A more efficient way of collecting and exporting data found in labs is needed.

Strongly Agree	16
Agree	66
Neither Agree nor	
Disagree	24
Disagree	4
Strongly Disagree	3

3) Collecting and organizing data for laboratory assignments detracts from the learning experience.

Strongly Agree	7
Agree	40
Neither Agree nor	
Disagree	32

Disagree	28
Strongly Disagree	6

4) How often do you go back to lab because you missed important data or make up data that you missed.

Almost Never	32
Not Very Often	34
Sometimes	21
Once or Twice	
Per Term	24
Almost Every Lab	2

5) How much text would you be willing to input on an onscreen keyboard before deciding to use a media other than a tablet device?

5	19.
0	47%
1	38.
00	05%
1	22.
50	12%
2	20.
00	35%

6) What do you believe could be improved upon in the labs through the addition of a tablet application?

According to the results of the short answer question in the survey, many students believed that there were aspects in the current labs that could be improved. Many keywords that were used are quicker, easier, organized, simplify, and mobility. Many of the students believed that the tablet application would make the labs easier and quicker to conduct. They also believed that all the information being located in one place, and collected through the same place would increase organization. They believed that the tablet would simplify the labs and allow the students to be more mobile.

Some quotes that gave the most feedback include:

*“Through the addition of a tablet application it would be easier to view the protocol in class as well as collect data and send it the the people in the group. Tablets are much smaller and easier to maneuver than computers, which would take up less space on the lab bench.”*

*“The use of tablets would make the procedure readily available and easy to keep track of, so it would be less likely that students would forget to do*

*certain experiments or record important data. Quickly and efficiently recording results would become second nature with tablets, and I think it would make it easier to share and compare these results with other classmates.”*

*“I believe that this would help students understand lab procedure and allow labs to go quicker. Also, it will greatly simplify data collection.”*

Even though most students believed that the labs could be improved with a tablet application, a few students were content with the way labs were or preferred a paper and pencil. This shows that even though technology may improve the learning atmosphere, it does not always apply for everyone.

Some quotes that gave the most feedback include:

*“Depends on the circumstances. I'm fond of netbooks for their ease in use and sharing systems such as google docs, but beyond that I don't necessarily believe anything else is needed.”*

*“It would be nice to have forms to put in all the data, however a tablet wouldn't be necessary. Just a print out would be helpful. (The lab print out we have is nice, but sometimes it isn't completely clear, or the data forms are embedded in text. I would love being able to just input the data and have it emailed to my wpi email though.”*

*“I think it would be more frustrating than beneficial. But to answer your question, if everything in terms of lab was on the tablet it would be beneficial, aka all information located in one place.”*

**7) Would you be willing to use a tablet device, provided by WPI at the start of the lab, as a means of data collection and exportation? Why or why not?**

An overwhelming number of students said that they would be willing to use a tablet device. Also there was a student who had used the app during the first testing stage, and gave great feedback.

Some quotes that gave the most feedback include:

*“Yes, because the current methods are very cumbersome. Data collection is currently very unorganized and sometimes completely organizing the data takes a long time.”*

*“Yes, it would be more efficient to collect and export data all electronically so it does not need to all be transferred from handwritten notes later.”*

*“Yes, using the tablet with the unfinished app in lab during C term was extremely helpful and I would love to try again with a finished app.”*

Some students were reluctant to use a tablet device in labs because of their own personal preferences. Also one student indicated that in order to be willing to use the tablet device, the improvements must overwhelmingly outweigh any negatives of the current process.

*“I mean I'd have to if everyone is using it, but I prefer paper and pencil. Then I don't have to worry about losing data because of technological stuff nor waste the time learning a new method, when it takes 3 hours to do the labs.”*

*“I would not be willing to use a tablet as I tend to have difficulties using the onscreen keyboard and find that laptops are much easier to use.”*

*“I would be willing because I feel as though the tablet provides a lot of positive improvements.”*

## Appendix F: Student Post-App Survey Results

### Question 1: True/False

Please choose either true or false:

I believe that the use of an iPad application improved the learning experience presented by this laboratory assignment.

Answers Percent Answered

True 71.429%

False 28.571%

Unanswered 0%

### Question 2: Multiple Answer

If you answered true above, please select why you believe the app improved the learning experience. Please check all that apply.

Answers Percent Answered

The app did not improve the learning experience. 22.449%

The technology makes data collection more fun, making me want to do it.  
38.776%

The technology makes data collection easier, so it allowed me to focus more on the material instead of struggling to get all of my data in one place.  
53.061%

It is quicker to record my data on an iPad than anything else.  
30.612%

It doesn't improve the learning experience, but I managed to install angry birds to make the lab pass quicker.  
6.122%

### Question 3: Opinion Scale/Likert

On a scale of 1:5 how much well do you think this application works? Please rate the application based on the user interface, ease of use, and reliability.

Percent Answered

The application works awful. It broke during our section and we had to use another means of collecting data.  
4.082%

The application was too difficult to figure out and poorly designed.  
6.122%

The application was OK. It feels unprofessionally developed or it had some hitches, but I think it is promising.

36.735%

It worked pretty good! I just want to see it developed a little more.

51.02%

It worked awesome! Why can't this be used in every lab!? 2.041% Unanswered 0%

#### Question 4: Opinion Scale/Likert

I think using the laptop in the lab would be preferable to using the ipad.

Percent Answered

Strongly Agree 12.245%

Agree 14.286%

Neither Agree nor Disagree 22.449%

Disagree 40.816%

Strongly Disagree 10.204%

Not Applicable 0%

Unanswered 0%

#### Question 5: Essay

Are there any features that you are looking for? What do you think we could

add (or remove) to make the application even more useful to you?

Unanswered Responses

1

Given Answers

- I like the overall idea of a touch-screen and portable device to use in a lab.
- I would have liked to have made entering the data into the data tables a little easier - our tablet did not work for this although others did.
- &nbsp;
- be able to add and send camera pictures (from data collection) through ipad to emails<br />
- n/a<br />
- get it so we could download it onto our own iPads (if we had them)<br />
- A couple of times the application froze and then crashed on me, but I was able to reboot the program without losing very much data. <br />
- no
- &nbsp;
- I used the iPad in BB 2903 and it helped so much. But this time, in BB 2904, the procedure was all jumbled into one paragraph which made it nearly impossible to decipher the steps. As



a result, my group immediately stopped using the iPad. The rest of the app was well-designed and I definitely like using it in the lab, but the kinks are getting in the way.<br />

- Add a clear way to export the data so I can get it onto my laptop to do the lab reports after I leave lab.
- More simple to use and easier to get.
- We couldn't even use the app due to a problem in its function.
- I believe that everything that is needed in included, it is just they need to be polished and perfected.
- <br />
- Exporting the data itself was easy but it was somewhat difficult to get the data into a usable form. I followed the directions of how to export the data into excel but I was never able to convert it back into a table format. Luckily, I could still see what my data values were but I feel that this will be a serious problem with future labs.
- &nbsp;It was helpful for reading the procedure, but some of the formatting was skewed, as there were large gaps of blank page during scrolling. Also, due to the nature of the lab, some of the&nbsp;questions&nbsp;which were asked needed atleast 24 to 48 hours of time before they could be answered so we were not able to actually use the full features of the lab.<div>The app should be geared to collect and organize all immediately available data. Perhaps another lab module or class (like it was in 2903) could give more assistance.</div>
- i found the tabs made navigating the procedure faster and more accessible. i found t problems when saving the data and decided to write down the info by hand because i was not confident in saving the data.<br />
- <br />
- Please change it so the format of the lab procedures more closely matches the word document versions of the procedures. In the iPad app, a lot of the words run together and create this big giant block of text that is not easy to read at all.
- N/A
- It would be better if the tabs would actually save your work, not just delete it. It would also need to actually export it properly, it was having trouble connecting.
- Nothing, I think it has all the features it needs. <br />
- some glitches.....
- I think it would be beneficial to have each step of the procedure be on its own tab. The more subcategories and organization within the app, the easier it will be to use. <br />
- I found it easier to use my lap top because I am more familiar with it than an iPad.
- The app wasn't loading on some of the iPads in lab and the TAs weren't trained to know how to fix them.
- &nbsp;It froze, thats my only complaint. I love the camera-card plug in--super easy
- <br />
- &nbsp;
- <Unanswered>
- I was really happy with the features of the application as is. <br />

- I tried clicking to enter my data on the table and it wouldn't work. Maybe it was just a glitch, but I ended up just copying the data in my notebook, which I think was kind of against the point of having the iPad.
- Nothing.
- The procedure format did not transfer well and was therefore hard to read on the iPad.
- Importing and extracting data is the most important, so improvement on that can be done.
- I couldn't really figure out how to keep the data I imputed from disappearing. I thought I hit enter but the data still didn't save. Don't know if it was the app or something I was doing, but if that could be corrected then it probably would have been more useful.
- It is still quite challenging to convert the data into tables from the email we send to ourselves from the iPad...
- &nbsp;
- When the data tables were emailed to my account the format changed completely! and was no longer of any use.
- <br />
- A clearer procedure would be better. The procedure was crammed in the application making it hard to read what to do.
- nothing in particular
- <p>It would almost be better to have excel on the iPad than the application since the text import wizard we were supposed to use was very difficult and I ended up having to put the data into tables by hand.</p>
- &nbsp;
- &nbsp;
- Word and Excel would be ideal.
- It is very promising and a good idea to use the iPad. It is smaller quicker to navigate on but it did not record any data. It would not send any data either. It was only good for having the procedure right in front of you.
- We couldn't edit the titles of the data tables, and the export feature seemed messy, so I ended up just saving my data in excel on my laptop like I usually do.<span class="Apple-tab-span" style="white-space: pre; ">&nbsp;</span>

#### Question 6: Essay

Is there any other comments you would like to make regarding the app? Is there a user interface change that you would like to see? If it broke in your section, what did it do? If it was too hard to figure out how to use, how long did it take you to figure out?

#### Given Answers

- There were problems with the iPad freezing. It would freeze and lose all the data that was input.
- A more reliable way to collect data without fear that everything will be lost by the click of one button or when you try to send it to your email. I just collected data on my own paper instead.

- Same as before- entering data into our tables was not accessible through our tablet app.
- &nbsp;
- tutorial on how to use the ipads would be useful<br />
- n/a<br />
- <br />
- It crashed when I was beginning to input data into the second table, but overall the app was great, I wish I could use it for more events. <br />
- the ipad did freeze a few times...but it fixed itself quite quickly
- I am the type of person to spill stuff randomly...so I hate having any type of electronics near me and use paper...
- <br />
- I like having the lab procedure right in front of me. I like to record my data by hand on paper data tables, so using the ipad for the procedure minimizes the amount of flipping through my copy of the lab I need to do.
- <Unanswered>
- When trying to use the app, nothing appeared on the screen and did not function so we didn't use it in the lab.
- When using the app I could only hold the ipad horizontally, not vertically because the text would not turn. Also, randomly when I would be typing it would begin to delete my text and I wasn't sure how to stop it and I hadn't been pressed backspace. Also, I think the data that is typed in should be automatically saved, even when switching from step to step or even when closing the app to check something on the internet. But overall this app is a very good idea.
- <br />
- No other comments
- I believe this app could be useful in a different environment, due to the nature of this lab it was not as useful as it could've been. My group only used it for reading the procedure, which was only a limited help since we also had our procedure printed out. Additionally, the procedure was straightforward and had an hour of waiting time before results could be compounded.
- the problem was that after filling out tables, data would be lost if we left the tab. <br />
- <br />
- The data collection was a bit buggy; even after entering the data and pressing enter for each cell on a table, it did not save in my section. In addition, I did not like that every time you had to enter something in, the iPad would revert back to alphabet inputs. In filling out a table for data, it would be much simpler and easier if the inputs remain whatever you had them as last. i.e. if you were inputting numbers into a table, and then moved to another cell on the table, your keypad would remain on the numbers. Having to switch back to number inputs every single time was a hassle. Finally, the fact that you have to hit enter after every cell block to save it is something I would like to see fixed. All of those things make data collection with the iPad rather slow and not as efficient as data collection with a laptop is; I think that it could potentially be just as fast but right now it isn't. It's also much easier to type quickly with a laptop than it is with the iPad, but that is more a fault of the iPad itself. <div>Those were really my big issues with the app. I think it is a good program and a promising idea that could help with data collection, but it still needs some work; right now I would prefer to use a

laptop instead. I like being able to test it in lab because I feel as though it is already much improved from the version that was tested in the previous term's lab.

- data wouldn't save. Work on fixing that other than that it was good.
- It just generally was being temperamental, mostly just not saving the work.
- It was really easy to figure out how to use the app. It would be easier to read if the procedures were in the same format as the word document version instead of one giant paragraph.
- The procedure was supposed to be broken up by numbers, but on the iPad the numbers got smooshed between the text making it difficult to read. Entering data was a pain because it wouldn't save unless you pressed the enter button every time you entered a new number. Also couldn't make the screen change from vertical to horizontal...
- I really enjoyed the idea of using iPads in lab, and it made data collection and keeping track of the procedure a lot easier. The biggest improvement that could be made would be better organization of the procedure. All of the steps were listed one after the other without any separation, which made it hard to keep track of what point in the lab we were working on.
- All that my group used the iPad for was to read the procedure on it so we didn't have to click back and forth, while the data was saved on my laptop because we were afraid we'd lose it on the iPad.
- I liked it, it just froze
- I do not have any other comments.
- I liked the embedded videos! But wireless in the lab was kinda shaky so that was a little annoying.
- No.
- The text layouts were hard to follow.
- Some data did not save when it was entered. Exporting was tedious since you had to convert to excel, as opposed to entering in excel to begin with.
- I mostly like the app because the iPad takes up less space on the bench than the laptop does. Not really sure if it makes actual data collection any better though.
- I would recommend an easier way to punch in the numbers/data. Every time I needed to punch in a number I had to click on the button that went from the keyboard (of letters) to the numbers. And after every number I punched in, it went back to the keyboard of letters. It would be great if it could just stay on the numbers section of the keyboard and not go back to the letters.
- The data that we exported from the lab didn't copy into excel right, even when we followed the instructions. This might be because we tried to open it on a PC instead of a Mac but I don't know.
- Make sure that when it's emailed the tables stay the same, if you can do that I feel like the app could be beneficial as I can also access the lab procedures on it, which is nice.
-

- It did not take too long to figure out how to use the application.
- -
- <p>Something that syncs to google docs would be useful</p>
- &nbsp;
- &nbsp;
- It was easy to use, just wish there was an app for word and excel so we could directly type in our results.
- It is a promising idea but last week no data could be recorded and it was missing a table.
- Although I didn't use it for data collection, I found it very useful for reading the procedure. Having it right there with those tabs at the side made it really easy to find the section of the procedure I needed. For someone who had never even used an ipad before, it was pretty easy to figure out.

## Appendix G: Faculty Post-App Interview Results

Question 1: Was the App Builder user friendly? (rating from 0 to 10, negative to positive evaluation)

Was the App Builder user friendly?			
Professor 1	5	Professor 5	8
Professor 2	8	Professor 6	8
Professor 3	5	Professor 7	8
Professor 4	7	Professor 8	5

Question 2: Was the Project Options menu intuitive? (rating from 0 to 10, negative to positive evaluation)

Was the Project Options menu intuitive?			
Professor 1	7	Professor 5	7
Professor 2	7	Professor 6	8
Professor 3	8	Professor 7	7
Professor 4	10	Professor 8	9

Question 3: Was the creation menu intuitive? (rating from 0 to 10, negative to positive evaluation)

Was the creation menu intuitive?			
Professor 1	6	Professor 5	8
Professor 2	7	Professor 6	8
Professor 3	4	Professor 7	8
Professor 4	7	Professor 8	7

Question 4: Was the App Builder intuitive? (rating from 0 to 10, negative to positive evaluation)

Was the App Builder intuitive?			
Professor 1	6	Professor 5	8
Professor 2	7	Professor 6	8
Professor 3	5	Professor 7	8
Professor 4	6	Professor 8	6

Question 5: As a whole were the Element Editor windows intuitive? (rating from 0 to 10, negative to positive evaluation)

As a whole were the Element Editor windows intuitive?			
Professor 1	7	Professor 5	9

Professor 2	7	Professor 6	8
Professor 3	5	Professor 7	7
Professor 4	9	Professor 8	7

Question 6: Did the Previewer give an accurate sense of what the final lab app would look like?  
(Yes/Not sure/No)

Did the Previewer give an accurate sense of what the final lab app would look like?							
	Yes	Not Sure	No		Yes	Not Sure	No
Professor 1			p	Professor 5	p		
Professor 2		p		Professor 6	p		
Professor 3	p			Professor 7		p	
Professor 4	p			Professor 8	p		

Question 7: Any final comments? (Open ended)

Professor 1	Nice idea and first implementation.
Professor 2	Easy to use authoring tool.
Professor 3	App must save to file - not keep data in app. Store labs on device with a list of labs - look at apps like keynote. If the app crashes after partially filling in lab, nobody will want to use it.
Professor 4	
Professor 5	I think this is a great idea and if it works across platforms it would be an essential teaching tool.
Professor 6	This is a great idea. It is a great first step. I am looking forward to see new improved versions.
Professor 7	Looks great so far.
Professor 8	Overall, the concept of a guided lab would be valuable. The lab generation will have to be convenient, the lab should be straightforward for the students to use while being flexible, would be great to be able to see student progress, and should have usable export options. Need to sell why tablet vs. fixed workstation PC. Is it fixed to station or group? Do you need to interface with devices? If not it would be much easier to be generic.  Lab suggestions: Can the lab be stopped, started later, relogged into later, etc Can the instructor see the status as progressing? Could the lab output the results and notes in a lab report template? A consistent format would be very helpful for reviewing.

Question 8: Would you use this App Builder as it is to make labs for your classes? (Yes/Not sure/No)

Would you use this App Builder as it is to make labs for your classes?							
	Yes	Not Sure	No		Yes	Not Sure	No
Professor 1	þ			Professor 5	þ		
Professor 2			þ	Professor 6		þ	
Professor 3			þ	Professor 7	þ		
Professor 4	þ			Professor 8			þ

Question 9: If improvements were made to the App Builder would you use it in your classes? (Yes/Not sure/No)

If improvements were made to the App Builder would you use it in your classes?							
	Yes	Not Sure	No		Yes	Not Sure	No
Professor 1	þ			Professor 5			þ
Professor 2	þ			Professor 6	þ		
Professor 3	þ			Professor 7	þ		
Professor 4	þ			Professor 8	þ		

Question 10: What improvements would make the App Builder more useful to you? (Open ended)

Professor 1	Full HTML text editor and the ability to see the formatted output in the preview window.
Professor 2	Broader device support. The ability for students to attach figures, images, videos, etc.
Professor 3	Standard menus. Full-time preview with editing on the bottom or right side. Tree view of document on left with D&D.
Professor 4	
Professor 5	Overall appearance, add video upload. add more detailed instructions. such as a step by step guide on the intro page.
Professor 6	I prefer to use my own tools. However, if this is a very stable application and will allow me to have more functionality I will consider using it.
Professor 7	Fix preview for text. Collecting photo and video data. Make getting exported data easier to convert to excel format.
Professor 8	The interface is intuitive, but not necessarily convenient.



	<p>It is very important to have templates available to base the labs off of.</p> <p>Make WYSIWYG - see what it looks like as you are going along. Model off of surveymonkey or google docs. Or, show something similar to your preview as developing.</p> <p>Would be very helpful to be able to copy and paste elements from other labs and then modify.</p> <p>How to do formatting in the lab text?</p> <p>Should be able to export data as excel or other usable format directly</p> <p>May be worthwhile considering a way to edited all at once (more like Latex/html coding). May want to have multiple ways to generate the common xml format. can you insert/reorder elements?</p> <p>Not everyone is going to want the same thing, so there is probably not 1 ideal solution. Having a standardized format and multiple ways to read/write to it may be useful.</p> <p>Why standalone vs web-based? Both could be OK, but should justify.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Question 11: What department are you a part of? (Open ended)

Professor 1	RBE
Professor 2	RBE
Professor 3	RBE
Professor 4	BIO
Professor 5	RBE
Professor 6	RBE
Professor 7	BIO
Professor 8	RBE

## Appendix H: XML Standard

- The purpose of this document is to outline general XML formatting rules for the Technology in Education IQP for the 2011-2012 academic year. This document will give a set standard for formatting and standard tags. Formatting will be unchangeable with in XML, but tags will be expandable
- Definitions
  - XML - a tag based language for machines and programs to interpret data from. This will be primarily generated by programs for programs.
  - Tags – a Tags are portions of the XML code that can represent elements of the application that can be made from the xml file.
  - App – short for application. An app is in this context run on an ipone/ipad and generated from XML code as described in this document

### Tag names, purposes, and arguments

NAME	Description	Attributes	Contains (TAGS)
LAB	the app tag will define the whole application		Author, version, title, icon, content, entry
ENTRY	the entry tag will define a section of the lab that is viewable by a tab in the main application		Text, picture, video
TEXT	the content tag will hold text information NOT in html format	length(lines of text)	
TEXTINPUT	text input	length(lines of text) question (text of question)	blank
VIDEO	the video tag will hold a reference to a video to be played MAKE SURE TO USE THE EMBEDDED LINK	height, width, source	
PICTURE	the picture tag will hold a picture reference to an image	height, width, source	
VERSION	The version tag will hold a date identifying when a tag was created or edited last	Date	
ICON	The icon tag will hold a picture reference to be used elsewhere	Source	
TABLE	package for ROW	Rows, columns	

ROW	Package for COLUMN		
COLUMN	each bit of data. If blank write @BLANK attribute boolean editable		

1. Formatting
2. General formatting rules
3. All apps will begin with APP tag immediately containing the VERSION tag and ICON tag.
4. The app will consist of entries defined by the ENTRY tag.
5. sample code

<TEXT length="1">This is some sample text</TEXT>

<IMAGE height="100" width="100">  
<http://a4.mzstatic.com/us/r1000/062/Purple/db/51/2e/mzl.edqotirc.100x100-75.png>  
 </IMAGE>

<VIDEO height="100" width="100">  
 youtube.com  
 </VIDEO>

<TABLE rows="3" columns="2">  
 <ROW>  
 <COLUMN>data1</COLUMN>  
 <COLUMN>data2</COLUMN>  
 </ROW>  
 <ROW>  
 <COLUMN>data3</COLUMN>  
 <COLUMN>data4</COLUMN>  
 <COLUMN>data5</COLUMN>  
 <COLUMN>data6</COLUMN>  
 </ROW>  
 <ROW>  
 <COLUMN>data7</COLUMN>  
 <COLUMN>data8</COLUMN>  
 <COLUMN>data9</COLUMN>  
 <COLUMN>data10</COLUMN>  
 </ROW>  
 </TABLE>

## Appendix I: App Source Code

### main.m

```
#import <UIKit/UIKit.h>

int main(int argc, char *argv[])
{
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

### AppDelegate.h

```
#import <UIKit/UIKit.h>

#import "ParseOperation.h"

extern NSString *AppDataDownloadCompleted;

@class ContentController;

@interface AppDelegate : NSObject <UIApplicationDelegate, ParseOperationDelegate>
{
    UIWindow                                *window;

    // the list of apps shared with "AppListViewController"
    NSMutableArray                          *appRecords;

    // the queue to run our "ParseOperation"
    NSOperationQueue                        *queue;

    // RSS feed network connection to the App Store
    NSURLConnection                        *appListFeedConnection;
    NSMutableData                           *appListData;

    ContentController                       *contentController;
}

@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) NSMutableArray *appRecords;
@property (nonatomic, retain) NSOperationQueue *queue;
@property (nonatomic, retain) NSURLConnection *appListFeedConnection;
@property (nonatomic, retain) NSMutableData *appListData;
@property (nonatomic, retain) IBOutlet ContentController *contentController;

@end
```

### AppDelegate.m

```
#import "AppDelegate.h"
#import "ParseOperation.h"
#import "AppRecord.h"

#import "ContentController.h"
```

```

// This framework was imported so we could use the kCFURLErrorNotConnectedToInternet error
code.
#import <CFNetwork/CFNetwork.h>

static NSString *const TopPaidAppsFeed =
    @"http://users.wpi.edu/~enniocd/bc8.xml";

NSString *AppDataDownloadCompleted = @"lab downloaded";

@implementation AppDelegate

@synthesize window, appRecords, queue, appListFeedConnection, appListData, contentController;

#pragma mark -

// -----
//   applicationDidFinishLaunching:application
// -----
- (void)applicationDidFinishLaunching:(UIApplication *)application
{
    // load the content controller object for Pad-based devices
    [[NSBundle mainBundle] loadNibNamed:@"Content_iPad" owner:self options:nil];

    [self.window addSubview:self.contentController.view];
    [window makeKeyAndVisible];

    // // Initialize the array of app records and pass a reference to that list to our root
    // view controller
    //   self.appRecords = [NSMutableArray array];

    NSURLRequest *urlRequest = [NSURLRequest requestWithURL:[NSURL
    URLWithString:TopPaidAppsFeed]];
    self.appListFeedConnection = [[NSURLConnection alloc] initWithRequest:urlRequest
    delegate:self] autorelease];

    NSAssert(self.appListFeedConnection != nil, @"Failure to create URL connection.");

    // show in the status bar that network activity is starting
    [UIApplication sharedApplication].networkActivityIndicatorVisible = YES;
}

// -----
//   handleLoadedApps:loadedApps
// -----
- (void)handleLoadedApps:(NSArray *)loadedApps
{
    [self.appRecords addObjectsFromArray:loadedApps];

    // tell our interested view controller reload its data, now that parsing has completed
    [[NSNotificationCenter defaultCenter] postNotificationName:AppDataDownloadCompleted
    object:loadedApps];
}

// -----
//   didFinishParsing:appList
// -----
- (void)didFinishParsing:(NSArray *)appList
{
    [self performSelectorOnMainThread:@selector(handleLoadedApps:) withObject:appList
    waitUntilDone:NO];

    self.queue = nil; // we are finished with the queue and our ParseOperation
}

- (void)parseErrorOccurred:(NSError *)error

```

```

{
    [self performSelectorOnMainThread:@selector(handleError:) withObject:error
waitUntilDone:NO];
}

#pragma mark -
#pragma mark NSURLConnection delegate methods

// -----
//   handleError:error
// -----
- (void)handleError:(NSError *)error
{
    NSString *errorMessage = [error localizedDescription];
    UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Cannot Show Top Paid Apps"
        message:errorMessage
        delegate:nil
        cancelButtonTitle:@"OK"
        otherButtonTitles:nil];
    [alertView show];
    [alertView release];
}

// The following are delegate methods for NSURLConnection. Similar to callback functions,
// this is how
// the connection object, which is working in the background, can asynchronously communicate
// back to
// its delegate on the thread from which it was started - in this case, the main thread.
//
// -----
//   connection:didReceiveResponse:response
// -----
- (void)connection:(NSURLConnection *)connection didReceiveResponse:(NSURLResponse *)response
{
    self.appListData = [NSMutableData data];    // start off with new data
}

// -----
//   connection:didReceiveData:data
// -----
- (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data
{
    [appListData appendData:data]; // append incoming data
}

// -----
//   connection:didFailWithError:error
// -----
- (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error
{
    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;

    if ([error code] == kCFURLErrorNotConnectedToInternet)
    {
        // if we can identify the error, we can present a more precise message to the user.
        NSDictionary *userInfo = [NSDictionary dictionaryWithObject:@"No Connection Error"
            forKey:NSLocalizedStringKey];
        NSError *noConnectionError = [NSError errorWithDomain:NSCocoaErrorDomain
            code:kCFURLErrorNotConnectedToInternet
            userInfo:userInfo];
        [self handleError:noConnectionError];
    }
}

```

```

    }
    else
    {
        // otherwise handle the error generically
        [self handleError:error];
    }
}
self.appListFeedConnection = nil; // release our connection
}

// -----
// connectionDidFinishLoading:connection
// -----
- (void)connectionDidFinishLoading:(NSURLConnection *)connection
{
    self.appListFeedConnection = nil; // release our connection

    [UIApplication sharedApplication].networkActivityIndicatorVisible = NO;

    // create the queue to run our ParseOperation
    self.queue = [[NSOperationQueue alloc] init];

    ParseOperation *parser = [[ParseOperation alloc] initWithData:appListData delegate:self];

    [queue addOperation:parser];

    [parser release];

    // ownership of appListData has been transferred to the parse operation
    // and should no longer be referenced in this thread
    self.appListData = nil;
}

- (void)dealloc
{
    [appRecords release];

    [appListFeedConnection release];
    [appListData release];

    [viewController release];

    [window release];

    [super dealloc];
}
@end

```

## ContentController.h

```

#import <Foundation/Foundation.h>
#import "AppRecord.h"

@interface ContentController : NSObject
{
    NSArray *detailItem;
}

@property (nonatomic, retain) NSArray *detailItem;

- (UIView *)view;

@end

```

## ContentController.m

```
#import "ContentController.h"
@implementation ContentController
@synthesize detailItem;
- (void)dealloc
{
    [detailItem release];
    [super dealloc];
}

- (UIView *)view
{
    return nil;
}
@end
```

## ContentController\_iPad.h

```
#import <Foundation/Foundation.h>

#import "ContentController.h"
#import <MessageUI/MessageUI.h>

@class DetailViewController;
@class AppListViewController;

@interface ContentController_iPad : ContentController <UIPopoverControllerDelegate,
UISplitViewControllerDelegate , MFMailComposeViewControllerDelegate>
{
    UISplitViewController *splitViewController;

    UINavigationController *navigationController; // contains our MasterViewController
    (UITableViewController)
    AppListViewController *masterViewController;
    DetailViewController *detailViewController;
    IBOutlet NSMutableArray *tableArray;
    IBOutlet UITextField *btext;
    UIPopoverController *popoverController;
    IBOutlet UIScrollView *scrollView1;
    IBOutlet UITextView *atext;
    IBOutlet UITextView *ctext;
    IBOutlet UIView *testV;
    IBOutlet UILabel *aline;
    IBOutlet NSArray *Current;
}

- (void)drawScreen:(NSArray *)newDetailItem;

@property (nonatomic, retain) IBOutlet UISplitViewController *splitViewController;
@property (nonatomic, retain) UIView *testV;
@property (nonatomic, retain) IBOutlet UINavigationController *navigationController;
@property (nonatomic, retain) UILabel *aline;
@property (nonatomic, retain) NSArray *Current;

@property (nonatomic, retain) IBOutlet AppListViewController *masterViewController;
@property (nonatomic, retain) NSMutableArray *tableArray;
@property (nonatomic, retain) IBOutlet DetailViewController *detailViewController;
@property (nonatomic, retain) UITextView *atext;
@property (nonatomic, retain) UITextView *ctext;
@property (nonatomic, retain) UIPopoverController *popoverController;
@property (nonatomic, retain) UIScrollView *scrollView1;
```



```
- (UIView *)view;
```

```
@end
```

## ContentController\_iPad.m

```
#import "ContentController_iPad.h"
#import "DetailViewController.h"
#import "AppListViewController.h"
#import "AppDelegate.h"
#import "DataClass.h"

static NSString *buttonTitle = @"Top Paid Apps";

@implementation ContentController_iPad

@synthesize splitViewController, navigationController, Current, masterViewController,
popoverController, detailViewController, scrollView1, tableArray, testV, aline, ctext;

- (void)mailComposeController:(MFMailComposeViewController *)controller
    didFinishWithResult:(MFMailComposeResult)result
    error:(NSError *)error {
    [self dismissModalViewControllerAnimated:YES];
}

- (void)awakeFromNib
{
    self.splitViewController.delegate = self;
    self.masterViewController.contentController = self;
    // configure the master popover button:
    //
    // In portrait mode (when the master view is not being displayed by the split view
    controller)
    // add a bar button item to display the master view in a popover; in landscape mode,
    remove the button.
    //
    if (UIInterfaceOrientationIsPortrait([[UIApplication sharedApplication]
statusBarOrientation]))
    {
        UIBarButtonItem *popoverMasterViewControllerBarButtonItem =
        [[UIBarItem alloc] initWithTitle:buttonTitle
        style:UIBarButtonItemStylePlain
        target:self

action:@selector(presentMasterInPopoverFromBarButtonItem:)];
        [self.detailViewController.navBar.topItem
setLeftBarButtonItem:popoverMasterViewControllerBarButtonItem animated:YES];
        [popoverMasterViewControllerBarButtonItem release];
    }
    else
    {
        [self.detailViewController.navBar.topItem setLeftBarButtonItem:nil animated:YES];
    }
}

- (IBAction)presentMasterInPopoverFromBarButtonItem:(UIBarButtonItem *)barButtonItem
{
    if (popoverController == nil)
    {
        Class cls = NSClassFromString(@"UIPopoverController");
        if (cls != nil)
        {
            UIPopoverController *aPopoverController =
            [[cls alloc] initWithContentViewController:self.masterViewController];
            self.popoverController = aPopoverController;

            [aPopoverController release];
        }
    }
}
```

```

        [popoverController presentPopoverFromBarButtonItem:barButtonItem
                          permittedArrowDirections:UIPopoverArrowDirectionUp
                          animated:YES];
    }
}

- (void)dealloc
{
    [splitViewController release];
    [masterViewController release];
    [popoverController release];
    [navigationController release];
    [detailViewController.scrollView10.subviews release];
    [detailViewController release];

    [super dealloc];
}

// when setting the detail item, update the view and dismiss the popover controller if it's
// showing

//put scrollview here
//
//
//

- (void)setDetailItem:(NSArray *)newDetailItem
{
    Current = newDetailItem;
    float height = 0;
    float heightTotal = 0;
    [self.detailViewController.scrollView10 setContentSize:CGSizeMake(700, 10000)];
    self.detailViewController.scrollView10.backgroundColor = [UIColor whiteColor];
    UIView *clearScree = [[UIView alloc] initWithFrame:CGRectMake(0, 0, 700, 10000) ];
    clearScree.backgroundColor = [UIColor whiteColor];
    [self.detailViewController.scrollView10 addSubview:clearScree];

    [self.detailViewController.scrollView10 setContentSize:CGSizeMake(700, heightTotal+250)];
    for (int countI = 0; countI < newDetailItem.count ; countI++) {
        if([[newDetailItem objectAtIndex:indexI] isEqualToString:@"Export Data"]){

            //          NSMutableArray *mainArray = [[NSMutableArray alloc] init];
            //          NSMutableDictionary *tableDic = [[NSMutableDictionary alloc] init];
            //          NSMutableDictionary *tableDic2 = [[NSMutableDictionary alloc]
init];
            //
            //          NSMutableDictionary *textDiction= [[NSMutableDictionary alloc]
init];
            //          NSMutableArray *tableArray = [[NSMutableArray alloc]
initWithObjects:@"shit (0,0)","shit (0,1)","shit (0,2)","shit (0,3)",
            //          @"shit (1,0)","shit (1,1)","shit
(1,2)","shit (1,3)",
            //          @"shit (2,0)","shit (2,1)","shit
(2,2)","shit (2,3)",nil];
            //          NSMutableArray *tableArray2 = [[NSMutableArray alloc]
initWithObjects:@"blah (0,0)","blah (0,1)","blah (0,2)","blah (0,3)",
            //          @"blah (1,0)","blah (1,1)","blah
(1,2)","blah (1,3)",
            //          @"blah (2,0)","blah (2,1)","blah
(2,2)","blah (2,3)",nil];
            //          [tableDic setObject:@"TABLE" forKey:@"type"];
            //          [tableDic setObject:@"4" forKey:@"col"];
            //          [tableDic setObject:@"3" forKey:@"row"];
            //          [tableDic setObject:@"title of lab" forKey:@"title"];
            //          [tableDic setObject:tableArray forKey:@"table"];
            //          [mainArray addObject:tableDic];
            //          [tableDic2 setObject:@"TABLE" forKey:@"type"];
            //          [tableDic2 setObject:@"4" forKey:@"col"];
            //          [tableDic2 setObject:@"3" forKey:@"row"];

```

```

//          [tableDic2 setObject:@"title2of lab" forKey:@"title"];
//          [tableDic2 setObject:tableArray2 forKey:@"table"];
//          [mainArray addObject:tableDic2];
//          [textDiction setObject:@"TEXTINPUT" forKey:@"type"];
//          [textDiction setObject:@"bkdsjfkdsjf?" forKey:@"question"];
//          [textDiction setObject:@"blah blah blah" forKey:@"text"];
//          [mainArray addObject:textDiction];
//////
//          if ([[mainArray objectAtIndex:1] objectForKey:@"type"]
isEqualToString:@"TABLE"]) {
//              int a = 0;
//          }
//          NSInteger *mainCount = [mainArray count];
//          NSString *dkfjdkj = [[[mainArray objectAtIndex:1] objectForKey:@"table"]
objectAtIndex:3]; //returns data
//          NSInteger *blah = [[[mainArray objectAtIndex:0] objectForKey:@"col"]
intValue]; //retrns #of col

NSMutableString *MyString;
NSMutableString *OpenText;
NSMutableString *CloseText;
NSMutableString *OpenTable;
NSMutableString *CloseTable;
NSMutableString *OpenRow;
NSMutableString *CloseRow;
NSMutableString *OpenCol;
NSMutableString *CloseCol;
NSMutableString *OpenResults;
NSMutableString *CloseResults;

MyString = [[NSMutableString alloc] init];
[MyString setString:@""];

////USE THIS FOR XML
//          OpenText = [[NSMutableString alloc] init];
//          [OpenText setString:@"<TEXT>"];
//          CloseText = [[NSMutableString alloc] init];
//          [CloseText setString:@"</TEXT>\n"];
//          OpenTable = [[NSMutableString alloc] init];
//          [OpenTable setString:@"<TABLE>\n"];
//          CloseTable = [[NSMutableString alloc] init];
//          [CloseTable setString:@"</TABLE>\n"];
//          OpenRow = [[NSMutableString alloc] init];
//          [OpenRow setString:@"<ROW>\n"];
//          CloseRow = [[NSMutableString alloc] init];
//          [CloseRow setString:@"</ROW>\n"];
//          OpenCol = [[NSMutableString alloc] init];
//          [OpenCol setString:@"<COLUMN>"];
//          CloseCol = [[NSMutableString alloc] init];
//          [CloseCol setString:@"</COLUMN>\n"];
//          OpenResults = [[NSMutableString alloc] init];
//          [OpenResults setString:@"<RESULTS>"];
//          CloseResults = [[NSMutableString alloc] init];
//          [CloseResults setString:@"</RESULTS>\n"];

//USE THIS FOR HTML
OpenText = [[NSMutableString alloc] init];
[OpenText setString:@"\nText Question and Answer:"];
CloseText = [[NSMutableString alloc] init];
[CloseText setString:@"\n"];
OpenTable = [[NSMutableString alloc] init];
[OpenTable setString:@"\nChart of Data(To import data into xml read instructions
at the bottom of the page)\n\nBeginning of Table\n\n"];
CloseTable = [[NSMutableString alloc] init];
[CloseTable setString:@"\nEnd of Table \n"];
OpenRow = [[NSMutableString alloc] init];

```

```

[OpenRow setString:@""];
CloseRow = [[NSMutableString alloc] init];
[CloseRow setString:@"\n"];
OpenCol = [[NSMutableString alloc] init];
[OpenCol setString:@""];
CloseCol = [[NSMutableString alloc] init];
[CloseCol setString:@";"];
OpenResults = [[NSMutableString alloc] init];
[OpenResults setString:@""];
CloseResults = [[NSMutableString alloc] init];
[CloseResults setString:@"\n\nyou can copy the content here (CTRL+C), go to your
excel sheet and use CTRL+V to paste the data and then click on the paste icon that appears at
the bottom of the pasted cell and select "use text import wizard" option. This will launch
the mildly powerful text import wizard of excel using which you can convert copied text to
columns by defining some simple parsing conditions."];

```

```

int count =0;
[MyString appendString:OpenResults];
for(int a = 1; a <([[newDetailItem count]-4]; a++){
    for (int i=1; i< [[newDetailItem objectAtIndex:a] count]; i++){
        NSString *kdjfdkjf = [[newDetailItem objectAtIndex:a] objectAtIndex:i ];
        if ([[newDetailItem objectAtIndex:a] objectAtIndex:i ]
isEqualToString:@"TABLE"] ) {
            //IF ITS A TABLE
            [MyString appendString:OpenTable];
            //for each row
            for(int x=0;x<[[[newDetailItem objectAtIndex:a] objectAtIndex:i+1 ]
objectForKey:@"rows"] intValue];x++){
                [MyString appendString:OpenRow];
                for(int y=0;y<[[[newDetailItem objectAtIndex:a]
objectAtIndex:i+1 ] objectForKey:@"columns"] intValue];y++){
                    [MyString appendString:OpenCol];
                    NSString *kdjfdkjf = [[[[newDetailItem objectAtIndex:a]
objectAtIndex:i+1 ] objectForKey:@"tableDataArray"] objectAtIndex:count]
objectForKey:@"value" ];
                    [MyString appendString:[[[[newDetailItem objectAtIndex:a]
objectAtIndex:i+1 ] objectForKey:@"tableDataArray"] objectAtIndex:count]
objectForKey:@"value" ]];
                    //[[[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"tableDataArray"] objectAtIndex:count] objectForKey:@"value"
                    count++;
                    [MyString appendString:CloseCol];
                }
                [MyString appendString:CloseRow];
            }
            [MyString appendString:CloseTable];
            count=0;
        }
        if ([[newDetailItem objectAtIndex:a] objectAtIndex:i ]
isEqualToString:@"TEXTINPUT" ) {
            //IF ITS A TEXT INPUT
            [MyString appendString:OpenText];
            [MyString appendString:[[newDetailItem objectAtIndex:a]
objectAtIndex:i+1 ] objectForKey:@"question"]];
            [MyString appendString:@": "];
            [MyString appendString:CloseText];
        }
        i++;
    }
}
[MyString appendString:CloseResults];

//[MyString appendString:OpenText];
//[MyString appendString:CloseText];

```

```

MFMailComposeViewController *picker = [[MFMailComposeViewController alloc] init]
autorelease];

```

```

        [picker setSubject:@"Bio lab Data"];
        [picker setToRecipients:[NSArray array]];
        // [picker setMessageBody:@"This is test text: " isHTML:NO];
        [picker setMessageBody:MyString isHTML:NO];
        [picker setMailComposeDelegate:self];
        countI =100;
        [self.detailViewController presentViewController:picker animated:YES];
    }
    else if([[newDetailItem objectAtIndex:countI] isEqualToString:@"TEXT"]){
        float width = 700;//[[UIScreen mainScreen] bounds].size.width;
        NSString *heighta = [[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"length"];
        height = [[newDetailItem objectAtIndex:(countI+1)] objectForKey:@"inputTxt"]
length]/3;
        atext = [[[UITextView alloc] initWithFrame:CGRectMake(0, heightTotal,
self.detailViewController.view.frame.size.width /*[[UIScreen mainScreen]
bounds].size.width*/, height)] autorelease];
        atext.textColor = [UIColor blackColor];
        atext.font = [UIFont fontWithName:@"Arial" size:18.0];
        // atext.delegate = self;
        atext.backgroundColor = [UIColor whiteColor];
        atext.text = [[newDetailItem objectAtIndex:(countI+1)] objectForKey:@"inputTxt"];
        atext.returnKeyType = UIReturnKeyDefault;
        atext.keyboardType = UIKeyboardTypeDefault; // use the default type input method
(entire keyboard)
        atext.scrollEnabled = FALSE;
        atext.editable = FALSE;
        atext.dataDetectorTypes = UIDataDetectorTypeAll;
        // this will cause automatic vertical resize when the table is resized
        atext.autoresizingMask = UIViewAutoresizingFlexibleHeight;

        // note: for UITextView, if you don't like autocompletion while typing use:
        // myTextView.autocorrectionType = UITextAutocorrectionTypeNo;

        [self.detailViewController.scrollView10 addSubview:atext];
        countI++;
        heightTotal = heightTotal + height + 50;
    }

    else if([[newDetailItem objectAtIndex:countI] isEqualToString:@"IMAGE"]){
        NSData *receivedData = [[NSData dataWithContentsOfURL:[NSURL
URLWithString:[[newDetailItem objectAtIndex:(countI+1)] objectForKey:@"imageLink"]]] retain];
        NSString *dsd = [[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"imageLink"];
        UIImage *imagea = [[UIImage alloc] initWithData:receivedData];
        UIImageView *img = [[UIImageView alloc] initWithImage:imagea];
        img.frame = CGRectMake(350-(imagea.size.width/2), heightTotal, imagea.size.width,
imagea.size.height+20);
        // [self.view addSubview:[UIImageView alloc] initWithImage:imagea];
        [self.detailViewController.scrollView10 addSubview:img];
        heightTotal = heightTotal + imagea.size.height + 50;

        countI++;
    }

    else if([[newDetailItem objectAtIndex:countI] isEqualToString:@"VIDEO"]){
        float videoHeight = [[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"height"] intValue];

        float videoWidth = [[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"width"] intValue];

        NSString *urlString = [[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"videoLink"];
        NSString *embedHTML = @"\
<html><head>\
<style type=\"text/css\">\

```

```

        body {\
        background-color: transparent;\
        color: white;\
        }\
    </style>\
</head><body style=\"margin:0\">\
<embed id=\"yt\" src=\"%@\" type=\"application/x-shockwave-flash\" \
width=\"%0.0f\" height=\"%0.0f\"></embed>\
</body></html>\";

    NSString *html = [NSString stringWithFormat:embedHTML, urlString, videoWidth,
videoHeight];
    UIWebView *videoView = [[UIWebView alloc] initWithFrame:CGRectMake((700-
videoWidth)/2, heightTotal, videoWidth, videoHeight)];
    [videoView loadHTMLString:html baseURL:nil];

    [self.detailViewController.scrollView10 addSubview:videoView];
    [videoView release];
    heightTotal = heightTotal + videoHeight + 50;
    countI++;
}
else if([[newDetailItem objectAtIndex:index:countI] isEqualToString:@"TEXTINPUT"]){
    NSString *heightA = [[newDetailItem objectAtIndex:index:(countI+1)]
objectForKey:@"length"];
    int height = [heightA intValue];

    CGRect frame1 = CGRectMake(0,10, 700, height*25);
    ctext = [[[UITextView alloc] initWithFrame:CGRectMake(10, heightTotal,
680,height*25)] autorelease];
    ctext.textColor = [UIColor blackColor];
    ctext.font = [UIFont fontWithName:@"Arial" size:18.0];
    ctext.backgroundColor = [UIColor colorWithRed:255/255.0 green:251/255.0
blue:204/255.0 alpha:1];
    ctext.returnKeyType = UIReturnKeyDefault;
    ctext.keyboardType = UIKeyboardTypeDefault; // use the default type input method
(entire keyboard)
    ctext.scrollEnabled = FALSE;
    ctext.editable = TRUE;
    // this will cause automatic vertical resize when the table is resized
    ctext.autoresizingMask = UIViewAutoresizingFlexibleHeight;
    ctext.text = [[newDetailItem objectAtIndex:index:(countI+1)] objectForKey:@"question"];
    NSString *dfd = [[newDetailItem objectAtIndex:index:(countI+1)] objectForKey:@"uid"];
    ctext.tag = [dfd intValue];
    ctext.delegate = self;

    [self.detailViewController.scrollView10 addSubview:ctext];
    heightTotal = heightTotal + height*25+10;
    countI++;
}

else if([[newDetailItem objectAtIndex:index:countI] isEqualToString:@"TABLE"]){
    NSString *row = [[newDetailItem objectAtIndex:index:(countI+1)] objectForKey:@"rows"];
    NSString *column = [[newDetailItem objectAtIndex:index:(countI+1)]
objectForKey:@"columns"];
    int numC = [column intValue];
    int numR = [row intValue];
    float tableHeight = numR*40;
    float tableWidth = numC*100;
    float totalWidth = tableWidth + 20;
    float totalHeight = tableHeight+20;
    float lineThickness = 2;
    int cells = numC*numR;
    testV = [[[UIView alloc] initWithFrame:CGRectMake((700-totalWidth)/2,
heightTotal, totalWidth, totalHeight+100)]autorelease];
    testV.backgroundColor = [UIColor whiteColor];

    //     aline = [[[UIView alloc]
initWithFrame:CGRectMake((self.testV.frame.size.width/2)+(tableWidth/2), 10, lineThickness,
tableHeight)] autorelease];
    //     aline.backgroundColor = [UIColor blackColor];

```

```

// [testV addSubview:aline];
for (int i = 0; i < numC+1; i++)
{
    aline = [[UIView alloc]
initWithFrame:CGRectMake(((self.testV.frame.size.width)/2)-(tableWidth/2))+i*100-1, 60,
lineThickness, tableHeight)] autorelease];
    aline.backgroundColor = [UIColor blackColor];
    [testV addSubview:aline];
};
for (int i = 0; i < numR+1; i++)
{
    aline = [[UIView alloc] initWithFrame:CGRectMake(10,
((((self.testV.frame.size.height))/2)-(tableHeight/2))+i*40-1), tableWidth,
lineThickness)] autorelease];
    aline.backgroundColor = [UIColor blackColor];
    [testV addSubview:aline];
}

    aline = [[UIView alloc]
initWithFrame:CGRectMake(((self.testV.frame.size.width)/2)-(tableWidth/2))-1, 10,
lineThickness, 50)] autorelease];
    aline.backgroundColor = [UIColor blackColor];
    [testV addSubview:aline];
    aline = [[UIView alloc]
initWithFrame:CGRectMake(((self.testV.frame.size.width)/2)+(tableWidth/2))-1, 10,
lineThickness, 50)] autorelease];
    aline.backgroundColor = [UIColor blackColor];
    [testV addSubview:aline];
    aline = [[UIView alloc]
initWithFrame:CGRectMake(((self.testV.frame.size.width)/2)-(tableWidth/2))-1, 10,
tableWidth, lineThickness)] autorelease];
    aline.backgroundColor = [UIColor blackColor];
    [testV addSubview:aline];

    tableArray = [[NSMutableArray alloc] init];

    for(int ce = 0; ce < cells; ce++)
    {
        [tableArray addObject:@"blank"];
    }
    int count = 0;
    for(int r = 0; r < numR; r++)
    {
        for(int c = 0; c < numC; c++)
        {
            NSInteger cell = c*r;
            CGRect frame = CGRectMake(((self.testV.frame.size.width)/2-
(tableWidth/2)+1)+(c*100), ((self.testV.frame.size.height)/2)-(tableHeight/2)+1)+(r*40)+5, 98,
28);

            btext = [[UITextField alloc] initWithFrame:frame];
            btext.textColor = [UIColor blackColor];
            btext.borderStyle = UITextBorderStyleNone;
            btext.font = [UIFont fontWithName:@"Arial" size:12.0];
            btext.delegate = self;
            btext.textAlignment = UITextAlignmentCenter;
            btext.backgroundColor = [UIColor whiteColor];
            // btext.placeholder = @"in";//[NSString
stringWithFormat:@"%d", (c*r)];
            // [[self.workingEntry objectAtIndex:(self.workingEntry count)-
1] objectForKey:@"tabledataArray"] removeObjectForKey:@"UID"];
            if (![[[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"tabledataArray"] objectAtIndex:count] objectForKey:@"value" ]
isEqualToString:@"@BLANK"]) {
                btext.text = [[[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"tabledataArray"] objectAtIndex:count] objectForKey:@"value"];
                NSString *dkjfkdfj= [[[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"tabledataArray"] objectAtIndex:count] objectForKey:@"value"];

```



```

        btext.enabled = FALSE;
    }

    btext.returnKeyType = UIReturnKeyDefault;
    NSString *dfd = [[[newDetailItem objectAtIndex:(countI+1)]
objectForKey:@"tableDataArray" objectAtIndex:(countI+1)] objectForKey:@"UID"];
    btext.tag = [dfd intValue];//= [[[newDetailItem
objectAtIndex:(countI+1)] objectForKey:@"tableDataArray" objectAtIndex:(countI+1)]
objectForKey:@"UID"];//count;

    btext.keyboardType = UIKeyboardTypeDefault; // use the default type input
method (entire keyboard)
    // this will cause automatic vertical resize when the table is resized
    // myTextView.autocorrectionType = UITextAutocorrectionTypeNo;
    btext.delegate = self;
    count++;

    [testV addSubview:btext];
}
}

CGRect frame = CGRectMake(((self.testV.frame.size.width/2)-(tableWidth/2)+2),
((self.testV.frame.size.height/2)-(tableHeight/2)+1)-40, tableWidth-20, 38);
btext = [[UITextField alloc] initWithFrame:frame];
btext.textColor = [UIColor blackColor];
btext.borderStyle = UITextBorderStyleNone;
btext.font = [UIFont fontWithName:@"Arial" size:20.0];
btext.delegate = self;
btext.textAlignment = UITextAlignmentCenter;
btext.backgroundColor = [UIColor whiteColor];
btext.enabled = FALSE;
btext.delegate = self;
btext.text = [[newDetailItem objectAtIndex:(countI+1)] objectForKey:@"TITLE"];
[testV addSubview:btext];
[testV setContentScaleFactor:1.5];
[self.detailViewController.scrollView10 addSubview:testV];
heightTotal = heightTotal + tableHeight + 100;
countI++;
}
}

[self.detailViewController.scrollView10 setContentSize:CGSizeMake(700, heightTotal+500)];
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
    return YES;
}
- (BOOL)textViewDidEndEditing:(UITextView *)textView
{
    [textView resignFirstResponder];
    NSUInteger cell = textView.tag;
    NSString *cellS = [NSString stringWithFormat:@"%d", cell];

    for (int countB = 1; countB < [Current count]; countB++) {
        NSString *skjfkdsdjf = [Current objectAtIndex:(countB)];
        if ([[Current objectAtIndex:(countB)] isEqualToString:@"TEXTINPUT"]) {
            NSString *skjfdkdsdjf = [[Current objectAtIndex:(countB+1)] objectForKey:@"uid"];
            NSString *skjfdkdsdjf = [[Current objectAtIndex:(countB+1)]
objectForKey:@"uid"];

            if ([[Current objectAtIndex:(countB+1)] objectForKey:@"uid"]
isEqualToString:cellS]) {
                [[Current objectAtIndex:(countB+1)] setObject:(textView.text)
forKey:@"question"];
                return YES;
            }
        }
    }
}

```



```

        }
        countB++;
    }

    return YES;
}

- (BOOL)textFieldDidEndEditing:(UITextField *)textField
{
    // the user pressed the "Done" button, so dismiss the keyboard
    [textField resignFirstResponder];
    NSUInteger cell = textField.tag;
    NSString *cellS = [NSString stringWithFormat:@"%d", cell];
    bool flag = TRUE;
    int countB = 0;
    for (int countB; countB < [Current count]; countB++) {
        if ([[Current objectAtIndex:index:countB] isEqualToString:@"TABLE"]){
            countB++;
            for(int i = 0; i < [[Current objectAtIndex:index:countB]
objectForKey:@"tabledataArray"] count]; i++) {
                if ([[Current objectAtIndex:index:(countB)] objectForKey:@"tabledataArray"]
objectAtIndex:i] objectForKey:@"UID" isEqualToString:cellS]){

                    //                      NSMutableDictionary *tempDic =
[[NSMutableDictionary alloc] init];
                    //                      [tempDic setObject:cell forKey:@"UID"];
                    //                      [tempDic setObject:(textField.text)
forKey:@"value"];
                    NSString *blah = [[Current objectAtIndex:(countB)]
objectForKey:@"tabledataArray"] objectAtIndex:i] objectForKey:@"value"];
                    [[[Current objectAtIndex:index:(countB)] objectForKey:@"tabledataArray"]
objectAtIndex:i] setObject:(textField.text) forKey:@"value"];
                    return YES;
                }
            }
        }
        if ([[Current objectAtIndex:index:countB] isEqualToString:@"TEXTINPUT"] && [[Current
objectAtIndex:countB] objectForKey:@"uid" ] ) {
            countB++;
            [[Current objectAtIndex:index:countB] setObject:(textField.text)
forKey:@"question"];
        }
    }

    //          [[[self.workingEntry objectAtIndex:[self.workingEntry count]-1]
objectForKey:@"tabledataArray"] objectForKey:@"value"] replaceObjectAtIndex:self.celltracker
withObject:trimmedString];
    //          [[[Current objectAtIndex:(textField.tag)] objectForKey:@"tabledataArray"]
objectAtIndex:countA] objectForKey:@"value"]

    ;
    //[[self.tableArray replaceObjectAtIndex:cell withObject:(textField.text)];
    // [SomeArrayName insertObject:SomeObjectName atIndex:12];

    return YES;
}

#pragma mark -
#pragma mark Split view support

- (UIView *)view
{
    return self.splitViewController.view;
}

- (void)splitViewController: (UISplitViewController *)svc
willHideViewController:(UIViewController *)aViewController
withBarButtonItem:(UIBarButtonItem *)barButtonItem

```

```

        forPopoverController:(UIPopoverController *)pc
    {
        barButtonItem.title = buttonTitle;
        [self.detailViewController.navBar.topItem setLeftBarButtonItem:barButtonItem
animated:NO];
        self.popoverController = pc;
    }

// called when the view is shown again in the split view, invalidating the button and popover
controller
//
- (void)splitViewController: (UISplitViewController *)svc
willShowViewController:(UIViewController *)aViewController
invalidatingBarButtonItem:(UIBarButtonItem *)barButtonItem
{
    [self.detailViewController.navBar.topItem setLeftBarButtonItem:nil animated:NO];
    self.popoverController = nil;
}

@end

```

## DetailViewController.h

```

#import <UIKit/UIKit.h>

@interface DetailViewController : UIViewController
{
    UINavigationController *navBar;

    UIImageView *appIcon;
    UILabel *appName;
    UILabel *appArtist;
    UILabel *appPrice;
    UILabel *appRights;
    UILabel *appReleaseDate;
    UITextView *appSummary;
    UIButton *appURLButton;
    NSURL *appURL;
    UIScrollView *scrollView10;
}

@property (nonatomic, retain) IBOutlet UINavigationController *navBar;
- (void)outputFile:(NSArray *)masterArray;

@property (nonatomic, retain) IBOutlet UIImageView *appIcon;
@property (nonatomic, retain) IBOutlet UILabel *appName;
@property (nonatomic, retain) IBOutlet UILabel *appArtist;
@property (nonatomic, retain) IBOutlet UILabel *appPrice;
@property (nonatomic, retain) IBOutlet UILabel *appRights;
@property (nonatomic, retain) IBOutlet UILabel *appReleaseDate;
@property (nonatomic, retain) IBOutlet UITextView *appSummary;
@property (nonatomic, retain) IBOutlet UIScrollView *scrollView10;
@property (nonatomic, retain) IBOutlet UIButton *appURLButton;
@property (nonatomic, retain) NSURL *appURL;

- (IBAction)appLinkAction:(id)sender;

@end

```

## DetailViewController.m

```

#import "DetailViewController.h"

```

```

#import "AppListViewController.h"
@implementation DetailViewController

@synthesize navBar, appIcon, appName, appArtist, appPrice, appRights, appReleaseDate,
appSummary, appURLButton, appURL, scrollView10;

- (void)dealloc
{
    [navBar release];
    [appIcon release];
    [appName release];
    [appArtist release];
    [appPrice release];
    [appRights release];
    [appReleaseDate release];
    [appSummary release];
    [appURL release];
    [appURLButton release];

    [super dealloc];
}

#pragma mark -
#pragma mark View lifecycle

// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor colorWithRed:197.0/255.0
                                                green:204.0/255.0
                                                blue:211.0/255.0
                                                alpha:1.0];

    // make the scroll view background a little lighter
    self.appSummary.backgroundColor = [UIColor colorWithRed:(197.0/255.0)*20
                                                green:(204.0/255.0)*20
                                                blue:(211.0/255.0)*20
                                                alpha:0.3];
}

- (void)outputFile:(NSArray *)masterArray{
    NSArray *skfjlsakfd = [[NSArray alloc] init];
    int a = 1;
}

- (void)viewDidUnload
{
    // release any retained subviews of the main view
    self.navBar = nil;
    self.appIcon = nil;
    self.appName = nil;
    self.appArtist = nil;
    self.appPrice = nil;
    self.appRights = nil;
    self.appReleaseDate = nil;
    self.appSummary = nil;
    self.appURLButton = nil;
    self.appURL = nil;
}

#pragma mark -
#pragma mark Rotation support

// Ensure that the view controller supports rotation and that the split view can therefore
show in

```

```

// both portrait and landscape.
//
- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
{
    return YES;
}

- (void)willRotateToInterfaceOrientation:(UIInterfaceOrientation)toInterfaceOrientation
duration:(NSTimeInterval)duration
{
    // no popouts for lanscape orientation (use the MasterViewController's table)
    //
    if (UIInterfaceOrientationIsLandscape(toInterfaceOrientation))
    {
        [self.navBar.topItem setLeftBarButtonItem:nil animated:NO];
    }
}

- (IBAction)appLinkAction:(id)sender
{
    [[UIApplication sharedApplication] openURL:self.appURL];
}

@end

```

## AppRecord.h

```

@interface AppRecord : NSObject
{
    NSString *appName;
    UIImage *appIcon;
    NSString *imageURLString;
    NSString *artist;

    NSString *appURLString;
    NSString *price;
    NSString *rights;
    NSString *releaseDate;
    NSMutableArray *sums;
    NSString *summary;
}

@property (nonatomic, retain) NSString *appName;
@property (nonatomic, retain) UIImage *appIcon;
@property (nonatomic, retain) NSString *artist;
@property (nonatomic, retain) NSString *imageURLString;

@property (nonatomic, retain) NSString *appURLString;
@property (nonatomic, retain) NSString *price;
@property (nonatomic, retain) NSString *rights;
@property (nonatomic, retain) NSString *releaseDate;
@property (nonatomic, retain) NSMutableArray *sums;
@property (nonatomic, retain) NSString *summary;

@end

```

## AppRecord.m

```

#import "AppRecord.h"

```

```

@implementation AppRecord

@synthesize appName;
@synthesize appIcon;
@synthesize imageURLString;
@synthesize artist;
@synthesize sums;
@synthesize appURLString;
@synthesize price;
@synthesize rights;
@synthesize releaseDate;
@synthesize summary;

- (void)dealloc
{
    [appName release];
    [appIcon release];
    [imageURLString release];
    [artist release];

    [appURLString release];
    [price release];
    [rights release];
    [releaseDate release];
    [summary release];

    [super dealloc];
}

@end

```

## ParseOperation.h

```

@class AppRecord;

@protocol ParseOperationDelegate;

@interface ParseOperation : NSOperation <NSXMLParserDelegate>
{
@private
    id <ParseOperationDelegate> delegate;

    NSData          *dataToParse;

    NSMutableArray  *workingArray;
    NSMutableArray  *workingEntry;
    NSMutableArray  *workingPropertyString;
    NSArray         *elementsToParse;
    NSMutableArray  *tagFlow;
    BOOL            storingCharacterData;
    NSString        *trackingCategoryName;
    NSString        *trackingReleaseDate;

    NSString *labTitle; //name of lab TITLE
    NSString *entryTab;
    NSString *tabTitle;
    NSString *txtView; //text to display in a view
    NSString *VideoView;
    NSString *tableView;
    int cellTracker;
    NSMutableArray *LAB;
}

- (id)initWithData:(NSData *)data delegate:(id <ParseOperationDelegate>)theDelegate;

```

```
@end
```

```
@protocol ParseOperationDelegate  
- (void)didFinishParsing:(NSArray *)appList;  
- (void)parseErrorOccurred:(NSError *)error;  
@end
```

## ParseOperation.m

```
#import "ParseOperation.h"  
#import "AppRecord.h"  
  
// string contants found in the RSS feed  
static NSString *kEntryStr = @"ENTRY"; // marker for each app entry  
//dictionary keys  
static NSString *ktxtView = @"TEXT";  
static NSString *kTxtViewLength = @"length";  
static NSString *kimageURL = @"imgURL";  
static NSString *kimageHeight = @"imgH";  
static NSString *kimageWidth = @"imgW";  
static NSString *ktitle = @"TITLE";  
static NSString *kvideo = @"VIDEO";  
static NSString *kvideoHeight = @"height";  
static NSString *kvideoWidth = @"width";  
static NSString *ktable = @"TABLE";  
static NSString *ktableRow = @"ROW";  
static NSString *ktableColumn = @"COLUMN";  
static NSString *ktableRowCount = @"rows";  
static NSString *ktableColumnsCount = @"columns";  
static NSString *ktableTitle = @"title";  
  
@interface ParseOperation ()  
@property (nonatomic, assign) id <ParseOperationDelegate> delegate;  
@property (nonatomic, retain) NSData *dataToParse;  
@property (nonatomic, retain) NSMutableArray *workingArray;  
@property (nonatomic, retain) NSMutableArray *workingEntry;  
@property (nonatomic, retain) NSMutableString *workingPropertyString;  
@property (nonatomic, retain) NSArray *elementsToParse;  
@property (nonatomic, retain) NSMutableArray *tagFlow;  
@property (nonatomic, assign) BOOL storingCharacterData;  
@property (nonatomic, assign) NSString *trackingCategoryName;  
@property (nonatomic, assign) NSString *trackingReleaseDate;  
@property (nonatomic, assign) int celltracker;  
@end  
  
@implementation ParseOperation  
  
@synthesize delegate, dataToParse, workingArray, workingEntry, workingPropertyString,  
elementsToParse,  
storingCharacterData, trackingCategoryName, trackingReleaseDate, tagFlow,  
celltracker;  
  
/*dictionary definitions  
summary  
[NSMutableDictionary dictionaryWithObjectsAndKeys:@"textcontent", summary, 00, summaryL,  
nil];  
image  
[NSMutableDictionary dictionaryWithObjectsAndKeys:@"www.url.com", imageURL, 00, imageHeight,  
00, imageWidth, nil];  
*/  
  
- (id)initWithData:(NSData *)data delegate:(id <ParseOperationDelegate>)theDelegate  
{  
    self = [super init];  
    if (self != nil)
```

```

    {
        self.dataToParse = data;
        self.delegate = theDelegate;

        self.tagFlow = [[NSMutableArray alloc] init];
    }

    return self;
}

// -----
//   dealloc:
// -----
- (void)dealloc
{
    [dataToParse release];
    [workingEntry release];
    [workingPropertyString release];
    [workingArray release];
    [trackingCategoryName release];
    [trackingReleaseDate release];

    [super dealloc];
}

- (void)main
{
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];

    self.workingArray = [NSMutableArray array];
    self.workingPropertyString = [NSMutableString string];

    storingCharacterData = true;
    NSXMLParser *parser = [[NSXMLParser alloc] initWithData:dataToParse];
    [parser setDelegate:self];
    [parser parse];

    if (![self isCancelled])
    {
        [self.delegate didFinishParsing:self.workingArray];
    }

    self.workingArray = nil;
    self.workingPropertyString = nil;
    self.dataToParse = nil;
    self.trackingCategoryName = nil;
    self.trackingReleaseDate = nil;
    self.celltracker = nil;
    [parser release];

    [pool release];
}

#pragma mark -
#pragma mark RSS processing
//did start used for when to get the attribute data
- (void)parser:(NSXMLParser *)parser didStartElement:(NSString *)elementName
    namespaceURI:(NSString *)namespaceURI
    qualifiedName:(NSString *)qName
    attributes:(NSDictionary *)attributeDict
{
    if ([elementName isEqualToString:kEntryStr])
    {
        self.workingEntry = [[[NSMutableArray alloc] init] autorelease];
    }
    if ([elementName isEqualToString:@"EXPORT"]) {
        [self.workingEntry addObject:@"Export Data"];
        [workingPropertyString setString:@""];
    }
}

```

```

    }
    if ([elementName isEqualToString:ktxtView])
    {
        NSMutableDictionary *txtViewD = [[NSMutableDictionary alloc] init];
        [txtViewD setObject:[attributeDict objectForKey:kTxtViewLength]
        forKey:kTxtViewLength];
        NSString *bIs = [txtViewD objectForKey:kTxtViewLength];
        [self.workingEntry addObject:@"TEXT"];
        [self.workingEntry addObject:txtViewD];
        [workingPropertyString setString:@""];
    }
    if ([elementName isEqualToString:ktable])
    {
        self.celltracker=0;
        NSMutableDictionary *tableViewD = [[NSMutableDictionary alloc] init];
        [tableViewD setObject:[attributeDict objectForKey:ktableRowCount]
        forKey:ktableRowCount];
        [tableViewD setObject:[attributeDict objectForKey:ktableColumnsCount]
        forKey:ktableColumnsCount];
        [tableViewD setObject:[attributeDict objectForKey:@"title"] forKey:ktitle];
        int rD = [[attributeDict objectForKey:ktableRowCount] intValue];
        int cD = [[attributeDict objectForKey:ktableColumnsCount] intValue];
        int cells = rD*cD;
        NSMutableArray *tableArrayD = [[NSMutableArray alloc] init];
        for(int ce= 0; ce < cells; ce++)
        {
            NSMutableDictionary *tableValDic = [[NSMutableDictionary alloc] init];
            // [tableValDic setObject:@"blank" forKey:@"value"];
            // [tableValDic setObject:@"place" forKey:@"UID"];
            [tableArrayD insertObject:tableValDic atIndex:ce];
        }
        [tableViewD setObject:tableArrayD forKey:@"tableDataArray"];
        [self.workingEntry addObject:@"TABLE"];
        [self.workingEntry addObject:tableViewD];
        [workingPropertyString setString:@""];

        // self.celltracker=0;
        // NSMutableDictionary *tableViewD = [[NSMutableDictionary alloc] init];
        // [tableViewD setObject:[attributeDict objectForKey:ktableRowCount]
        forKey:ktableRowCount];
        // [tableViewD setObject:[attributeDict objectForKey:ktableColumnsCount]
        forKey:ktableColumnsCount];
        // [tableViewD setObject:[attributeDict objectForKey:@"title"] forKey:ktitle];
        // int rD = [[attributeDict objectForKey:ktableRowCount] intValue];
        // int cD = [[attributeDict objectForKey:ktableColumnsCount] intValue];
        // int cells = rD*cD;
        // NSMutableDictionary *tableValDic = [[NSMutableDictionary alloc] init];
        // [tableViewD setObject:tableValDic forKey:@"tableDataArray"];
        // [self.workingEntry addObject:@"TABLE"];
        // [self.workingEntry addObject:tableViewD];
        // [workingPropertyString setString:@""];
    }
    if([elementName isEqualToString:ktableColumn])
    {
        [self.workingEntry objectAtIndex:([self.workingEntry count]-1)]
        objectForKey:@"tableDataArray" objectAtIndex:self.celltracker ] setObject:[attributeDict
        objectForKey:@"uid" forKey:@"UID"];
    }
    if ([elementName isEqualToString:@"IMAGE"])
    {
        NSMutableDictionary *imageViewD = [[NSMutableDictionary alloc] init];
        [imageViewD setObject:[attributeDict objectForKey:@"height"] forKey:kimageHeight];
        // [imageViewD setObject:[attributeDict objectForKey:kimageWidth] forKey:kimageWidth];
        [self.workingEntry addObject:@"IMAGE"];
        [self.workingEntry addObject:imageViewD];
        [workingPropertyString setString:@""];
    }
    if ([elementName isEqualToString:kvideo])

```



```

{
    NSMutableDictionary *videoViewD = [[NSMutableDictionary alloc] init];
    [videoViewD setObject:[attributeDict objectForKey:kvideoHeight] forKey:kvideoHeight];
    [videoViewD setObject:[attributeDict objectForKey:kvideoWidth] forKey:kvideoWidth];
    [self.workingEntry addObject:@"VIDEO"];
    [self.workingEntry addObject:videoViewD];
    [workingPropertyString setString:@""];
}
if ([elementName isEqualToString:@"TEXTINPUT"])
{
    NSMutableDictionary *videoViewD = [[NSMutableDictionary alloc] init];
    [videoViewD setObject:[attributeDict objectForKey:@"length"] forKey:@"length"];
    [videoViewD setObject:[attributeDict objectForKey:@"uid"] forKey:@"uid"];
    [self.workingEntry addObject:@"TEXTINPUT"];
    [self.workingEntry addObject:videoViewD];
    [workingPropertyString setString:@""];
}
}
- (void)parser:(NSXMLParser *)parser didEndElement:(NSString *)elementName
        namespaceURI:(NSString *)namespaceURI
        qualifiedName:(NSString *)qName
{
    NSString *trimmedString = [workingPropertyString stringByTrimmingCharactersInSet:
        [NSCharacterSet whitespaceCharacterSet]];
    if([elementName isEqualToString:kEntryStr]){
        [self.workingArray addObject:self.workingEntry];
        [workingPropertyString setString:@""];
    }
    if ([elementName isEqualToString:ktableColumn]) {
        [[[self.workingEntry objectAtIndex:[self.workingEntry count]-1]
        objectForKey:@"tableDataArray" objectAtIndex:self.celltracker ] setObject:trimmedString
        forKey:@"value"];
        [workingPropertyString setString:@""];
        self.celltracker++;
    }
    if([elementName isEqualToString:ktitle]){
        [self.workingEntry addObject:(trimmedString)];
        [workingPropertyString setString:@""];
    }
    if ([elementName isEqualToString:ktxtView]) {
        [[[self.workingEntry objectAtIndex:[self.workingEntry count]-1]
        setObject:trimmedString forKey:@"inputTxt"];
        // NSString *bls = [[self.workingEntry objectAtIndex:[self.workingEntry count]
        objectAtIndex:@"inputTxt"];
        [workingPropertyString setString:@""];
    }
    if ([elementName isEqualToString:kvideo]) {
        [[[self.workingEntry objectAtIndex:[self.workingEntry count]-1]
        setObject:trimmedString forKey:@"videoLink"];
        [workingPropertyString setString:@""];
    }
    if ([elementName isEqualToString:@"IMAGE"]) {
        [[[self.workingEntry objectAtIndex:[self.workingEntry count]-1]
        setObject:trimmedString forKey:@"imageLink"];
        [workingPropertyString setString:@""];
    }
    if ([elementName isEqualToString:@"TEXTINPUT"]) {
        [[[self.workingEntry objectAtIndex:[self.workingEntry count]-1]
        setObject:trimmedString forKey:@"question"];
        [workingPropertyString setString:@""];
    }
}
}

```

```

- (void)parser:(NSXMLParser *)parser foundCharacters:(NSString *)string
{
    if (storingCharacterData)
    {
        NSString *blah = [string stringByTrimmingCharactersInSet:
                           [NSCharacterSet whitespaceAndNewlineCharacterSet]];
        [workingPropertyString appendString:blah];
    }
}

- (void)parser:(NSXMLParser *)parser parseErrorOccurred:(NSError *)parseError
{
    [delegate parseErrorOccurred:parseError];
}

@end

```

## AppListViewController.h

```

#import <UIKit/UIKit.h>
#import "IconDownloader.h"

@class DetailViewController;
@class ContentController;

@interface AppListViewController : UITableViewController <UIScrollViewDelegate,
IconDownloaderDelegate>
{
    NSMutableArray *entries; // the main data model for our UITableView
    NSMutableDictionary *imageDownloadsInProgress; // the set of IconDownloader objects for
each app

    ContentController *contentController;
}

@property (nonatomic, retain) NSArray *entries;
@property (nonatomic, retain) NSMutableDictionary *imageDownloadsInProgress;

@property (nonatomic, retain) ContentController *contentController;

- (void)appImageDidLoad:(NSIndexPath *)indexPath;

@end

```

## AppListViewController.m

```

#import "AppDelegate.h"
#import "AppListViewController.h"
#import "AppRecord.h"
#import "ContentController.h"

#define kCustomRowHeight 60.0
#define kCustomRowCount 7

#pragma mark -

//handles navigation controller and download image

@interface AppListViewController ()

```

```

- (void)startIconDownload:(AppRecord *)appRecord forIndexPath:(NSIndexPath *)indexPath;
@end

@implementation AppListViewController

@synthesize entries;
@synthesize imageDownloadsInProgress;
@synthesize contentController;

#pragma mark -
#pragma mark UIViewController

- (void)viewDidLoad
{
    [super viewDidLoad];

    self.imageDownloadsInProgress = [NSMutableDictionary dictionary];
    self.tableView.rowHeight = kCustomRowHeight;

    // start listening for download completion
    [[NSNotificationCenter defaultCenter] addObserver:self
                                                selector:@selector(downloadCompleted:)
                                                name:AppDataDownloadCompleted
                                                object:nil];
}

- (void)viewDidUnload
{
    [super viewDidUnload];

    // no longer wanting to listen for download completion
    [[NSNotificationCenter defaultCenter] removeObserver:self
                                                name:AppDataDownloadCompleted
                                                object:nil];
}

- (void)dealloc
{
    [entries release];
    [imageDownloadsInProgress release];
    [contentController release];

    [super dealloc];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];

    // terminate all pending download connections
    NSArray *allDownloads = [self.imageDownloadsInProgress allValues];
    [allDownloads performSelector:@selector(cancelDownload)];
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)toInterfaceOrientation
{
    if(UIInterfaceOrientationIsLandscape(toInterfaceOrientation)){
        return YES;
    }
    else {
        return NO;
    }
}

#pragma mark -
#pragma mark Table view creation (UITableViewDataSource)

```

```

- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    if (entries != nil && entries.count > 0)
    {
        NSInteger *dkfjksdfjks = indexPath.row;
        NSInteger *ksdfkdjf = [entries count];
        if (indexPath.row == [entries count]-2) {
            [entries insertObject:@"Export Data" atIndex:0];
            contentController.detailItem = entries;
        }
        NSMutableArray *app = [entries objectAtIndex:indexPath.row];
        contentController.detailItem = app;
    }
}

// customize the number of rows in the table view
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    int count = [entries count];
    count--;
    // if there's no data yet, return enough rows to fill the screen
    if (count == 0)
        count = kCustomRowCount;

    return count;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    // customize the appearance of table view cells
    //
    static NSString *CellIdentifier = @"TableCell";
    static NSString *PlaceholderCellIdentifier = @"PlaceholderCell";

    // add a placeholder cell while waiting on table data
    int nodeCount = [self.entries count];

    if (nodeCount == 0 && indexPath.row == 0)
    {
        UITableViewCell *cell = [tableView
        dequeueReusableCellWithIdentifier:PlaceholderCellIdentifier];
        if (cell == nil)
        {
            cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle
            reuseIdentifier:PlaceholderCellIdentifier] autorelease];
            cell.detailTextLabel.textAlignment = UITextAlignmentCenter;
            cell.selectionStyle = UITableViewCellSelectionStyleNone;
        }

        cell.detailTextLabel.text = @"Loading...";

        return cell;
    }

    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil)
    {
        cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle
        reuseIdentifier:CellIdentifier] autorelease];
    }

    if (nodeCount > 0)
    {
        NSMutableArray *tab = [self.entries objectAtIndex:indexPath.row];
        NSString *sas = [[NSString alloc] initWithString:[tab objectAtIndex:0]];
        cell.textLabel.text = sas;
    }
}

```

```

//
//      // Only load cached images; defer new downloads until scrolling ends
//      if (TRUE)
//      {
//          if (self.tableView.dragging == NO && self.tableView.decelerating == NO)
//          {
//              // [self startIconDownload:appRecord forIndexPath:indexPath];
//          }
//          // if a download is deferred or in progress, return a placeholder image
//          //cell.imageView.image = [UIImage imageNamed:@"Placeholder.png"];
//      }
//      else
//      {
//          //cell.imageView.image = appRecord.appIcon;
//      }
//      cell.accessoryType = UITableViewCellAccessoryNone;
//  }
//
//  return cell;
}

#pragma mark -
#pragma mark Table cell image support

- (void)startIconDownload:(AppRecord *)appRecord forIndexPath:(NSIndexPath *)indexPath
{
    IconDownloader *iconDownloader = [imageDownloadsInProgress objectForKey:indexPath];
    if (iconDownloader == nil)
    {
        iconDownloader = [[IconDownloader alloc] init];
        iconDownloader.appRecord = appRecord;
        iconDownloader.indexPathInTableView = indexPath;
        iconDownloader.delegate = self;
        [imageDownloadsInProgress setObject:iconDownloader forKey:indexPath];
        [iconDownloader startDownload];
        [iconDownloader release];
    }
}

// this method is used in case the user scrolled into a set of cells that don't have their
// app icons yet
- (void)loadImagesForOnscreenRows
{
    if ([self.entries count] > 0)
    {
        NSArray *visiblePaths = [self.tableView indexPathsForVisibleRows];
        for (NSIndexPath *indexPath in visiblePaths)
        {
            AppRecord *appRecord = [self.entries objectAtIndex:indexPath.row];
            if (!appRecord.appIcon) // avoid the app icon download if the app already has
an icon
            {
                [self startIconDownload:appRecord forIndexPath:indexPath];
            }
        }
    }
}

// called by our ImageDownloader when an icon is ready to be displayed
- (void)appImageDidLoad:(NSIndexPath *)indexPath
{
    IconDownloader *iconDownloader = [imageDownloadsInProgress objectForKey:indexPath];
    if (iconDownloader != nil)
    {
        UITableViewCell *cell = [self.tableView
cellForRowAtIndexPath:indexPathInTableView];
    }
}

```

```

        // Display the newly loaded image
        //cell.imageView.image = iconDownloader.appRecord.appIcon;
    }
}

#pragma mark -
#pragma mark Deferred image loading (UIScrollViewDelegate)

- (void)downloadCompleted:(NSNotification *)notification
{
    self.entries = [notification object]; // incoming object is an NSArray of AppRecords
    [self.tableView reloadData];
}

// Load images for all onscreen rows when scrolling is finished
- (void)scrollViewDidEndDragging:(UIScrollView *)scrollView willDecelerate:(BOOL)decelerate
{
    if (!decelerate)
    {
        [self loadImagesForOnscreenRows];
    }
}

- (void)scrollViewDidEndDecelerating:(UIScrollView *)scrollView
{
    // [self loadImagesForOnscreenRows];
}

@end

```

## Appendix J: App-Builder Source Code

```
//  
// $codepro.audit.disable missingBlock  
package IQP;  
  
import java.awt.EventQueue;  
/**  
 * the main window for the Technology In Education IQP  
 * @author Ben Leone & Nikolaos Matthiopoulos  
 *  
 */  
public class MainWindow {  
    XMLTree project;  
    JPanel userInputPanel;  
    Box userEditingBox;  
    AutoComboBox tabComboBox;  
    AutoComboBox elementComboBox;  
    JPanel userSelectionPanel;  
    Box comboBoxHolder;  
    ItemListener tabComboBoxStateListener;  
    ItemListener elementComboBoxStateListener;  
    ActionListener tabComboBoxActionListener;  
    ActionListener elementComboBoxActionListener;  
    Box creationMenu;  
    Box projectOptions;  
    JLabel tabComboBoxLabel, elementComboBoxLabel, project_title;  
    JLabel idTab;  
    int state = 0;  
    JLabel idPr;  
    JPanel projectMenuPanel;  
  
    private JFrame mainWindowFrame;  
  
    /**  
     * Launch the application.  
     *  
     * @param args arguments  
     */  
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    final MainWindow window = new MainWindow();  
                    window.mainWindowFrame.setVisible(true);  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
    }  
  
    /**  
     * Create the application.  
     */  
    public MainWindow() {  
        initialize();  
    }  
  
    /**  
     * Initialize the contents of the frame.
```

```

*/
private void initialize() {
    mainWindowFrame = new JFrame();
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    mainWindowFrame.setBounds(0, 0, screenSize.width, screenSize.height);

    mainWindowFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    elementComboBoxLabel = new JLabel("Elements: ");
    tabComboBoxLabel = new JLabel("Tabs: ");
    project_title = new JLabel("no current Project");

    JPanel panel = new JPanel();
    mainWindowFrame.getContentPane().add(panel, BorderLayout.NORTH);
    panel.setLayout(new GridLayout(1, 0, 0, 0));

    projectOptions = Box.createVerticalBox();
    panel.add(projectOptions);
    panel.setBorder(BorderFactory.createLineBorder(Color.black));

    loadProjectMenu();

    projectMenuPanel = new JPanel();
    mainWindowFrame.getContentPane().add(projectMenuPanel, BorderLayout.WEST);
    projectMenuPanel.setPreferredSize(new Dimension(150, 190));

    creationMenu = Box.createVerticalBox();
    projectMenuPanel.add(creationMenu);
    loadCreationMenu(0);
    projectMenuPanel.setBorder(BorderFactory.createLineBorder(Color.black));

    userSelectionPanel = new JPanel();
    mainWindowFrame.getContentPane().add(userSelectionPanel, BorderLayout.CENTER);

    userEditingBox = Box.createVerticalBox();
    userSelectionPanel.add(userEditingBox);

    comboBoxHolder = Box.createHorizontalBox();
    userEditingBox.add(comboBoxHolder);
    userEditingBox.add(Box.createVerticalStrut(12));

    comboBoxHolder.add(tabComboBoxLabel);
    tabComboBox = new AutoComboBox(new ArrayList<String>());
    tabComboBox.setEditable(true);
    tabComboBoxActionListener = new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            int selection = tabComboBox.getSelectedIndex();
            setUserEditPanel(loadTabCreator(project.getTab(selection), selection));
            resetDropdowns();
            tabComboBox.setSelectedIndex(selection);
            loadElements();
            resetComboboxListeners();
        }
    };
    tabComboBox.addActionListener(tabComboBoxActionListener);

    tabComboBoxStateListener = new ItemListener() {
        /**
         * this method will select an existing tab to be loaded.
         * it loads the tab editor window with the selected tab.

```



```

        */
        public void itemStateChanged(ItemEvent arg0) {
            if (arg0.getStateChange() == ItemEvent.SELECTED) {
                int selection = tabComboBox.getSelectedIndex();
                setUserEditPanel(loadTabCreator(project.getTab(selection), selection));
                resetDropdowns();
                tabComboBox.setSelectedIndex(selection);
                loadElements();
                resetComboboxListeners();
            }
        }
    };

    tabComboBox.addItemListener(tabComboBoxStateListener);
    comboBoxHolder.add(tabComboBox);
    comboBoxHolder.add(Box.createHorizontalStrut(15));
    comboBoxHolder.add(elementComboBoxLabel);

    elementComboBox = new AutoComboBox(new ArrayList<String>());
    elementComboBox.setEditable(true);
    elementComboBoxStateListener = new ItemListener() {

        @Override
        public void itemStateChanged(ItemEvent e) {
            if(e.getStateChange() == ItemEvent.SELECTED){

                if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("text")){

                    setUserEditPanel(loadTextInPut((TextElement)project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementCom
boBox.getSelectedIndex()), elementComboBox.getSelectedIndex(),
project.getTab(tabComboBox.getSelectedIndex())));
                }

                if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("Video")){

                    setUserEditPanel(loadLinkIn(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.get
SelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
                }

                if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("picture")){

                    setUserEditPanel(loadLinkIn(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.get
SelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
                }

                if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("Table")){

                    setUserEditPanel(loadTable((TableElement)project.getTab(tabComboBox.getSelectedIndex()).elements.get(element
ComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
                }

                if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("intext")){

                    setUserEditPanel(loadTextBox((InText)project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementCom
boBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
                }
            }
        }
    };

```

```

        resetComboboxListeners();
    }
}
};
elementComboBox.addItemListener(elementComboBoxStateListener);

elementComboBoxActionListener = new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {

        if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("text")){

            setUserEditPanel(loadTextInPut((TextElement)project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex(),
project.getTab(tabComboBox.getSelectedIndex())));
        }

        if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("Video")){

            setUserEditPanel(loadLinkIn(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
        }

        if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("picture")){

            setUserEditPanel(loadLinkIn(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
        }

        if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("Table")){

            setUserEditPanel(loadTable((TableElement)project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
        }

        if(project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()).getType().
equals("intext")){

            setUserEditPanel(loadTextBox((InText)project.getTab(tabComboBox.getSelectedIndex()).elements.get(elementComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()), elementComboBox.getSelectedIndex()));
        }
        resetComboboxListeners();
    }
};

elementComboBox.addActionListener(elementComboBoxActionListener);
comboBoxHolder.add(elementComboBox);

userInputPanel = new JPanel();
userEditingBox.add(userInputPanel);
userInputPanel.setSize(new Dimension(300, 300));
resetComboboxListeners();
tabComboBox.addActionListener(tabComboBoxActionListener);
setUserEditPanel(new MessagePanel("hello, \nWelcome to the Technology in Education App builder! \n" +

```

```

Load an existing file.\n\n" +
+
your project\n" +
        "\nWe recommend you begin by creating a new project from the project options or
        "after you have created a project you will need to add a tab from the creation menu.\n" +
        "you may the add elements to your tab from the creation menu or add more tabs to
        "when you are finished you may report your project using the Save Project button.\n" +
        "please note you must retain the .xml extension\n" +
        "\nWe hope you find this builder useful in your educational endeavors." +
        "\nyou can access this menu from the help button in the project options menu.");

```

```

        resetDropdowns();
    }
    /**
    * this method loads a jpanel n to the work area
    * @param place the new panel to load.
    */
    public void setUserEditPanel(JPanel place){
        userEditingBox.remove(userInputPanel);
        userInputPanel = place;
        userEditingBox.add(userInputPanel);
        userInputPanel.setBorder(BorderFactory.createLineBorder(Color.black));
        userInputPanel.revalidate();
        userInputPanel.repaint();
        mainWindowFrame.repaint();
    }

    /**
    *
    */
    public void clearDropdowns(){
        AutoComboBox temp = new AutoComboBox(new ArrayList<String>(0));
        AutoComboBox temp2 = new AutoComboBox(new ArrayList<String>(0));
        comboBoxHolder.removeAll();
        tabComboBox = temp;
        elementComboBox = temp2;
        comboBoxHolder.add(tabComboBoxLabel);
        comboBoxHolder.add(tabComboBox);
        comboBoxHolder.add(Box.createHorizontalStrut(15));
        comboBoxHolder.add(elementComboBoxLabel);
        comboBoxHolder.add(elementComboBox);
        comboBoxHolder.revalidate();
        comboBoxHolder.repaint();
        userSelectionPanel.revalidate();
        userSelectionPanel.repaint();
        tabComboBox.revalidate();
        tabComboBox.repaint();
        elementComboBox.revalidate();
        elementComboBox.repaint();
        mainWindowFrame.repaint();
    }

    /**
    * this function resets the first drop down box with the list of current tabs in the project. the second one is left blank.
    * this function is meant to ready the boxes for tab selection.
    */
    public void resetDropdowns(){
        if(project != null){
            comboBoxHolder.setVisible(true);
            int hold = project.getSize();

```

```

        List<String> TabList = new ArrayList<String>();
        for(int i =0; i < hold; i++){
            TabList.add(project.getTab(i).title);
        }
        tabComboBox.removeItemListener(tabComboBoxStateListener);
        elementComboBox.removeItemListener(elementComboBoxStateListener);
        AutoComboBox temp = new AutoComboBox(TabList);
        AutoComboBox temp2 = new AutoComboBox(new ArrayList<String>());
        comboBoxHolder.removeAll();
        tabComboBox = temp;
        elementComboBox = temp2;
        comboBoxHolder.add(tabComboBoxLabel);
        comboBoxHolder.add(tabComboBox);
        comboBoxHolder.add(Box.createHorizontalStrut(15));
        comboBoxHolder.add(elementComboBoxLabel);
        comboBoxHolder.add(elementComboBox);
        comboBoxHolder.revalidate();
        comboBoxHolder.repaint();
        userSelectionPanel.revalidate();
        userSelectionPanel.repaint();
        tabComboBox.revalidate();
        tabComboBox.repaint();
        elementComboBox.revalidate();
        elementComboBox.repaint();
        mainWindowFrame.repaint();
    }else{
        comboBoxHolder.setVisible(false);
    }
    comboBoxHolder.revalidate();
    comboBoxHolder.repaint();
}

/**
 * loads elements to element combobox
 */
public void loadElements(){
    int selectTabSize = project.getTab(tabComboBox.getSelectedIndex()).elements.size();
    List<String> elementList = new ArrayList<String>();
    for(int i =0; i < selectTabSize; i++){
        elementList.add(project.getTab(tabComboBox.getSelectedIndex()).elements.get(i).getTitle());
    }
    comboBoxHolder.removeAll();
    elementComboBox.removeItemListener(elementComboBoxStateListener);
    AutoComboBox temp = new AutoComboBox(elementList);
    elementComboBox = temp;
    comboBoxHolder.add(tabComboBoxLabel);
    comboBoxHolder.add(tabComboBox);
    comboBoxHolder.add(Box.createHorizontalStrut(15));
    comboBoxHolder.add(elementComboBoxLabel);
    comboBoxHolder.add(elementComboBox);
    comboBoxHolder.revalidate();
    comboBoxHolder.repaint();
    userSelectionPanel.revalidate();
    userSelectionPanel.repaint();
    tabComboBox.revalidate();
    tabComboBox.repaint();
    elementComboBox.revalidate();
    elementComboBox.repaint();
    mainWindowFrame.repaint();
}

/**
 * clears combo boxes after deltion of tabs

```

```

*/
public void cleanUpDelete(){
    loadElements();
    resetComboboxListeners();
}

/**
 * resets combobox listeners. this method should be called after loading the comboboxes.
 */
public void resetComboboxListeners(){
    tabComboBox.removeActionListener(tabComboBoxActionListener);
    elementComboBox.removeActionListener(elementComboBoxActionListener);
    tabComboBox.removeItemListener(tabComboBoxStateListener);
    elementComboBox.removeItemListener(elementComboBoxStateListener);
    tabComboBox.addItemListener(tabComboBoxStateListener);
    elementComboBox.addItemListener(elementComboBoxStateListener);
    tabComboBox.addActionListener(tabComboBoxActionListener);
    elementComboBox.addActionListener(elementComboBoxActionListener);
    tabComboBox.setEditable(true);
    elementComboBox.setEditable(true);

}

/**
 * loads a project to the editor
 * @param tree a prject as defined in the XMLTree package
 */
public void Loadproject(XMLTree tree){
    project = tree;
    resetDropdowns();
    resetComboboxListeners();

}

/**
 * makes a LinkIn element editing window for a LinkIn element
 *
 * @param element the LinkIn element
 * @param tab the tab this element is a part of
 * @param loc the index this elemetrn is in the tab
 * @return a LinkIn edit window for the specific element
 */
public LinkIn loadLinkIn(Ielement element, Tab tab, int loc){
    return new LinkIn(this, element, tab, loc);
}

/**
 * builds a tab creator window
 *
 * @param tab the tab to be edited
 * @param loc the tabs index in the project
 * @return a tab creator window for the tab given
 */
public TabCreator loadTabCreator(Tab tab, int loc){

    resetDropdowns();
    tabComboBox.setSelectedIndex(loc);
    resetComboboxListeners();
    return new TabCreator(this, tab, project, loc);
}

```

```

/**
 * builds a text display editor window
 *
 * @param txt the text element to be edited
 * @param loc the index into the tab that the element is
 * @param tab the tab the element is
 * @return a text display element editor window
 */
public TextDisplay loadTextInPut(TextElement txt, int loc, Tab tab){
    return new TextDisplay(this, tab, txt, loc);
}

/**
 * builds a table edit window
 *
 * @param t the table to be edited
 * @param tab the tab this element is in
 * @param loc the index into the table that this element is at
 * @return a teable edit window for the given element
 */
public Table loadTable(TableElement t, Tab tab, int loc){
    return new Table(t, this, tab, loc);
}

/**
 * builds a textbox edit window
 *
 * @param in the textbox to be edited
 * @param tab the tab that the text box is in
 * @param loc the index in the tab that the element is located at
 * @return a textbox edit window for the specified element.
 */
public TextBox loadTextBox(InText in, Tab tab, int loc){
    return new TextBox(this, in, tab, loc);
}

/**
 * builds a save panel for this project
 * @return a savepanel for the project
 */
public SavePanel loadSavePanel(){
    return new SavePanel(this);
}

/**
 * returns the main window
 * @return the main window
 */
public MainWindow getThis(){
    return this;
}

/**
 * Adds a tab tot he project
 * @return the tab tat was added
 */
public Tab addNewTab(){
    return project.newTab();
}

/**
 * loads the editor area with a message

```

```

* @param mess the message
* @return the messagepanel
*/
public static MessagePanel loadMessage(String mess){
    return new MessagePanel(mess);
}

/**
* loads project menu
*/
public void loadProjectMenu(){

    projectOptions.removeAll();

    JLabel title = new JLabel("Lab Builder");
    projectOptions.add(title);
    projectOptions.add(Box.createHorizontalGlue());
    title.setFont(new Font("Dialog", Font.PLAIN, 24));
    projectOptions.add(Box.createVerticalStrut(10));
    JLabel projectOptionsLabel = new JLabel("Project options");
    projectOptionsLabel.setFont(new Font("Dialog", Font.PLAIN, 16));
    projectOptions.add(Box.createHorizontalGlue());
    projectOptions.add(projectOptionsLabel);
    projectOptions.add(Box.createHorizontalGlue());

    Box horizontalBox_1 = Box.createHorizontalBox();
    projectOptions.add(Box.createHorizontalGlue());
    projectOptions.add(horizontalBox_1);

    if(project != null){
        JButton btnNewButton_5 = new JButton("Save Project");
        btnNewButton_5.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                if(project != null){
                    JFileChooser chooser = new JFileChooser();
                    chooser.setCurrentDirectory( new File( "." + ".xml" ) );
                    int alertDialog = chooser.showSaveDialog(mainWindowFrame);
                    if ( alertDialog == JFileChooser.APPROVE_OPTION )
                    {
                        File fileName = new File( chooser.getSelectedFile(
).getAbsolutePath() );

                        if(fileName == null){
                            return;
                        }
                        String temp = fileName.getAbsolutePath();
                        temp += ".xml";
                        fileName = new File(temp);
                        if(fileName.exists())
                        {
                            alertDialog =
                                JOptionPane.showConfirmDialog(mainWindowFrame, "Replace existing file?");
                                // may need to check for cancel option as well
                                if ( alertDialog ==
                                    JOptionPane.NO_OPTION)
                                    return; // $codepro.audit.disable
                        }
                        // okay to write file
                        BufferedWriter outFile = null;
                    }
                }
            }
        });
    }
}

```





```

        if(fileName == null)
            return;
        // okay to write file

        FileReader reader = new FileReader();
        String s = reader.getFileAsString(fileName);
        Loadproject(XMLParser.readln(s));
    }

    setUserEditPanel(new ProjectEdit(getThis(), project));
    loadProjectMenu();
    if(project.getSize() == 0){
        loadCreationMenu(1);
    }
    if(project.getSize() > 0){
        loadCreationMenu(2);
    }
}

});

horizontalBox_1.add(btnNewButton_6);

Box horizontalBox_2 = Box.createHorizontalBox();
projectOptions.add(Box.createHorizontalGlue());
projectOptions.add(horizontalBox_2);

if(project != null){
    JButton btnNewButton_8 = new JButton("edit project name");
    horizontalBox_2.add(btnNewButton_8);

    JButton btnNewButton_7 = new JButton("Delete Project");
    horizontalBox_2.add(btnNewButton_7);

    btnNewButton_7.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            project = null;
            clearDropdowns();
            resetComboboxListeners();
            loadCreationMenu(0);
            loadProjectMenu();
            setUserEditPanel(new MessagePanel("hello, \nWelcome to the Technology
in Education App builder! \n" +
"from the project options or Load an existing file.\n\n" +
"tab from the creation menu.\n" +
"menu or add more tabs to your project\n" +
using the Save Project button.\n" +
educational endeavors." +
project options menu."));

            resetDropdowns();
        }
    });
}

```

```

    });
    btnNewButton_8.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            if(project != null){
                setUserEditPanel(new ProjectEdit(getThis(), project));
            }else{
                setUserEditPanel(loadMessage("you need a project for this"));
            }
        }
    });
}

JButton btnHelp = new JButton("help");
btnHelp.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        setUserEditPanel(new MessagePanel("hello, \nWelcome to the Technology in
Education App builder! \n" +
                                        "\nWe recommend you begin by creating a new project from the
project options or Load an existing file.\n\n" +
                                        "after you have created a project you will need to add a tab from
the creation menu.\n" +
                                        "you may the add elements to your tab from the creation menu or
add more tabs to your project\n" +
                                        "when you are finished you may report your project using the
Save Project button.\n" +
                                        "please note you must retain the .xml extension\n" +
                                        "\nWe hope you find this builder useful in your educational
endeavors." +
                                        "\nyou can access this menu from the help button in the project
options menu."));
    }
});
horizontalBox_2.add(btnHelp);

if(project != null){
    JButton preview = new JButton("Preview Project");
    preview.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            setUserEditPanel(new preview.preview(project));
        }
    });

    projectOptions.add(preview);
}
if(project == null){
    project_title = new JLabel("no current Project");
    projectOptions.add(project_title);
}
projectOptions.add(Box.createVerticalStrut(20));
if(project != null){
    project_title = new JLabel("Current Project: \n" + project.Title + "\n");
    project_title.setFont(new Font("Dialog", Font.PLAIN, 16));
    projectOptions.add(project_title);
}

projectOptions.add(Box.createVerticalStrut(10));

```

```

projectOptions.setBorder(BorderFactory.createLineBorder(Color.black));
projectOptions.revalidate();
projectOptions.repaint();
mainWindowFrame.repaint();

}

/**
 * this program load the creation menu based on state
 * state 0 = nothing, state 1 = title and tab, state 2 = all elements
 *
 * @param nstate the state to set the creation menu to
 */
public void loadCreationMenu(int nstate){
    state = nstate;
    creationMenu.removeAll();
    if(state > 0){
        JLabel lblNewLabel = new JLabel("Creation Menu");
        creationMenu.add(lblNewLabel);

        //creates new tabs
        idPr = new JLabel("add to " + project.Title + " :");
        creationMenu.add(idPr);
        JButton btnNewButton = new JButton("Tab");
        btnNewButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                if(project != null){
                    setUserEditPanel(loadTabCreator(addNewTab(), project.getSize()
- 1));

                    resetDropdowns();
                    tabComboBox.setSelectedIndex(project.getSize() - 1);
                    resetComboboxListeners();
                    loadCreationMenu(2);
                }else{
                    setUserEditPanel(loadMessage("you must creat a project first"));
                }
            }
        });
        creationMenu.add(btnNewButton);

        if(state > 1){
            idTab = new JLabel("add to " + project.getTab(tabComboBox.getSelectedIndex()).title
+ " :");

            creationMenu.add(idTab);

            JLabel txt1 = new JLabel("Text Display");
            JLabel txt2 = new JLabel("Element");
            Box txtB = Box.createVerticalBox();
            txtB.add(txt1);
            txtB.add(txt2);
            JButton btnNewButton_1 = new JButton();
            btnNewButton_1.add(txtB);
            btnNewButton.setPreferredSize(new Dimension((projectMenuPanel.getWidth()), 50));
            btnNewButton_1.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                    if(project != null){
                        if(project.getSize() != 0){
                            TextElement temp = new TextElement("Text

```

```

" + (project.getTab(tabComboBox.getSelectedIndex()).elements.size()), "", 0);

        project.getTab(tabComboBox.getSelectedIndex()).addElement(temp);
        setUserEditPanel(loadTextInPut(temp,
project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1, project.getTab(tabComboBox.getSelectedIndex()));
        loadElements();

        elementComboBox.setSelectedIndex(project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1);
        resetComboboxListeners();
    }else{
        setUserEditPanel(loadMessage("you must
create a tab first"));
    }
}
}
setUserEditPanel(loadMessage("you must creat a
project first"));
}
});
creationMenu.add(btnNewButton_1);

JLabel img1 = new JLabel("Image");
JLabel img2 = new JLabel("Element");
Box imgB = Box.createVerticalBox();
imgB.add(img1);
imgB.add(img2);
JButton btnNewImage = new JButton();
btnNewImage.add(imgB);
btnNewImage.addMouseListener(new MouseAdapter(){
    @Override
    public void mouseClicked(MouseEvent e){
        if(project != null){
            if(project.getSize() != 0){
                PictureElement temp = new
PictureElement("picture " + (project.getTab(tabComboBox.getSelectedIndex()).elements.size()), "", 200, 200);

                project.getTab(tabComboBox.getSelectedIndex()).addElement(temp);
                setUserEditPanel(loadLinkIn(temp,
project.getTab(tabComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1));
                loadElements();

                elementComboBox.setSelectedIndex(project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1);
                resetComboboxListeners();
            }else{
                setUserEditPanel(loadMessage("you must
create a tab first"));
            }
        }else{
            setUserEditPanel(loadMessage("you must creat a
project first"));
        }
    }
});

creationMenu.add(btnNewImage);

// Creates new videos
JLabel vid1 = new JLabel("Video");
JLabel vid2 = new JLabel("Element");
Box vidB = Box.createVerticalBox();
vidB.add(vid1);
vidB.add(vid2);

```

```

        JButton btnNewVideo = new JButton();
        btnNewVideo.add(vidB);
        btnNewVideo.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                if(project != null){
                    if(project.getSize() != 0){
                        VideoElement temp = new
VideoElement("video " + (project.getTab(tabComboBox.getSelectedIndex()).elements.size()), "", 200, 200);

                        project.getTab(tabComboBox.getSelectedIndex()).addElement(temp);
                        setUserEditPanel(loadLinkIn(temp,
project.getTab(tabComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1));
                        loadElements();

                        elementComboBox.setSelectedIndex(project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1);
                        resetComboboxListeners();
                    }else{
                        setUserEditPanel(loadMessage("you must
create a tab first"));
                    }
                }else{
                    setUserEditPanel(loadMessage("you must creat a
project first"));
                }
            }
        });
        creationMenu.add(btnNewVideo);

//add a new table
JLabel t1 = new JLabel("Table");
JLabel t2 = new JLabel("Element");
Box tB = Box.createVerticalBox();
tB.add(t1);
tB.add(t2);
JButton btnNewTable = new JButton();
btnNewTable.add(tB);
btnNewTable.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        if(project != null){
            if(project.getSize() != 0){
                TableElement temp = new
TableElement("Table " + (project.getTab(tabComboBox.getSelectedIndex()).elements.size()), new String[0][0], 0, 0);

                project.getTab(tabComboBox.getSelectedIndex()).addElement(temp);
                setUserEditPanel(loadTable(temp,
project.getTab(tabComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1));
                loadElements();

                elementComboBox.setSelectedIndex(project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1);
                resetComboboxListeners();
            }else{
                setUserEditPanel(loadMessage("you must
create a tab first"));
            }
        }else{
            setUserEditPanel(loadMessage("you must creat a
project first"));
        }
    }
});

```

```

    }
    });
    creationMenu.add(btnNewTable);

    JLabel q1 = new JLabel("Question");
    JLabel q2 = new JLabel("Element");
    Box qB = Box.createVerticalBox();
    qB.add(q1);
    qB.add(q2);
    JButton btnNewButton_2 = new JButton();
    btnNewButton_2.add(qB);
    btnNewButton_2.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            if(project != null){
                if(project.getSize() != 0){
                    InText temp = new InText("Text Box " +
(project.getTab(tabComboBox.getSelectedIndex()).elements.size()), "", 0);

                    project.getTab(tabComboBox.getSelectedIndex()).addElement(temp);
                    setUserEditPanel(loadTextBox(temp,
project.getTab(tabComboBox.getSelectedIndex()), project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1));
                    loadElements();

                    elementComboBox.setSelectedIndex(project.getTab(tabComboBox.getSelectedIndex()).elements.size() - 1);
                    resetComboboxListeners();
                }else{
                    setUserEditPanel(loadMessage("you must
create a tab first"));
                }
            }else{
                setUserEditPanel(loadMessage("you must creat a
project first"));
            }
        }
    });
    creationMenu.add(btnNewButton_2);
}
}

creationMenu.revalidate();
creationMenu.repaint();
mainWindowFrame.repaint();
}
/**
 * loads titles for the creation menu
 */
public void loadCreationMenuTitles(){
    loadCreationMenu(state);
}
}
}

```

\* This class is for displaying messages to users  
package IQP;

import javax.swing.JPanel;

```

public class MessagePanel extends JPanel {

    /**
     * constructor for the message panel
     * @param Message the message to be displayed
     */
    public MessagePanel(String Message) {

        JTextArea textArea = new JTextArea();
        textArea.setText(Message);
        textArea.setEditable(false);
        add(textArea);

    }

}

```

```

// $codepro.audit.disable methodJavadoc
/**
 * Auto comboBox is an automatically completing combo box
 */

package IQP;

import java.awt.event.ItemEvent;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JComboBox;
import javax.swing.plaf.basic.BasicComboBoxEditor;

/**
 * an auto-completing combobox
 *
 */
public class AutoComboBox extends JComboBox {
    /**
     * public constructor
     */
    private class AutoTextFieldEditor extends BasicComboBoxEditor {

        private AutoTextField getAutoTextFieldEditor() {
            return (AutoTextField) editor;
        }

        AutoTextFieldEditor(java.util.List list) {
            editor = new AutoTextField(list, AutoComboBox.this);
        }
    }

    /**
     * constructor
     * @param list a list of components to load the box with
     */
    public AutoComboBox(java.util.List list) {

```

```

isFired = false;
autoTextFieldEditor = new AutoTextFieldEditor(list);
setEditable(true);
setModel(new DefaultComboBoxModel(list.toArray()) {

    protected void fireContentsChanged(Object obj, int i, int j) {
        if (!isFired)
            super.fireContentsChanged(obj, i, j);
    }

});
setEditor(autoTextFieldEditor);
}

public boolean isCaseSensitive() {
    return autoTextFieldEditor.getAutoTextFieldEditor().isCaseSensitive();
}

public void setCaseSensitive(boolean flag) {
    autoTextFieldEditor.getAutoTextFieldEditor().setCaseSensitive(flag);
}

public boolean isStrict() {
    return autoTextFieldEditor.getAutoTextFieldEditor().isStrict();
}

public void setStrict(boolean flag) {
    autoTextFieldEditor.getAutoTextFieldEditor().setStrict(flag);
}

public java.util.List getDataList() {
    return autoTextFieldEditor.getAutoTextFieldEditor().getDataList();
}

public void setDataList(java.util.List list) {
    autoTextFieldEditor.getAutoTextFieldEditor().setDataList(list);
    setModel(new DefaultComboBoxModel(list.toArray()));
}

void setSelectedValue(Object obj) {
    if (isFired) {
        return;
    } else {
        isFired = true;
        setSelectedItem(obj);
        fireItemStateChanged(new ItemEvent(this, 701, selectedItemReminder,
            1));
        isFired = false;
        return;
    }
}

protected void fireActionEvent() {
    if (!isFired)
        super.fireActionEvent();
}

private AutoTextFieldEditor autoTextFieldEditor;

private boolean isFired;

```



```

}

// $codepro.audit.disable emptyCatchClause, methodJavadoc
/**
 * AutoTextFeild is a text field that can auto complete.
 *
 */
package IQP;

import java.util.List;
import javax.swing.JTextField;
import javax.swing.text.*;

public class AutoTextField extends JTextField {
    class AutoDocument extends PlainDocument {

        public void replace(int i, int j, String s, AttributeSet attributeset)
            throws BadLocationException {
            super.remove(i, j);
            insertString(i, s, attributeset);
        }

        public void insertString(int i, String s, AttributeSet attributeset)
            throws BadLocationException {
            if (s == null || "".equals(s))
                return;
            String s1 = getText(0, i);
            String s2 = getMatch(s1 + s);
            int j = (i + s.length()) - 1;
            if (isStrict && s2 == null) {
                s2 = getMatch(s1);
                j--;
            } else if (!isStrict && s2 == null) {
                super.insertString(i, s, attributeset);
                return;
            }
            if (autoComboBox != null && s2 != null)
                autoComboBox.setSelectedValue(s2);
            super.remove(0, getLength());
            super.insertString(0, s2, attributeset);
            setSelectionStart(j + 1);
            setSelectionEnd(getLength());
        }

        public void remove(int i, int j) throws BadLocationException {
            int k = getSelectionStart();
            if (k > 0)
                k--;
            String s = getMatch(getText(0, k));
            if (!isStrict && s == null) {
                super.remove(i, j);
            } else {
                super.remove(0, getLength());
                super.insertString(0, s, null);
            }
            if (autoComboBox != null && s != null)
                autoComboBox.setSelectedValue(s);
            try {
                setSelectionStart(k);
                setSelectionEnd(getLength());
            } catch (Exception exception) {
            }
        }
    }
}

```

```

    }
}

public AutoTextField(List list) {
    isCaseSensitive = false;
    isStrict = true;
    autoComboBox = null;
    if (list == null) {
        throw new IllegalArgumentException("values can not be null");
    } else {
        dataList = list;
        init();
        return;
    }
}

AutoTextField(List list, AutoComboBox b) {
    isCaseSensitive = false;
    isStrict = true;
    autoComboBox = null;
    if (list == null) {
        throw new IllegalArgumentException("values can not be null");
    } else {
        dataList = list;
        autoComboBox = b;
        init();
        return;
    }
}

private void init() {
    setDocument(new AutoDocument());
    if (isStrict && dataList.size() > 0)
        setText(dataList.get(0).toString());
}

private String getMatch(String s) {
    for (int i = 0; i < dataList.size(); i++) {
        String s1 = dataList.get(i).toString();
        if (s1 != null) {
            if (!isCaseSensitive
                && s1.toLowerCase().startsWith(s.toLowerCase()))
                return s1;
            if (isCaseSensitive && s1.startsWith(s))
                return s1;
        }
    }
}

return null;
}

public void replaceSelection(String s) {
    AutoDocument _lb = (AutoDocument) getDocument();
    if (_lb != null)
        try {
            int i = Math.min(getCaret().getDot(), getCaret().getMark());
            int j = Math.max(getCaret().getDot(), getCaret().getMark());
            _lb.replace(i, j - i, s, null);
        } catch (Exception exception) {
        }
}
}

```

```

public boolean isCaseSensitive() {
    return isCaseSensitive;
}

public void setCaseSensitive(boolean flag) {
    isCaseSensitive = flag;
}

public boolean isStrict() {
    return isStrict;
}

public void setStrict(boolean flag) {
    isStrict = flag;
}

public List getDataList() {
    return dataList;
}

public void setDataList(List list) {
    if (list == null) {
        throw new IllegalArgumentException("values can not be null");
    } else {
        dataList = list;
        return;
    }
}

private List dataList;

private boolean isCaseSensitive;

private boolean isStrict;

private AutoComboBox autoComboBox;
}

/**
 * this is a class that is a window for a text display that gives a message to the user.
 */
package preview;

import javax.swing.Box;

public class DisplayText extends JPanel {

    /**
     * DisplayText public constructor
     * @param txt the text to be displayed
     */
    public DisplayText(TextElement txt) {
        Box verticalBox = Box.createVerticalBox();
        add(verticalBox);

        JLabel lblNewLabel = new JLabel(txt.Title);
        verticalBox.add(lblNewLabel);
    }
}

```

```

        JTextArea lblNewLabel_1 = new JTextArea(txt.Text);
        lblNewLabel_1.setSize(200, 500);
        verticalBox.add(lblNewLabel_1);
    }
}

```

\* this class is a window for editing tables  
package IQP;

```
import javax.swing.JButton;
```

```

public class Table extends JPanel {
    int row;
    int col;
    int i;
    TableElement table_el;
    JComboBox comboBox;
    JComboBox comboBox_1;
    private JTextField titleField;

    private JButton btnNewButton;
    private JButton btnNewButton_1;
    private JButton btnNewButton_2;
    //first row
    private JTextField textField0;
    private JTextField textField1;
    private JTextField textField2;
    private JTextField textField3;
    private JTextField textField4;
    private JTextField textField5;
    private JTextField textField6;
    private JTextField textField7;
    //row 2
    private JTextField textField8;
    private JTextField textField9;
    private JTextField textField10;
    private JTextField textField11;
    private JTextField textField12;
    private JTextField textField13;
    private JTextField textField14;
    private JTextField textField15;
    //row 3
    private JTextField textField16;
    private JTextField textField17;
    private JTextField textField18;
    private JTextField textField19;
    private JTextField textField20;
    private JTextField textField21;
    private JTextField textField22;
    private JTextField textField23;
    //row 4
    private JTextField textField24;
    private JTextField textField25;
    private JTextField textField26;
    private JTextField textField27;
    private JTextField textField28;
    private JTextField textField29;
    private JTextField textField30;
    private JTextField textField31;
    //row 5

```

```

private JTextField textField32;
private JTextField textField33;
private JTextField textField34;
private JTextField textField35;
private JTextField textField36;
private JTextField textField37;
private JTextField textField38;
private JTextField textField39;
//row 6
private JTextField textField40;
private JTextField textField41;
private JTextField textField42;
private JTextField textField43;
private JTextField textField44;
private JTextField textField45;
private JTextField textField46;
private JTextField textField47;
//row 7
private JTextField textField48;
private JTextField textField49;
private JTextField textField50;
private JTextField textField51;
private JTextField textField52;
private JTextField textField53;
private JTextField textField54;
private JTextField textField55;
//row 8
private JTextField textField56;
private JTextField textField57;
private JTextField textField58;
private JTextField textField59;
private JTextField textField60;
private JTextField textField61;
private JTextField textField62;
private JTextField textField63;
private JComboBox comboBox_2;
Tab tab;
int loc;

private MainWindow main;
ItemListener llistener1, llistener2;

/**
 * public constructor for the table editor
 * @param t the table to be edited
 * @param sup the mainwindow this editor is a part of
 * @param tab the the tab the element is in
 * @param loc the index into the tab that the element is
 */
public Table(TableElement t, MainWindow sup, Tab tab, int loc) {
    row = t.rows;
    col = t.columns;
    table_el = t;
    main = sup;
    this.tab = tab;
    this.loc = loc;
    Box verticalBox = Box.createVerticalBox();
    add(verticalBox);
    Box titleBox = Box.createHorizontalBox();

```

```

titleBox.add(new JLabel("Table Edit Window"));
verticalBox.add(titleBox);
verticalBox.add(Box.createVerticalStrut(10));

Box horizontalBox = Box.createHorizontalBox();
verticalBox.add(horizontalBox);

JLabel lblNewLabel = new JLabel("Rows:");
horizontalBox.add(lblNewLabel);
horizontalBox.add(Box.createHorizontalStrut(5));
String[] rowSel = new String[9];
for(i=0;i < 9;i++){
    rowSel[i] = Integer.toString(i);
}
comboBox_2 = new JComboBox(rowSel);
listener2 = new ItemListener() {
    public void itemStateChanged(ItemEvent arg0) {
        if (arg0.getStateChange() == ItemEvent.SELECTED) {
            changeRow(comboBox_2.getSelectedIndex());
        }
    }
};
comboBox_2.setSelectedIndex(row);
comboBox_2.addItemListener(listener2);
horizontalBox.add(comboBox_2);

Component horizontalStrut = Box.createHorizontalStrut(10);
horizontalBox.add(horizontalStrut);

JLabel lblNewLabel_1 = new JLabel("Columns:");
horizontalBox.add(lblNewLabel_1);
horizontalBox.add(Box.createHorizontalStrut(5));

String[] colSel = new String[9];
for(i=0;i < 9;i++){
    colSel[i] = Integer.toString(i);
}
comboBox_1 = new JComboBox(colSel);
listener1 = new ItemListener() {
    public void itemStateChanged(ItemEvent arg0) {
        if (arg0.getStateChange() == ItemEvent.SELECTED) {
            changeCol(comboBox_1.getSelectedIndex());
        }
    }
};
comboBox_1.setSelectedIndex(col);
comboBox_1.addItemListener(listener1);
horizontalBox.add(comboBox_1);

verticalBox.add(Box.createVerticalStrut(10));
Box horizontalBox_1 = Box.createHorizontalBox();
verticalBox.add(horizontalBox_1);

JLabel lblNewLabel_2 = new JLabel("Title:");
horizontalBox_1.add(lblNewLabel_2);

Component horizontalStrut_1 = Box.createHorizontalStrut(10);
horizontalBox_1.add(horizontalStrut_1);

titleField = new JTextField();

```

```

horizontalBox_1.add(titleField);
titleField.setText(table_el.title);
titleField.setColumns(20);

verticalBox.add(Box.createVerticalStrut(7));

if(row >= 1){
    Box Row1 = Box.createHorizontalBox();
    verticalBox.add(Row1);
    if(col >= 1){
        textField0 = new JTextField();
        Row1.add(textField0);
    }
    if(col >= 2){
        textField1 = new JTextField();
        Row1.add(textField1);
    }
    if(col >= 3){
        textField2 = new JTextField();
        Row1.add(textField2);
    }
    if(col >= 4){
        textField3 = new JTextField();
        Row1.add(textField3);
    }
    if(col >= 5){
        textField4 = new JTextField();
        Row1.add(textField4);
    }
    if(col >= 6){
        textField5 = new JTextField();
        Row1.add(textField5);
    }
    if(col >= 7){
        textField6 = new JTextField();
        Row1.add(textField6);
    }
    if(col >= 8){
        textField7 = new JTextField();
        Row1.add(textField7);
    }
}

if(row >= 2){
    Box Row2 = Box.createHorizontalBox();
    verticalBox.add(Row2);
    if(col >= 1){
        textField8 = new JTextField();
        Row2.add(textField8);
    }
    if(col >= 2){
        textField9 = new JTextField();
        Row2.add(textField9);
    }
    if(col >= 3){
        textField10 = new JTextField();
        Row2.add(textField10);
    }
    if(col >= 4){
        textField11 = new JTextField();
        Row2.add(textField11);
    }
}

```

```

    }
    if(col >= 5){
        textField12 = new JTextField();
        Row2.add(textField12);
    }
    if(col >= 6){
        textField13 = new JTextField();
        Row2.add(textField13);
    }
    if(col >= 7){
        textField14 = new JTextField();
        Row2.add(textField14);
    }
    if(col >= 8){
        textField15 = new JTextField();
        Row2.add(textField15);
    }
}

if(row >= 3){
    Box Row3 = Box.createHorizontalBox();
    verticalBox.add(Row3);
    if(col >= 1){
        textField16 = new JTextField();
        Row3.add(textField16);
    }
    if(col >= 2){
        textField17 = new JTextField();
        Row3.add(textField17);
    }
    if(col >= 3){
        textField18 = new JTextField();
        Row3.add(textField18);
    }
    if(col >= 4){
        textField19 = new JTextField();
        Row3.add(textField19);
    }
    if(col >= 5){
        textField20 = new JTextField();
        Row3.add(textField20);
    }
    if(col >= 6){
        textField21 = new JTextField();
        Row3.add(textField21);
    }
    if(col >= 7){
        textField22 = new JTextField();
        Row3.add(textField22);
    }
    if(col >= 8){
        textField23 = new JTextField();
        Row3.add(textField23);
    }
}

if(row >= 4){
    Box Row4 = Box.createHorizontalBox();
    verticalBox.add(Row4);
    if(col >= 1){
        textField24 = new JTextField();
        Row4.add(textField24);
    }
}

```



```

    }
    if(col >= 2){
        textField25 = new JTextField();
        Row4.add(textField25);
    }
    if(col >= 3){
        textField26 = new JTextField();
        Row4.add(textField26);
    }
    if(col >= 4){
        textField27 = new JTextField();
        Row4.add(textField27);
    }
    if(col >= 5){
        textField28 = new JTextField();
        Row4.add(textField28);
    }
    if(col >= 6){
        textField29 = new JTextField();
        Row4.add(textField29);
    }
    if(col >= 7){
        textField30 = new JTextField();
        Row4.add(textField30);
    }
    if(col >= 8){
        textField31 = new JTextField();
        Row4.add(textField31);
    }
}

if(row >= 5){
    Box Row5 = Box.createHorizontalBox();
    verticalBox.add(Row5);
    if(col >= 1){
        textField32 = new JTextField();
        Row5.add(textField32);
    }
    if(col >= 2){
        textField33 = new JTextField();
        Row5.add(textField33);
    }
    if(col >= 3){
        textField34 = new JTextField();
        Row5.add(textField34);
    }
    if(col >= 4){
        textField35 = new JTextField();
        Row5.add(textField35);
    }
    if(col >= 5){
        textField36 = new JTextField();
        Row5.add(textField36);
    }
    if(col >= 6){
        textField37 = new JTextField();
        Row5.add(textField37);
    }
    if(col >= 7){
        textField38 = new JTextField();
        Row5.add(textField38);
    }
}

```

```

        if(col >= 8){
            textField39 = new JTextField();
            Row5.add(textField39);
        }
    }

    if(row >= 6){
        Box Row6 = Box.createHorizontalBox();
        verticalBox.add(Row6);
        if(col >= 1){
            textField40 = new JTextField();
            Row6.add(textField40);
        }
        if(col >= 2){
            textField41 = new JTextField();
            Row6.add(textField41);
        }
        if(col >= 3){
            textField42 = new JTextField();
            Row6.add(textField42);
        }
        if(col >= 4){
            textField43 = new JTextField();
            Row6.add(textField43);
        }
        if(col >= 5){
            textField44 = new JTextField();
            Row6.add(textField44);
        }
        if(col >= 6){
            textField45 = new JTextField();
            Row6.add(textField45);
        }
        if(col >= 7){
            textField46 = new JTextField();
            Row6.add(textField46);
        }
        if(col >= 8){
            textField47 = new JTextField();
            Row6.add(textField47);
        }
    }

    if(row >= 7){
        Box Row7 = Box.createHorizontalBox();
        verticalBox.add(Row7);
        if(col >= 1){
            textField48 = new JTextField();
            Row7.add(textField48);
        }
        if(col >= 2){
            textField49 = new JTextField();
            Row7.add(textField49);
        }
        if(col >= 3){
            textField50 = new JTextField();
            Row7.add(textField50);
        }
        if(col >= 4){
            textField51 = new JTextField();
            Row7.add(textField51);
        }
    }

```

```

        if(col >= 5){
            textField52 = new JTextField();
            Row7.add(textField52);
        }
        if(col >= 6){
            textField53 = new JTextField();
            Row7.add(textField53);
        }
        if(col >= 7){
            textField54 = new JTextField();
            Row7.add(textField54);
        }
        if(col >= 8){
            textField55 = new JTextField();
            Row7.add(textField55);
        }
    }

    if(row >= 8){
        Box Row8 = Box.createHorizontalBox();
        verticalBox.add(Row8);
        if(col >= 1){
            textField56 = new JTextField();
            Row8.add(textField56);
        }
        if(col >= 2){
            textField57 = new JTextField();
            Row8.add(textField57);
        }
        if(col >= 3){
            textField58 = new JTextField();
            Row8.add(textField58);
        }
        if(col >= 4){
            textField59 = new JTextField();
            Row8.add(textField59);
        }
        if(col >= 5){
            textField60 = new JTextField();
            Row8.add(textField60);
        }
        if(col >= 6){
            textField61 = new JTextField();
            Row8.add(textField61);
        }
        if(col >= 7){
            textField62 = new JTextField();
            Row8.add(textField62);
        }
        if(col >= 8){
            textField63 = new JTextField();
            Row8.add(textField63);
        }
    }

    if(row >= 1){
        if(col >= 1){
            textField0.setText(table_el.Table[0][0] );
        }
        if(col >= 2){

```

```

        textField1.setText(table_el.Table[0][1] );
    }
    if(col >= 3){
        textField2.setText(table_el.Table[0][2] );
    }
    if(col >= 4){
        textField3.setText(table_el.Table[0][3] );
    }
    if(col >= 5){
        textField4.setText(table_el.Table[0][4] );
    }
    if(col >= 6){
        textField5.setText(table_el.Table[0][5] );
    }
    if(col >= 7){
        textField6.setText(table_el.Table[0][6] );
    }
    if(col >= 8){
        textField7.setText(table_el.Table[0][7] );
    }
}

if(row >= 2){
    if(col >= 1){
        textField8.setText(table_el.Table[1][0] );
    }
    if(col >= 2){
        textField9.setText(table_el.Table[1][1] );
    }
    if(col >= 3){
        textField10.setText(table_el.Table[1][2] );
    }
    if(col >= 4){
        textField11.setText(table_el.Table[1][3] );
    }
    if(col >= 5){
        textField12.setText(table_el.Table[1][4] );
    }
    if(col >= 6){
        textField13.setText(table_el.Table[1][5] );
    }
    if(col >= 7){
        textField14.setText(table_el.Table[1][6] );
    }
    if(col >= 8){
        textField15.setText(table_el.Table[1][7] );
    }
}

if(row >= 3){
    if(col >= 1){
        textField16.setText(table_el.Table[2][0] );
    }
    if(col >= 2){
        textField17.setText(table_el.Table[2][1] );
    }
    if(col >= 3){
        textField18.setText(table_el.Table[2][2] );
    }
    if(col >= 4){
        textField19.setText(table_el.Table[2][3] );
    }
}

```

```

        if(col >= 5){
            textField20.setText(table_el.Table[2][4] );
        }
        if(col >= 6){
            textField21.setText(table_el.Table[2][5] );
        }
        if(col >= 7){
            textField22.setText(table_el.Table[2][6] );
        }
        if(col >= 8){
            textField23.setText(table_el.Table[2][7] );
        }
    }

    if(row >= 4){
        if(col >= 1){
            textField24.setText(table_el.Table[3][0] );
        }
        if(col >= 2){
            textField25.setText(table_el.Table[3][1] );
        }
        if(col >= 3){
            textField26.setText(table_el.Table[3][2] );
        }
        if(col >= 4){
            textField27.setText(table_el.Table[3][3] );
        }
        if(col >= 5){
            textField28.setText(table_el.Table[3][4] );
        }
        if(col >= 6){
            textField29.setText(table_el.Table[3][5] );
        }
        if(col >= 7){
            textField30.setText(table_el.Table[3][6] );
        }
        if(col >= 8){
            textField31.setText(table_el.Table[3][7] );
        }
    }

    if(row >= 5){
        if(col >= 1){
            textField32.setText(table_el.Table[4][0] );
        }
        if(col >= 2){
            textField33.setText(table_el.Table[4][1] );
        }
        if(col >= 3){
            textField34.setText(table_el.Table[4][2] );
        }
        if(col >= 4){
            textField35.setText(table_el.Table[4][3] );
        }
        if(col >= 5){
            textField36.setText(table_el.Table[4][4] );
        }
        if(col >= 6){
            textField37.setText(table_el.Table[4][5] );
        }
        if(col >= 7){
            textField38.setText(table_el.Table[4][6] );
        }
    }

```

```

        }
        if(col >= 8){
            textField39.setText(table_el.Table[4][7] );
        }
    }

    if(row >= 6){
        if(col >= 1){
            textField40.setText(table_el.Table[5][0] );
        }
        if(col >= 2){
            textField41.setText(table_el.Table[5][1] );
        }
        if(col >= 3){
            textField42.setText(table_el.Table[5][2] );
        }
        if(col >= 4){
            textField43.setText(table_el.Table[5][3] );
        }
        if(col >= 5){
            textField44.setText(table_el.Table[5][4] );
        }
        if(col >= 6){
            textField45.setText(table_el.Table[5][5] );
        }
        if(col >= 7){
            textField46.setText(table_el.Table[5][6] );
        }
        if(col >= 8){
            textField47.setText(table_el.Table[5][7] );
        }
    }

    if(row >= 7){
        if(col >= 1){
            textField48.setText(table_el.Table[6][0] );
        }
        if(col >= 2){
            textField49.setText(table_el.Table[6][1] );
        }
        if(col >= 3){
            textField50.setText(table_el.Table[6][2] );
        }
        if(col >= 4){
            textField51.setText(table_el.Table[6][3] );
        }
        if(col >= 5){
            textField52.setText(table_el.Table[6][4] );
        }
        if(col >= 6){
            textField53.setText(table_el.Table[6][5] );
        }
        if(col >= 7){
            textField54.setText(table_el.Table[6][6] );
        }
        if(col >= 8){
            textField55.setText(table_el.Table[6][7] );
        }
    }

    if(row >= 8){

```

```

        if(col >= 1){
            textField56.setText(table_el.Table[7][0] );
        }
        if(col >= 2){
            textField57.setText(table_el.Table[7][1] );
        }
        if(col >= 3){
            textField58.setText(table_el.Table[7][2] );
        }
        if(col >= 4){
            textField59.setText(table_el.Table[7][3] );
        }
        if(col >= 5){
            textField60.setText(table_el.Table[7][4] );
        }
        if(col >= 6){
            textField61.setText(table_el.Table[7][5] );
        }
        if(col >= 7){
            textField62.setText(table_el.Table[7][6] );
        }
        if(col >= 8){
            textField63.setText(table_el.Table[7][7] );
        }
    }
}

```

```

Box horizontalBox_2 = Box.createHorizontalBox();
verticalBox.add(horizontalBox_2);

```

```

btnNewButton = new JButton("save " + table_el.title);
btnNewButton.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        table_el.title = titleField.getText();
        if(row >= 1){
            if(col >= 1){
                table_el.Table[0][0] = textField0.getText();
            }
            if(col >= 2){
                table_el.Table[0][1] = textField1.getText();
            }
            if(col >= 3){
                table_el.Table[0][2] = textField2.getText();
            }
            if(col >= 4){
                table_el.Table[0][3] = textField3.getText();
            }
            if(col >= 5){
                table_el.Table[0][4] = textField4.getText();
            }
            if(col >= 6){
                table_el.Table[0][5] = textField5.getText();
            }
            if(col >= 7){
                table_el.Table[0][6] = textField6.getText();
            }
            if(col >= 8){
                table_el.Table[0][7] = textField7.getText();
            }
        }
    }
}

```

```

}

if(row >= 2){
    if(col >= 1){
        table_el.Table[1][0] = textField8.getText();
    }
    if(col >= 2){
        table_el.Table[1][1] = textField9.getText();
    }
    if(col >= 3){
        table_el.Table[1][2] = textField10.getText();
    }
    if(col >= 4){
        table_el.Table[1][3] = textField11.getText();
    }
    if(col >= 5){
        table_el.Table[1][4] = textField12.getText();
    }
    if(col >= 6){
        table_el.Table[1][5] = textField13.getText();
    }
    if(col >= 7){
        table_el.Table[1][6] = textField14.getText();
    }
    if(col >= 8){
        table_el.Table[1][7] = textField15.getText();
    }
}

if(row >= 3){
    if(col >= 1){
        table_el.Table[2][0] = textField16.getText();
    }
    if(col >= 2){
        table_el.Table[2][1] = textField17.getText();
    }
    if(col >= 3){
        table_el.Table[2][2] = textField18.getText();
    }
    if(col >= 4){
        table_el.Table[2][3] = textField19.getText();
    }
    if(col >= 5){
        table_el.Table[2][4] = textField20.getText();
    }
    if(col >= 6){
        table_el.Table[2][5] = textField21.getText();
    }
    if(col >= 7){
        table_el.Table[2][6] = textField22.getText();
    }
    if(col >= 8){
        table_el.Table[2][7] = textField23.getText();
    }
}

if(row >= 4){
    if(col >= 1){
        table_el.Table[3][0] = textField24.getText();
    }
    if(col >= 2){
        table_el.Table[3][1] = textField25.getText();
    }
}

```



```

    }
    if(col >= 3){
        table_el.Table[3][2] = textField26.getText();
    }
    if(col >= 4){
        table_el.Table[3][3] = textField27.getText();
    }
    if(col >= 5){
        table_el.Table[3][4] = textField28.getText();
    }
    if(col >= 6){
        table_el.Table[3][5] = textField29.getText();
    }
    if(col >= 7){
        table_el.Table[3][6] = textField30.getText();
    }
    if(col >= 8){
        table_el.Table[3][7] = textField31.getText();
    }
}

if(row >= 5){
    if(col >= 1){
        table_el.Table[4][0] = textField32.getText();
    }
    if(col >= 2){
        table_el.Table[4][1] = textField33.getText();
    }
    if(col >= 3){
        table_el.Table[4][2] = textField34.getText();
    }
    if(col >= 4){
        table_el.Table[4][3] = textField35.getText();
    }
    if(col >= 5){
        table_el.Table[4][4] = textField36.getText();
    }
    if(col >= 6){
        table_el.Table[4][5] = textField37.getText();
    }
    if(col >= 7){
        table_el.Table[4][6] = textField38.getText();
    }
    if(col >= 8){
        table_el.Table[4][7] = textField39.getText();
    }
}

if(row >= 6){
    if(col >= 1){
        table_el.Table[5][0] = textField40.getText();
    }
    if(col >= 2){
        table_el.Table[5][1] = textField41.getText();
    }
    if(col >= 3){
        table_el.Table[5][2] = textField42.getText();
    }
    if(col >= 4){
        table_el.Table[5][3] = textField43.getText();
    }
    if(col >= 5){

```

```

        table_el.Table[5][4] = textField44.getText();
    }
    if(col >= 6){
        table_el.Table[5][5] = textField45.getText();
    }
    if(col >= 7){
        table_el.Table[5][6] = textField46.getText();
    }
    if(col >= 8){
        table_el.Table[5][7] = textField47.getText();
    }
}

if(row >= 7){
    if(col >= 1){
        table_el.Table[6][0] = textField48.getText();
    }
    if(col >= 2){
        table_el.Table[6][1] = textField49.getText();
    }
    if(col >= 3){
        table_el.Table[6][2] = textField50.getText();
    }
    if(col >= 4){
        table_el.Table[6][3] = textField51.getText();
    }
    if(col >= 5){
        table_el.Table[6][4] = textField52.getText();
    }
    if(col >= 6){
        table_el.Table[6][5] = textField53.getText();
    }
    if(col >= 7){
        table_el.Table[6][6] = textField54.getText();
    }
    if(col >= 8){
        table_el.Table[6][7] = textField55.getText();
    }
}

if(row >= 8){
    if(col >= 1){
        table_el.Table[7][0] = textField56.getText();
    }
    if(col >= 2){
        table_el.Table[7][1] = textField57.getText();
    }
    if(col >= 3){
        table_el.Table[7][2] = textField58.getText();
    }
    if(col >= 4){
        table_el.Table[7][3] = textField59.getText();
    }
    if(col >= 5){
        table_el.Table[7][4] = textField60.getText();
    }
    if(col >= 6){
        table_el.Table[7][5] = textField61.getText();
    }
    if(col >= 7){
        table_el.Table[7][6] = textField62.getText();
    }
}

```

```

        }
        if(col >= 8){
            table_el.Table[7][7] = textField63.getText();
        }
        table_el.title = titleField.getText();
        main.loadElements();
        main.resetComboboxListeners();
    }
    Save();
    main.setUserEditPanel(new MessagePanel(table_el.getTitle() + " was saved"));
}

});
horizontalBox_2.add(btnNewButton);

btnNewButton_2 = new JButton("delete " + table_el.title);
btnNewButton_2.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        delete();
    }
});
horizontalBox_2.add(btnNewButton_2);

btnNewButton_1 = new JButton("move");
}
/**
 * changes the rows of the table
 * @param Rows the number of rows the new table will have
 */
public void changeRow(int Rows){
    table_el.rows = Rows;
    table_el.Table = new String[table_el.rows][table_el.columns];
    main.setUserEditPanel(main.loadTable(table_el, tab, loc));
}

/**
 * changes the number of columns the table will have
 * @param col the number of columns the new table will have
 */
public void changeCol(int col){
    table_el.columns = col;
    table_el.Table = new String[table_el.rows][table_el.columns];
    main.setUserEditPanel(main.loadTable(table_el, tab, loc));
}

/**
 * deletes the table element
 */
public void delete(){
    tab.elements.remove(loc);
    main.setUserEditPanel(new JPanel());
    main.loadElements();
    main.resetComboboxListeners();
}

/**
 * saves the table element
 */
public void Save(){

```

```

        main.setUserEditPanel(new Table(table_el, main, tab, loc));
    }
}

```

\* this is a class for a window that edits tables

```

package IQP;
import javax.swing.JPanel;

```

```

public class TableMod extends JPanel {//(RxC) // $codepro.audit.disable codeInComments
    private JTextField textField; //(1x1)
    private JTextField textField_1; //(1x2)
    private JTextField textField_2; //(1x3)
    private JTextField textField_3; //(1x4)
    private JTextField textField_4; //(1x5)
    private JTextField textField_5; //(1x6)
    private JTextField textField_6; //(2x1)
    private JTextField textField_7; //(2x2)
    private JTextField textField_8; //(2x3)
    private JTextField textField_9; //(2x4)
    private JTextField textField_10; //(2x5)
    private JTextField textField_11; //(2x6)
    private JTextField textField_12; //(3x1)
    private JTextField textField_13; //(3x2)
    private JTextField textField_14; //(3x3)
    private JTextField textField_15; //(6x2)
    private JTextField textField_16; //(3x5)
    private JTextField textField_17; //(5x6)
    private JTextField textField_18; //(3x4)
    private JTextField textField_19; //(4x1)
    private JTextField textField_20; //(6x6)
    private JTextField textField_21; //(6x5)
    private JTextField textField_22; //(6x4)
    private JTextField textField_23; //(6x3)
    private JTextField textField_24; //(6x1)
    private JTextField textField_25; //(5x2)
    private JTextField textField_26; //(4x6)
    private JTextField textField_27; //(5x5)
    private JTextField textField_28; //(5x4)
    private JTextField textField_29; //(5x3)
    private JTextField textField_30; //(4x5)
    private JTextField textField_31; //(5x1)
    private JTextField textField_32; //(4x2)
    private JTextField textField_33; //(3x6)
    private JTextField textField_34; //(4x4)
    private JTextField textField_35; //(4x3)
        private JPanel panel_1;
        private JPanel panel_2;
        private Box horizontalBox;
        private Box horizontalBox_1;
        private JLabel lblTitle;
        private JTextField textField_36;
        private Component horizontalStrut;
        private JButton btnNewButton;
        private JButton btnNewButton_1;
        private JButton btnNewButton_2;

    /**
     * Create the panel.

```

```

*/
public TableMod() {

    Box verticalBox = Box.createVerticalBox();
    add(verticalBox);

    panel_1 = new JPanel();
    verticalBox.add(panel_1);

    horizontalBox = Box.createHorizontalBox();
    panel_1.add(horizontalBox);

    lblTitle = new JLabel("Title:");
    horizontalBox.add(lblTitle);

    horizontalStrut = Box.createHorizontalStrut(20);
    horizontalBox.add(horizontalStrut);

    textField_36 = new JTextField();
    horizontalBox.add(textField_36);
    textField_36.setColumns(10);

    JPanel panel = new JPanel();
    verticalBox.add(panel);
    panel.setLayout(new GridLayout(6, 6, 0, 0));

    textField = new JTextField();
    panel.add(textField);
    textField.setColumns(6);

    textField_1 = new JTextField();
    panel.add(textField_1);
    textField_1.setColumns(6);

    textField_2 = new JTextField();
    panel.add(textField_2);
    textField_2.setColumns(6);

    textField_3 = new JTextField();
    panel.add(textField_3);
    textField_3.setColumns(6);

    textField_4 = new JTextField();
    panel.add(textField_4);
    textField_4.setColumns(6);

    textField_5 = new JTextField();
    panel.add(textField_5);
    textField_5.setColumns(6);

    textField_6 = new JTextField();
    panel.add(textField_6);
    textField_6.setColumns(6);

    textField_7 = new JTextField();
    panel.add(textField_7);
    textField_7.setColumns(6);

    textField_8 = new JTextField();
    panel.add(textField_8);
    textField_8.setColumns(6);
}

```

```
textField_9 = new JTextField();
panel.add(textField_9);
textField_9.setColumns(6);

textField_10 = new JTextField();
panel.add(textField_10);
textField_10.setColumns(6);

textField_11 = new JTextField();
panel.add(textField_11);
textField_11.setColumns(6);

textField_12 = new JTextField();
panel.add(textField_12);
textField_12.setColumns(6);

textField_13 = new JTextField();
panel.add(textField_13);
textField_13.setColumns(6);

textField_14 = new JTextField();
panel.add(textField_14);
textField_14.setColumns(6);

textField_18 = new JTextField();
panel.add(textField_18);
textField_18.setColumns(6);

textField_16 = new JTextField();
panel.add(textField_16);
textField_16.setColumns(6);

textField_33 = new JTextField();
panel.add(textField_33);
textField_33.setColumns(6);

textField_19 = new JTextField();
panel.add(textField_19);
textField_19.setColumns(6);

textField_32 = new JTextField();
panel.add(textField_32);
textField_32.setColumns(6);

textField_35 = new JTextField();
panel.add(textField_35);
textField_35.setColumns(6);

textField_34 = new JTextField();
panel.add(textField_34);
textField_34.setColumns(6);

textField_30 = new JTextField();
panel.add(textField_30);
textField_30.setColumns(6);

textField_26 = new JTextField();
panel.add(textField_26);
textField_26.setColumns(6);

textField_31 = new JTextField();
panel.add(textField_31);
```

```

textField_31.setColumns(6);

textField_29 = new JTextField();
panel.add(textField_29);
textField_29.setColumns(6);

textField_25 = new JTextField();
panel.add(textField_25);
textField_25.setColumns(6);

textField_28 = new JTextField();
panel.add(textField_28);
textField_28.setColumns(6);

textField_27 = new JTextField();
panel.add(textField_27);
textField_27.setColumns(6);

textField_17 = new JTextField();
panel.add(textField_17);
textField_17.setColumns(6);

textField_24 = new JTextField();
panel.add(textField_24);
textField_24.setColumns(6);

textField_15 = new JTextField();
panel.add(textField_15);
textField_15.setColumns(6);

textField_23 = new JTextField();
panel.add(textField_23);
textField_23.setColumns(6);

textField_22 = new JTextField();
panel.add(textField_22);
textField_22.setColumns(6);

textField_21 = new JTextField();
panel.add(textField_21);
textField_21.setColumns(6);

textField_20 = new JTextField();
panel.add(textField_20);
textField_20.setColumns(6);

panel_2 = new JPanel();
verticalBox.add(panel_2);

horizontalBox_1 = Box.createHorizontalBox();
panel_2.add(horizontalBox_1);

btnNewButton = new JButton("save");
horizontalBox_1.add(btnNewButton);

btnNewButton_2 = new JButton("delete");
horizontalBox_1.add(btnNewButton_2);

btnNewButton_1 = new JButton("move");
}

```

```
}
```

\* this class is for creating tabs and naming them

```
package IQP;
```

```
import javax.swing.JPanel;
```

```
public class TabCreator extends JPanel {
    private JTextField textField;
    private MainWindow main;
    private Tab tab;
    private XMLTree project;
    private int loc;

    /**
     * public constructor for the TabCreator
     * @param sup The mainWindow this is a part of
     * @param in the tab to be created/edited
     * @param proj the project this tab is a part of
     * @param pos the index this tab will be in the project
     */
    public TabCreator(MainWindow sup, Tab in, XMLTree proj, int pos) {
        main = sup;
        project = proj;
        tab = in;
        loc = pos;
        setLayout(new BorderLayout(0, 0));

        JPanel panel = new JPanel();
        add(panel, BorderLayout.SOUTH);

        JButton btnNewButton = new JButton("save " + tab.title);
        btnNewButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                tab.title = textField.getText();
                main.resetDropdowns();
                main.tabComboBox.setSelectedIndex(loc);
                main.setUserEditPanel(main.loadTabCreator(tab, loc));
                main.setUserEditPanel(new MessagePanel(tab.title + " was saved"));
                main.loadElements();
                main.resetComboboxListeners();
            }
        });
        panel.add(btnNewButton);

        //this is a delete button
        JButton btnNewButton_1 = new JButton("delete " + tab.title);
        btnNewButton_1.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                project.deleteTab(tab);
                main.resetDropdowns();
                main.resetComboboxListeners();
                if(main.project.getSize() == 0){
                    main.loadCreationMenu(1);
                }
                main.setUserEditPanel(new JPanel());
            }
        });
    }
};
```



```

panel.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("move");

JPanel panel_1 = new JPanel();
add(panel_1, BorderLayout.CENTER);

Box verticalBox = Box.createVerticalBox();
panel_1.add(verticalBox);
Box horizontalBox = Box.createHorizontalBox();
verticalBox.add(new JLabel("Tab Edit Window"));
verticalBox.add(Box.createVerticalStrut(5));
verticalBox.add(horizontalBox);

JLabel lblNewLabel = new JLabel("Title of Tab:");
horizontalBox.add(lblNewLabel);
horizontalBox.add(Box.createHorizontalStrut(5));

textField = new JTextField();
horizontalBox.add(textField);
textField.setColumns(10);
textField.setText(tab.getName());

}

}

* this class is for a window that will save the project through a file chooser.
package IQP;

import javax.swing.JFileChooser;

public class SavePanel extends JPanel {

    /**
     * Create the panel.
     * @param sup the mainwindow the panel is a part of
     */
    public SavePanel(MainWindow sup) {

        Box verticalBox = Box.createVerticalBox();
        add(verticalBox);
        final JFileChooser fc = new JFileChooser();
        fc.showSaveDialog(sup.userInputPanel);
        verticalBox.add(fc);

    }

}

```

\* this is a clas for a window that allows for a text display box to be edited

```

package IQP;
import javax.swing.JPanel;

```

```

public class TextDisplay extends JPanel {
    private MainWindow main;
    private JTextField textField;
    private JTextArea textArea;
    private TextElement txt;
    private Tab tab;
    private int loc;

    /**
     * TextDisplay public constructor
     * @param sup the main window this is a part of
     * @param myTab the tab this element is owned by
     * @param text the text element that is to be edited
     * @param index the index of the element in the tab
     */
    public TextDisplay(MainWindow sup, Tab myTab, TextElement text, int index) {
        main=sup;
        txt = text;
        tab = myTab;
        loc = index;

        setLayout(new BorderLayout(0, 0));

        JPanel panel_2 = new JPanel();
        add(panel_2, BorderLayout.NORTH);

        Box verticalBox = Box.createVerticalBox();
        panel_2.add(verticalBox);

        Box horizontalBox = Box.createHorizontalBox();
        verticalBox.add(new JLabel("Text Edit Window"));
        verticalBox.add(Box.createVerticalStrut(5));
        verticalBox.add(horizontalBox);

        JLabel lblTitle = new JLabel("Title:");
        horizontalBox.add(lblTitle);

        Component horizontalStrut = Box.createHorizontalStrut(20);
        horizontalBox.add(horizontalStrut);

        textField = new JTextField();
        textField.setText(txt.Title);
        horizontalBox.add(textField);
        textField.setColumns(10);

        Box horizontalBox_1 = Box.createHorizontalBox();
        verticalBox.add(horizontalBox_1);

        JPanel panel = new JPanel();
        add(panel, BorderLayout.SOUTH);

        JButton btnNewButton = new JButton("save " + txt.getTitle());
        btnNewButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                txt.Title = textField.getText();
                txt.Text = textArea.getText();
                txt.lines = textArea.getText().length() / 30;
                if(txt.lines < 2){
                    txt.lines=2;
                }
            }
        });
    }
}

```

```

        }
        save();
        main.setUserEditPanel(new MessagePanel(txt.getTitle() + " was saved"));
    }
});
panel.add(btnNewButton);

JButton btnNewButton_1 = new JButton("delete " + txt.getTitle());
btnNewButton_1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        tab.elements.remove(loc);
        main.cleanUpDelete();
        main.setUserEditPanel(new JPanel());
    }
});
panel.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("move");

JPanel panel_1 = new JPanel();
add(panel_1, BorderLayout.CENTER);

textArea = new JTextArea(20, 40);
textArea.setEditable(true);
textArea.setText(txt.Text);
textArea.setBackground(Color.WHITE);
textArea.setBorder(BorderFactory.createLineBorder(Color.black));
Box verticalBox_1 = Box.createVerticalBox();
verticalBox_1.add(Box.createVerticalGlue());
panel_1.add(verticalBox_1);
verticalBox_1.add(new JLabel("Enter text:"));
verticalBox_1.add(textArea);
}
/**
 * saves the textdisplay element
 */
public void save(){
    main.setUserEditPanel(new TextDisplay(main, tab, txt, loc);
    main.loadElements();
    main.elementComboBox.setSelectedIndex(loc);
    main.resetComboboxListeners();
}
}

```

\* This class is for handling editing of individual elements that are Internet links  
package IQP;  
import javax.swing.JPanel;

```

public class LinkIn extends JPanel {
    JPanel panel;
    private int place;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private MainWindow main;
    private JTextField textField_3;

```

```

private VideoElement video;
private PictureElement picture;
private Tab project;
JLabel lblTitle;

/**
 * public constructor
 * @param sup the mainwindow this is a part of
 * @param element the element being edited
 * @param tree the tab that this element is a part of
 * @param loc the index into the tab that this element is at
 */
public LinkIn(MainWindow sup, Ielement element, Tab tree, int loc) {
    place = loc;
    main = sup;
    if(element.getType().equals("Video")){
        video = (VideoElement) element;
        picture = null;
    }else if(element.getType().equals("picture")){
        video = null;
        picture = (PictureElement) element;
    }else{
        return;
    }
    project = tree;
    setLayout(new BorderLayout(0, 0));

    panel = new JPanel();
    add(panel, BorderLayout.SOUTH);

    JButton btnNewButton = new JButton("save " + element.getTitle());
    btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            int t;
            int e2;
            Ielement locale = null;
            if(video != null){
                video.title = textField_3.getText();
                video.link = textField.getText();
                video.width = Integer.parseInt(textField_2.getText());
                video.hight = Integer.parseInt(textField_1.getText());
                locale = video;
            }
            if(picture != null){
                picture.title = textField_3.getText();
                picture.link = textField.getText();
                picture.width = Integer.parseInt(textField_2.getText());
                picture.hieght = Integer.parseInt(textField_1.getText());
                locale = picture;
            }

            t = main.tabComboBox.getSelectedIndex();
            e2 = main.elementComboBox.getSelectedIndex();

            main.loadElements();
            main.tabComboBox.setSelectedIndex(t);
            main.elementComboBox.setSelectedIndex(e2);
            main.setUserEditPanel(redraw());
            main.setUserEditPanel(new MessagePanel(locale.getTitle() + " was saved"));
            main.resetComboboxListeners();
        }
    }
}

```

```

});
panel.add(btnNewButton);

//delete method
JButton btnNewButton_1 = new JButton("delete " + element.getTitle());
btnNewButton_1.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        if(video != null){
            project.elements.remove(video);
        }
        if(picture != null){
            project.elements.remove(picture);
        }
        main.setUserEditPanel(new JPanel());
        main.loadElements();
        main.resetComboboxListeners();
    }
});
panel.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("move");

JPanel panel_2 = new JPanel();
add(panel_2, BorderLayout.CENTER);

Box verticalBox = Box.createVerticalBox();
panel_2.add(verticalBox);
Box titleBox = Box.createHorizontalBox();
titleBox.add(new JLabel("Media Edit Window"));
verticalBox.add(titleBox);
verticalBox.add(Box.createVerticalStrut(10));

Box horizontalBox = Box.createHorizontalBox();
verticalBox.add(horizontalBox);

if(element.getType().equals("Video")){
    lblTitle = new JLabel("Video Title");
}else{
    lblTitle = new JLabel("Picture Title");
}

horizontalBox.add(lblTitle);

Component horizontalStrut_3 = Box.createHorizontalStrut(20);
horizontalBox.add(horizontalStrut_3);

textField_3 = new JTextField();
horizontalBox.add(textField_3);
textField_3.setColumns(10);

Box horizontalBox_1 = Box.createHorizontalBox();
verticalBox.add(horizontalBox_1);

JLabel lblHeight = new JLabel("Height:");
horizontalBox_1.add(lblHeight);

Component horizontalStrut_2 = Box.createHorizontalStrut(20);
horizontalBox_1.add(horizontalStrut_2);

textField_2 = new JTextField();
horizontalBox_1.add(textField_2);

```

```

textField_2.setColumns(10);

Box horizontalBox_2 = Box.createHorizontalBox();
verticalBox.add(horizontalBox_2);

JLabel lblNewLabel_1 = new JLabel("Width:");
horizontalBox_2.add(lblNewLabel_1);

Component horizontalStrut_1 = Box.createHorizontalStrut(20);
horizontalBox_2.add(horizontalStrut_1);

textField_1 = new JTextField();
horizontalBox_2.add(textField_1);
textField_1.setColumns(10);

Box horizontalBox_3 = Box.createHorizontalBox();
verticalBox.add(horizontalBox_3);

JLabel lblNewLabel = new JLabel("Input full link address:");
horizontalBox_3.add(lblNewLabel);
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);

Component horizontalStrut = Box.createHorizontalStrut(20);
horizontalBox_3.add(horizontalStrut);

textField = new JTextField();
horizontalBox_3.add(textField);
textField.setColumns(10);

if(video != null){
    textField_3.setText(video.title);
    textField.setText(video.link);
    textField_2.setText(Integer.toString(video.width));
    textField_1.setText(Integer.toString(video.height));
}
if(image != null){
    textField_3.setText(image.title);
    textField.setText(image.link);
    textField_2.setText(Integer.toString(image.width));
    textField_1.setText(Integer.toString(image.height));
}
}
/**
 * redraws the panel
 * @return the the new panel
 */
public LinkIn redraw(){
    if(image == null){
        return new LinkIn(main, video, project, place);
    }else{
        return new LinkIn(main, image, project, place);
    }
}
}

```

\* This class is for renaming projects  
package IQP;

import javax.swing.JPanel;

```

public class ProjectEdit extends JPanel {
    private JTextField textField;
    private MainWindow main;
    private XMLTree project;

    /**
     * constructor for the project edit window
     * @param sup the main window this is a part of
     * @param tree the XMLTree that is the project to be edited
     */
    public ProjectEdit(MainWindow sup, XMLTree tree) {
        main = sup;
        project = tree;

        Box verticalBox = Box.createVerticalBox();
        add(verticalBox);
        verticalBox.add(Box.createVerticalStrut(15));
        verticalBox.add(new JLabel("Project name edit window"));
        verticalBox.add(Box.createVerticalStrut(10));
        Box horizontalBox = Box.createHorizontalBox();
        verticalBox.add(horizontalBox);

        JLabel lblNewLabel = new JLabel("project name:");
        horizontalBox.add(lblNewLabel);

        Component horizontalStrut = Box.createHorizontalStrut(15);
        horizontalBox.add(horizontalStrut);

        textField = new JTextField();
        horizontalBox.add(textField);
        textField.setText(project.Title);
        textField.setColumns(30);
        verticalBox.add(Box.createVerticalStrut(10));

        JButton btnNewButton = new JButton("save changes");
        btnNewButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                project.Title = textField.getText();
                main.resetDropdowns();
                main.resetComboboxListeners();
                main.setUserEditPanel(new JPanel());
                main.setUserEditPanel(new MessagePanel(project.Title + " was saved"));
                main.loadProjectMenu();
                main.loadCreationMenuTitles();
            }
        });
        verticalBox.add(btnNewButton);
    }
}

```

\* this is a class for a window that allows for a question to be edited.

```
package IQP;
```

```
import javax.swing.JOptionPane;
```

```
public class TextBox extends JPanel {
    private JTextField textField;
```

```

private JTextField textField_1;
private JTextField textField_2;
private MainWindow main;
private InText input;
private Tab tab;
private int index;

/**
 * TextBox public constructor
 * @param sup the mainwindow this is a part of
 * @param in the element to be edited
 * @param myTab the tab the element is a part of
 * @param index the index of the element in the tab
 */
public TextBox(MainWindow sup, InText in, Tab myTab, int index) {
    main = sup;
    input = in;
    tab = myTab;
    this.index = index;

    Box verticalBox = Box.createVerticalBox();
    add(verticalBox);
    Box titleBox = Box.createHorizontalBox();
    titleBox.add(new JLabel("Question Edit Window"));
    verticalBox.add(titleBox);

    Box horizontalBox = Box.createHorizontalBox();
    verticalBox.add(horizontalBox);

    JLabel lblNewLabel = new JLabel("Title:");
    horizontalBox.add(lblNewLabel);

    Component horizontalStrut = Box.createHorizontalStrut(10);
    horizontalBox.add(horizontalStrut);

    textField = new JTextField();
    horizontalBox.add(textField);
    textField.setText(in.title);
    textField.setColumns(10);

    Box horizontalBox_1 = Box.createHorizontalBox();
    verticalBox.add(horizontalBox_1);

    JLabel lblNewLabel_1 = new JLabel("Question:");
    horizontalBox_1.add(lblNewLabel_1);

    Component horizontalStrut_1 = Box.createHorizontalStrut(10);
    horizontalBox_1.add(horizontalStrut_1);

    textField_1 = new JTextField();
    horizontalBox_1.add(textField_1);
    textField_1.setText(in.question);
    textField_1.setColumns(10);

    Box horizontalBox_2 = Box.createHorizontalBox();
    verticalBox.add(horizontalBox_2);

    JLabel lblNewLabel_2 = new JLabel("Lines for answer:");
    horizontalBox_2.add(lblNewLabel_2);

    Component horizontalStrut_2 = Box.createHorizontalStrut(10);
    horizontalBox_2.add(horizontalStrut_2);

```



```

        textField_2 = new JTextField();
        horizontalBox_2.add(textField_2);
        textField_2.setText(Integer.toString(in.lines));
        textField_2.setColumns(10);

        Box horizontalBox_3 = Box.createHorizontalBox();
        verticalBox.add(horizontalBox_3);

        JButton btnNewButton = new JButton("save " + in.getTitle());
        btnNewButton.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                save();
                main.setUserEditPanel(new MessagePanel(input.getTitle() + " was saved"));
            }
        });
        horizontalBox_3.add(btnNewButton);

        JButton btnNewButton_1 = new JButton("delete " + input.getTitle());
        btnNewButton_1.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                delete();
            }
        });
        horizontalBox_3.add(btnNewButton_1);

        JButton btnNewButton_2 = new JButton("move");
    }
    /**
     * saves the text box element
     */
    public void save(){
        input.title = textField.getText();
        input.question = textField_1.getText();
        try{
            input.lines = Integer.parseInt(textField_2.getText());
        }catch(Exception e1){
            textField_2.setText("1");
            input.lines = Integer.parseInt(textField_2.getText());
            JOptionPane.showMessageDialog(null, "Lines must be a number!");
        }
        main.setUserEditPanel(main.loadTextBox(input, tab, index));
        main.loadElements();
        main.elementComboBox.setSelectedIndex(index);
        main.resetComboboxListeners();
    }
    /**
     * deletes the textbox element
     */
    public void delete(){
        tab.elements.remove(index);
        main.loadElements();
        main.resetComboboxListeners();
        main.setUserEditPanel(new JPanel());
    }
}

```

\* this is a window that will preview a project  
package preview;

```
import java.awt.Dimension;
```

```
public class preview extends JPanel {
    XMLTree project;
    List<JButton> Tabs;
    List<JScrollPane> Displays;
    List<Box> boxes;
    JScrollPane scrollPane;
    int temp;
    JPanel panel_1;
    /**
     * Create the panel.
     * @param project the project that is being previewed
     */
    public preview(XMLTree project) {
        this.project = project;
        Tabs = new ArrayList<JButton>();
        Displays = new ArrayList<JScrollPane>();
        boxes = new ArrayList<Box>();
        for(int i=0;i < project.getSize();i++){
            Tabs.add(new JButton(project.getTab(i).title));
            Displays.add(new JScrollPane());
            Displays.get(i).setPreferredSize(new Dimension(500, 375));
            boxes.add(Box.createVerticalBox());
            boxes.get(i).setPreferredSize(new Dimension(500, 375));

            for(int k=0;k < project.getTab(i).elements.size();k++){
                boxes.get(i).add(Dispaly(project.getTab(i).elements.get(k)));
            }
            temp = new Integer(i);
            Displays.get(i).getViewport().add(boxes.get(i));
            Tabs.get(i).addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                    int t;
                    t = Tabs.indexOf(e.getComponent());
                    setScroll(Displays.get(t));
                }
            });
        }
    }

    Box horizontalBox = Box.createHorizontalBox();
    add(horizontalBox);

    JPanel panel = new JPanel();
    horizontalBox.add(panel);

    Box verticalBox = Box.createVerticalBox();
    for(int i = 0; i < Tabs.size();i++){
        verticalBox.add(Tabs.get(i));
    }
}
```

```

        panel.add(verticalBox);

        panel_1 = new JPanel();
        panel_1.setSize(600, 400);
        panel_1.setPreferredSize(panel_1.getSize());
        horizontalBox.add(panel_1);

        scrollPane = new JScrollPane();
        scrollPane.setPreferredSize(new Dimension(500, 375));
        panel_1.add(scrollPane);
    }

    /**
     * sets up display
     * @param el the element that need to be displayed
     * @return the jpanel that will display the element
     */
    public static JPanel Display(Ielement el){
        if(el.getType().equals("text")){
            return new DisplayText((TextElement) el);
        }
        if(el.getType().equals("Video")){
            return new PlayVideo((VideoElement) el);
        }
        if(el.getType().equals("picture")){
            return new ShowPicture((PictureElement) el);
        }
        if(el.getType().equals("Table")){
            return new ShowTable((TableElement) el);
        }
        if(el.getType().equals("intext")){
            return new Question((InText) el);
        }
        return new JPanel();
    }

    public void setScroll(JScrollPane scr){
        panel_1.removeAll();
        scrollPane = scr;
        scrollPane.setVerticalScrollBarPolicy(javax.swing.JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
        scrollPane.setSize(200, 100);
        scrollPane.setPreferredSize(new Dimension(500, 375));

        panel_1.add(scr);
        scrollPane.revalidate();
        scrollPane.repaint();
        panel_1.revalidate();
        panel_1.repaint();
        this.revalidate();
        this.repaint();
        scrollPane.revalidate();
        scrollPane.repaint();
    }

    /** sets the view box in scroll panel
     *
     * @param box the box that will be viewed
     */
    public void setBox(Box box){
        panel_1.removeAll();
        panel_1.add(box);
    }

```

```

        box.revalidate();
        box.repaint();
        panel_1.revalidate();
        panel_1.repaint();
        this.revalidate();
        this.repaint();
        scrollPane.revalidate();
        scrollPane.repaint();
    }

}

* this is a window that displays a question
package preview;

import javax.swing.JPanel;

public class Question extends JPanel {
    private JTextField textField;

    /**
     * Create the panel.
     *
     * @param question to be displayed
     */
    public Question(InText question) {

        Box verticalBox = Box.createVerticalBox();
        add(verticalBox);

        JLabel lblNewLabel = new JLabel(question.question);
        verticalBox.add(lblNewLabel);

        textField = new JTextField();
        verticalBox.add(textField);
        textField.setColumns(30);

    }
}

```

\* this is a window that shows a table  
package preview;

import javax.swing.Box;

```

public class ShowTable extends JPanel {

    int row;
    int col;
    int i;
    TableElement table;
    private JLabel titleField;

    //first row

```

```
private JTextField textField0;
private JTextField textField1;
private JTextField textField2;
private JTextField textField3;
private JTextField textField4;
private JTextField textField5;
private JTextField textField6;
private JTextField textField7;
//row 2
private JTextField textField8;
private JTextField textField9;
private JTextField textField10;
private JTextField textField11;
private JTextField textField12;
private JTextField textField13;
private JTextField textField14;
private JTextField textField15;
//row 3
private JTextField textField16;
private JTextField textField17;
private JTextField textField18;
private JTextField textField19;
private JTextField textField20;
private JTextField textField21;
private JTextField textField22;
private JTextField textField23;
//row 4
private JTextField textField24;
private JTextField textField25;
private JTextField textField26;
private JTextField textField27;
private JTextField textField28;
private JTextField textField29;
private JTextField textField30;
private JTextField textField31;
//row 5
private JTextField textField32;
private JTextField textField33;
private JTextField textField34;
private JTextField textField35;
private JTextField textField36;
private JTextField textField37;
private JTextField textField38;
private JTextField textField39;
//row 6
private JTextField textField40;
private JTextField textField41;
private JTextField textField42;
private JTextField textField43;
private JTextField textField44;
private JTextField textField45;
private JTextField textField46;
private JTextField textField47;
//row 7
private JTextField textField48;
private JTextField textField49;
private JTextField textField50;
private JTextField textField51;
private JTextField textField52;
private JTextField textField53;
private JTextField textField54;
private JTextField textField55;
```

```

//row 8
private JTextField textField56;
private JTextField textField57;
private JTextField textField58;
private JTextField textField59;
private JTextField textField60;
private JTextField textField61;
private JTextField textField62;
private JTextField textField63;

/**
 * Create the panel.
 *
 * @param table the table element to be displayed
 */
public ShowTable(TableElement table) {
    row = table.rows;
    col = table.columns;
    this.table = table;

    Box verticalBox = Box.createVerticalBox();
    add(verticalBox);

    Box horizontalBox_1 = Box.createHorizontalBox();
    verticalBox.add(horizontalBox_1);

    titleField = new JLabel();
    titleField.setText(table.title);

    horizontalBox_1.add(titleField);
    titleField.setText(table.title);

    if(row >= 1){
        Box Row1 = Box.createHorizontalBox();
        verticalBox.add(Row1);
        if(col >= 1){
            textField0 = new JTextField();
            Row1.add(textField0);
        }
        if(col >= 2){
            textField1 = new JTextField();
            Row1.add(textField1);
        }
        if(col >= 3){
            textField2 = new JTextField();
            Row1.add(textField2);
        }
        if(col >= 4){
            textField3 = new JTextField();
            Row1.add(textField3);
        }
        if(col >= 5){
            textField4 = new JTextField();
            Row1.add(textField4);
        }
        if(col >= 6){
            textField5 = new JTextField();

```

```

        Row1.add(textField5);
    }
    if(col >= 7){
        textField6 = new JTextField();
        Row1.add(textField6);
    }
    if(col >= 8){
        textField7 = new JTextField();
        Row1.add(textField7);
    }
}

if(row >= 2){
    Box Row2 = Box.createHorizontalBox();
    verticalBox.add(Row2);
    if(col >= 1){
        textField8 = new JTextField();
        Row2.add(textField8);
    }
    if(col >= 2){
        textField9 = new JTextField();
        Row2.add(textField9);
    }
    if(col >= 3){
        textField10 = new JTextField();
        Row2.add(textField10);
    }
    if(col >= 4){
        textField11 = new JTextField();
        Row2.add(textField11);
    }
    if(col >= 5){
        textField12 = new JTextField();
        Row2.add(textField12);
    }
    if(col >= 6){
        textField13 = new JTextField();
        Row2.add(textField13);
    }
    if(col >= 7){
        textField14 = new JTextField();
        Row2.add(textField14);
    }
    if(col >= 8){
        textField15 = new JTextField();
        Row2.add(textField15);
    }
}

if(row >= 3){
    Box Row3 = Box.createHorizontalBox();
    verticalBox.add(Row3);
    if(col >= 1){
        textField16 = new JTextField();
        Row3.add(textField16);
    }
    if(col >= 2){
        textField17 = new JTextField();
        Row3.add(textField17);
    }
    if(col >= 3){
        textField18 = new JTextField();

```

```

        Row3.add(textField18);
    }
    if(col >= 4){
        textField19 = new JTextField();
        Row3.add(textField19);
    }
    if(col >= 5){
        textField20 = new JTextField();
        Row3.add(textField20);
    }
    if(col >= 6){
        textField21 = new JTextField();
        Row3.add(textField21);
    }
    if(col >= 7){
        textField22 = new JTextField();
        Row3.add(textField22);
    }
    if(col >= 8){
        textField23 = new JTextField();
        Row3.add(textField23);
    }
}

if(row >= 4){
    Box Row4 = Box.createHorizontalBox();
    verticalBox.add(Row4);
    if(col >= 1){
        textField24 = new JTextField();
        Row4.add(textField24);
    }
    if(col >= 2){
        textField25 = new JTextField();
        Row4.add(textField25);
    }
    if(col >= 3){
        textField26 = new JTextField();
        Row4.add(textField26);
    }
    if(col >= 4){
        textField27 = new JTextField();
        Row4.add(textField27);
    }
    if(col >= 5){
        textField28 = new JTextField();
        Row4.add(textField28);
    }
    if(col >= 6){
        textField29 = new JTextField();
        Row4.add(textField29);
    }
    if(col >= 7){
        textField30 = new JTextField();
        Row4.add(textField30);
    }
    if(col >= 8){
        textField31 = new JTextField();
        Row4.add(textField31);
    }
}

if(row >= 5){

```



```

Box Row5 = Box.createHorizontalBox();
verticalBox.add(Row5);
if(col >= 1){
    textField32 = new JTextField();
    Row5.add(textField32);
}
if(col >= 2){
    textField33 = new JTextField();
    Row5.add(textField33);
}
if(col >= 3){
    textField34 = new JTextField();
    Row5.add(textField34);
}
if(col >= 4){
    textField35 = new JTextField();
    Row5.add(textField35);
}
if(col >= 5){
    textField36 = new JTextField();
    Row5.add(textField36);
}
if(col >= 6){
    textField37 = new JTextField();
    Row5.add(textField37);
}
if(col >= 7){
    textField38 = new JTextField();
    Row5.add(textField38);
}
if(col >= 8){
    textField39 = new JTextField();
    Row5.add(textField39);
}
}

if(row >= 6){
    Box Row6 = Box.createHorizontalBox();
    verticalBox.add(Row6);
    if(col >= 1){
        textField40 = new JTextField();
        Row6.add(textField40);
    }
    if(col >= 2){
        textField41 = new JTextField();
        Row6.add(textField41);
    }
    if(col >= 3){
        textField42 = new JTextField();
        Row6.add(textField42);
    }
    if(col >= 4){
        textField43 = new JTextField();
        Row6.add(textField43);
    }
    if(col >= 5){
        textField44 = new JTextField();
        Row6.add(textField44);
    }
    if(col >= 6){
        textField45 = new JTextField();
        Row6.add(textField45);
    }
}

```

```

    }
    if(col >= 7){
        textField46 = new JTextField();
        Row6.add(textField46);
    }
    if(col >= 8){
        textField47 = new JTextField();
        Row6.add(textField47);
    }
}

if(row >= 7){
    Box Row7 = Box.createHorizontalBox();
    verticalBox.add(Row7);
    if(col >= 1){
        textField48 = new JTextField();
        Row7.add(textField48);
    }
    if(col >= 2){
        textField49 = new JTextField();
        Row7.add(textField49);
    }
    if(col >= 3){
        textField50 = new JTextField();
        Row7.add(textField50);
    }
    if(col >= 4){
        textField51 = new JTextField();
        Row7.add(textField51);
    }
    if(col >= 5){
        textField52 = new JTextField();
        Row7.add(textField52);
    }
    if(col >= 6){
        textField53 = new JTextField();
        Row7.add(textField53);
    }
    if(col >= 7){
        textField54 = new JTextField();
        Row7.add(textField54);
    }
    if(col >= 8){
        textField55 = new JTextField();
        Row7.add(textField55);
    }
}

if(row >= 8){
    Box Row8 = Box.createHorizontalBox();
    verticalBox.add(Row8);
    if(col >= 1){
        textField56 = new JTextField();
        Row8.add(textField56);
    }
    if(col >= 2){
        textField57 = new JTextField();
        Row8.add(textField57);
    }
    if(col >= 3){
        textField58 = new JTextField();
        Row8.add(textField58);
    }
}

```

```

    }
    if(col >= 4){
        textField59 = new JTextField();
        Row8.add(textField59);
    }
    if(col >= 5){
        textField60 = new JTextField();
        Row8.add(textField60);
    }
    if(col >= 6){
        textField61 = new JTextField();
        Row8.add(textField61);
    }
    if(col >= 7){
        textField62 = new JTextField();
        Row8.add(textField62);
    }
    if(col >= 8){
        textField63 = new JTextField();
        Row8.add(textField63);
    }
}

if(row >= 1){
    if(col >= 1){
        textField0.setText(table.Table[0][0] );
    }
    if(col >= 2){
        textField1.setText(table.Table[0][1] );
    }
    if(col >= 3){
        textField2.setText(table.Table[0][2] );
    }
    if(col >= 4){
        textField3.setText(table.Table[0][3] );
    }
    if(col >= 5){
        textField4.setText(table.Table[0][4] );
    }
    if(col >= 6){
        textField5.setText(table.Table[0][5] );
    }
    if(col >= 7){
        textField6.setText(table.Table[0][6] );
    }
    if(col >= 8){
        textField7.setText(table.Table[0][7] );
    }
}

if(row >= 2){
    if(col >= 1){
        textField8.setText(table.Table[1][0] );
    }
    if(col >= 2){
        textField9.setText(table.Table[1][1] );
    }
    if(col >= 3){
        textField10.setText(table.Table[1][2] );
    }
}

```

```

        if(col >= 4){
            textField11.setText(table.Table[1][3] );
        }
        if(col >= 5){
            textField12.setText(table.Table[1][4] );
        }
        if(col >= 6){
            textField13.setText(table.Table[1][5] );
        }
        if(col >= 7){
            textField14.setText(table.Table[1][6] );
        }
        if(col >= 8){
            textField15.setText(table.Table[1][7] );
        }
    }

    if(row >= 3){
        if(col >= 1){
            textField16.setText(table.Table[2][0] );
        }
        if(col >= 2){
            textField17.setText(table.Table[2][1] );
        }
        if(col >= 3){
            textField18.setText(table.Table[2][2] );
        }
        if(col >= 4){
            textField19.setText(table.Table[2][3] );
        }
        if(col >= 5){
            textField20.setText(table.Table[2][4] );
        }
        if(col >= 6){
            textField21.setText(table.Table[2][5] );
        }
        if(col >= 7){
            textField22.setText(table.Table[2][6] );
        }
        if(col >= 8){
            textField23.setText(table.Table[2][7] );
        }
    }

    if(row >= 4){
        if(col >= 1){
            textField24.setText(table.Table[3][0] );
        }
        if(col >= 2){
            textField25.setText(table.Table[3][1] );
        }
        if(col >= 3){
            textField26.setText(table.Table[3][2] );
        }
        if(col >= 4){
            textField27.setText(table.Table[3][3] );
        }
        if(col >= 5){
            textField28.setText(table.Table[3][4] );
        }
        if(col >= 6){
            textField29.setText(table.Table[3][5] );
        }
    }

```

```

    }
    if(col >= 7){
        textField30.setText(table.Table[3][6] );
    }
    if(col >= 8){
        textField31.setText(table.Table[3][7] );
    }
}

if(row >= 5){
    if(col >= 1){
        textField32.setText(table.Table[4][0] );
    }
    if(col >= 2){
        textField33.setText(table.Table[4][1] );
    }
    if(col >= 3){
        textField34.setText(table.Table[4][2] );
    }
    if(col >= 4){
        textField35.setText(table.Table[4][3] );
    }
    if(col >= 5){
        textField36.setText(table.Table[4][4] );
    }
    if(col >= 6){
        textField37.setText(table.Table[4][5] );
    }
    if(col >= 7){
        textField38.setText(table.Table[4][6] );
    }
    if(col >= 8){
        textField39.setText(table.Table[4][7] );
    }
}

if(row >= 6){
    if(col >= 1){
        textField40.setText(table.Table[5][0] );
    }
    if(col >= 2){
        textField41.setText(table.Table[5][1] );
    }
    if(col >= 3){
        textField42.setText(table.Table[5][2] );
    }
    if(col >= 4){
        textField43.setText(table.Table[5][3] );
    }
    if(col >= 5){
        textField44.setText(table.Table[5][4] );
    }
    if(col >= 6){
        textField45.setText(table.Table[5][5] );
    }
    if(col >= 7){
        textField46.setText(table.Table[5][6] );
    }
    if(col >= 8){
        textField47.setText(table.Table[5][7] );
    }
}
}

```

```

if(row >= 7){
    if(col >= 1){
        textField48.setText(table.Table[6][0] );
    }
    if(col >= 2){
        textField49.setText(table.Table[6][1] );
    }
    if(col >= 3){
        textField50.setText(table.Table[6][2] );
    }
    if(col >= 4){
        textField51.setText(table.Table[6][3] );
    }
    if(col >= 5){
        textField52.setText(table.Table[6][4] );
    }
    if(col >= 6){
        textField53.setText(table.Table[6][5] );
    }
    if(col >= 7){
        textField54.setText(table.Table[6][6] );
    }
    if(col >= 8){
        textField55.setText(table.Table[6][7] );
    }
}

if(row >= 8){
    if(col >= 1){
        textField56.setText(table.Table[7][0] );
    }
    if(col >= 2){
        textField57.setText(table.Table[7][1] );
    }
    if(col >= 3){
        textField58.setText(table.Table[7][2] );
    }
    if(col >= 4){
        textField59.setText(table.Table[7][3] );
    }
    if(col >= 5){
        textField60.setText(table.Table[7][4] );
    }
    if(col >= 6){
        textField61.setText(table.Table[7][5] );
    }
    if(col >= 7){
        textField62.setText(table.Table[7][6] );
    }
    if(col >= 8){
        textField63.setText(table.Table[7][7] );
    }
}

}

}

```

\* this is a window that displays a video element

```

package preview;

import javax.swing.JPanel;

public class PlayVideo extends JPanel {

    /**
     * Create the panel.
     * @param vid the video element
     */
    public PlayVideo(VideoElement vid) {

        Box verticalBox = Box.createVerticalBox();
        add(verticalBox);

        JLabel lblNewLabel = new JLabel(vid.title);
        verticalBox.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel(vid.link);
        verticalBox.add(lblNewLabel_1);

    }
}

```

\* this is a window that displays a picture  
package preview;

```

import java.io.IOException;

public class ShowPicture extends JPanel {
    java.awt.Image image;
    ImageIcon im;

    /**
     * Create the panel.
     *
     * @param picture the picture element to be displayed
     */
    public ShowPicture(PictureElement picture) {

        Box verticalBox = Box.createVerticalBox();
        add(verticalBox);

        JLabel lblNewLabel = new JLabel(picture.title);
        verticalBox.add(lblNewLabel);

        try {
            // Create a URL for the image's location
            URL url = new URL(picture.link);

            // Get the image
            image = java.awt.Toolkit.getDefaultToolkit().getDefaultToolkit().createImage(url);
            im = new ImageIcon(image);
        } catch (MalformedURLException e) {
            return;
        } catch (IOException e) {
            return;
        }
    }
}

```

```

        JLabel lblNewLabel_1 = new JLabel();
        lblNewLabel_1.setIcon(im);
        verticalBox.add(lblNewLabel_1);

    }

}

/**
 * this class takes a loaded file and parses it
 */
package XMLParse;

import XMLTree.*;

public class XMLParser {

    /**
     * reads in a file and converts it to an xml tree
     * @param file the file to be parsed
     * @return the XMLTree representation of the file
     */
    public static XMLTree readIn(String file){
        String temp;
        String[] tabs = null;
        XMLTree result = null;

        if(file.contains("<APP>") && file.contains("</APP>")){

            file = file.replaceAll("<APP>", "");
            file = file.replaceAll("</APP>", "");
            tabs = file.split("</ENTRY>");
            for(int i=0;i < tabs.length;i++){
                tabs[i] = tabs[i].replaceAll("<ENTRY>", "");
            }
            tabs[0] = tabs[0].replaceFirst("<TITLE>", "");
            temp = tabs[0].split("</TITLE>", 2)[0];
            result = new XMLTree(temp);
            tabs[0] = tabs[0].replaceFirst(temp, "");
            tabs[0] = tabs[0].replaceFirst("</TITLE>", "");
            for(int i=0;i < tabs.length;i++){
                if(tabs[i].replaceAll(" ", "").equals("")){

                }else{
                    result.addTab(getTab(tabs[i]));
                }
            }
            result.deleteTab(tabs.length - 1);

        }

    }

    return result;
}

```



```

}

/**
 * this method will take a string and parse a tab out of it.
 * the method assumes that the string is formatted in xml that is used by the XMLTree standard.
 * this method also fills the tab with its elements
 * @param str the string that has the tab in XML
 * @return the tab and its components
 */
public static Tab getTab(String str){
    Tab tab;
    String temp;
    str = str.replaceFirst("<TITLE>", "");
    temp = str.split("</TITLE>", 2)[0];
    tab = new Tab(temp);
    str = str.replaceFirst(temp, "");
    str = str.replaceFirst("</TITLE>", "");
    buildElements(tab, str);

    return tab;
}

/**
 * this method will take a string that contains elements and a tab to add them to. the
 * string should be XML formatted.
 * @param tab the tab the elements will be added to
 * @param str the string that contains the elements
 */
public static void buildElements(Tab tab, String str){
    if(str.startsWith("<TEXT")){
        str = fetchText(tab, str);
        buildElements(tab, str);
        return;
    }
    if(str.startsWith("<IMAGE")){
        str = fetchImage(tab, str);
        buildElements(tab, str);
        return;
    }
    if(str.startsWith("<TABLE")){
        str = fetchTable(tab, str);
        buildElements(tab, str);
        return;
    }
    if(str.startsWith("<VIDEO")){
        str = fetchVideo(tab, str);
        buildElements(tab, str);
        return;
    }
    if(str.startsWith("<INPUTTEXT")){
        str = fetchInText(tab, str);
        buildElements(tab, str);
        return;
    }
}

/**
 * parses an inText element from xml
 * @param tab the tab to add it to
 * @param str the string to be parsed

```

```

* @return the remaining string after the element was parsed
*/
public static String fetchInText(Tab tab, String str){
    String temp;
    InText in = new InText("", "", 0);
    str = str.replaceFirst("<INPUTTEXT length = \\\"\", \"\");
    in.lines = Integer.parseInt(str.split("\\\"\", 2)[0].replaceAll(" ", ""));
    str = str.split("\\\"\", 2)[1].replaceFirst(">", "");
    in.question = str.split("</INPUTTEXT>", 2)[0];
    str = str.split("</INPUTTEXT>", 2)[1];
    in.title = "question";
    tab.addElement(in);
    return str;
}

/**
* parses a Video element from a string and returns what is left of the string
* @param tab the tab the element is to be added to
* @param str the string the element will be parsed from
* @return the remaining string after the element was parsed
*/
public static String fetchVideo(Tab tab, String str){
    String temp;
    VideoElement video = new VideoElement("", "", 0, 0);
    str = str.replaceFirst("<VIDEO height = \\\"\", \"\");
    temp = str.split("\\\"\", 2)[0];
    video.hight = Integer.parseInt(temp.replaceAll(" ", ""));
    str = str.split("\\\"\", 2)[1].replaceFirst("width = \\\"\", \"\");
    temp = str.split("\\\"\", 2)[0];
    video.width = Integer.parseInt(temp.replaceAll(" ", ""));
    str = str.split("\\\"\", 2)[1].replaceFirst("title = \\\"\", \"\");
    temp = str.split("\\\"\", 2)[0];
    video.title = temp.trim();
    str = str.split("\\\"\", 2)[1].replaceFirst(">", "");
    video.link = str.split("</VIDEO>", 2)[0];
    tab.addElement(video);
    return str;
}

/**
* parses a table from xaml code passed as a string
* @param tab the tab to add the table element to
* @param str the string to parse the table from
* @return the remaining string after the table was parsed
*/
public static String fetchTable(Tab tab, String str){
    String temp;
    TableElement t = new TableElement("", new String[0][0], 0, 0);
    str = str.replaceFirst("<TABLE rows = \\\"\", \"\");
    temp = str.split("\\\"\", 2)[0];
    t.rows = Integer.parseInt(temp.replaceAll(" ", ""));
    str = str.split("\\\"\", 2)[1].replaceFirst("columns = \\\"\", \"\");
    temp = str.split("\\\"\", 2)[0];
    t.columns = Integer.parseInt(temp.replaceAll(" ", ""));
    str = str.split("\\\"\", 2)[1];
    str = str.replaceFirst("title = \\\"\", \"\");
    temp = str.split("\\\"\", 2)[0];
    t.title = temp;
    str = str.split("\\\"\", 2)[1];
    str = str.replaceFirst(">", "");
    t.Table = new String[t.rows][t.columns];
    for(int i=0;i < t.rows;i++){

```

```

        str = str.replaceFirst("<ROW>", "");
        for(int k=0;k < t.columns;k++){
            str = str.replaceFirst("<COLUMN>", "");
            temp = str.split("</COLUMN>", 2)[0];
            if(temp.contains("@BLANK")){
                t.Table[i][k] = "";
            }else{
                t.Table[i][k] = temp.replaceAll(" ", "");
            }
            str = str.split("</COLUMN>", 2)[1];
        }
        str = str.replaceFirst("</ROW>", "");
    }
    str = str.replaceFirst("</TABLE>", "");
    tab.addElement(t);
    return str;
}

```

```

/**
 * parses an image form a string and adds it to a tab.
 * @param tab the tab to have the string added to
 * @param str the string to be parsed
 * @return the string that was left after parsing
 */
public static String fetchImage(Tab tab, String str){
    String temp;
    PictureElement pic = new PictureElement("", "", 0, 0);
    str = str.replaceFirst("<IMAGE ", "");
    str = str.replaceFirst("height = \\", "");
    temp = str.split("\\", 2)[0];
    pic.hieght = Integer.parseInt(temp.replaceAll(" ", ""));
    str = str.split("\\", 2)[1];
    str = str.replaceFirst("width = \\", "");
    temp = str.split("\\", 2)[0];
    pic.width = Integer.parseInt(temp.replaceAll(" ", ""));
    str = str.split("\\", 2)[1];
    str = str.replaceFirst("title= \\", "");
    temp = str.split("\\", 2)[0];
    pic.title = temp;
    str = str.split("\\>", 2)[1];
    temp = str.split("</IMAGE>", 2)[0];
    pic.link = temp;
    str = str.split("</IMAGE>", 2)[1];
    tab.addElement(pic);
    return str;
}

```

```

/**
 * parses a text element from the given string and adds it to a tab
 * @param tab the tab to add to
 * @param str the string to parse
 * @return the string left after parsing
 */
public static String fetchText(Tab tab, String str){

```

```

    String temp;
    TextElelment txt = new TextElelment("text", "", 0);
    str = str.replaceFirst("<TEXT length = \\", "");
    temp = str.split("\\", 2)[0];
    str = str.split("\\", 2)[1];

```

```

        txt.lines = Integer.parseInt(temp.replaceAll(" ", ""));
        str = str.replaceFirst(">", "");
        temp = str.split("</TEXT>", 2)[0];
        txt.Text = temp;
        str = str.split("</TEXT>", 2)[1];
        txt.Title = "TEXT";
        tab.addElement(txt);
        return str;
    }

}

/**
 * this is a class that handles reading XML files and turning them into a project
 */
package XMLParse;

import java.io.*;

public class FileReader {

    /**
     * gets a file and converts it to a string for reading
     *
     * @param file the file to be read
     * @return the string interpretation of the file
     */
    @SuppressWarnings("deprecation")
    public static String getFileAsString(File file){
        FileInputStream fis = null;
        BufferedInputStream bis = null;
        DataInputStream dis = null;
        StringBuffer sb = new StringBuffer();
        try {
            fis = new FileInputStream(file);
            bis = new BufferedInputStream(fis);
            dis = new DataInputStream(bis);

            while (dis.available() != 0) {
                sb.append( dis.readLine() + "\n");
            }
            fis.close();
            bis.close();
            dis.close();

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return sb.toString();
    }

}

/**

```

```

* this is the XMLTree Class
*/
package XMLTree;

import java.util.ArrayList;
import java.util.List;

/**
 * this is the XMLTree class for holding a Java object representation of the XML file
 * @author Ben Leone & Nikolaos Matthiopoulos
 */
public class XMLTree {
    private List<Tab> tree;
    public String Title;

    /**
     * builds a new xml tree
     *
     * @param title the title of the tree
     */
    public XMLTree(String title){
        tree = new ArrayList<Tab>();
        Title = title;
    }

    /**
     * adds a new blank tab to the tree
     *
     * @return the new tab
     */
    public Tab newTab(){
        int size = tree.size();
        Tab tab = new Tab("Tab " + size);
        tree.add(tab);
        return tab;
    }

    /**
     * adds a tab to the tree
     * @param tab the tab to add
     */
    public void addTab(Tab tab){
        tree.add(tab);
    }

    /**
     * gets the indicated tab
     * @param index the index of the tab
     * @return a tab if the index is valid, else null
     */
    public Tab getTab(int index){
        if(index >= 0 && index < tree.size()){
            return tree.get(index);
        }else{
            return null;
        }
    }

    /**
     * deletes a Tab from the tree if the given index is a real tab.
     * @param index the index of the tab to be deleted
     */
}

```

```

public void deleteTab(int index){
    if(index >= 0 && index < tree.size()){
        tree.remove(index);
    }
}

/**
 * deletes a tab from a tree
 *
 * @param tab the tab to be deleted
 */
public void deleteTab(Tab tab){
    tree.remove(tab);
}

/**
 * gets the size of the project by tab
 * @return the number of tabs in the project
 */
public int getSize(){
    return tree.size();
}

/**
 * converts the tree to an xml string
 * @return the xml string for file writing
 */
public String toXML(){
    String out = "";
    out = "<APP>" +
        "<TITLE>" + Title + "</TITLE>"
        ;
    for(int i =0; i < tree.size(); i++){
        out = out + tree.get(i).toXML();
    }
    out = out +
        "</APP>";
    return out;
}

}

/**
 * this represents a input text box in XML
 */
package XMLTree;

public class InText implements Ielement{

    public String title;
    public String question;
    public int lines;

    /**
     * creates an InText element
     * @param title the title of the element
     * @param question the question as as string
     * @param lines the number of lines that the answer will have
     */
    public InText(String title, String question, int lines){
        this.title = title;
        this.question = question;
    }
}

```

```

        this.lines = lines;
    }

    @Override
    public String toXML() {
        return "<INPUTTEXT length = \"\" + lines + "\">" + question + "</INPUTTEXT>";
    }

    @Override
    public String getType() {
        return "intext";
    }

    @Override
    public String getTitle() {
        return title;
    }
}

/**
 * this represents a table element
 */
package XMLTree;

public class TableElement implements Ielement{
    public String[][] Table;
    public String title;
    public int rows, columns;

    /**
     * public constructor for TableElement
     * @param Title the title element
     * @param input the table element as an array
     * @param row the number of rows
     * @param col the number of columns
     */
    public TableElement(String Title, String[][] input, int row, int col){
        Table = new String[row][col];
        for(int i = 0; (i < row) && (i < input.length); i++){
            for(int k = 0; (k < input[i].length) && k < col; k++){
                Table[i][k] = new String(input[i][k]);
            }
        }
        title = Title;
    }

    @Override
    public String toXML() {
        String out;
        out = "<TABLE rows = \"\" + rows + "\" columns = \"\" + columns + "\" title = \"\" + title + "\" >";
        for(int i=0; i < rows; i++){
            out = out +
                "<ROW>";
            for(int k =0;k < columns;k++){
                out = out +
                    "<COLUMN>";
                if(Table[i][k] == null){
                    out = out + "@BLANK";
                }else if(Table[i][k].equals("")){

```

```

        out = out + "@BLANK";
    }else{
        out = out + Table[i][k];
    }
    out = out + "</COLUMN>"
        ;
    }
    out = out +
        "</ROW>";
    }
    out = out +
        "</TABLE>";
    return out;
}

@Override
public String getType() {
    return "Table";
}

@Override
public String getTitle() {
    return title;
}
}

/**
 * this is the VideoElement class
 */
package XMLTree;

public class VideoElement implements Ielement{
    public String link;
    public String title;
    public int hight, width;

    /**
     * the VideoElement public constructor
     * @param Title the title of the element
     * @param string the link to the video on the Internet (youtube embedded link)
     * @param width the pixel width
     * @param height the pixel height
     */
    public VideoElement(String Title, String string, int width, int height){
        link = string;
        title = Title;
        hight = height;
        this.width = width;
    }

    @Override
    public String toXML() {
        return "<VIDEO height = \"" + hight + "\" " + "width = \"" + width + "\" " + "title = \"" + title + "\" >" + link +
"</VIDEO>";
    }

    @Override
    public String getType() {
        return "Video";
    }
}

```



```

        @Override
        public String getTitle() {
            return title;
        }
    }

/**
 * this is a comment to identify this as the TextElement class
 *
 */
package XMLTree;
/**
 * this is the TextElement class for editing text element
 * @author Ben Leone & Nikolaos Matthiopoulos
 *
 */
public class TextElement implements Ielement{
    public String Text;
    public String Title;
    public int lines;

    /**
     * public constructor
     * @param Title the title of the element
     * @param Text the text to display
     * @param lines the number of lines to be displayed
     */
    public TextElement(String Title, String Text, int lines){
        this.Text = Text;
        this.Title = Title;
        this.lines = lines;
    }

    @Override
    public String toXML() {
        return "<TEXT " + "length = " + "\"" + lines + "\" > " + Text + " </TEXT>";
    }

    @Override
    public String getType() {
        return "text";
    }

    @Override
    public String getTitle() {
        return Title;
    }
}

/**
 * this is a tab that holds elements and is part of a tree
 */
package XMLTree;

import java.util.ArrayList;
import java.util.List;

```

```

public class Tab {
    public String title;
    public List<Ielement> elements;

    /**
     * constructs a tab
     * @param title the title of the tab
     */
    public Tab(String title){
        this.title=title;
        elements = new ArrayList<Ielement>();
    }
    /**
     * adds an element to the tab
     * @param add the element to be added
     */
    public void addElement(Ielement add){
        elements.add(add);
    }
    /**
     * converts the tab to XML
     * @return a string that is the xml code of the tab and its elements
     */
    public String toXML(){
        String out = "<ENTRY>" +
                    "<TITLE>" + title + "</TITLE>"
                    ;
        for(int i = 0; i < elements.size(); i++){
            out = out +
                    elements.get(i).toXML();
        }
        out = out +
            "</ENTRY>"
            ;
        return out;
    }
    /**
     * renames the tab
     * @param name the new name for the tab
     */
    public void rename(String name){
        title = name;
    }

    public String getName(){
        return title;
    }
}

/**
 * this is a class that holds an XML project in java
 */
package XMLTree;

public class PictureElement implements Ielement{
    public String link;
    public String title;
    public int hieght;
    public int width;
}

```

```

/**
 * public constructor for PictureElement
 * @param Title the title of the element
 * @param string the link to the element on the Internet
 * @param width the pixel width of the picture
 * @param height the pixel height of the element
 */
public PictureElement(String Title, String string, int width, int height){
    link = string;
    title = Title;
    this.width = width;
    hieght = height;
}

@Override
public String toXML() {
    return "<IMAGE height = \" " + hieght + "\" + \" width = \" + width + \" title= \" + title + \" \">> " + link +
"</IMAGE>";
}

@Override
public String getType() {
    return "picture";
}

@Override
public String getTitle() {
    return title;
}
}

```

```

/**
 * this is an interface for XML elements to be store in the XMLTree
 */
package XMLTree;

public interface Ielement {
    /**
     * this method creates the xml string of the element
     * @return a string of the xml code
     */
    String toXML();
    /**
     * returns the type of the element as a string
     * @return the element type
     */
    String getType();
    /**
     * gets the element title
     * @return the elements title
     */
    String getTitle();
}

```

## Appendix K: Test App XML code

```
<LAB>
  <TITLE>
    BB 2904 Lab 2: Pollution Detection Using Bioassays
  </TITLE>
<ENTRY>
  <TITLE>Objectives</TITLE>
  <TEXT length="7">
    Understand what a Bioassay is and how it works Understand Lethal Concentration
    50%(LC50) Understand the difference between quantitative and qualitative data
    Learn serial dilutions Learn some basics of microbiological sterile technique Learn to
    streak a plate Use Analysis of Variance (ANOVA) to analyze data.
  </TEXT>
</ENTRY>
<ENTRY>
  <TITLE>Introduction</TITLE>
  <TEXT length="10">
    This week in lab you will use two different biological assay systems (bioassays) to
    assess the toxicity of some common household chemicals that could end up as
    pollutants in a water supply. One assay will follow the survival of the fresh water
    crustacean Daphnia during a 1-hour exposure to different concentrations of the
    products being tested. The other assay will assess chemical toxicity by examining the
    light production of a bioluminescent bacteria, Vibrio fisheri, when exposed to varying
    concentrations of the products being tested. You will use the data derived from these
    assays to determine the LC50.
  </TEXT>
</ENTRY>
<ENTRY>
  <TITLE>Background</TITLE>
  <TEXT length="100">
    Bioassays: Biological assays (bioassays) are methods to assess the potential harmful
    effects of a compound by observing its activity on an indicator organism. The
    indicator organism, some times called a biological indicator, can be any living
    organism from bacteria to humans. Bioassays have been used for some time in testing
    chemicals for their carcinogenic and mutagenic potentials, toxicity to animals, and
    potential for immunological reactivity. The initial testing of drugs on cells and
    animals is a form of bioassay for the compounds toxicity. The testing of other
    compounds and products for immunologic affect is another form of bioassay. One of
    the original assays for testing a chemicals mutagenic potential is called the Ames test.
    This test scores a compound's potentials to change DNA by examining the gain or
    loss of specific growth characteristics in a population of bacteria exposed to it. It is
    easy to test many compounds at varying concentrations in this type of test. Bioassays
    have been used to determine to key points in a compounds affects, threshold and
    Lethal Concentration 50%(LC50).

    The threshold is the concentration below which the agent in question has no
    observable effects. This means that for every compound or chemical there is an
    exposure level, which can be tolerated by the organism under study. Te threshold may
    be so small as to be essentially zero for an extremely toxic compound.
```

Lethal Concentration 50% (LC50) is the concentration of a chemical that when a population of test subjects is exposed to it or 50% of them will die. This is the level, which is most often used to compare the toxicities of different compounds. Later in the term when we look at material safety data sheets it is the LC50 of different test animals that is reported, when talking specifically about drug products the term LD50 or Lethal Dose 50% is used instead. LD50 is used for these comparisons because it is an easily quantifiable and identifiable point in survivorship curves when plotting the number of test subjects surviving versus exposure level.

Everyday many types of chemical are released in to the environment. Industry, agriculture, and even household waste contribute. It is obviously important to identify these hazards and determine if they are accumulating to dangerous levels. Some very sophisticated techniques have been developed to test for and monitor compounds in the environment, but sometimes a quick, efficient, and simple test is needed in the field to answer the simple question is there something harmful here or not? This is where bioassays come in, in particular the bioluminescent bacteria. Dr. Kenneth Thomulka developed an inexpensive and convenient field-testing method to determine if hazardous agents are present in a sample. His method uses the loss of light emission from cultures of bioluminescent bacteria to indicate the presence of harmful compounds. The test you will run in lab this week works in a similar fashion.

Daphnia: One of your test organisms for the week will be Daphnia. Daphnia are small fresh water crustaceans. They are only a few millimeters long and vary in size depending on species and age. They are filter feeders, meaning they eat by moving through the water and pulling in smaller organisms in the water like bacteria and algae. They propel them selves through the water by beating their antennae. They are good for these kinds of studies for a number of reasons. First they are small and reproduce quickly so you can easily grow them in the lab, Second, since most pollution finds it way water using an aquatic organism is helpful and because they are aquatic it is easy to expose them to water soluble chemicals. Third though they are small they are still animals and you can observe them for signs of expose other than just death when using them in an assay. For instance, you can see their heart beating, so changes in heart rate can be observed. Fourth, they are filter feeders so they will tend to adsorb all of the toxins present in the food they eat, termed bioaccumulation. Lastly, they are a primary food for many small fish so if something happens to the Daphnia the fish may follow. This is why we Daphnia are commonly used for testing and why we will use them this week.

Bioluminescent Bacteria: There are 9 species of bioluminescent bacteria know at this time. Because it is such a unique attribute they were all originally classified as a single genus with the following general characteristics:

“Plump, straight rods, 0.8-1.3 um in diameter and 1.8-2.4 um in length.  
Accumulate poly-beta-hydroxybutyrate under suitable conditions; do not use b-hydroxybutyrate monomer in the medium.  
Gram-negative.  
Motile by 1-3 unsheathed polar flagella, or non-motile.  
Chemoorganotrophs capable of respiratory and fermentative metabolism.  
Can grow in absence of oxygen when suitable electron-acceptor present.  
Do not denitrify.  
Cannot use molecular nitrogen.

Fermentation of glucose produces acid.  
Sodium ions required for growth; K<sup>+</sup> will not substitute for Na<sup>+</sup>.  
Most strains will grow in synthetic seawater containing glucose and NH<sub>4</sub>Cl;  
Some strains also require L-methionine.  
All strains can use glucose, mannose, fructose, or glycerol as the sole carbon source.  
All strains grow at 20 deg C. Also slow growth at refrigerator temperature”.  
(reference: Isolation of Pure Culture of Bacteria, Vibrio phosphorem from Squid or Ocean Fish, [http://www.disknet.com/indiana\\_biolab/b203.htm](http://www.disknet.com/indiana_biolab/b203.htm), 3/19/05)

Because of newer techniques and closer examination the bioluminescent bacteria (BL) have been reassigned to four genera in which they more closely match the other members than they do each other. The 9 species along with some unique attributes are listed below:

“Alteromonas hanedai - only obligate aerobe emitting light; grows at 4°C not at 35°C  
?? (Photobacterium) leiognathi - fish light organs; no growth at 4°C; growth at 35°C  
Vibrio cholera - a few, rare strains emit faint light; growth at 40°C  
Vibrio fischeri - yellow-orange growth at 30°C; no growth at 4°C; emits light weakly  
Vibrio (Photobacterium) phosphoreum - growth at 4°C, no growth 35°C; brightest light emitter of all bacteria; isolated from fish light organs and seawater  
Vibrio (Lucibacterium) harveyi - many strains emit some light  
Vibrio logei - growth at 4°C; most strains emit light  
Vibrio (Beneckea) natriegens - survives years at room temperature; even drying  
Vibrio splendidus I - most strains emit strong light; growth at 40°C  
Xenorhabdus luminescens - isolated from Heterorhabditis, genus of nematodes”

Reference: Isolation of Pure Culture of Bacteria, Vibrio phosphorem from Squid or Ocean Fish, [http://www.disknet.com/indiana\\_biolab/b203.htm](http://www.disknet.com/indiana_biolab/b203.htm), 3/19/05)

This weeks experiment will utilize Vibrio fischeri as our bioluminescent biological indicator. There are conflicting reports about the intensity of the light produced by Vibrio fischeri, however it is sufficient for our purposes. Vibrio fischeri was originally isolated from fish, hence the name. It is reported to come from a bioluminescing patch from the fish however you can isolate Vibrio fischeri from many non luminescent marine species and even directly from sea water so it is difficult to pinpoint its actual origin. Like all of the other BL and many other organisms, V. fischeri uses the enzyme luciferase to produce its light. The light emitting reaction requires three things, the light emitting molecule (generally termed a luciferin), the enzyme that facilitates the reaction (luciferase) and oxygen. Luciferase facilitates the addition of oxygen to luciferin to produce oxyluciferin and a photon. The bacterial luciferin is related to molecules of the electron transport chain and has the structure seen below in Figure 1:

Figure 1: Structure of Bacterial Luciferin (reference: The Bioluminescence Web Page, <http://www.lifesci.ucsb.edu/~biolum/chem/detail2.html>, 3/19/05)

</TEXT>

<IMAGE height="111">

[http://users.wpi.edu/~enniocd/bacterial\\_luciferin.png](http://users.wpi.edu/~enniocd/bacterial_luciferin.png)

</IMAGE>

<TEXT length="30">

Bacterial luciferin is a reduced riboflavin phosphate (FMNH<sub>2</sub>, pictured here) which is oxidized in association with a long-chain aldehyde, oxygen, and a luciferase..

For further details of the chemistry of bacterial or other bioluminescence see <http://www.lifesci.ucsb.edu/~biolum/>.

One final interesting characteristic of *V. fischeri* is that it exhibits a phenomenon called quorum sensing. If you think of the definition of quorum you may get an idea of what that means. The outward demonstration of the quorum sensing is that individual bacteria or cultures at low concentrations emit little or no light. However, when a culture reaches a certain density (quorum) then the light produced increase markedly. It is not that the light level was undetectable and rises to an observable level, but that the bacteria do not produce light until there are a sufficient number of them. Compounds that the bacteria secrete into the surrounding environment control this phenomenon. When a high enough concentration of the signaling compound is reached the bacteria increase light production. This serves a communication mechanism between bacteria. (Reference: Miyamoto, C.M., Lin, Y.H. and Meighen, E.A. (2000) Control of bioluminescence in *Vibrio fischeri* by the LuxO signal response regulator. *Mol. Microbiol.* 36, 594-607). For more information on quorum sensing in *V. fischeri* and other organisms see <http://www.nottingham.ac.uk/quorum/fischeri2.htm>.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Aseptic Techniques</TITLE>

<TEXT length="20">

Since we will be working with bacteria and you will be streaking a plate to simulate maintaining your own culture of bacteria we need to review some general methods and techniques used in working with bacteria. Sterile, or aseptic technique is at the heart of any microbiological work, both for establishing and maintaining pure cultures and for preventing contamination of the environment by laboratory cultures. The following procedures are used to handle microorganisms in such a way that the culture itself is neither contaminated nor spread to the equipment and people in the vicinity of the person using these techniques. Your TA's will demonstrate proper sterile techniques. Following are some reminders. You can also see some demonstrations at

</TEXT>

<VIDEO height="315" width="560">

[http://www.youtube.com/user/WPI#p/c/E597F22DB929D8FD/4/w1t\\_IP3XXug](http://www.youtube.com/user/WPI#p/c/E597F22DB929D8FD/4/w1t_IP3XXug)

</VIDEO>

<TEXT length="100">

The main sources of alien organisms are dust and humans. You are often more dangerous to your experiment than it is to you. Dust descends onto horizontal surfaces and is stirred up by air currents. Humans collect dust on their skin and clothes, harbor a variety of skin bacteria, and create bacteria laden aerosols when speaking. Some procedures that will greatly decrease the likelihood of contaminated cultures are the following:

1. Keep sterile containers (e.g. plates containing agar) closed or covered as much as possible. When working with a plate, hold the lid above it so it shelters the plate from above.

2. Never put anything sterile, including the lid of a test tube or petri plate, down on your working surface.
3. Don't talk when sterile surfaces and objects are exposed.
4. Before you withdraw a sterile implement from its wrapper, decide exactly what you are going to do with it, and have all the materials you will need laid out exactly in the proper spot.
5. Try to imagine that every non-sterile surface is coated with a fine invisible coating of red dye. Think in terms of preventing any red dye from finding its way into your culture.

#### Basic Procedures of Aseptic Technique

1. The wire inoculating loop or needle is heated to redness (flaming)(Figure 3) by holding it at an angle in the flame of a Bunsen burner before and after using it to transfer bacteria. The heated loop is permitted to cool before touching the bacteria (no matter how tasty they are you don't want fried bacteria), but it is not permitted to touch any non sterile surface such as a bench top or the outside of a Petri dish or flask. After the transfer is made, excess matter or drops of liquid are gently shaken off the loop into the culture tube or plate, and then the loop is re-flamed before being returned to its holder. Flaming loops with clumps of bacteria adhering to the wire may cause these bacteria to spatter into the vicinity, and is an obvious hazard to be avoided when working with pathogens.
2. Solid media in Petri dishes have many uses in the bacteriology laboratory and students must learn how to pick bacteria from solid media for transfer, and how to streak solid media so as to obtain many isolated colonies. Both procedures are accomplished with a wire-inoculating loop or needle, which is flamed, permitted to cool (the process can be sped up by touching the wire to droplets of condensation on the lid of the dish, or touching a portion of the agar that has no growth on it). Each isolated colony usually represents the progeny of a single cell; colonies growing from cells of a pure culture will look alike, although their size may differ depending on how close they are to other colonies. In taking inoculum from a Petri dish culture for transfer, it is best to pick from an isolated colony, one that could have arisen from a single cell, rather than to swipe the loop across masses of growth, which could contain hidden contaminating bacteria. The young healthy cells will be at the outer edge of the colony, and the loop can be manipulated to pick up only these. It is not necessary to pick up the entire colony -- sufficient numbers of bacteria to inoculate any medium can be carried by the wire with no visible mass. It only takes one bacteria to make many more in a hurry (see the exponential growth section)

The sterile Petri dish containing medium to be streaked by the inoculating loop is usually held in one of two ways: either the lid is tipped up slightly opening the dish just enough for the loop to be inserted and brushed over the surface of the medium, or the dish is placed upside down on the bench and the bottom half is picked up and held in the hand while being streaked. The latter method is commonly used for non pathogenic bacteria. Petri dishes are normally stored and handled upside down i.e., the lid on the bottom to prevent moisture condensed on the lid from dripping on the cultures, to ensure that the lid is flush as possible due to extra weight of the medium pressing down, and to make stacks of dishes easier to handle since the bottom can't fall off if inadvertently only the lid is held. It is important to write all identifying words and symbols on the bottom of the dish rather than on the lid, since lids can fall off and get mixed up.



THE IDENTITY OF THE CULTURE AND THE DATE OF TRANSFER SHOULD BE ON EVERY INOCULATED DISH.

The actual streaking to inoculate the fresh plate can be done in a variety of ways, all of which are intended to ensure that individual cells will be separated on the surface of the medium so that the colonies growing from them will be separate, discrete masses of cells (Figure 2).

One effective method is to rub the inoculating loop back and forth over a small region of the dish, knocking the bacteria adhering to the wire onto the agar. The loop is then flamed, dragged once through this heavily streaked area, and streaked in a tight zigzag fashion across one third of the plate. The loop is again flamed, dragged once through this zigzag area, and streaked in a second zigzag over the rest of the plate (see Figures 2 and 3). It is most efficient to keep the zigzags tight, and to use the entire surface of the plate, from edge to edge, without touching previously streaked areas, since the objective of this procedure is to dilute the number of cells dragged by the wire until single ones drop off at well spaced intervals. Thus only the last third of the plate may have isolated colonies: experienced technicians can produce 20-100 isolated colonies on each plate, but a beginner should be pleased to obtain between 15 and 30 isolated colonies after incubation of the streaked plate.

</TEXT>

<IMAGE height="581">

[http://users.wpi.edu/~enniocd/streaking\\_for\\_colony\\_isolation.png](http://users.wpi.edu/~enniocd/streaking_for_colony_isolation.png)

</IMAGE>

<TEXT length="1">

Figure 3: Loops and sterilizing

</TEXT>

<IMAGE height="300">

[http://users.wpi.edu/~enniocd/loops\\_and\\_serilizing\\_1.png](http://users.wpi.edu/~enniocd/loops_and_serilizing_1.png)

</IMAGE>

<IMAGE height="300">

[http://users.wpi.edu/~enniocd/loops\\_and\\_serilizing\\_2.png](http://users.wpi.edu/~enniocd/loops_and_serilizing_2.png)

</IMAGE>

<IMAGE height="400">

[http://users.wpi.edu/~enniocd/loops\\_and\\_serilizing\\_3.png](http://users.wpi.edu/~enniocd/loops_and_serilizing_3.png)

</IMAGE>

<IMAGE height="400">

[http://users.wpi.edu/~enniocd/loops\\_and\\_serilizing\\_4.png](http://users.wpi.edu/~enniocd/loops_and_serilizing_4.png)

</IMAGE>

<TEXT length="50">

3. It is also frequently useful to inoculate liquid media in tubes, either from a colony on a Petri plate or from a tube culture. These transfers are handled in the following manner: the tubes to be inoculated are held in the left hand and the inoculating loop in the right hand (reverse this if you are left-handed). The caps or plugs of the tubes are removed and held between the fingers of the right hand (see Figure 4) throughout the entire procedure. The caps should not be placed on the bench top, and should be replaced on the tube immediately after the transfer is completed. Many technicians pass the top of the open tubes through a flame to sterilize the outside of the opening before and after inoculation. The wire transfer loop is then flamed and used to pick up the inoculum (if the inoculum is from a colony on a petri dish this step will precede

uncapping and flaming the tubes) and the bacteria are either swished off the loop into liquid medium, or if an agar slant is to be inoculated, the loop is inserted to the bottom of the slant and then zigzagged over the surface to the top. Either way, no visible mass of bacteria should adhere to the loop at the end of the procedure, and the loop should be flamed before being put away in its holder. An experienced technician can inoculate 4 or 5 tubes at a time by gripping the tubes between the fingers of one hand, while holding their caps between the fingers of the other hand.

Note the location of the caps - in the right hand, held between the fingers. Also note that the flame is nearby, ready for use as soon as the inoculation is completed. The openings of both test tubes should be flamed before the caps are replaced.

Figure 4. Inoculating a Test Tube

</TEXT>

<IMAGE height="156">

[http://users.wpi.edu/~enniocd/inoculating\\_a\\_test\\_tube.png](http://users.wpi.edu/~enniocd/inoculating_a_test_tube.png)

</IMAGE>

<TEXT length="50">

4. Another commonly used tool in the microbiology laboratory is the sterile pipette. Pipettes can be glass or plastic, sterilized in individual wrappings of paper or all together in a canister, and either serological (Blow out) or Mohr pipettes. Serological pipettes are calibrated to deliver volume down to the last drop in the tip of the pipette, and thus must either be drained against the side of the tube receiving the pipetted material, or a bulb can be used to "blow out" the pipette into the tube. These pipettes have a larger orifice than Mohr pipettes and are frequently used when viscous liquids or thick cultures are being transferred. They are, however, not as accurate in delivering measured quantities as the Mohr pipettes. They usually have an etched or colored band around the top of the pipette and thus can be identified without removing their sterile cover further down. Mohr pipettes are calibrated to hold a specific volume within the barrel of the pipette and are used by permitting the bottom of the meniscus of the liquid to descend to the desired measuring line. They are never entirely drained, since the extra space at the tip of the pipette is not included in the measured volume. They have smaller orifices than serological pipettes and can be used to deliver very precise quantities.

It is important to remember that sterile pipettes must remain in their sterile holders until used, and can never be laid on the bench top. If wrapped in paper, the wrapping can be opened at the upper end of the pipette so that it will be easily accessible when needed; if a canister of pipettes is being used it is opened and placed on its side, with the open end at the edge of the bench top. The pipettes are placed in the canister tip down, so that they can be pulled out one at a time as needed, by the top. Replace the cap on the canister when done removing pipettes.

Sterile pipettes are frequently plugged with a bit of cotton before sterilization to avoid contaminating the pipette bulbs and to prevent contamination of the pipette with "stuff" from the air in the bulb. A rubber bulb or pipettor is used to draw liquid into the pipette. Mouth pipetting is no longer used and occasionally led to unpleasant surprises.

Pipetting aseptically into the tubes is done in the same manner as wire loop transferring -- the tubes are held in one hand, the pipettes and caps in the other, and

the tubes are usually flamed before and after transfer. Many people flame the outside of the pipette, but it is important not to overheat it since glass does not cool rapidly and plastic melts.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Culture Media</TITLE>

<TEXT length="50">

The nutritional requirements of bacteria are extremely variable, although bacteria, as living things, have certain basic needs in common. In addition to water, four essential elements, carbon, nitrogen, phosphorous, and sulfur, must be present in the media. These four elements may be provided by inorganic or organic compounds, depending upon the kind of organism one wishes to cultivate. For example, photoautotrophs and chemoautotrophs require the necessary elements in inorganic form as they are capable of synthesizing all of their cellular constituents by using either sunlight or the oxidation of inorganic compounds as a source of energy. All organisms also require vitamins and trace mineral elements such as magnesium, zinc, copper, and molybdenum. Autotrophs can synthesize their own vitamins, while many heterotrophs require preformed vitamins (or precursors of vitamins) since they are unable to synthesize their own.

A culture medium is the material in which or on which bacteria are grown. The media used for the cultivation of microorganisms may be liquid, semisolid, or solid. The ingredients included in any one medium may be limited to well defined inorganic or organic compounds. A medium such as this, whose chemical constituents are well known, is said to be a chemically defined medium. On the other hand, media (especially those designed for cultivation of nutritionally fastidious organisms, such as certain human pathogens) may contain exceedingly complex materials (digest of plants, animal tissue extract, or blood serum) whose composition is not clearly defined.

Nutrient broth, or nutrient agar, is the medium most commonly used in bacteriology. The medium supports the growth of a large number of organisms, hence it is called all-purpose or general-purpose medium in contrast to selective or differential media. Selective media contain chemical substances, which will prevent growth of one or more groups of bacteria without inhibiting the growth of the desired one. Differential media contain certain chemicals, which permit the observer to differentiate between types of bacteria.

If a solid medium is desired, 15 g/l of agar may be added to the broth medium.

Although the primary constituent of agar is a complex carbohydrate, galactan, extracted from the marine alga of the genus *Gelidium*, it is not used as food by the vast majority of microorganisms but acts only as a solidifying agent. Agar goes into solution or melts when heated to nearly 100°C and remains liquid until cooled to about 43°C. Unlike gelatin, once solidified it must be reheated to approximately 100°C to cause liquefaction. You will pour some plates later in this lab.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Sterilization of media and equipment</TITLE>

<TEXT length="50">

In bacteriology, the terms "sterile" and "sterilized" mean free of all life including viruses. Culture media, with the exception of a few, must be sterilized in proper containers immediately after preparation. The sterilized medium must be kept in

sterile condition if it is to be of any service in bacteriological studies. Remember, bacteria and molds are always present in the water and on glass and other utensils. Storing non-sterile media in a refrigerator merely delays the growth of the organism. A number of different sterilization methods are described below.

Most media and equipment used in bacteriology are sterilized by steam sterilization. The principal lethal agent in steam sterilization is heat. Microorganisms are killed when their enzymes and the cellular proteins are irreversibly destroyed. The higher the temperature, the shorter the time required for sterilization of the media.

The autoclave is an instrument that uses steam sterilization to sterilize bacteriological media and surgical equipment. It is based on the same principle as the pressure cooker: the ability to increase temperature by building up steam pressure, thus preventing boiling. In normal steam sterilization, the temperature is kept at 121°C (at this temperature the pressure gauge should read between 15 and 16 pounds per square inch) for 15 to 20 minutes. Although a higher temperature can be obtained by increasing the steam pressure, further increase in temperature and time may drastically alter the ingredients in the media, thereby making them unsuitable for a particular use. The time necessary for complete sterilization will vary somewhat, according to the kind and amount of material to be sterilized. For equipment such as pipettes and petri plates, dry heat sterilization can be used. The temperature is kept at 170°C for one hour while equipment is in the hot air sterilizer. It is important to remember that most types of media, cloth, rubber, and other material would be destroyed by dry heat.

Many materials for example certain sugars and blood sera are destroyed by heating at temperatures normally used for sterilization. To sterilize such heat labile materials that are liquids or substances in solution, filtration may be used. The filters in the method remove bacteria in two ways: by the mechanical sieve-like action of the minute pores of the filter, and by the adsorption of the microbes to the filter because of the difference in their electrical charges. A filter widely used in microbiology is the membrane filter. The membrane filter is a cellulose or plastic membrane with a pore size sufficiently small (usually 0.45  $\mu$ m to 0.22  $\mu$ m) to trap and thereby remove bacteria and even viruses (0.22  $\mu$ m) from a liquid. These filters are used by being attached to a suction flask and having suction applied to draw the liquid materials through them leaving any contaminating microbes behind. Ethylene oxide and other gaseous vapors are becoming increasingly important as sterilizing agents. Although many laboratories do not have the specialized equipment necessary to demonstrate this method of gaseous sterilization, you should be aware of its use and advantages. The use of ethylene oxide vapors under pressure in special equipment resembling a modified autoclave is becoming a common method of "cold" sterilization.

Ethylene oxide is highly toxic for viral particles, bacterial, and fungal cells, and the most heat resistant bacterial endospores. As a sterilizing agent it is easy to handle with the proper equipment and is relatively inexpensive. Unlike most toxic chemicals, it is relatively non-corrosive and non deleterious to materials being sterilized. Residual amounts of it are readily removed by aeration. Although ethylene gas itself is flammable, 10% ethylene oxide with 90% carbon dioxide or mixtures with freon are not only effective sterilizing agents, but also nonflammable and nonexplosive. Its disadvantages include relatively long exposure periods for sterilization (several hours), reactivity with components of media and certain type of plastics, and the

residue of ethylene oxide that remains after sterilization, which must be removed by aeration or by allowing the sterilized material to stand.

Another method for the sterilization of certain materials (for example, pharmaceuticals, and these days occasionally the US mail) is irradiation that is, the use of high energy ionizing radiation. These radiations include gamma rays from a cobalt 60 or cesium 139 source and cathode rays from electron generators and accelerators. Irradiation with ultraviolet light is not a satisfactory means of sterilization because of the low penetrability of the wavelengths of the ultraviolet portion of the spectrum.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Serial Dilutions</TITLE>

<TEXT length="50">

This section contains just a few notes on what serial dilutions are some basic terms. Like so many other things in lab it will be easier to get a handle on them by doing them. Serial dilutions are a means of achieving a large dilution of concentration with a small amount of reagent and great accuracy. They are called serial dilutions because they are performed one after another in a step wise fashion. The previous dilution in the series is used to make the next one down. It should be clear that for this to work well you have to be careful and concentrate through out. Any mistake shows up in each subsequent step. If you mess up the first dilution all of the rest are wrong. The most commonly performed serial dilutions in biology are what we call 2 fold and 10 fold dilutions. A 2 fold dilution means that you are cutting the concentration in  $\frac{1}{2}$  with each dilution or step in the series (figure 5). So you as you see in Figure 5 after 5 dilution steps the concentration in that solution is  $\frac{1}{32}$  of the original concentration or we refer to it as a 32 fold dilution. Each step is a 2 fold dilution and cumulatively you get a 32 fold dilution. The same thing applies to a 10 fold dilution series. Each step dilutes the previous step to a  $\frac{1}{10}$  concentration. So after 5 of these steps you have a concentration that is  $\frac{1}{100,000}$  of the original. We would call this a 100,000 fold or 105 dilution. Serial dilution is a very convenient way to get solutions of low concentration, particularly using the 10 fold dilution series. Some times if a large dilution is necessary some 100 fold steps will be used. Operationally a 2 fold dilution is made by putting 1 part of the solution to be diluted into a total of 2 parts, cutting the concentration in  $\frac{1}{2}$ . This means mixing equal parts of the compound to be diluted and the diluent (the liquid, often water, that is being used to dilute). A 10 fold dilution is 1 part in a total of 10 parts or 1 part + 9 part diluent. When calculating an original concentration of a substance from a dilution we use a term we call the Dilution Factor (DF) which is equal to the fold of the dilution. Below is an example. You can see a video demonstration of 10 fold dilutions at

</TEXT>

<VIDEO height="315" width="560">

<http://www.youtube.com/user/WPI#p/c/E597F22DB929D8FD/18/18HgDPajsMI>

</VIDEO>

<TEXT length="6">

DF = # of fold dilution

Ex. 1:2 dilution=2 fold=DF of 2

1:4 dilution=4 fold=DF of 4

1:10 dilution=10 fold=DF of 10

1:100 dilution=100 fold=DF of 100

</TEXT>  
</ENTRY>  
<ENTRY>  
<TITLE>Collection and Analysis of Data</TITLE>  
<TEXT length="50">

Becoming a scientist entails learning a number of new things. It requires learning how to use equipment, perform procedures, design experiments and collect data. Collection of data is only the first part of what is done with the results of your experiments. As a scientist you also need to learn how to look at your data and find meaning and understand if it is convincing. We usually do this through the use of statistics. Starting with the previous lab and continuing through the term you will learn some very basic statistical methods for examining your data. Through this you will begin to see how to analyze your data and determine if it is meaningful and believable. Lastly, knowing what you are looking for in your data will also help you to appropriately design your experiment.

### Analysis of Variance

One of the most common statistical analyses used to test scientific hypothesis is the Analysis Of Variance or ANOVA. Actually, the analysis of variance comprises a whole group of related analyses, but the simplest one, and the one we will use here, is the one-way ANOVA. This tests the validity of a hypothesis that different treatments will all affect our dependent variable equally. This form of prediction is called the null hypothesis, that is, it predicts there will be NO effect on our dependent variable of changing our independent variable.

The outcome statistic of the ANOVA is a p value. Again remember that you are testing the null hypothesis, that there will be no difference in your dependent variables despite changes in your independent variables (sample treatments). If you DO observe a difference, the question that needs to be answered to determine whether that difference is "real", that is whether it is a result of treatment changes, or whether the differences observed in your data might be due to random chance alone. The p value provides us with a measure of that probability. If the chances of observing those differences when the change in treatment truly has no effect are less than 5%, then we say the differences you observe are statistically significant. In this case, the p value less than 0.05 (5%). This being the case, we then REJECT the null hypothesis. That is, we assert that the treatment change did, in fact, have an effect on the dependent variable.

As in always the case for statistical analyses, certain assumptions about your data must be met in order for the analysis to be valid. In this case, again, one of those is that your data are normally distributed. You should recall what this means from an earlier discussion of means and standard deviations. For our purposes here, we will assert that this assumption and others required for the ANOVA to be valid, are true. But in general, you must test the validity of those assumptions before proceeding with your ANOVA.

In this laboratory, you are testing the effect of various products on the survival of daphnia. Of course the underlying lesson is about toxicity and bioassays but to be able to say in a bioassay that different compounds are toxic or that one is more toxic than you need to see if the results are really due to the actual differences in toxicity or

random fluctuations. This is done using statistical tests. The ANOVA is a good method in this case because it allows you to compare the changes of a number of independent variables (products) at once. If you compare the survival at the same concentration for several products using ANOVA you will be able to say that there is definitely a difference in toxicity.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Procedure Pt.1</TITLE>

<TEXT length="20">

Part 1: Dilutions

In the back of the room you will find 8-10 household products to test for toxicity. You will be making serial dilutions of 3 of these to be used in the two bioassays so that you can test and compare their toxicity. For the purposes of getting meaningful class data you will be assigned one of the products numbered 1-4 to test. You are free to choose any of the others that you want to examine for your 2nd and 3rd products. The use of serial or stepwise dilutions will enable you to make accurate dilutions to low concentrations using only small volumes of product and reagents. You can see a video demonstration of 10 fold dilutions at

</TEXT>

<VIDEO height="315" width="560">

<http://www.youtube.com/user/WPI#p/c/E597F22DB929D8FD/18/l8HgDPajsMI>

</VIDEO>

<TEXT length="4">

You will be using micropipetors extensively this week. If you are unfamiliar with micropipetors or haven't used them for a while you should review the document on how to use them in the course info section of the website and in particular the demonstration video at

</TEXT>

<VIDEO height="315" width="420">

<http://youtu.be/FaK1v4N5tzA>

</VIDEO>

<TEXT length="50">

Prepare 10 fold serial dilutions using the following procedure:

1. Obtain a sample of 3 different products to test. Two are your choice so pick something interesting to you, the other will be assigned to you. Also be sure you have a bottle of 3% saline (salt water) and a bottle of pond/spring water, these will probably be at your bench already.
2. Obtain 5, 1.5 ml microfuge tubes for each product you will dilute and label them with the product name or some code of your choosing using a lab marker. Five for each product.
3. Then label one of each set of 5 with F (full strength), 10-1, 10-2, 10-3, and 10-4 to indicate the dilution in each one.
4. Set the tubes in a microfuge tube rack in the above order.
5. Using a micropipettor add 1 ml of product (X) to the appropriately labeled F tube. This will serve as your stock solution so you can return the class stock to the front of the room and is not really a dilution.
6. Add 900 µl of 3% saline to each of the 4 remaining tubes (10-1, 10-2, 10-3, and 10-4). The saline is your diluent (the "stuff" you are diluting with). Note: You will use saline for diluting the products to be tested on the bacteria since it is a marine

organism and pond water for the dilution of the one product you will test on the Daphnia.

7. To make your first 10 fold dilution transfer 100  $\mu$ l of the stock from the F tube into the 10-1 tube, cap the tube and mix using the vortex. (This is a 10 fold dilution because you now have 1 part, 100  $\mu$ l, in a total of 10 parts,  $900 + 100 \mu\text{l} = 1000 \mu\text{l}$ , the solution in the 10-1 tube is 1/10 its original concentration.

8. Make your second 10 fold dilution by transferring 100  $\mu$ l from the 10-1 tube into the 10-2 tube, capping the tube and mixing using the vortex. (This is again a 10 fold dilution because the solution in the 10-2 tube is 1/10 the concentration of that in the 10-1 tube. Since you have now done 10 fold dilutions serially, or stepwise, the solution in the 10-2 tube is 1/100 the concentration of the F or stock tube)

9. Make the 10-3 dilution by transferring 100  $\mu$ l from the 10-2 tube into the 10-3 tube, capping the tube and mixing using the vortex. [This creates a solution that is 1/1000 of the original stock,  $(1/10) (1/10) (1/10) = 1/1000$ ]

10. Make the 10-4 dilution by transferring 100  $\mu$ l from the 10-3 tube into the 10-4 tube, capping the tube and mixing using the vortex. [This creates a solution that is 1/10,000 of the original stock,  $(1/10) (1/10) (1/10) (1/10) = 1/10,000$ ]

11. Follow steps 4-10 above for each of your products using 3% saline. These will be for the Bacteria experiment.

12. Get 15 more microfuge tubes and repeat steps 3-10 for the products you have chosen and been assigned, only use pond/spring water as your diluent instead of saline. This will be for the Daphnia experiment.

13. See Fig 5 and Table 1 for further clarification on the dilution scheme.

Congratulations you have now mastered the useful and important skill of serial dilution. All serial dilutions no matter what the “fold” are variations on the above theme. Your dilutions are now ready to use for your bioassays.

</TEXT>

<TABLE rows="7" columns="3" title="Table 1: Dilution Scheme">

<ROW>

<COLUMN>Tube/dilution</COLUMN>

<COLUMN>Fractional dilution</COLUMN>

<COLUMN>% Concentration</COLUMN>

</ROW>

<ROW>

<COLUMN>F</COLUMN>

<COLUMN>0-no dilution</COLUMN>

<COLUMN>100</COLUMN>

</ROW>

<ROW>

<COLUMN> $10^{-1}$ </COLUMN>

<COLUMN>1/10</COLUMN>

<COLUMN>10</COLUMN>

</ROW>

<ROW>

<COLUMN> $10^{-2}$ </COLUMN>

<COLUMN>1/100</COLUMN>

<COLUMN>1.0</COLUMN>

</ROW>

<ROW>

<COLUMN> $10^{-3}$ </COLUMN>



```

        <COLUMN>1/1000</COLUMN>
        <COLUMN>0.10</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>10^-4</COLUMN>
        <COLUMN>1/10,000</COLUMN>
        <COLUMN>0.01</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>NC (negative control)</COLUMN>
        <COLUMN>No product added</COLUMN>
        <COLUMN>0</COLUMN>
    </ROW>
</TABLE>
<TEXT length="1">Figure 5: Dilution scheme</TEXT>
<IMAGE height="221">http://users.wpi.edu/~enniocd/dilution_scheme.png</IMAGE>
</ENTRY>
<ENTRY>

```

<TITLE>Procedure Pt.1</TITLE>

<TEXT length="3">  
 Part 2: Daphnia Bioassay

(Adapted from Ex. 39 Ecological Relationships, Explorations in Basic Biology 10th Ed., Stanely E. Gunstream Prentice Hall Pub. 2005, pp. 481-492)

</TEXT>

<TEXT length="15">

You want to set this assay up first because it needs to run for 60 minutes before you examine it for results. While it is running you can work on the rest of the lab. Of the two assays you will be using today this one is an example of a quantitative assay. The bacteria assay is more qualitative and will be discussed later. The Daphnia assay is quantitative because you will be able to count the Daphnia that are alive or dead. This like other experiments, particularly those involving live organisms, does not mean there is no subjectivity. You will see there is some interpretation of the affect of the compound on the Daphnia. You can observe a demonstration of the Daphnia assay at

</TEXT>

<VIDEO height="315" width="560">  
<http://youtu.be/3Dgqg9qOa5k>

</VIDEO>

<TEXT length="50">

Set up your assay as follows:

1. Obtain 16 12 X 100 mm test tubes from the table at the front of the lab
2. Add 900 µl of pond/spring water to each
3. With a marker (you are allowed to write on the tubes) or tape label 15 of them to match the three dilution series (F-10-4 )you created and the last NC for negative control. This tube will serve as the no product added tube so that you will have something to compare the survival of your other tubes to. It will have only diluent added, in this case pond/spring water, so that it is an exact duplicate of the others except it has no added toxins. This will show us if any of the mortality is due to just being in the small tube or from our water.

4. With a marker, put a mark at the meniscus of the liquid level in each tube. This will be important later to ensure that each tube has exactly the same volume of water in it and that you know what the volume is.
5. From the small tanks/jars in the back capture 5 Daphnia for each of your 16 tubes using a transfer pipette. Try to use animals that are of equal size. This is another control. It is best to have all animals as identical as possible. In our case size could play a big role in resistance to environmental toxins, If the animals aren't size matched it could skew your results. When you release the Daphnia into your tube be sure to do it beneath the surface of the liquid so you don't trap air under the animal's carapace which will cause it to float on the surface. Discard any floating animals and replace them. Note be sure to rinse the pipette between each transfer to avoid carrying toxic products into the stock cultures.
6. Using a micropipettor remove water from each tube until the liquid level is back to the line you marked previously. This will ensure that each tube has an equal and known volume of water in it. This allows you to know exactly what the concentration on product in the tube is.
7. Begin your assay by adding 100  $\mu$ l from each of your dilution tubes to the correspondingly labeled tube of Daphnia. Add 100  $\mu$ l of pond/spring water to the NC tube. Mix the solution by gently pipetting up and down. Record the time. Do not vortex. The Daphnia don't have crash helmets.
8. After 60 minutes observe each tube and record the number of daphnia surviving. If you think you can be consistent you may score some as being partially alive or affected as opposed to unaffected. Meaning you could have 2.5 or 2.75 surviving if you think you can make that judgment.
9. Record your results in Tables 2-4 and also record the data for your assigned solution in the class google doc table. Use the group number row that matches the number on the drawers in front of you and add your data to the appropriate column. If you are not sure how to log on see appendix II. For your lab report you will calculate the LC50 by plotting the percentage surviving vs product concentration. (Remember that because you put 100  $\mu$ l in a total of 1000  $\mu$ l you have actually performed another 10-fold dilution. This means that the actual concentration in your assay tube, the one that really counts, is 1/10 of what your dilution says. i.e. the full strength tube is really 10-1 and so forth.)
10. After your test is complete use a dissecting (stereo) microscope to sketch or take a photograph of one of the Daphnia in your NC tube. You may also want to include photos or sketches of examples of Daphnia from the other tubes if it will help support and explain your results.
11. After observing and recording one, return your Daphnia from the NC tube to the tank. Dispose of your other Daphnia test cultures in the blue capped biohazard waste jars in the center of each bench. Because these Daphnia may not be native to the area we as good, concerned biologists cannot allow them to be released into the wild.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Procedure Pt.3</TITLE>

<TEXT length="4">

Part 3: Bioluminescence Assay

(Adapted from Ex. 19 Using Biotechnology to Assess Ecosystem Damage, Thinking About Biology an introductory Laboratory Manual, 2nd Ed. Mimi Bres and Arnold Weisshaar, Prentice Hall Pub.,2005, pp 367-382 )

</TEXT>

<TEXT length="20">

You will test the environmental toxicity of 3 products using this assay. These should be the same as the ones you tested on the Daphnia so that you can compare the results. This assay is a qualitative assay. This is because you will be scoring the light emitted by different cultures on a 0-5 scale. This makes the assay subjective and based somewhat variable from observer to observer. This does not make it an invalid test. Many tests can only be run this way. This bioassay is only qualitative because we do not have the instrument required to accurately measure the light output, called a luminometer. The light output from these bacteria is very low and requires a luminometer to quantify it. A normal light meter is not sensitive enough. The human eye can be very sensitive and you will be able to see the light emitted by the cultures after first acclimatizing your eyes in a completely dark room for 5 minutes. We will also try to make this a little easier by taking long exposure photographs that should be able to capture the light signal produced. Set up your bioluminescence assay using the following procedure:

1. Keep in mind that you are working with bacteria so we will be operating using aseptic technique (see background above)
2. Before you begin the experiment make a prediction and form a hypothesis about what you believe will happen and which compound you think will be the most toxic. Record this in your notebook and be sure to include it in your report.
3. Obtain a working stock of bacteria from the class stock culture flasks by aseptically transferring 20 ml into a sterile 50 ml conical tube (This may already be done for you). Remember if it is presterilized and plastic you don't need to flame it.
4. Setup 6 assay tubes for each product you are testing using sterile 1.5 ml microfuge tubes. Label them with the product being tested or a representative code and for each concentration F, NC, and  $10^{-1}$ - $10^{-4}$ . Place the tubes in a rack
5. Using a sterile micropipette tip aliquot 900  $\mu$ l of bacteria suspension into each tube.
6. Begin the assay by pipetting 100  $\mu$ l of each of the product dilutions in saline as well as the full strength solution into the matching appropriately labeled assay tube. Be sure to add 100  $\mu$ l of saline to the NC tube.
7. Cap and gently vortex each tube to be sure it is mixed. Record the time.
8. Be sure you have all of your tubes in some kind of order you will know without being able to read the labels. You will be observing them in the darkroom and you will probably not be able to read the labels.
9. The assay needs to run for 15 minutes. Your eyes need to adjust in the dark room for 5 minute so go to the dark room with your tubes after 10 minutes of incubation. Do not wear gloves out of the lab. If you are wearing gloves remove them before you leave and place them in the biohazard waste.
10. The dark room is just down the hall. The TAs will show you where it is and how it is setup at the beginning of class. Be sure to take table 3 or your notebook with you to record your results. You should be able to see well enough to see because the red light tin the dark room will be on. It does not interfere with viewing the light emission but should allow you enough light to write.
11. Enter and leave the darkroom by the revolving door so that you don't shine light into someone's' already light adapted eyes. Be careful because there may already be groups in the dark room. It is not very big and may get a little crowded.

12. After approximately 5 minutes in the darkroom observe all of your tubes and record the results. Score the light emission on a scale of 0-5. Use the NC tubes as the basis for a 5 since they should be the maximum light producers.
13. Then mix all of your tubes by inverting then several times. You can probably do all of them at once in the rack if you are careful. Observe your tubes and again score them for light emission on a scale of 0-5
14. Be sure to exit the darkroom through the revolving door. We don't want to blind anybody.
15. Add your data to the master list if you have been told to do so.
16. For your lab report you will graph dilution/concentration of product vs. light intensity. This will allow you to compare toxicities and calculate the LC50 for each.
17. Dispose of your tubes in the biohazard bins on the bench or the big ones on the floor.
18. Clean up your work area and be sure to wipe down your bench with ethanol. Wash your hands before leaving. This is always good practice but is particularly important when working with biological agents, no matter how safe.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Procedure Pt.4</TITLE>

<TEXT length="15">

#### Part 4: Creating a Bacterial Stock by Streaking

When running experiments that require bacteria it is common practice to make a number of different kinds of master stocks so you can go back to them and grow up more bacteria when you need it. Each type has a varying lifetime and is used for different kinds of master stocks. Some times frozen stocks are made and stored at -80°C, these can last for many years to indefinitely. Sometimes agar slants are made by putting solid media in a screw cap tube and inoculating it with bacteria. This can be kept several years at 4°C. Another simple method that is often used is streaking a plate of solid medium. This culture is grow first at the optimum temperature for the organism in question and will be viable for several months at 4°C. A streak plate also allows you to see and use isolated colonies, which can be very useful when looking for contamination of your cultures or doing genetic experiments. This is the type of culture you will be making this week. You can see a demonstration of plate streaking at

</TEXT>

<VIDEO height="315" width="560">

<http://youtu.be/LTs7vkHp4Xo>

</VIDEO>

<TEXT length="20">

1. Obtain one Luminescence plate for each member of your group as well as a master plate of previously streaked bacteria.
2. Each person should streak a plate by following the directions in the background above.
3. Be sure to label your plate with your name, the date, and the organism on the back. There is a reason for labeling the back, what do you think it is?

</TEXT>

<TEXTINPUT length="5">There is a reason for labeling the back, what do you think it is?</TEXTINPUT>

<TEXT length="5">

4. Place your plate back up in the 30°C
5. Check your plate in 24 and 48 hours to look for growth of bacterial colonies. Does your plate resemble the master plate? Did you get isolated colonies? Are there any spots or colonies that appear odd that might be contamination? The digital camera will be available if you would like to record your results photographically.

</TEXT>

<TEXTINPUT length="5">Does your plate resemble the master plate?</TEXTINPUT>

<TEXTINPUT length="5">Did you get isolated colonies?</TEXTINPUT>

<TEXTINPUT length="5">Are there any spots or colonies that appear odd that might be contamination?</TEXTINPUT>

<TEXT length="5">

6. When everyone has observed the plate dispose of it in the large biohazard bins on the floor.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Procedure Pt.5</TITLE>

<TEXT length="12">

#### Part 5: Safety Scavenger Hunt

Find the phone, the emergency number, eyewash (all 3), safety shower, fire extinguisher, MSDS/safety material, and first-aid kit and write out what or where they are. A diagram could be useful for this. Also, not including jumping out the window, using the elevator or jumping off the building, find 4 different routes down from the 2nd floor and out of the building (this does not include anything about jumping from anything). Write out a description or use a map for each location or route. You will turn this in as part of your lab report.

</TEXT>

<TEXTINPUT length="5">Description</TEXTINPUT>

<TEXT length="1">

**REMEMBER TO CLEAN UP YOUR BENCH AND PUT AWAY ANY EQUIPMENT OR SUPPLIES THAT YOU USED! Thank You**

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Appendix 1: Data tables</TITLE>

<TEXT length="1">

the following tabs hold the tables for holding your data.

</TEXT>

</ENTRY>

<ENTRY>

<TITLE>Table 2</TITLE>

<TABLE rows="7" columns="7" title="Table 2:Daphnia Survival after Exposure to (solution name)">

<ROW>

<COLUMN>Tube/dilution</COLUMN>

<COLUMN>Fractional dilution</COLUMN>

<COLUMN>% Concentration</COLUMN>

<COLUMN>Actual Dil.</COLUMN>

<COLUMN>Actual concentration</COLUMN>

<COLUMN># of surviving Daphnia</COLUMN>

```

        <COLUMN>% survival</COLUMN>
</ROW>
<ROW>
    <COLUMN>F</COLUMN>
    <COLUMN>0-no dilution</COLUMN>
    <COLUMN>100</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-1</COLUMN>
    <COLUMN>1/10</COLUMN>
    <COLUMN>10</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-2</COLUMN>
    <COLUMN>1/100</COLUMN>
    <COLUMN>1.0</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-3</COLUMN>
    <COLUMN>1/1000</COLUMN>
    <COLUMN>0.10</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-4</COLUMN>
    <COLUMN>1/10,000</COLUMN>
    <COLUMN>0.01</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>NC (negative control)</COLUMN>
    <COLUMN>No product added</COLUMN>
    <COLUMN>0</COLUMN>

```

```

        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
</TABLE>
</ENTRY>
<ENTRY>
    <TITLE>Table 3</TITLE>
    <TABLE rows="7" columns="7" title="Table 3:Daphnia Survival after Exposure to (solution
    name)">
        <ROW>
            <COLUMN>Tube/dilution</COLUMN>
            <COLUMN>Fractional dilution</COLUMN>
            <COLUMN>% Concentration</COLUMN>
            <COLUMN>Actual Dil.</COLUMN>
            <COLUMN>Actual concentration</COLUMN>
            <COLUMN># of surviving Daphnia</COLUMN>
            <COLUMN>% survival</COLUMN>
        </ROW>
        <ROW>
            <COLUMN>F</COLUMN>
            <COLUMN>0-no dilution</COLUMN>
            <COLUMN>100</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
        </ROW>
        <ROW>
            <COLUMN>10-1</COLUMN>
            <COLUMN>1/10</COLUMN>
            <COLUMN>10</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
        </ROW>
        <ROW>
            <COLUMN>10-2</COLUMN>
            <COLUMN>1/100</COLUMN>
            <COLUMN>1.0</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
        </ROW>
        <ROW>
            <COLUMN>10-3</COLUMN>
            <COLUMN>1/1000</COLUMN>
            <COLUMN>0.10</COLUMN>

```

```

        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>10^-4</COLUMN>
        <COLUMN>1/10,000</COLUMN>
        <COLUMN>0.01</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>NC (negative control)</COLUMN>
        <COLUMN>No product added</COLUMN>
        <COLUMN>0</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
</TABLE>
</ENTRY>
<ENTRY>
    <TITLE>Table 4</TITLE>
    <TABLE rows="7" columns="7" title="Table 4:Daphnia Survival after Exposure to (solution
name)">
        <ROW>
            <COLUMN>Tube/dilution</COLUMN>
            <COLUMN>Fractional dilution</COLUMN>
            <COLUMN>% Concentration</COLUMN>
            <COLUMN>Actual Dil.</COLUMN>
            <COLUMN>Actual concentration</COLUMN>
            <COLUMN># of surviving Daphnia</COLUMN>
            <COLUMN>% survival</COLUMN>
        </ROW>
        <ROW>
            <COLUMN>F</COLUMN>
            <COLUMN>0-no dilution</COLUMN>
            <COLUMN>100</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
            <COLUMN>@BLANK</COLUMN>
        </ROW>
        <ROW>
            <COLUMN>10^-1</COLUMN>
            <COLUMN>1/10</COLUMN>
            <COLUMN>10</COLUMN>

```



```

        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>10^-2</COLUMN>
        <COLUMN>1/100</COLUMN>
        <COLUMN>1.0</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>10^-3</COLUMN>
        <COLUMN>1/1000</COLUMN>
        <COLUMN>0.10</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>10^-4</COLUMN>
        <COLUMN>1/10,000</COLUMN>
        <COLUMN>0.01</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
        <COLUMN>NC (negative control)</COLUMN>
        <COLUMN>No product added</COLUMN>
        <COLUMN>0</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
        <COLUMN>@BLANK</COLUMN>
    </ROW>
</TABLE>
</ENTRY>
<ENTRY>
    <TITLE>Table 5</TITLE>
    <TABLE rows="7" columns="7" title="Table 5:Bioluminescent Intensity After Exposure to
(solution name)">
        <ROW>
            <COLUMN>Tube/dilution</COLUMN>
            <COLUMN>Fractional dilution</COLUMN>
            <COLUMN>% Concentration</COLUMN>

```

```

<COLUMN>Actual Dil.</COLUMN>
<COLUMN>Actual concentration</COLUMN>
<COLUMN>Light Rank after agitation (0-5)</COLUMN>
<COLUMN>Light emission as % NC From light rank after
agitation</COLUMN>
</ROW>
<ROW>
<COLUMN>F</COLUMN>
<COLUMN>0-no dilution</COLUMN>
<COLUMN>100</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
<COLUMN>10-1</COLUMN>
<COLUMN>1/10</COLUMN>
<COLUMN>10</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
<COLUMN>10-2</COLUMN>
<COLUMN>1/100</COLUMN>
<COLUMN>1.0</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
<COLUMN>10-3</COLUMN>
<COLUMN>1/1000</COLUMN>
<COLUMN>0.10</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
<COLUMN>10-4</COLUMN>
<COLUMN>1/10,000</COLUMN>
<COLUMN>0.01</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
<COLUMN>@BLANK</COLUMN>
</ROW>

```

```

    <ROW>
      <COLUMN>NC (negative control)</COLUMN>
      <COLUMN>No product added</COLUMN>
      <COLUMN>0</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>100</COLUMN>
    </ROW>
  </TABLE>
</ENTRY>
<ENTRY>
  <TITLE>Table 6</TITLE>
  <TABLE rows="7" columns="7" title="Table 6:Biolumenescent Intensity After Exposure to
  (solution name)">
    <ROW>
      <COLUMN>Tube/dilution</COLUMN>
      <COLUMN>Fractional dilution</COLUMN>
      <COLUMN>% Concentration</COLUMN>
      <COLUMN>Actual Dil.</COLUMN>
      <COLUMN>Actual concentration</COLUMN>
      <COLUMN>Light Rank after agitation (0-5)</COLUMN>
      <COLUMN>Light emission as % NC From light rank after
      agitation</COLUMN>
    </ROW>
    <ROW>
      <COLUMN>F</COLUMN>
      <COLUMN>0-no dilution</COLUMN>
      <COLUMN>100</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
      <COLUMN>10-1</COLUMN>
      <COLUMN>1/10</COLUMN>
      <COLUMN>10</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
    </ROW>
    <ROW>
      <COLUMN>10-2</COLUMN>
      <COLUMN>1/100</COLUMN>
      <COLUMN>1.0</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
    </ROW>
  </TABLE>
</ENTRY>

```

```

</ROW>
<ROW>
  <COLUMN>10^-3</COLUMN>
  <COLUMN>1/1000</COLUMN>
  <COLUMN>0.10</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
  <COLUMN>10^-4</COLUMN>
  <COLUMN>1/10,000</COLUMN>
  <COLUMN>0.01</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
<ROW>
  <COLUMN>NC (negative control)</COLUMN>
  <COLUMN>No product added</COLUMN>
  <COLUMN>0</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>@BLANK</COLUMN>
  <COLUMN>100</COLUMN>
</ROW>
</TABLE>
</ENTRY>
<ENTRY>
  <TITLE>Table 7</TITLE>
  <TABLE rows="7" columns="7" title="Table 7:Bioluminescent Intensity After Exposure to
  (solution name)">
    <ROW>
      <COLUMN>Tube/dilution</COLUMN>
      <COLUMN>Fractional dilution</COLUMN>
      <COLUMN>% Concentration</COLUMN>
      <COLUMN>Actual Dil.</COLUMN>
      <COLUMN>Actual concentration</COLUMN>
      <COLUMN>Light Rank after agitation (0-5)</COLUMN>
      <COLUMN>Light emission as % NC From light rank after
      agitation</COLUMN>
    </ROW>
    <ROW>
      <COLUMN>F</COLUMN>
      <COLUMN>0-no dilution</COLUMN>
      <COLUMN>100</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
      <COLUMN>@BLANK</COLUMN>
    </ROW>
  </TABLE>

```

```

        <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-1</COLUMN>
    <COLUMN>1/10</COLUMN>
    <COLUMN>10</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-2</COLUMN>
    <COLUMN>1/100</COLUMN>
    <COLUMN>1.0</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-3</COLUMN>
    <COLUMN>1/1000</COLUMN>
    <COLUMN>0.10</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>10^-4</COLUMN>
    <COLUMN>1/10,000</COLUMN>
    <COLUMN>0.01</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
</ROW>
<ROW>
    <COLUMN>NC (negative control)</COLUMN>
    <COLUMN>No product added</COLUMN>
    <COLUMN>0</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>@BLANK</COLUMN>
    <COLUMN>100</COLUMN>
</ROW>
</TABLE>
</ENTRY>
<ENTRY>
    <TITLE>Appendix II: Using google docs</TITLE>

```

<TEXT length="3">  
WPI Biology Department  
Instructions for use of Google Docs lab data collection  
1. Open Microsoft Internet Explorer  
</TEXT>  
<IMAGE height="111">  
http://users.wpi.edu/~enniocd/internet\_explorer.png  
</IMAGE>  
<TEXT length="1">  
2. Go to <http://docs.google.com>  
</TEXT>  
<IMAGE height="111">  
http://users.wpi.edu/~enniocd/google\_search.png  
</IMAGE>  
<TEXT length="5">  
3. Enter in the username and password for the section. Username will be  
“WPI.BB” plus the two-digit section number as posted on the board;  
password is “password” plus the two-digit section number.  
</TEXT>  
<IMAGE height="111">  
http://users.wpi.edu/~enniocd/google\_login.png  
</IMAGE>  
<TEXT length="4">  
4. Insure “Stay signed in” is not selected and click “Sign in”  
5. Once signed in, a list of lab documents should appear. Click the document  
for particular lab to open in new window.  
</TEXT>  
<IMAGE height="111">  
http://users.wpi.edu/~enniocd/document.png  
</IMAGE>  
<TEXT length="4">  
6. Your row assignment is the number designated on the drawers at your  
bench. For example,  
Bench number “4” will use group number 4  
</TEXT>  
<IMAGE height="111">  
http://users.wpi.edu/~enniocd/row\_assignment.png  
</IMAGE>  
<TEXT length="2">  
7. Enter lab data in the appropriate cells.  
8. Once complete, go to File Save and close  
</TEXT>  
<IMAGE height="111">  
http://users.wpi.edu/~enniocd/file\_save.png  
</IMAGE>  
<TEXT length="20">

Troubleshooting  
I can't login.

Make sure caps lock is not enabled and that you are using the right section password and username. If problem continues, log out and close then reopen your browser. If problems persist, contact your lab instructor.

I can't see the lab documents.

Make sure that you are logged into the Documents application of Google. You can verify this by clicking on the "Documents" link located on the top of your screen. Make sure that you are also viewing "All items" located on the left side of the screen.

I am unable to input data.

Refresh the page. If problem continues, log out and close then reopen your browser. Insure that you are using a valid username and password.

My data is disappearing.

Insure that you are using the correct row. Insure you have adequate internet connection. Attempt to refresh the page, close the document, and reopen

</TEXT>

</ENTRY>

</LAB>