

# Physics and Education Interdisciplinary Qualifying Project

Terms A09, B09, C10

Christopher Wellington, Justin Pombrio, and Guy Mann

## **Abstract**

Our purpose was to create tools for the WPI Physics Department to facilitate learning through the web. We designed Java Applets as web based learning tools, redesigned the layout of the lab website, and created a tool for managing the site's data and design. With these tools the WPI Physics Department will be better able to serve the students' needs through the medium of the web.

## Table of Contents

Abstract .....	2
Table of Contents .....	3
Introduction .....	4
Chapter 1 – Applet Development.....	5
Chapter 2 Redesign of WPI Physics Lab Site and Maintenance Software Package.....	9
Conclusion.....	18
Appendix A - Proposal for Physics and Education Interdisciplinary Qualifying Project .....	19
Appendix B - Suggested Improvements for Existing Web Applets.....	24
Appendix C - Review of Lab Content.....	26
Appendix D - Student Survey .....	27
Appendix E - Summary of Survey Responses .....	29
Appendix F - Potential Changes to Applets .....	32
Appendix G - Database Design .....	33
Appendix H- Website Builder Manual.....	48
Appendix I - Database Diagram .....	80
Appendix J - UI Diagram .....	81
Appendix K- Timeline.....	82
Attachments (contained in ZIP archive of project):	
WebBuilder Executable (WebBuilder.jar)	
WebBuilder Source Code (WebBuilder.zip)	
Database Diagram (Database_Design.jpg)	
User Interface Diagram (UI_diagram.jpg)	
Applet Source Code (AppletSource folder)	
Applet Executables (Applets folder)	
Screenshots of Original Lab Site (Original_Lab_Site_Screenshots.zip)	
Screenshots of Redesigned Lab Site (Redesigned_Lab_Site_Screenshots.zip)	

## **Introduction**

The purpose of our project was to create tools for the the WPI Physics Department to facilitate learning through the web. To this end we split into two groups and tackled the problem from different angles. The group comprised of Justin and Christopher focused on creating Java Applets, while Guy worked on a redesigned web layout for the lab site and a tool for maintaining and updating the data. Our plan for accomplishing these goals is outlined in our proposal in Appendix A.

The applets were created as web based interactive software simulations to model physics problems. They prompt the user to solve randomly generated problems in order to engage the student and improve the learning experience. The redesign of the website allows the users to obtain the vital course information through a more intuitive and user focused layout. To maintain our design a software package was built to manage the information about the site and provide the site maintainer the ability to work with the data. The software package regenerates from that data a website based on our redesign.

# Chapter 1 - Applet Development

Terms A09, B09, C10

By Chris Wellington and Justin Pombrio

## Planning Phase

Applet development was the part of the project we pursued. We wanted to create a well-rounded set of applets that would be useful and interesting to physics students. This consisted of both improving existing applets and developing new ones. It was clear that the existing applets needed improved interfaces and layouts and more understandable explanations of the concepts being demonstrated. As we stated in the proposal, there were also other parts of the physics curriculum in which applets could effectively help students learn. Our plan was to focus on easily visualized concepts in the PH1110 and PH1120 courses and develop a set of tools and simulations that would complement a textbook or lecture. To accomplish this, we proposed interviewing lab staff and faculty in order to identify the areas in which students most often have difficulty. We could then focus our attention on topics that were both challenging for students and lent themselves to simulation.

We took an in-depth look at the existing applets before we began improving them. We noticed that most had poor and inconsistent layouts. For example, some had several text input fields arranged horizontally in a row, instead of vertically, as would normally be expected. Additionally, none of the applets could be re-sized. While these may seem to be minor details, a poor interface can immediately turn off users and make the applet difficult to use. We were also surprised by some of the behavior we noticed during testing. The Powers of Ten applet was a matching game which taught the prefixes for different powers of ten. Students were given a power of ten and asked to select the button with the correct prefix. Unexpectedly, the buttons sometimes had to be pressed more than once, and more than one button could be selected at a time. We compiled a list of the problems we noticed along with potential solutions to them, which can be found in Appendix B. Instead of modifying the existing code, which would have been difficult because of the way it was written, we decided to rewrite new versions of the existing applets.

We also wanted to develop new applets to augment the existing ones. As proposed, we sent out a survey to lab staff and faculty asking them to identify areas in the curriculum where they noticed students repeatedly having difficulty. Despite the brief nature of the survey, we received only two responses. A graduate student suggested we focus on simulations that show the evolution of a physical system over time. One physics professor said that she always sees students having trouble with basic math skills, especially trigonometry, vector arithmetic, and solving systems of equations. The only applet we ended up developing to teach math skills was one showing vector addition. We entertained ideas for other applets to help with math skills, but decided instead to focus on concepts specific to physics.

Later in the year, we realized that we needed to figure out how the applets we had been developing could fit into the physics curriculum. A number of possibilities were suggested by Professor Iannacchione. We ultimately decided that students would benefit most if the applets were each tied to a specific lab. We had the further idea to incorporate questions, which could be used as pre-lab exercises, into some of the applets. To determine which labs our applets

could be linked to, we took an in-depth look at the lab procedures for the introductory physics courses. PH1120 introduces electricity and magnetism, but its labs already make use of a software tool for modeling charged particles called EM Field. PH1130 is a course introducing twentieth century physics, primarily special relativity and quantum mechanics, while PH1140 introduces waves and oscillations. Applets could potentially aid in the learning of some of the topics associated with these courses. We determined, however, that it was too ambitious to start developing them at the time. This left the PH1110 curriculum, which teaches classical mechanics. At this point we had written applets which fit with three of its labs, namely the ones that dealt with simple harmonic motion, collisions, and rotational motion. Several of the other labs dealt with the acceleration of a cart on a ramp, so we decided to develop an applet to simulate this. Our notes about the lab contents can be found in Appendix C.

## Execution Phase

We began by writing a common set of tools that all of the applets would be built from. This allowed us to keep their design consistent, and have them share as much code as possible. We developed four sets of tools. First, we made the observation that most applets could be thought of as consisting of three parts; a model, a view, and a controller. The model is responsible for keeping track of what goes on behind the scenes. It stores values relevant to the simulation and performs necessary calculations. The view is responsible for drawing to the screen. It gets the current state of the simulation from the model, and draws a graphical representation of it. The controller is responsible for displaying a set of controls and getting input from the user. When the user inputs data, the controller passes it on to the model so that it can be reflected in the simulation. The "Model, View, Controller" separation is a common theme in software design. We chose to use it primarily to keep our code organized in a logical way. To implement this, Justin wrote a set of three Java Interfaces to be used by each applet's model, view, and controller. The interfaces specified functionality that all models, all views, and all controllers would share.

Second, Justin wrote a Graph class that stands between each applet and the window it draws on. It allows each applet to define its own coordinate system, rather than use the default one. In the default coordinate system, the origin is in the upper-left corner of the screen and distances are measured in pixels. Having a custom coordinate system also enables the window to be re-sized easily. The Graph class also lets each applet define a number of Plots, each of which is responsible for drawing one or more components of the display. For instance, we created a plot that draws the curve defined by a list of points. This plot could then be modified or added to a Graph as a single entity. Other useful plots include one to draw a single point, one to draw an arrow, and one to draw labeled or unlabeled axes.

Third, we jointly designed a class called ControlPanel that makes it easy for each applet to lay out its controls. The applet specifies what controls it wants, and the ControlPanel determines how to lay them out. This keeps the applets visually consistent. It also allows an applet to change a control from one type to another, such as from a text field to a scroll-bar, by changing just one line of code. The Graph and ControlPanel classes captured a lot of code that would otherwise be duplicated for each applet, and made them more easily modifiable.

Fourth, we wrote a class to generate and display problems to the user. The Problem class first displays a problem in a pop-up window. When the user enters an answer, it checks it against the correct answer. If the answer is correct, it shows a dialog congratulating them.

Otherwise it gives them the option to try again or to cancel. Unlike the Graph and ControlPanel classes, the purpose of the Problem class is to extend the functionality that an applet can have. The inclusion of problems in two of the applets played an important role in making them useful to students.

While designing the tools in A and B term, we simultaneously wrote some simple applets to make use of them. In fact, most of our ideas for tools came from this development. The four applets we developed during this initial period were the Vector Arithmetic, Hooke's Law, Projectile Motion, and Rotational Motion applets. The first three of these were just reimplementations of existing ones. In Vector Arithmetic, the user picks two vectors by clicking within a region, and their sum and dot product are displayed both graphically and in component form. Hooke's Law simulates a mass hanging from a spring in simple harmonic motion. As the mass oscillates, it displays a graph of its displacement versus time. The Projectile Motion applet lets the user choose initial positions and velocities for projectiles, and then launch them. Rotational Motion allows the user to specify a radius and a period, and shows an arrow rotating with the given radius and period. Justin wrote the vector and projectile applets, while Christopher wrote the spring and rotational ones. While these initial applets were well written and improved upon their existing counterparts, in the end they seemed more like toys than useful learning instruments.

In order to make the applets more useful, we had them generate questions. We developed two new applets during C term that generate and display problems to the student, called Collision and Inclined Plane. As mentioned before, many of the PH1110 labs deal with the acceleration of a cart on a ramp. This led to our development of the Inclined Plane applet. It lets the user specify attributes such as the inclination of the ramp, static and kinetic friction, and the acceleration due to gravity. It then displays the forces acting on the cart. The Collision applet, on the other hand, was another reimplementations of an existing applet. It simulates the collision between two bodies with user-specified masses, initial velocities, and coefficient of restitution. Besides showing the bodies colliding graphically, it also displays their final velocities and the total energy and momentum before and after the collision. The problems these applets generate vary in difficulty, but were written to be representative of the types of problems students see in class. For instance, the easiest problem for the Inclined Plane applet asks the user to calculate the acceleration of a cart on a ramp with no friction, while the hardest requires solving a more complicated system of equations.

## **Evaluation Phase**

After completing the applets, we took a number of steps to evaluate our work. Most importantly, we conducted a survey to assess the Inclined Plane and Collision applets. We chose to evaluate only these two applets because they were the most developed and complex ones. The goal of the survey was to collect a modest amount of feedback from perspectives other than our own. We were primarily interested in finding problems that we were unable to notice, such as unintuitive aspects of the interface or incorrect behavior. We also wanted to see how the applets related to the physics course; for instance whether the problems were relevant to what students encountered in class, and how they could be incorporated into the curriculum. We wrote a survey to address these questions. We administered it first in person during two two-hour sessions and then through an online survey service called SurveyMonkey. As an

incentive to students, we held a drawing for a thirty dollar Amazon.com gift card. The full text of the survey can be found in Appendix D.

Altogether, we received seven responses. The majority of respondents found the controls of the two applets intuitive. The primary exception was that some students did not realize they needed to press enter after changing values. The only other unexpected behavior was a bug that caused the cart to move up the ramp under certain circumstances, which we quickly fixed. We found that students were generally receptive to the idea of using the applets as a graded pre-lab activity. They thought that the combination of a simulation and a randomly generated problem was an effective way of teaching the concept. One student did point out that the problems could feel repetitive if the concept is already familiar, and that it would add another assignment to an already busy work load. An overview of the survey responses can be found in Appendix E.

In response to this feedback from the survey, we made a list of potential improvements. In the limited amount of time we had left, we implemented the improvements that were both important and easy to make. These included adding help text that tells the user to press enter after changing a value, fixing the bug in the Inclined Plane applet, and showing the equations window while a problem is being displayed. Two changes we thought would have been valuable but did not make were adding hot-keys, and allowing text fields to be modified without pressing enter. The complete list of potential changes, and whether we implemented them, can be found in Appendix F.

We also made a number of changes which will not affect the applets from the user's perspective, but will instead make it easier for future IQP students to improve upon our work. These changes include updating all the applets to use the latest versions of the Model, View, Controller, and ControlPanel classes, as well as improving code documentation. During the initial stages of applet development, we created improved versions of some of these tools, but not all of the applets were brought up to date. We wanted all the applets to use the same versions of the tools in order to reduce the amount of code that needs to be read and understood by future IQP students. Our improved documentation will also help with this.

We hope future IQP students will continue our work. We did not implement all of the potential changes we identified, or develop all of the applets we thought would be useful. We were also unsure whether certain recommendations from the survey would have been beneficial or detrimental if they were actually implemented. A more thorough study, however, would be able to distinguish between what individual students want and what the majority wants. Any of these areas could provide the basis for future work.



# Chapter 2 - Redesign of WPI Physics Lab Site and Maintenance Software Package

Terms A09, B09, C10

By Guy Mann

## 1.0 Table of Contents

- 1.0 Table of Contents
- 2.0 Executive Summary
- 3.0 Statement of Needs
  - 3.1 Web Design
  - 3.2 Maintenance of Website
- 4.0 Project Description
  - 4.1 Web Design
    - 4.1.A Navigation
    - 4.1.B Layout
    - 4.1.C Calendars
  - 4.2 Website Builder Software
    - 4.2.A Database Design
      - 4.2.A.i Pages
      - 4.2.A.ii Documents
      - 4.2.A.iii Courses
      - 4.2.A.iv Menu
    - 4.2.B UI Design
    - 4.2.C XML Generation
    - 4.2.D Website Generation from XML
- 5.0 Conclusion

## 2.0 Executive Summary

The WPI Physics Lab Website was in dire need of redesign. The previous design gave only a single entry point for navigation of the site and did not have a consistent visual style. A redesign to add navigation bars to pages and make the layout more readable was not very complex but it was far more complex than what had previously existed. As the saying goes, "teach a man to fish", if we did a redesign on the website and then it was maintained by someone who did not understand it then the benefits would be quickly depleted by broken links and nonuniform designs. To solve this problem we created a content management system for building and maintaining the WPI Physics Lab Website. The program allows direct HTML manipulation when needed but provides for HTML free sailing by including a WYSIWYG (What You See Is What You Get) interface as well as doing all the heavy lifting of creating schedule pages, maintaining link consistency for the menus and keeping track of documents. The software package is a single operating system agnostic executable which stores website data in a database and generates a static website which can be uploaded to a webserver and used. By doing the redesign we are able to increase the usefulness of the tools provided to

students and with the software package the new accessible design is maintainable by a user without web design or programming knowledge.

### **3.0 Statement of Needs**

#### **3.1 Web Design**

We found the WPI Physics Department Labs website in a sad state. The site was a collection of disparate pages strung together using a single page, like stones on a thong. The site was the bare minimum which could have been done to display the needed information on the internet and have users navigate to it. The site shows a lack of thought about layout or design. There was no thought to the need to support deep linking or that color choice is important to page readability. Because none of these considerations were taken into account the website is difficult to navigate and not as useful or helpful as it could be.

The navigation of the site is only possible from a single page and once you have navigated from there, there is often no link to go back. This is a big problem because not only does it make the page more difficult to navigate but it also means that any user linked to any page on the site other than the main page is not provided with a way to navigate to other pages. The concept of deep link, which is the idea that someone may enter your site on any page, suggests that to give the most usability to the site navigation must be present on every page.

The lack of visual consistency between pages was also in issue with the site. Some pages contained headers, while other did not. Some pages had background color, while other did not. Some pages had their text centered, while other did not. This inconsistency can be confusing to users and give them a less pleasant and useful experience on the site.

There are 2 types of schedules seen on the site a personal schedule and course schedule. Both are hard to read and interpret and the course schedule uses eye wrenching colors which seem to compel the viewer to look away. The course calendars were designed as tables but the day cells were not consistent in their size and so the calendar took more time than necessary to interpret and read. These are issues of layout and forethought, both of which can be remedied with a well thought out design.

#### **3.2 Maintenance of Website**

Once we understand the reasons for the lack of proper design for the website then it behooves us to address the underlying problem. This problem is that the person tasked with maintaining the website has only the most rudimentary knowledge of HTML and web design which makes designing and maintain a complex design impossible. In a corporate environment this would be solved by hiring someone with the appropriate skills to do this task. In the world of academics this task is merely one of the many chores of an academician with too many duties within his subject matter to have the time to learn a whole new one in depth. So instead of bringing the mountain to Moses we must instead bring Moses to the mountain; which is to say we need a tool that will allow the data on the website to be maintained by a nontechnical user.

The usual technical solution for a project of this kind would be to build a dynamic website using a server side language, such as PHP or ASP, with a database backend which a nontechnical administrator could login to and update as needed. The difficulty with this

approach was that the WPI CCC has an inhibition against dynamic applications on campus servers which they don't maintain. This left us the choice of setting up a new server and linking there from the current webspace. This is a poor choice. An equally unappealing option was begging the CCC for permission. Begging did not seem dignified. Left with only one solution to the problem of maintaining the website design and content, which was to make it work, we made a pot of tea and got down to business.

#### **4.0 Project Description**

To address the triple pronged problem of inadequate design, inadequate knowledge and inadequate server permissions we attacked redesign of the website first. The new website design was created to provide visual consistency to the site and a navigation system which would allow for simple traversal of the site. However, because of the need to update the data periodically a single redesign of the website is not enough. We proceeded to create the new design with drop-down menus and visual framing and centering. Once a workable design was agreed upon we had to then focus on the issue of maintenance.

The solution to the problem of maintaining a dynamic website using static languages was the main difficulty. The solution, beyond just making it work, was to build a piece of software to store and access the data which will be dynamic and then based on the stored data generate a static page which is ready for upload to the webserver. This emulates the standard solution, mentioned above, but instead of the administration being done directly on the website it is abstracted one level to our standalone application. To do this we break the problem conceptually into 3 pieces: website generation, persistent data storage and data input.

Essentially we had 3 problems to solve: how to get the data about the website, how to store the data for later use and how to make that data into a website.

While the final touches were being put on the website redesign we addressed the issue of website generation. We built an application which would read an XML file and generate the pages we designed. XML is an agreed upon file structure for storing data which allows for user tag definition. We used this flexibility in the format to create a set of tags which could be read by our software and output the website. The original XML script was written by hand to mimic the layout in the website redesign. Once the XML file could be interpreted, we had a full understanding of the set of the data needed to generate the website in a format which our software could interpret.

With the knowledge of the data needed to generate the website we addressed the issue of persistent data storage. We found a free relational database system with a small footprint and designed a database structure to store the data in a sensible and accessible manner. This means we broke the data down into a table structure which could be used to store the data in separate tables which could be referenced by ID number so that data could be stored a single time and referenced in many locations. The relational database system allowed us to run queries on the data, in order to insert, update, delete and obtain data from the table structure we defined. By wrapping the relational database library into our own program and implementing the database structure into our own initialization library we were ready to obtain data from the user.

To obtain the information from the user we designed our user interface to give the user structured access to the functionality of the WebBuilder software in an intuitive workflow. This means that we structured the UI to provide forms for the user to enter the data and to keep

the structure consistent with what we need to build the website and structured those forms so the use of each form flows intuitively from the previous form. The forms include data validation to check that data is input and that it is in the proper format for eventual output.

With the user interface working and data about the website being stored we implemented the final component which pulls the data from the database and writes it to an XML file which we can feed into our original XML website generation application to generate the desired website. The generation of the XML file is a matter of traversing the data entered by the user and then selected for display in the page. With all these components working together we are able to obtain course and website data from the user and generate our redesigned website for easy upload.

## **4.1 Website Design**

### **4.1.A Navigation**

To introduce a navigational system we first categorized the pages available. By understanding the categories of pages we were able to group the pages so that the top level of navigation had only a few options, but can be navigated to find its child pages and groups. We took these groupings and built them as a horizontal menu bar at the top of each page, which allows navigation to all other pages on the site. By giving the menu bar drop down features we are able to expand our categories visually and minimize the screen real estate used by the menu while still giving access to all pages. By placing this menu at the top of each page we address the issue of deep linking, because no matter what page the user initially arrives on they are able to get to the rest of the site.

### **4.1.B Layout**

Since there was no consistency in layout to speak of before hand, we were able to start from scratch on a usable layout for the website. Based on discussions with the websites current maintainer and an eye toward readability and simplicity, we created a page layout with dark borders which would center on the screen and provide framing for the content. To give a stronger sense of visual consistency to the site we created header and footer content to put on each page. The header containing a banner for the site and the footer containing site information. Having these visual cues to the open and close of each page, and placing them consistently on the page makes for a better user experience because transitions between pages are more natural and the data which is changing is bookended and highlighted by the visual elements which stay the same.

### **4.1.C Calendars**

Two calendar types existed on the site requiring 2 different approaches. The personal schedule needed to display time blocks for the different days of the week. While the course schedule was date based. We addressed the calendar readability issue by creating new calendar layouts. The layout for the course calendar shows the full calendar at the top of the page and a summary of each lab below with links between the days in the calendar and the lab titles and documents. For the personal schedule layout we implemented the hour based grid layout seen in the old design but to improve on it made the boxes in the grid larger and fixed with so they

could contain the data about the events occurring at different times. This allowed us to remove the color based time coding seen in the old layout which was difficult to read and interpret.

This new layout allows the user to create a fixed width calendar which takes up minimal screen real estate while still being readable and usable.

## 4.2 Website Builder Software

The website builder software allows the user to enter all the information needed to build a website specific to lab coordination with little to no experience with html or webdesign. It is made up of 4 components which interlock to allow the user to build and maintain a website of the format of the WPI Physics Lab site redesign and the purposes of lab coordination as seen in the WPI Physics Labs site with no need for HTML coding. We have achieved this by understanding the needs of the website and designing a database which can store the data needed for the vital duties of the WPI Physics Labs Site, such as disseminating lab times and documents, but also be extensible enough to store information for various pages which might be needed such as lab rules or lab staff schedules. The 4 components that make up the core of the Website Builder are the relational database, the user interface, the database to XML component and the XML to website component.

The database stores the data the WebBuilder uses to generate the website and when not in use is compressed in a zip archive with files to be displayed or linked on the website. The user interface component allows the user to work with the database through a visual interface, adding to and modifying the contents to build a website. Once the user has entered all their data the last two components come into play. First the data which has been entered and selected for display on the website is written into an XML file then the file is interpreted to generate the final HTML.

One aspect of building this software, which made it possible for the project to be completed by only a single developer within a limited amount of time, is that we were able to leverage many different technologies and preexisting libraries to expedite development. Firstly, the program is developed in the Java language, which provides executables which are operating system agnostic, meaning they will run on any operating system that has a java virtual machine written for it. The relational database system used, HSQLDB, a free open source database project for Java provided us with the means for data storage without building our own framework. WYSIWYG HTML editing is obtained using the eKit Java library, another freely available library. The intermediate files for web building use the XML 1.0 specification for data modelling to give a structure to the data and to allow for the use of preexisting xml interpretation libraries in Java. The database is stored in an archive when not in use. This archive uses the ZIP format which allows for exploring the database archives using tools standard on most computers.

### 4.2.A Database Design

The database is powered by HSQLDB. HSQLDB is a publicly available open source project, maintained at <http://hsqldb.org/>, which we used as the database backend. It has a small footprint and has advanced features. *"HSQLDB is included with OOo, and downloaded over 100 million times. Some specific information relating "*(<http://hsqldb.org/>). By leveraging this

preexisting technology we designed a database which could contain the data needed to represent the WPI Physics Labs Site.

The database design is structured to store 4 major components which make up the website. Pages, Documents, Courses and the Menu. The Page, Document and Course components store the data used to create the pages. The Menu component is the glue that sticks them together; defining what Pages, Documents and Courses will be generated and accessible.

#### **4.2.A.i Pages**

The goal of the design of the page was to provide a dynamic method for creating and storing pages for the website. Because of the varied uses a page could be put to the database representation must be very mutable. This led to a design where pages are made up of various elements of different types which are indexed in the *page\_element* table, but have their specific data stored in other tables. As you can see in the Database Diagram (Appendix G) the *page* table contains very little data. Its ID however is referenced in the *ref\_page\_pElement* table. The *ref\_page\_pElement* table contains a row for each element in the page and tracks its order within the page. This table then references the *page\_element* table using the *pElementID* which refers to the ID in the *page\_element* table. The *page\_element* table in turn stores an element type and an element ID. then reference which table to look in for the specific data about the element.

The types of elements that a page can contain are HTML, personal schedule, image and applets. Each of these element types has at least one table to represent it. HTML elements are contained in the *html\_element* table. Image elements are stored in the *image\_element* table and their files are stored in the directory where the database is stored and are included in the archive. Applet and personal schedule elements each require 2 tables to represent them. They each have a main table, *applet\_element* and *p\_sched\_element*, which is used in the same way as the other element tables, i.e. storing data about the element and providing an ID value to link to in the *ref\_page\_pElement* table. The secondary table needed for the applet element is called *applet\_files* and stores the information about the files needed on the server for an applet to execute. This functionality is necessary because many times applets will need to make reference to multiple library files, these files therefore must be included in our archive to be copied to the final HTML directory for upload to the web. The secondary table for the schedule element is the *sched\_event* table. This table stores the events which are to be displayed on the page where the schedule element is included.

This table structure provides us with maximum flexibility in page creation by providing the user with a single page component to be placed in the menu structure while allowing that page to contain elements of various types. We are able to arrange these elements within the page by changing their *page\_order* in the *ref\_page\_pElement* table and modify the contents of those elements in their own tables.

#### **4.2.A.ii Documents**

The document component is the simplest of the components, being made up of only the *document* table. It is used to store documents needed for the creation of the website. The main purpose of the design of this table was to store documents which were unrelated to a specific

lab for display on a documents page. The document component is still used for that but it is also used to store lab document. To distinguish between those documents which the user added and those used for lab documents the boolean field *user\_doc* field is used. The files referenced in the *document* table are stored in the directory which contains the database files and archived in the zip file for persistence.

#### 4.2.A.iii Courses

The most complex of the components is the course component which spans 9 tables. This component stores the information about the courses, the lab sections, the labs, the instructors for the courses and labs as well as the schedule relating the lab to the time each section is scheduled for that lab. The main table of this structure is the *course* table which stores the information which is common to every section of the course such as the start and end date, the course staff. Information about labs and sections are stored in the *lab* and *section* table respectively. The *lab* table stores only the title of the lab but its ID is used to reference documents related to the lab in the *lab\_document* table and to connect the labs to the sections in the *ref\_section\_lab* table. The section table stores the IDs of the lab instructors, which reference the ID in the *lab\_instructor* table, and the section number, which is essentially a title for the section. By using the ID fields in the *section* and *lab* tables we use the *ref\_section\_lab* table to connect the section and the lab and define a date and time. This provides a dynamic method for storing the scheduling information for the courses. The staff for the course are referenced in the *course* table using the fields *lecturer\_id*, *lab\_coordinator\_id* and *coordinator\_id*. Each of these fields references the ID field in the *lecturer*, *lab\_coordinator* or *coordinator* depending on the prefix of the ID field in the course table. Therefore the *lecturer\_id* in the *course* table references the ID field in the *lecturer* table.

The *lab\_document* table stores the information about specific documents related to a lab. The *lab\_id* in this table references the ID in the lab field linking each document to a lab. These documents can be either URLs (Universal Resource Locator) or files. The *doc\_type* field can have the values "URL" or "FILE" and defines the type of the file being stored by the row. If the type is "URL" then the URL is stored in the *external\_URL* field and the *document\_id* is NULL. When the type is "FILE" the *external\_URL* field is null and the *document\_id* contains the value of the ID field in the *document* table which contains location information for the file.

#### 4.2.A.iv Menu

The menu is made up of 3 tables. The *menu* table stores the structure of the menu as it will be displayed. The *courses\_selected*, *documents\_selected* tables are used to store the user selected courses and documents which the site will display. The menu is built by traversing the data in the *menu* table, starting with the those rows that have a *parent\_groupID* equal to -1. -1 in the *parent\_groupID* defines the top level of the menu. There are 4 possible values in the *type* field of the *menu* table, these values are "page", "group", "courses" and "docs". These types define the behaviour of the menu traversal. The "group" type denotes a category in the drop down menu. When a "group" typed row is found the traversal uses the *group\_name* to define the title for the group and then queries the table for rows with a *parent\_groupID* equal to the ID of the "group" row. The "page" type references a page which will be linked in the menu system. The *pageID* field in the menu table references the ID in the *page* table to

generate and link. The "courses" and "docs" types are place holder and tell the traversal code to check the *courses\_selected* and *documents\_select* tables for courses and documents selected by the user for display.

The *courses\_selected* table stores the ID from the *course* table so that when a "courses" row is found during the menu traversal the courses selected by the user can be called up and their pages and links in the menus can be generated. The *documents\_selected* table stores the ID from the *document* table and is used in the same fashion as the *courses\_selected* table.

#### **4.2.B User Interface Design**

Once the database design was complete the User Interface(UI) design was a matter of building a means to allow structured access to the database. The structure of the forms was in many ways defined by the database structure because data input relates directly to the type of data being stored and the structure it needed to be stored in. The upshot of this is that there is a direct correlation in many case between table structure and UI structure.

As the main interface to the program we provided a dashboard window containing interfaces to control the creation of the website. This main window is conceptually split, with the pieces required for website creation of the left side of the window and those pieces which are optional on the right side. The pieces required for website generation are the header, footer, home page and menu. The optional pieces are additional pages, courses and documents. The required pieces are initialized in the database and cannot be removed. While optional pieces can be created and removed. Once optional pieces are created they can be added to the generation of the website by including them in the menu system.

Most of the functionality of the forms is data input of flowing text or data information however the HTML page element page uses a library for WYSIWYG HTml editing. The library, called eKit and available a <http://www.hexidec.com/ekit.php>, provides the user a graphical interface for building html fragments to be included in page.

#### **4.2.C XML Generation**

From the data entered by the user through the UI into the database we generate the XML file to define the website we wish to create. This is the defining moment for the generation of the website, when the data is transformed from a theoretical representation in the database into a data structure which represents it's final structure.

The generation of the XML file starts with the creation of pages from the menu table. As previously discussed in section 4.2A.iv, the top level of the menu has a parent\_groupID of -1. The top level of the *menu* table is queried and each row is checked in order of its group\_order. The type of the *menu* row is checked first to determine how the row is to be used. If the type is "page" then the the pageID, in the *menu* table, is used to obtain the page information from the *page* table. This information is used to sequentially acces the ref\_page\_pElement table and then the specific element tables to construct the XML page tag. When the type is "courses" the traversal of the *selected\_courses* table is performed and an XML page tag is created for each section of each course referenced in the *selected\_courses* table. For the type of "docs" the *select\_documents* table is traversed to create the documents page XML tag. During the creation of the documents page the selected\_courses table is



traversed again to obtain the lab\_documents for the selected courses and display them on the documents page. If the type in the *menu* row is group then the ID of the row is used to iteratively traverse all rows in the *menu* table with a parent\_groupID equal to the group rows ID and perform the above procedure on them. By this method the XML tags for the pages are created.

Once the pages are created the header tag is generated. First the header row in the *page* table is used to generate any user created header element. Then the *menu* table is traversed a second time. This time generating the menu to link to the pages which were created in the first traversal of the table. The traversal procedure is much the same as for the page creation discussed above but in this case there is an added step when a row of type "group" is found. For a "group" the group\_name field is used to create a drop down group in the menu system prior to the traversal of the pages in the group.

#### **4.2.D XML Website Generation**

The final generation of the HTML pages occurs when the XML file is interpreted and the files are written. The XML file is first read by the interpreter library into an easily searchable tree structure. Once the data is interpreted we can traverse it to build HTML pages. The traversal starts with the header and footer. These two tags are traversed and html is built and stored to be placed at the top and bottom of each page. During the traversal of the header the menu is also generated. Once the header and footer are generated then the rest of the page tags are iterated through and generated converting our XML tagged data into HTML designed to the specifications of our website redesign. During the generation of the HTML the files referenced in the pages are copied into the directory with the HTML so they can be referenced by the pages after uploading.

### **5.0 Conclusion**

The redesign of the WPI Physics Lab Website is vital to its continued usefulness to students as a resource. Updating the website with a menu system and consistent styles across the pages we are able to provide the students with a more accessible resource. By these steps we hope to improve their learning experience overall. Our WebBuilder software package allows the site to be maintained with this new design by a user without specific skills in web design. This allows the redesign to continue to be useful in the hectic WPI 7 week term environment where the site's data must change at a regular interval.

## **Conclusion**

In the end, a well-rounded set of tools was created to facilitate learning through the web. Seven applets were developed to help students with various topics in the PH1110 curriculum. The major topics covered included acceleration on an inclined plane with friction, inelastic collisions, and projectile motion. The combination of simulations of physical systems and related problems effectively engages students.

The redesign of the web page added a level of visual consistency to the WPI Physics Lab site which was previously conspicuously lacking. By adding horizontal drop down menus at the top of all pages and a centered and consistent page layout the usability and usefulness of the site was improved for students.

The website builder software maintains information about lab schedules and documents, as well as fully user-created content pages for hosting applets and important course and lab information. This software package allows the maintainer of the site to create a usable website based on our redesign without technical expertise in website design.

# **Appendix A - Proposal for Physics and Education Interdisciplinary Qualifying Project**

## **1.0 Table of Contents**

- 1.0 Table of Contents
- 2.0 Executive Summary
- 3.0 Statement of Needs
  - 3.1 Improvement of Existing Applets
  - 3.2 Development of New Applets
  - 3.3 Web Design
  - 3.4 Maintenance of Website
- 4.0 Project Description
  - 4.1 Improvement of Existing Applets
  - 4.2 Development of New Applets
  - 4.3 Web Design
  - 4.4 Website Builder Software
- 5.0 Project Time Line
  - 5.1 Improvement of Existing Applets
  - 5.2 New Applet Development
  - 5.3 Website Design
  - 5.4 Website Builder Software
  - 5.5 Paper
- 6.0 Conclusion

## **2.0 Executive Summary**

The intent of this project is to create a better web based tool set for the WPI Physics Department Lab website as well as improving usability and maintainability of the website to access those tools and lab information. We will improve upon the currently available applets as well as identifying needs of the lab staff and students which could be met with custom applets. The website will be redesigned to improve navigability and accessibility while maintaining a consistent style through each page. To make maintenance of this new format simple and provide a means for easily changing course material, a software package will be developed. This package will provide a user interface for updating the lab site with minimal HTML editing. Through our applet development and website usability improvements we will give physics students a much improved set of tools for taking full advantage of their physics education.

## **3.0 Statement of Needs**

### **3.1 Improvement of Existing Applets**

Many of the current applets provided for users on the WPI Physics Department Lab website are somewhat lacking in sophistication. Problems which we have notes include a lack of information in the interface to make the tool easy to use, poor layouts of pages and elements within the applets which often cause the instructions to be unreadable. There is also a marked lack of reference to the concepts being displayed on the pages which contain the applets;

without a means for further understanding the concepts the learning applets are merely toys. Beyond just interface problems some of the applets could provide much improved functionality for data input which would make them more usable teaching tools.

### **3.2 Development of New Applets**

Many of the topics in the introductory physics curriculum are mathematically simple and easily conceptualized through visual representations such as drawings, diagrams and animations. These topics therefore lend themselves well to applet creation. Given that the current selection of applets is quite limited, especially in relation to PH1120, an examination of the PH1110 and PH1120 curricula should yield possibilities for new simulation and learning applets.

To identify other areas on which to focus applet development we will interview the lab staff. Unlike a professor giving only lectures, the lab staff has the ability to work with students one on one and field real questions and problems that the students face on a daily basis. We are confident these staff members will be able to identify the questions that consistently come up, as well as the topics that students continually have problems understanding. A set of interviews with staff will likely be the most effective method in determining the areas in which the students as well as instructors would benefit from teaching methods and tools beyond the standard combination of lecture, lab, and conference.

### **3.3 Web Design**

The current website for the WPI Physics Department Lab is woefully in need of a make over. The site has a single page from which to navigate to each section, thus requiring users to return to the front page between viewing other sections. This is needless and time consuming. In the current layout many pages do not provide a link to return to the main page, this deficiency could be a problem for a user who is linked directly to a subpage of the site and wants to go to another section. Another fairly major problem is that the schedules for labs are incredibly hard to read due to the use of variable width cells, vertical text and vibrant clashing colors. These issues are indicative of a lack of attention to the needs of students who are increasingly used to obtaining all information through the web. By leveraging modern web design principals we will update the WPI Physics Department Lab website to better serve the students by providing a more usable interface and better accessibility to important course materials.

### **3.4 Maintenance of Website**

Increased complexity of the website will be an inherent side effect of the design changes we intend to make. This increased complexity would normally be dealt with by creating a dynamic page which could include other files for the header or layout information. The WPI servers restrict developers from using server side languages such as PHP or ASP, thus each page will have to include all needed page data in a static format. This will easily get out of synch with even the most conscientious developer. To maintain consistency across the pages and to make new page creation and updating simpler we will develop a stand alone software package. This package will allow the website, with it's new design and layout, to be maintained by a user with only basic HTML knowledge.

## **4.0 Project Description**

### **4.1 Improvement of Existing Applets**

The first step in improving existing applets will be to perform a careful review off the available applets to determine what problems currently exist. We have already noted that there are a number of issues with the layout of some of the applets, this will be the most obvious of the categories. I expect however on close inspection we will find bugs in at least a few of the current applets which we will be able to fix. As well as layout and bug fixes we will use this review to determine what new functionality could be added to the present applets to make them more usable.

In the arena of functionality improvements we have already had a request for Sharepoint integration for data retrieval to be added to the Statistics Applet . While adding this functionality to the applet other improvements occurred due to the necessity of the new data acquisition process or because it seemed to be lacking. Examples of secondary improvements include storing multiple datasets, increased size of dataset, improved labeling of control buttons and the addition of the median and quartiles to the analysis data. Also during the upgrade process a bug was found in the graphing of negative numbers which was corrected.

The Statistics Applet itself is a gold mine of possibilities. It could be improved to display multi-variable analysis as well as import from a multitude of data sources. Given that this is only one of a half dozen applets there should be many improvements available to be found in the other applets.

### **4.2 New Applet Development**

To determine the teaching needs of the lab personnel, we will perform interviews with the staff. In these interviews we will attempt to understand the concepts that students often have trouble grasping as well as what tools the staff themselves need to better facilitate their jobs. We will analyze the information drawn from the interviews to define areas of focus for applets.

Once we have identified areas of focus for possible new applets, we will select a number of topics from our list based on their ability to be visualized as well as the level of need of the staff and/or student population. The next step will be to determine the general functionality of the applets within our chosen areas of focus. While it would certainly be possible to design a multi-faceted applet to teach many different concepts related to an overarching topic in the physics curriculum, it may be more effective to keep the purpose of each applet clear to the students by focusing on specific concepts that would typically be taught in a lecture or two.

In fact, we can set two specific goals for each newly designed applet to ensure that they remain simple and focused. First, the input method should be intuitive enough that a student can use an applet for the first time without instruction. Second, a student using the applet for the first time should be able to explain the output, either from the applet itself or explanatory text on the page. Both of these goals can be tested during the applet review phase.

A common area of difficulty for introductory physics students that we can already identify is rotational motion. We can present this topic effectively with an interactive visualization applet. This will provide students the chance to experiment with the concepts and see relationships between frequency, period, angular velocity and tangential velocity that can be difficult to grasp on a purely conceptual level.

Beyond the high-level design, each applet will be created using standard software development processes and integrated into the newly designed WPI Physics Department Lab website.

### **4.3 Website Design**

As mentioned previously the current WPI Physics Department Lab website has only one point of entry from which you can access the rest of the page. This front page for the site is somewhat haphazard, with links in no specific order or grouping. To improve navigation through out the site we will design the site to show a horizontal menu bar at the top of each page, which will allow navigation to the other major pages. By categorizing the current pages which are accessible from the frontpage we can create a small set of groups which is optimal for creation of the menu bar; as a large set of top level groups makes for poor readability. We will create the menu bar drop-down functionality so that within the categories users will be able to navigate quickly and directly to the page they would like to view.

We will address the calendar readability issue by creating a new calendar layout. This layout will show the full calendar at the top of the page and a summary of each lab below with links between from the days in the calendar to the lab summary. The summary section will include links to all relevant documents, which are now linked to from within the cells of the calendar. This new layout will allow us to create a fixed width calendar which takes up minimal screen real estate while still being readable and usable. We will continue the use of color within the cells of the document, however moving forward our color choices will be more muted in contrast with the current eye watering scheme.

One beneficial side effect of this redesign process is that since the new design will be crafter by hand we will be able to remove a great deal of cruft left behind by the various HTML editors used at various times in the creation of the current website. These code artifacts may never cause problems but by removing them we guarantee they never will. This will also mean a minimalistic decrease in load times as the pages are smaller so less data needs to be transmitted. The removal of these artifacts will also make the site simpler to modify in the future, as only useful code will be present.

### **4.4 Website Builder Software**

The website builder will be made up of 3 components. The first will be a store of HTML fragments and stored static document which make up the whole set of static pages available to the Website Builder. This repository of HTML will contain HTML fragments for the menu bar, footer data and static informational pages such as the Lab Rules, as well as the static applet pages and their necessary java files. These files will all be specially crafted before hand to fit with the structure of the Website Builder.

The second component of Website Builder will be a script interpreter. The interpreter will read in a specially crafted script which will define the entire layout of the website. This script will make reference to the pre-existing HTML pieces as well as containing stored parameters to auto-generate HTML for pages needing more dynamic layouts such as calendars, schedules and course materials.

The final component will be the user interface. Through this interface the user will define the data which should appear on the website. Course schedules will be entered as well as the lab coordinator's schedule. These parameters will all be written to a script which upon

completion of the user input will be passed to the script interpreter. The finally generated website will be saved to a user defined location. Thus allowing the user to simply upload the files to the server to perform a full update of the WPI Physics Department Lab website.

## 5.0 Timeline

The time line below can be found represented in Appendix A visualized as a gantt chart.

### 5.1 Improvement of Existing Applets

- Identify Deficiencies in Applets (week 1 of A term)
- Fix Page Layout Problems (week 2 of A term)
- Upgrade Applets (week 3 - 7 of A term)
- Debug, Review Applets - Round 1 (week 1 of B term)
- Upgrade Applets (week 2 of 5 B term)
- Debug, Review Applets - Round 2 (week 6 of B term)
- Finalize Applet Updates (week 7 of B term)

### 5.2 New Applet Development

- Interviews with Lab Staff (week 1 & 2 of A term)
- Identify Needs and Brainstorm new Applets (week 3 of A term)
- Build New Applets (week 4 of A term - week 5 of B term )
- Debug, Review Applets - Round 1 (week 6 & 7 of B term)
- Improve Applets (week 1 of C term)
- Debug, Review Applets - Round 2 (week 2 of C term)
- Finalize Applet Updates (week 3 of C term)

### 5.3 Website Design

- Redesign Website (week 1 - 3 of A term)
- Finalize Design and Colors (week 4 of A term)

### 5.4 Website Builder Software

- Build Script Based Website Builder (week 1 - 7 of A term)
- Build UI for Script Creation (week 1 - 7 of B term)
- finalized program (week 1 & 2 of C term)

### 5.5 Paper

- Project Proposal (complete by start of A term)
- First Draft of Paper (week 1 to 7 of A term)
- Second Draft of Paper (B term)
- Third Draft of Paper (week 1 to 3 of C term)
- Final Paper (week 4 to 7 of C term)

## 6.0 Conclusion

Our ultimate goal is to facilitate undergraduate physics students' education through our work on the website and physics applets. We will accomplish this by improving the navigability and consistency of the website and by increasing the usability and variety of the applications available on the website. We must make sure to obtain feedback from students, lab assistants, and professors to ensure that our work stays relevant and fulfills the needs of the user population.

## Appendix B - Suggested Improvements for Existing Web Applets

### All Applets

- Consider rewording or rewriting the descriptions after changes have been made.
- Use double buffering.
- Align input boxes vertically rather than horizontally.

### Vector Calculus

- Rename as Vector Arithmetic.
- Consider allowing vectors to be input by coordinate. The description says that this feature was purposefully left out to give students an intuitive feel for the vectors. Was that a sensible decision?
- Round vector coordinates to the nearest integer so that the relation between the vectors and their coordinates is clearer.
- Allow vectors to be dragged after they have been added.
- Follow the cursor even when it is outside of the  $[-10, -10]$ ,  $[10, 10]$  box.
- Show the dot product.

### Powers of Ten

- You should only be able to select one choice at a time.
- You should only need to click on a button *once* to select it.
- The reset button should reset rather than pause.

### Greek

- The interface is unintuitive. How should this matching game work?
- Consider just redirecting to a better-designed Greek alphabet learning applet elsewhere on the internet.

### Hook's Law

- This applet is very narrow. We should try to make it more general.
- Allow any number of springs to be input instead of exactly five.
- Show the standard deviation or some other measure of variability so that it is evident if someone inputs inconsistent data. Or just raise a flag if the data varies too much.
- Move the spring picture into the description, and fix the text.

### Elastic Collision Simulator

- Allow the balls to move either left or right at the start, with negative velocities indicating leftward movement.
- Either rescale the picture to deal well with various starting velocities or do not run when the input is out of bounds. Silently modifying the input to fit within the bounds is poor behavior.



### Projectile Motion

- Consider allowing more or less than four trials. Perhaps there should be a reset button that eliminates all trials, or a button that eliminates just the oldest trial.
- Reorder the input boxes to have y:position, then x:velocity, then y:velocity.
- Display x and y scales on the grid.
- Automatically rescale the grid to fit any inputs.
- Have the projectile stop when it hits the ground, rather than when it passes through the ground. Show its final x position.
- Display a table of relevant information about the trials shown similar to the following:

Trial	Final x pos	Max height
1	30	50
2	30	40
3	-	-
4	-	-

### Statistics Applet

- Added Quartile graph to analysis output
- Import data from various locations directly from the sharepoint site excel files
- Display units in the scatterplot graph at the max and min of Y axis
- implement multi variable comparison
- implement outlier detection

## Appendix C - Review of Lab Content

### PH 1110 Labs

- Cart is placed on a sloped ramp and undergoes constant acceleration (Inclined Plane applet)
- Force plate experiments and practice with free body diagrams
- Cart is placed on a sloped ramp. Normal force and friction are calculated (Inclined Plane applet)
- Cart is placed on ramp with various amounts of weight added (Inclined Plane applet)
- Measuring the impulse of a sand bag, and comparing the behavior of a spring to an elastic band (Hooke applet)
- Carts are placed on a flat ramp and collided with each other (Collision applet)
- Popsicle stick experiment involving balancing forces and moments
- Cart is placed on a sloped ramp, and its motion is measured by the rotation of a pulley (Rotational Motion applet)

### PH 1120 Labs

These labs already use EMField.

### PH 1130 Labs

Modelling relativity and quantum mechanics is too ambitious.

### PH 1140 Labs

Many of the experiments with complicated oscillating systems would lend themselves well to simulation. There's some potential for applets here, but we don't have time.

## Appendix D - Student Survey

What is your major?

In what year are you graduating?

### Collision Applet

*Download the collision applet from <http://www.mediafire.com/?cadaddnzznu>. Spend a couple minutes adjusting the various parameters and see the effects they have on the collision.*

Do the controls behave as you expect them to? If not, how do you think they should behave?

Does the applet do everything you want it to? If not, what functionality should we add?

*Try out the problem buttons. Then pick a problem and try to solve it. You may want to use the Equations button for help.*

Which problem did you choose?

Were you successful?

Was the problem too easy? Was it too difficult?

Did you need the Equations to solve the problem?

Was the problem representative of the type of problem you would encounter in class?

Do you think this applet would be useful as a graded pre-lab activity? Why or why not?

Did you notice any bugs, or do you have any other comments?

### Inclined Plane Applet

*Download the inclined plane applet from <http://www.mediafire.com/?ozqj2mnjmq2>. Spend a couple minutes adjusting the various parameters and see the effects they have on the cart's motion.*

Do the controls behave as you expect them to? If not, how do you think they should behave?

Does the applet do everything you want it to? If not, what functionality should we add?

*Try out the problem buttons. Then pick a problem and try to solve it. You may want to use the Equations button for help.*

Which problem did you choose?

Were you successful?

Was the problem too easy? Was it too difficult?

Did you need the Equations to solve the problem?

Was the problem representative of the type of problem you would encounter in class?

Do you think this applet would be useful as a graded pre-lab activity? Why or why not?

Did you notice any bugs, or do you have any other comments?

## Appendix E - Summary of Survey Responses

### Collision Applet

*Do the controls behave as you expect them to? If not, how do you think they should behave?*

Generally they thought the controls were intuitive.

Two of the three respondents did not realize they had to press enter to change a text field.

One respondent expected the start button to stop the simulation while it was running.

*Does the applet do everything you want it to? If not, what functionality should we add?*

The balls could reset to their original positions after a time.

We could implement two-dimensional collisions.

*Which problem did you choose?*

All three respondents chose the elastic collision problem.

*Were you successful?*

All three were successful

*Was the problem too easy? Was it too difficult?*

They all thought the problem was reasonable with the equations at hand, but still somewhat tedious.

Some difficulty was caused by the fact that they couldn't see the equations window while working on the problem.

One respondent suggested choosing problems with integer solutions.

*Did you need the Equations to solve the problem?*

Yes.

*Was the problem representative of the type of problem you would encounter in class?*

Yes.

*Do you think this applet would be useful as a graded pre-lab activity? Why or why not?*

They all thought it would work well as a graded pre-lab activity.

They reflected positively on the combination of a simulation with randomly generated problems.

*Did you notice any bugs, or do you have any other comments?*

They found no bugs.

Two had positive comments.

### **Inclined Plane Applet**

*Do the controls behave as you expect them to? If not, how do you think they should behave?*

All respondents found the controls intuitive.

*Does the applet do everything you want it to? If not, what functionality should we add?*

Change what's on the ramp (e.g. box to goat in box, etc.)

Display the equations used in the calculations with numbers plugged in.

Show the numbers along with the acceleration vectors.

Spacebar as a shortcut to release the cart.

Be able to input the angle in radians.

Add a reset button.

One person expected the cart to start moving as you increased the angle of the ramp.

*Which problem did you choose?*

Three chose frictionless acceleration, and one chose the Which Planet? problem.

*Were you successful? Was the problem too easy? Was it too difficult?*

The three respondents who chose the frictionless acceleration problem solved it, and said it had a reasonable difficulty.

The person attempting the 'Which Planet?' problem was unsuccessful because they could not keep the equations window visible while attempting the problem.

*Did you need the Equations to solve the problem?*

Half of them did, and the other half didn't.

*Was the problem representative of the type of problem you would encounter in class?*

They all said it was.

One described the problem as fun.

One pointed out that while the problem was randomly generated, it did not pose a challenge once the governing equation has been determined.

*Do you think this applet would be useful as a graded pre-lab activity? Why or why not?*

Two said it would be useful as a graded pre-lab activity.

One said it would be useful if the bugs (referring to the 'nudge bug') were fixed.

One said it might be useful if you are new to the material, but if you are familiar with it, the activity would feel like a repeat of the lab

*Did you notice any bugs, or do you have any other comments?*

Under certain circumstances, pressing the nudge button caused the cart to move up the ramp.

One student was more used to seeing vector arrows drawn from the center of the box.

One student didn't realize that you had to press enter when editing a text field.

The equations window couldn't stay up when viewing a problem.

## Appendix F - Potential Changes to Applets

### Suggested By Survey And Implemented

- Mention in the help text that you should press enter after changing a text field.
- Fix a bug in the Inclined Plane applet (under certain circumstances, nudge button caused cart to move up the ramp).
- Make equations window visible when solving a problem.
- Draw the normal acceleration vector starting from the center of the cart to be consistent with the other arrows.
- Added a reset button to the Inclined Plane applet (called "setup").

### Suggested By Survey But Not Implemented

- Have the start button pause the cart simulation if it is running.
- Make a two-dimensional collision applet.
- Have the balls reset to their original positions after a certain period of time.
- Allow text fields to be changed without pressing enter (this was a common source of confusion).
- Show numbers alongside force vectors.
- Introduce hotkeys, such as having spacebar start the simulation.
- Have the cart start moving when you increase the angle of the ramp.

### Other Unimplemented Changes

- Allow for variable mass, spring constant, and frequency in Hooke applet.
- Display angular velocity, tangential velocity vector in Rotational Motion applet.
- Add arrows showing location and direction of launch of projectiles prior to pressing the "launch" button in Projectile applet (see the Projectile Game applet).
- Show the cross product vector (in  $i$ - $j$ - $k$  notation) in the Vector applet.



# Appendix G - Database Design

## 1.0 Table of Contents

- 1.0 Table of Contents
- 2.0 menu
  - 2.1 Usage Description
  - 2.2 Design
  - 2.3 Description
- 3.0 courses\_selected
  - 3.1 Usage Description
  - 3.2 Design
  - 3.3 Description
- 4.0 documents\_selected
  - 4.1 Usage Description
  - 4.2 Design
  - 4.3 Description
- 5.0 page
  - 5.1 Usage Description
  - 5.2 Design
  - 5.3 Description
- 6.0 page\_element
  - 6.1 Usage Description
  - 6.2 Design
  - 6.3 Description
- 7.0 ref\_page\_pElement
  - 7.1 Usage Description
  - 7.2 Design
  - 7.3 Description
- 8.0 image\_element
  - 8.1 Usage Description
  - 8.2 Design
  - 8.3 Description
- 9.0 html\_element
  - 9.1 Usage Description
  - 9.2 Design
  - 9.3 Description
- 10.0 p\_schedule\_element
  - 10.1 Usage Description
  - 10.2 Design
  - 10.3 Description
- 11.0 sched\_event
  - 11.1 Usage Description
  - 11.2 Design
  - 11.3 Description
- 12.0 applet\_element

- 12.1 Usage Description
- 12.2 Design
- 12.3 Description
- 13.0 applet\_files
  - 13.1 Usage Description
  - 13.2 Design
  - 13.3 Description
- 14.0 course
  - 14.1 Usage Description
  - 14.2 Design
  - 14.3 Description
- 15.0 section
  - 15.1 Usage Description
  - 15.2 Design
  - 15.3 Description
- 16.0 lab
  - 16.1 Usage Description
  - 16.2 Design
  - 16.3 Description
- 17.0 lab\_document
  - 17.1 Usage Description
  - 17.2 Design
  - 17.3 Description
- 18.0 ref\_section\_lab
  - 18.1 Usage Description
  - 18.2 Design
  - 18.3 Description
- 19.0 lecturer
  - 19.1 Usage Description
  - 19.2 Design
  - 19.3 Description
- 20.0 coordinator
  - 20.1 Usage Description
  - 20.2 Design
  - 20.3 Description
- 21.0 lab\_coordinator
  - 21.1 Usage Description
  - 21.2 Design
  - 21.3 Description
- 22.0 lab\_instructor
  - 22.1 Usage Description
  - 22.2 Design
  - 22.3 Description
- 23.0 document
  - 23.1 Usage Description
  - 23.2 Design

## 23.3 Description

### **2.0 menu**

#### 2.1 Usage Description

The menu table stores the structure of the webpages menu. The menu drives the creation of the webbuilder because only pages referenced by the menu are generated. The documents and courses pages are referenced with place holders in the type field. If the "docs" or "courses" place holder is found in the type field then the documents\_selected or courses\_selected tables are referenced.

#### 2.2 Design

ID INTEGER IDENTITY

type VARCHAR(10)

parent\_groupID INTEGER

group\_name VARCHAR(25)

pageID INTEGER

group\_order INTEGER

#### 2.3 Variable Description

*ID* is a unique integer identifier for the row

*type* can be "page", "group", "courses" or "docs"

*parent\_groupID* reference the parent group rows ID are used for storing the nested menu

*parent\_groupID* is -1 for the top level of the menu

*group\_name* is used to store the name of the group when the *type* is "group"

*pageID* references the ID in the page table

*group\_order* determines the order of the menu item within its group

### **3.0 courses\_selected**

#### 3.1 Usage Description

The courses\_selected table is the list of courses that the webbuilder uses to generate the course schedule section of the webpage. This table is only referenced if a menu item of the type "courses" is found during the generation of the webpage.

#### 3.2 Design

ID INTEGER IDENTITY

courseID INTEGER

### 3.3 Description

*ID* is a unique integer identifier for the row

*courseID* is the unique integer identifier in the courses table for selected courses.

## **4.0 documents\_selected**

### 4.1 Usage Description

The documents\_selected table is the list of documents that the webbuilder uses to generate the documents page. This table is only referenced if a menu item of the type "docs" is found during the generation of the webpage.

### 4.2 Design

ID INTEGER IDENTITY

documentID INTEGER

### 4.3 Description

*ID* is a unique integer identifier for the row

*documentID* is the unique integer identifier in the documents table for selected documents.

## **5.0 page**

### 5.1 Usage Description

The page table stores the data for the pages created. It is primed at the creation of a new webpage with the the required pages header, footer and home page.

### 5.2 Design

ID INTEGER IDENTITY

title VARCHAR(256)

name VARCHAR(256)

required\_page BOOLEAN

### 5.3 Description

*ID* is a unique integer identifier for the row

*title* in a user defined human readable title

*name* is a system generated name that will be used for the filename of the page

*required\_page* if this value is true then this is a system defined page which cannot be removed

## **6.0 page element**

### 6.1 Usage Description

The *page\_element* table stores the type and specific index of a page element in its typed table.

### 6.2 Design

ID INTEGER IDENTITY

element\_type VARCHAR(50)

elementID INTEGER

### 6.3 Description

*ID* is a unique integer identifier for the row

*element\_type* can be "html", "psched", "applet"

*elementID* references the ID field in the specific element type table

(html\_element, p\_sched\_element, applet\_element)

## **7.0 ref\_page pElement**

### 7.1 Usage Description

The *ref\_page\_pElement* table connects page elements to pages and defines the order in which they appear in the page.

### 7.2 Design

ID INTEGER IDENTITY

pageID INTEGER

pElementID INTEGER

page\_order INTEGER

### 7.3 Description

*ID* is a unique integer identifier for the row

*pageID* references the ID in the page table

*pElementID* references the ID in the page\_element table

*page\_order* defines the location in the order of elements the page element referenced by pElementID

## **8.0 image\_element**

### 8.1 Usage Description

The image\_element table stores information about an image for use in a page.

### 8.2 Design

ID INTEGER IDENTITY

title VARCHAR(30)

filename VARCHAR(256)

alt\_text VARCHAR(50)

align VARCHAR(20)

width INTEGER

height INTEGER

### 8.3 Description

*ID* is a unique integer identifier for the row

*title* stores the title of the image

*filename* stores the filename of the image

*alt\_text* stores the text to display in the alt\_text tag for the image

*align* defines the alignment of the image on the page

*width* defines the width of the image

*height* defines the height of the image

## **9.0 html\_element**

### 9.1 Usage Description

The html\_element table stores the data for html elements of a page.

### 9.2 Design

ID INTEGER IDENTITY

title VARCHAR(30)  
html\_text VARCHAR(32000)

### 9.3 Description

*ID* is a unique integer identifier for the row  
*title* stores the title of the element for UI display  
*html\_text* contains the html to be displayed

## **10.0 p\_schedule\_element**

### 10.1 Usage Description

The p\_schedule\_element table is more a place hold table to keep consistency. The only thing stored in this table is a title for display in the UI. However by having this row we can keep a consistent design and use it as an index in the sched\_event table.

### 10.2 Design

ID INTEGER IDENTITY  
schedule\_title VARCHAR(30)

### 10.3 Description

*ID* is a unique integer identifier for the row  
*schedule\_title* contains the title of the schedule for display in the interface

## **11.0 sched\_event**

### 11.1 Usage Description

The sched\_event table stores the date, time and human readable data about personal schedules.

### 11.2 Design

ID INTEGER IDENTITY  
schedID INTEGER  
MON BOOLEAN  
TUE BOOLEAN  
WED BOOLEAN  
THU BOOLEAN  
FRI BOOLEAN

starttime VARCHAR(5)  
endtime VARCHAR(5)  
event\_text VARCHAR(30)

### 11.3 Description

*ID* is a unique integer identifier for the row

*schedID* is the ID of the p\_sched\_element row this event belongs to

*MON* is a boolean value which says whether the event occurs on Monday

*TUE* is a boolean value which says whether the event occurs on Tuesday

*WED* is a boolean value which says whether the event occurs on Wednesday

*THU* is a boolean value which says whether the event occurs on Thursday

*FRI* is a boolean value which says whether the event occurs on Friday

*starttime* is the time the event begins

*endtime* is the time the event ends

*event\_text* is the human readable text of the event

## **12.0 applet element**

### 12.1 Usage Description

The applet\_element table stores the information about an applet.

### 12.2 Design

ID INTEGER IDENTITY

title VARCHAR(30)

type VARCHAR(8)

height INTEGER

width INTEGER

code VARCHAR(50)

src\_applet\_file\_id INTEGER

directory VARCHAR(256)

### 12.3 Description

*ID* is a unique integer identifier for the row

*type* can be "class" or "archive"

*height* and *width* refer to the size the applet will appear on the page



*code* is the code value for the archive type

*src* is the source for the class type

*filename* contains the file which should be referenced in the applet call

### **13.0 applet files**

#### **13.1 Usage Description**

The `applet_files` table stores the information about the files which make up the applet.

#### **13.2 Design**

ID INTEGER IDENTITY

applet\_id INTEGER

directory VARCHAR(256)

filename VARCHAR(50)

#### **13.3 Description**

*ID* is a unique integer identifier for the row

*applet\_id*: references the ID in the `applet_element` table, defines the applet the file belongs to

*directory*: the directory in which the applet file is stored

*filename*: the filename of the applet file

### **14.0 course**

#### **14.1 Usage Description**

The `course` table course the base information for a course.

#### **14.2 Design**

ID INTEGER IDENTITY

title VARCHAR(256)

year VARCHAR(2)

department VARCHAR(2)

term VARCHAR(1)

course\_number VARCHAR(25)

start\_date DATE

end\_date DATE

lecturer\_id INTEGER  
coordinator\_id INTEGER  
lab\_coordinator\_id INTEGER

### 14.3 Description

*ID*: unique integer identifier for the row

title: title of the course

year: the year in which the course is being held

department: The department offering the course

term: the term in which the course is being held

course\_number: the course number

start\_date: the start date of the course

end\_date: the end date of the course

lecturer\_id: references the ID in the lecturer table defining the lecturer for the course

coordinator\_id: references the ID in the coordinator table defining the coordinator for the course

lab\_coordinator\_id: references the ID in the lab\_coordinator table defining the lab coordinator for the course

## **15.0 section**

### 15.1 Usage Description

The section table stores the data for each section in a course.

### 15.2 Design

ID INTEGER IDENTITY

courseID INTEGER

section\_number VARCHAR(10)

lab\_instructor\_id1 INTEGER

lab\_instructor\_id2 INTEGER

### 15.3 Description

*ID*: unique integer identifier for the row

courseID: references the ID in the course table to which this section belongs

section\_number: a human readable section number, this can be characters as well as numerals

lab\_instructor\_id1: references the ID in the lab\_instructor table defining the primary lab instructor

lab\_instructor\_id2: references the ID in the lab\_instructor table defining the secondary lab instructor

## **16.0 lab**

### 16.1 Usage Description

The lab table stores the titles of the labs that relate to a course. The ID for this table is used in the to schedule the lab for each section.

### 16.2 Design

ID INTEGER IDENTITY

courseID INTEGER

title VARCHAR(256)

### 16.3 Description

*ID*: unique integer identifier for the row

*courseID*: refers to the ID of the course in the course table which this lab is a part of

*title*: title of the lab

## **17.0 lab document**

### 17.1 Usage Description

The lab\_document table stores information about the lab documents. They can be actual files or URLs. the doc\_type is either "FILE" or "URL" depending on which type of data is stored in the row

### 17.2 Design

ID INTEGER IDENTITY

lab\_id INTEGER

title VARCHAR(256)

doc\_type VARCHAR(5)

external\_URL VARCHAR(256)

document\_id INTEGER

### 17.3 Description

*ID*: unique integer identifier for the row

*lab\_id*: refers to the ID of the lab in the lab table which this document is linked to

*title*: title of the lab document

*doc\_type*: contains either "URL" or "FILE" depending on the type of the lab document

*external\_URL*: if the doc\_type is "URL" then this field will contain the URL for the document

*document\_id*: if the doc\_type is "FILE" then this field contain the ID of the document in the document table

## **18.0 ref\_section\_lab**

### 18.1 Usage Description

The ref\_section\_lab table defines the date and time when each section is scheduled for each lab.

### 18.2 Design

ID INTEGER IDENTITY

sectionID INTEGER

labID INTEGER

lab\_date DATE

lab\_time VARCHAR(20)

### 18.3 Description

*ID* is a unique integer identifier for the row

*sectionID*: ID for the section which is referenced

*labID*: ID for the lab which is referenced

*lab\_date*: date on which the referenced section has the referenced lab

*lab\_time*: time at which the referenced section has the referenced lab

## **19.0 lecturer**

### 19.1 Usage Description

The lecturer table stores the information about the lecturers.

### 19.2 Design

ID INTEGER IDENTITY

lecturer\_name VARCHAR(100)

lecturer\_email VARCHAR(100)

lecturer\_phone1 VARCHAR(12)

lecturer\_phone2 VARCHAR(12)

### 19.3 Description

*ID*: unique integer identifier for the row

*lecturer\_name*: name of the lecturer

*lecturer\_email*: email address of the lecturer

*lecturer\_phone1*: lecturer primary phone number

*lecturer\_phone2*: lecturer secondary phone number

## **20.0 coordinator**

### 20.1 Usage Description

The coordinator table stores the information about the coordinators.

### 20.2 Design

ID INTEGER IDENTITY

coordinator\_name VARCHAR(100)

coordinator\_email VARCHAR(100)

coordinator\_phone1 VARCHAR(12)

coordinator\_phone2 VARCHAR(12)

### 20.3 Description

*ID*: unique integer identifier for the row

*coordinator\_name*: name of the coordinator

*coordinator\_email*: email address of the coordinator

*coordinator\_phone1*: coordinator primary phone number  
*coordinator\_phone2*: coordinator secondary phone number

## **21.0 lab\_coordinator**

### 21.1 Usage Description

The lab\_coordinator table stores the information about the lab coordinators.

### 21.2 Design

ID INTEGER IDENTITY

lab\_coordinator\_name VARCHAR(100)

lab\_coordinator\_email VARCHAR(100)

lab\_coordinator\_phone1 VARCHAR(12)

lab\_coordinator\_phone2 VARCHAR(12)

### 21.3 Description

*ID*: unique integer identifier for the row

*lab\_coordinator\_name*: name of the lab coordinator

*lab\_coordinator\_email*: email address of the lab coordinator

*lab\_coordinator\_phone1*: lab coordinator primary phone number

*lab\_coordinator\_phone2*: lab coordinator secondary phone number

## **22.0 lab\_instructor**

### 22.1 Usage Description

The lab\_instructor table stores the information about the lab instructors.

### 22.2 Design

ID INTEGER IDENTITY

lab\_instructor\_name VARCHAR(100)

lab\_instructor\_email VARCHAR(100)

### 22.3 Description

*ID*: unique integer identifier for the row

*lab\_instructor\_name*: name of the lab instructor

*lab\_instructor\_email*: email address of the lab instructor

## **23.0 document**

### 23.1 Usage Description

The document table stores the information about the documents which could be selected for display on the documents page.

### 23.2 Design

ID INTEGER IDENTITY

title VARCHAR(256)

filename VARCHAR(256)

document\_group VARCHAR(256)

user\_doc BOOLEAN

### 23.3 Description

*ID*: unique integer identifier for the row

*title*: human readable title of the document

*filename*: name of the file that has been copied in to the database

*document\_group*: name of the group the file will be listed under in the documents page

*user\_doc*: boolean value which allows the program to store files in the document table without displaying them for selection in the document list.

# Appendix H - Website Builder Manual

## 1.0 Table of Contents

- 1.0 Table of Contents
- 2.0 Quick Start Guide
- 3.0 Running the Program
- 4.0 Webpage Load Window
  - 4.1 Creating a new webpage
  - 4.2 Load a webpage file
- 5.0 Main Interface Window
  - 5.1 Modifying the Header
    - 5.2 Modifying the Footer
  - 5.3 Modifying the Home Page
    - 5.4 Adding Pages
    - 5.5 Modifying Pages
    - 5.6 Removing Pages
    - 5.7 Adding Documents
    - 5.8 Modifying Documents
    - 5.9 Removing Documents
    - 5.10 Adding Courses
    - 5.11 Modifying Courses
    - 5.12 Removing Courses
    - 5.13 Updating the Menu
    - 5.14 Save Webpage and Close
    - 5.15 Generate Webpage
- 6.0 Page Window
  - 6.1 Entering the Title
  - 6.2 Adding a Page Element
  - 6.3 Modifying a Page Element
  - 6.4 Removing a Page Element
  - 6.5 Changing the order of the Page Elements
- 7.0 Page Element Window
  - 7.1 Creating a HTML Element
    - 7.1.a Entering the Title
    - 7.1.b Entering HTML through the WYSIWYG Interface
    - 7.1.c Entering HTML directly
    - 7.1.d Save Data
    - 7.1.e Cancel Entry
  - 7.2 Creating an Image Element
    - 7.2.a Entering the Title
    - 7.2.b Selecting the File
    - 7.2.c Entering the Alt Text
    - 7.2.d Selecting the Image Alignment
    - 7.2.e Entering the Height
    - 7.2.f Entering the Width



- 7.2.g Save Data
  - 7.2.h Cancel Entry
- 7.3 Creating an Applet Element
  - 7.3.a Entering the Title
  - 7.3.b Selecting the Executable
  - 7.3.c Adding Files
  - 7.3.d Entering the Height
  - 7.3.e Entering the Width
  - 7.3.f Selecting a Class or an Archive
  - 7.3.g Save Data
  - 7.3.h Cancel Entry
- 7.4 Creating a Personal Schedule Element
  - 7.4.a Entering the Title
  - 7.4.b Adding an Event
  - 7.4.c Modifying an Event
  - 7.4.d Removing an Event
  - 7.4.e Save Data
  - 7.4.f Cancel Entry
- 7.5 Schedule Event Window
  - 7.5.a Setting up the Schedule for the Event
  - 7.5.b Entering the Text of the Event
  - 7.5.c Save Data
  - 7.5.d Cancel Entry
- 8.0 Menu Window
  - 8.1 Adding Pages to the Menu
  - 8.2 Adding Groups to the Menu
  - 8.3 Removing a Menu Item
  - 8.4 Changing the Order of Menu Items
  - 8.5 Enabling Courses
  - 8.6 Adding and Removing Courses
  - 8.7 Enabling Document
  - 8.7 Adding and Removing Documents
  - 8.8 Closing the form
- 9.0 Document Window
  - 9.1 Entering the Title
  - 9.2 Selecting the Group
  - 9.3 Selecting the File
  - 9.4 Save Data
  - 9.5 Cancel Entry
- 10.0 Course Window
  - 10.1 Entering the Title
  - 10.2 Entering the Course Number
  - 10.3 Entering Academic Year
  - 10.4 Selecting the Term
  - 10.5 Entering the Start Date
  - 10.6 Entering the End Date

- 10.7 Selecting a Lecturer
- 10.8 Adding a Lecturer
- 10.9 Modifying a Lecturer
- 10.10 Removing a Lecturer
- 10.11 Selecting a Coordinator
- 10.12 Adding a Coordinator
- 10.13 Modifying a Coordinator
- 10.14 Removing a Coordinator
- 10.15 Selecting a Lab Coordinator
- 10.16 Adding a Lab Coordinator
- 10.17 Modifying a Lab Coordinator
- 10.18 Removing a Lab Coordinator
- 10.19 Adding a Lab
- 10.20 Modifying a Lab
- 10.21 Removing a Lab
- 10.22 Adding a Section
- 10.23 Modifying a Section
- 10.24 Removing a Section
- 10.25 Save Data
- 10.26 Cancel Entry
- 11.0 Section Window
  - 11.1 Entering the Section Number
  - 11.2 Select Primary Lab Instructor
  - 11.3 Select Secondary Lab Instructor
  - 11.4 Adding a Lab Instructor
  - 11.5 Modifying a Lab Instructor
  - 11.6 Removing a Lab Instructor
  - 11.7 Setting the Date of a Lab for the Section
  - 11.8 Modifying the Date of a Lab for the Section
  - 11.9 Save Data
  - 11.10 Cancel Entry
- 12.0 Set Lab Date Window
  - 12.1 Enter Lab Date
  - 12.2 Enter Lab Time
  - 12.3 Save Data
  - 12.4 Cancel Entry
- 13.0 Lab Window
  - 13.1 Entering the Title
  - 13.2 Adding a Lab Document
  - 13.3 Modifying a Lab Document
  - 13.4 Removing a Lab Document
  - 13.5 Save Data
  - 13.6 Cancel Entry
- 14.0 Lab Document Window
  - 14.1 URL
    - 14.1.a Entering the Title

- 14.1.b Entering the URL
- 14.1.c Save Data
- 14.1.d Cancel Entry
- 14.2 File
  - 14.2.a Entering the Title
  - 14.2.b Selecting the File
  - 14.2.c Save Data
  - 14.2.d Cancel Entry
- 15.0 Lab Instructor Window
  - 15.1 Entering the Name
  - 15.2 Entering the Email Address
  - 15.3 Save Data
  - 15.4 Cancel Entry
- 16.0 Lab Coordinator Window
  - 16.1 Entering the Name
  - 16.2 Entering the Email Address
  - 16.3 Entering the Phone Number
  - 16.4 Save Data
  - 16.5 Cancel Entry
- 17.0 Coordinator Window
  - 17.1 Entering the Name
  - 17.2 Entering the Email Address
  - 17.3 Entering the Phone Number
  - 17.4 Save Data
  - 17.5 Cancel Entry
- 18.0 Lecturer Window
  - 18.1 Entering the Name
  - 18.2 Entering the Email Address
  - 18.3 Entering the Phone Number
  - 18.4 Save Data
  - 18.5 Cancel Entry

## 2.0 Quick Start Guide

1. Start the program by double clicking on the WebBuilder.jar file.
2. Select "New Webpage" from the **Main** drop-down menu.
3. Enter a filename in the popup window and press *Enter*.
4. Create a **Header**, **Footer** and **Home Page** for the webpage by clicking the ~ button.
5. Create **Pages** and **Courses** for the webpage by clicking the + button.
6. Setup the **Menu** by clicking the ~ button.
7. Generate the website by clicking the **Generate Webpage**
8. To view the output open the HTML directory and open the index.html file in a web browser.
9. Upload the contents of the HTML directory to your webspace.

10. Open a browser and view your URL.

Button Labels

+: adds items

-: removes items

~: modifies items

### 3.0 Running the Program

To run the program, navigate to the directory where you have stored the WebBuilder.jar and double click on it. You will need to have java installed to run the application.

### 4.0 Webpage Load Window

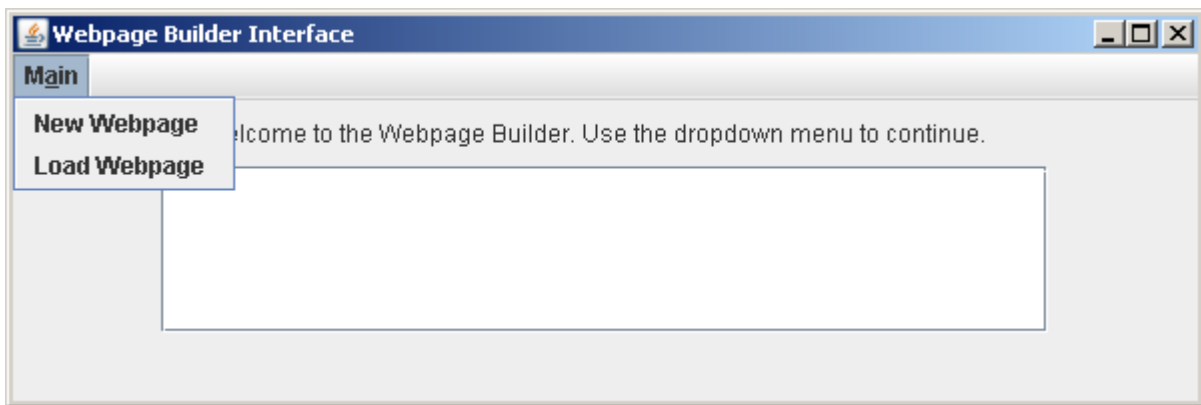


Figure 1: Webpage Builder Interface

Once you have successfully started the application you will see the screen shown in Figure 1. From this screen you can start a new website or load a website you have previously created with the application.

#### 4.1 Creating a new webpage

To create a new webpage, use the "Main" drop down menu and select "New Webpage". You will be prompted for a name. When the name is entered and the *Enter* key is pressed on your keyboard or the *OK* button is pressed the Main Interface Window, see section 6.0, will open.

#### 4.2 Load a webpage file

To load a website previously created by the WebBuilder, use the drop down menu and select "Load Webpage". You will be prompted to browse to the archive for the website you wish to load. The file extension is .zip and if it has not been moved since the last use of the software it will be contained in the same directory as the WebBuilder executable. When the file is opened the Main Interface Window, see section 6.0, will open with your existing data.

## 5.0 Main Interface Window

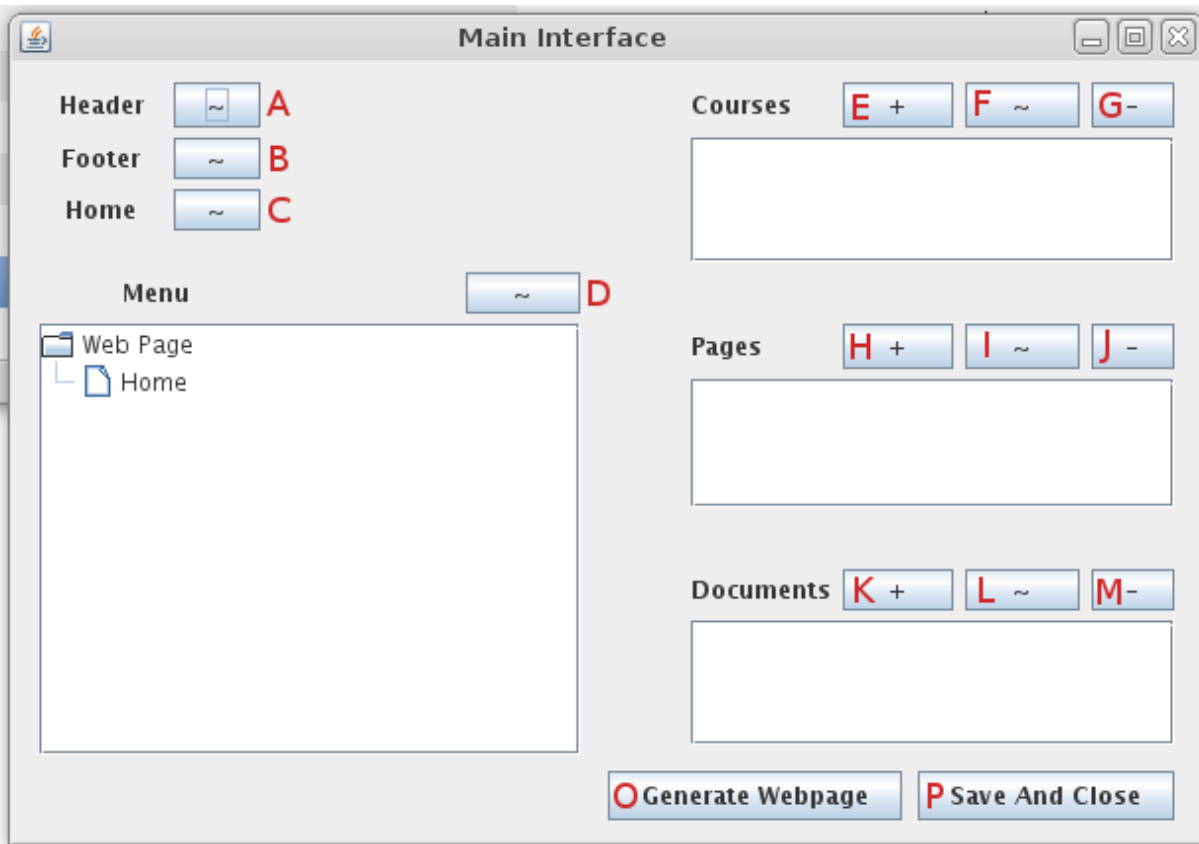


Figure 2: Main Interface

The Main Interface Window is the gateway to the Webpage Builder. From here you have access to all the functionality of the application. The required page information is all on the left side of the form while data that is optional and user created is on the right.

### 5.1 Modifying the Header

To modify the Header for all the pages on the website click the ~ button to the right of the "Header" title on the left side of the form. This button is labeled in Figure 2 as **A**. This will bring up the Page Window. The Header is considered a page and thus can be modified in the same way as any other page. For instructions on using the Page Window, to go to section 6.

### 5.2 Modifying the Footer

To modify the Header for all the pages on the website click the ~ button to the right of the "Footer" title on the left side of the form. This button is labeled in Figure 2 as **B**. This will

bring up the Page Window. The Header is considered a page and thus can be modified in the same way as any other page. For instructions on using the Page Window, to go to section 6.

### **5.3 Modifying the Home Page**

To modify the Home Page for each page on the webpage click the ~ button to the right of the Home Page title on the left side of the form. This button is labeled in Figure 2 as **C**. This will bring up the Page Window. The Home Page is the same as any other user created page except that it is required for the creation of the website. For instructions on using the Page Window, to go to section 6.

### **5.4 Adding Pages**

To add a page to the website click the + button to the right of the Pages title on the right side of the form. This button is labeled in Figure 2 as **H**. This will bring up the Page Window. For instructions on using the form which you are brought to go to section 6.

### **5.5 Modifying Pages**

To modify a page to the website, select it from the list of existing pages and click the ~ button to the right of the Pages title on the right side of the form. This button is labeled in Figure 2 as **I**. This will bring up the Page Window. For instructions on using the form which you are brought to go to section 6.

### **5.6 Removing Pages**

To remove a page from the website, select it from the list of existing pages and click the - button to the right of the Pages title on the right side of the form. This button is labeled in figure 2 as **J**.

### **5.7 Adding Documents**

To add a document to the website click the + button to the right of the Documents title on the right side of the form. This button is labeled in Figure 2 as **K**. This will bring up the Document Window. For instructions on using the form which you are brought to go to section 9.

### **5.8 Modifying Documents**

To modify a document, select it from the list of existing documents and click the ~ button to the right of the Documents title on the right side of the form. This button is labeled in Figure 2 as **L**. This will bring up the Document Window. For instructions on using the form which you are brought to go to section 9.

## 5.9 Removing Documents

To remove a document from the website, select it from the list of existing documents and click the - button to the right of the Documents title on the right side of the form. This button is labeled in figure 2 as **M**.

## 5.10 Adding Courses

To add a course to the website click the + button to the right of the Courses title on the right side of the form. This button is labeled in Figure 2 as **E**. This will bring up the Courses Window. For instructions on using the form which you are brought to go to section 10.

## 5.11 Modifying Courses

To modify a course, select it from the list of existing courses and click the ~ button to the right of the Courses title on the right side of the form. This button is labeled in Figure 2 as **F**. This will bring up the Course Window. For instructions on using the form which you are brought to go to section 10.

## 5.12 Removing Courses

To remove a course from the website, select it from the list of existing courses and click the - button to the right of the Courses title on the right side of the form. This button is labeled in figure 2 as **G**.

## 5.13 Updating the Menu

To update the menu click the ~ button to the right of the Menu title half way down the left side of the form. This button is labeled in Figure 2 as **C**. For instructions on using the form which you are brought to go to section 7.

## 5.14 Save Webpage and Close

To save and close the website click the **Save and Close** button, labeled **P**.

## 5.15 Generate Webpage

To generate the webpage, click the **Generate Webpage** button on the bottom edge of the form. This button is labeled in Figure 2 as **O**. The webpage will be generated in a directory titled "HTML" in the same directory as the program executable, i.e. WebBuilder.jar. Contained within the HTML directory you will find the HTML and other files. To view the generated webpage open the index.html file in a web browser by double clicking the file. Once you are satisfied with the generated page the contents of the HTML directory can be uploaded to your webspace to be displayed.

## 6.0 Page Window

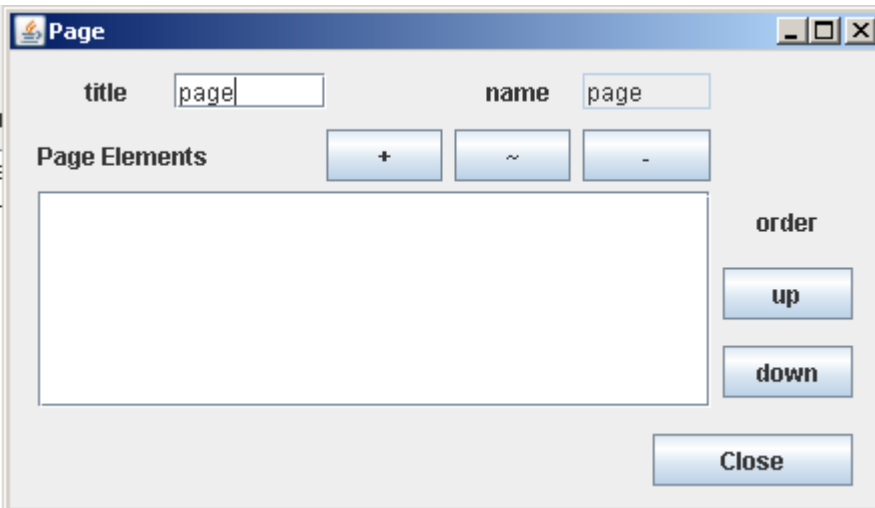


Figure 3: Page Interface

Pages are collections of page elements. To create a page first title it and then begin adding elements to the page. The order of the elements can be changed if needed.

### 6.1 Entering the Title

To enter the title type in the title field and press Enter. Pressing Enter is important as it will unlock the page element controls.

### 6.2 Adding a Page Element

To add a page element click the + button. This will open the Page Element Windows, see section 7.

### 6.3 Modifying a Page Element

To modify a page element, select a page element from the list and click the ~ button. This will open the Page Element Windows, see section 7.

### 6.4 Removing a Page Element

To remove a page element, select a page element from the list and click the - button.

### 6.5 Changing the order of the Page Elements

To change the order of the page elements in the page, select a page element and then press either the **up** or **down** button on the right side of the form.



## 7.0 Page Element Window

The page element window is a tabbed window which allows access to the 4 possible page element types. These 4 types are shown below with instructions.

### 7.1 Creating a HTML Element

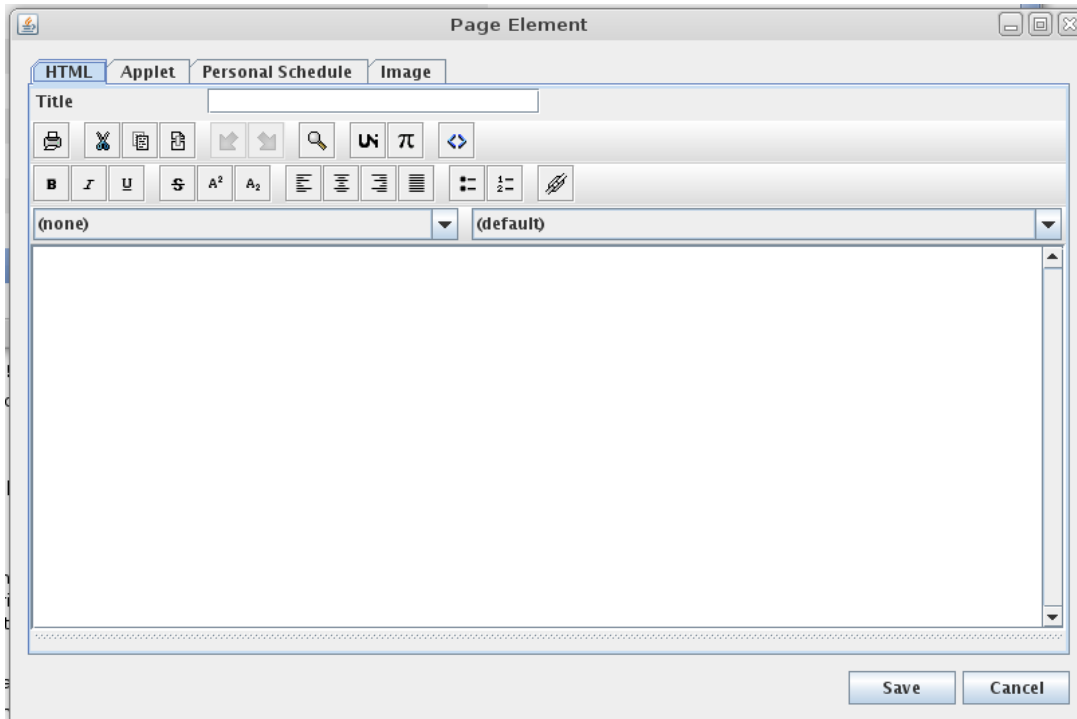


Figure 4: HTML Page Element Interface

#### 7.1.a Entering the Title

To enter the title type in the title field.

#### 7.1.b Entering HTML through the WYSIWYG Interface

To use the WYSIWYG interface type into the large provided field in the center in the HTML Page Element Window. This text and formatting will be converted to HTML upon saving.

#### 7.1.c Entering HTML directly

To enter HTML directly press the blue button containing "<>" in the top bar of the HTML Window. The field will change and the HTML field will be provided for you to enter the HTML directly.

### 7.1.d Save Data

To save the data, press the **Save** button on the lower edge of the in the Lab Document Window.

### 7.1.e Cancel Entry

If you wish to cancel the entry or updating of the Lab Document, press the **Cancel** button in the lower right of the in the HTML Element Window. No data will be saved.

## 7.2 Creating an Image Element

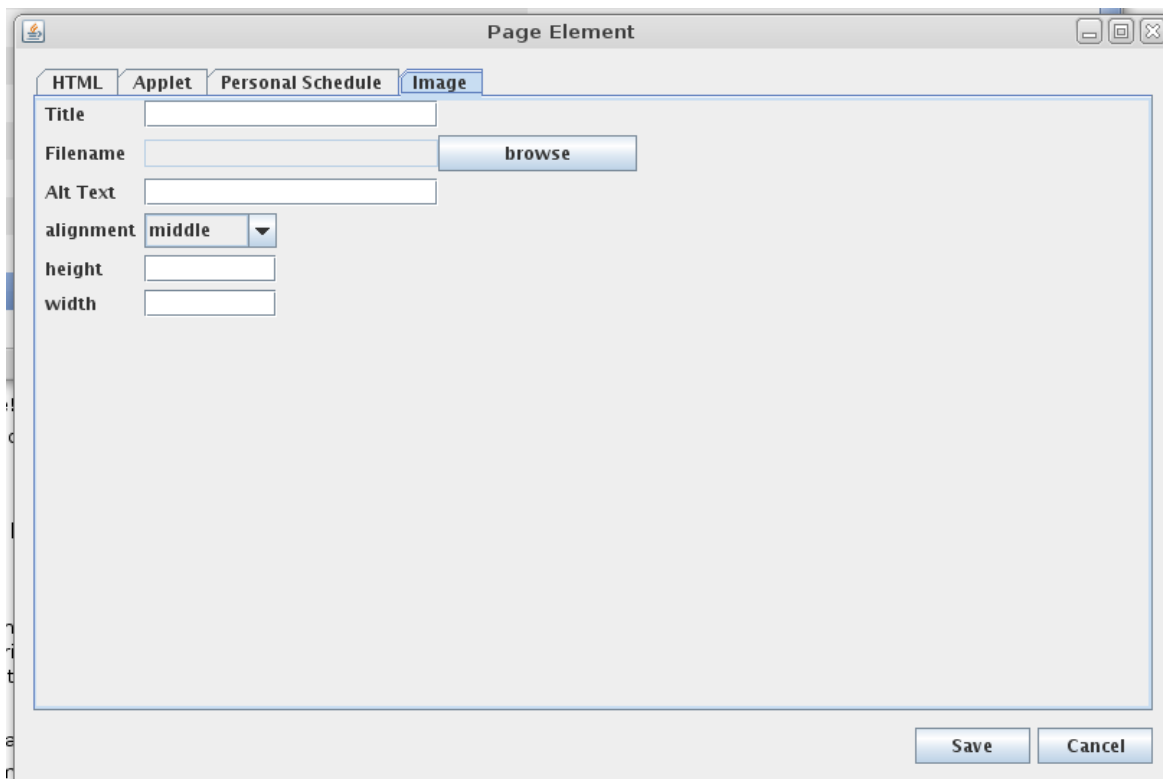


Figure 5: Image Page Element Interface

### 7.2.a Entering the Title

To enter the title type in the title field.

### 7.2.b Selecting the File

To select a file press the browse button and then navigate to the file you wish to include and press the open button in the browser window. The documents name will be displayed next to the file location title.

### **7.2.c Entering the Alt Text**

To enter the alt text type in the title field. The alt text is the text which is displayed when the user mouses over the image in the browser

### **7.2.d Selecting the Image Alignment**

Select the appropriate image alignment from the drop down menu labeled "alignment".

### **7.2.e Entering the Height**

To enter the height type, enter the images height in pixels in the height field.

### **7.2.f Entering the Width**

To enter the widthtype, enter the images width in pixels in the width field.

### **7.2.g Save Date**

To save the data, press the **Save** button on the lower edge of the in the Image Element Window.

### **7.2.h Cancel Entry**

If you wish to cancel the entry or updating of the Image Page Element, press the **Cancel** button in the lower right of the in the Page Element Window. The data will not be saved.

## 7.3 Creating an Applet Element

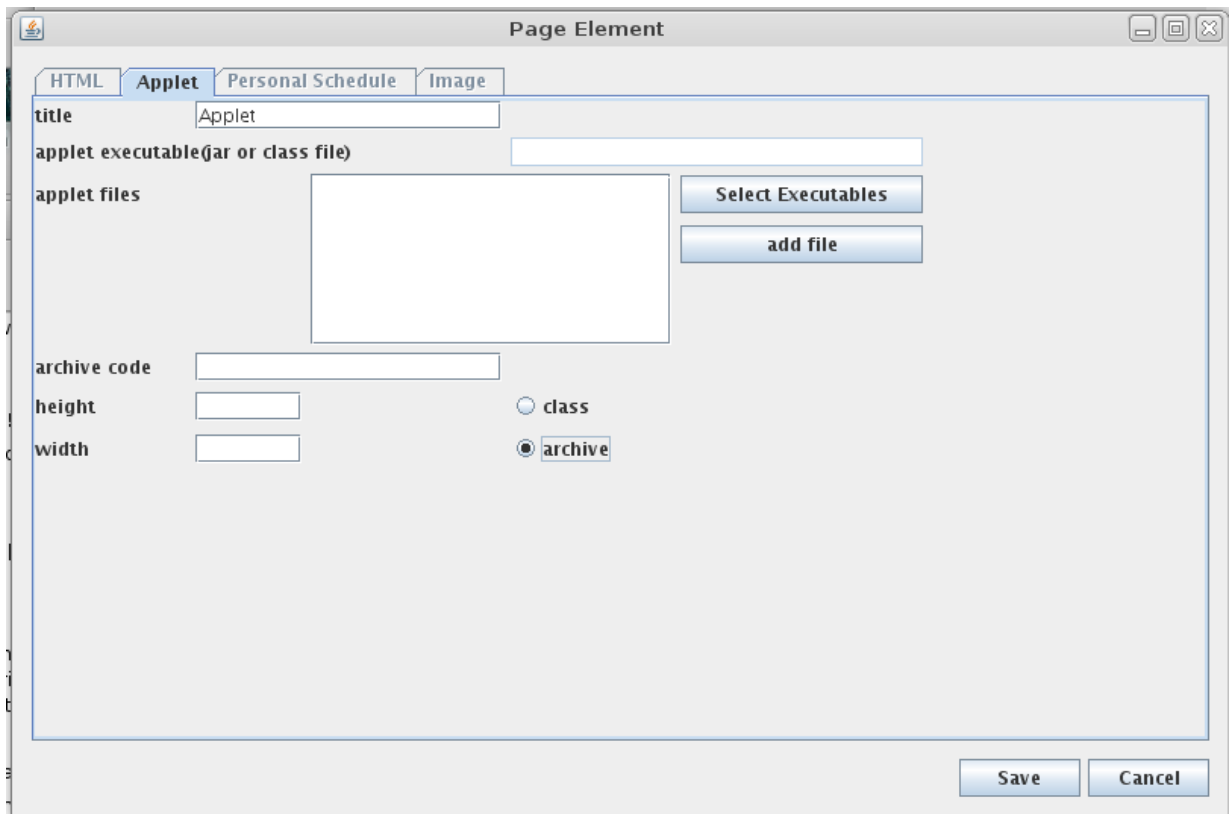


Figure 6: Applet Page Element Interface

### 7.3.a Entering the Title

To enter the title type in the title field.

### 7.3.b Selecting the Executable

To select the applet executable, i.e. the main file that will be called to initiate the applet, choose a file from the applet files list and click the button labeled "Select Executable". If no files are available in the list then you will need to add files to the applet, for instructions on adding a file to the applet see section 7.3.C.

### 7.3.c Adding a File

To select a file press the browse button and then navigate to the file you wish to include and press the open button in the browser window. The documents name will be displayed next to the applet files list.

### 7.3.d Entering the Height

To enter the height type the height in pixels in the height field.

### 7.3.e Entering the Width

To enter the width type the width in pixels in the width field.

### 7.3.f Selecting a Class or an Archive

Applets come in 2 different types, archive and class. You should know what type of applet you are using but the usual give away is the class type have more files and they have the file extension ".class". Select the applet type from the radio buttons provided. If the type is archive then the archive code field will be enabled for the codebase value for your applet to be entered.

### 7.3.g Save Data

To save the data, press the **Save** button on the lower edge of the in the Page Element Window

### 7.3.h Cancel Entry

If you wish to cancel the entry or updating of the Applet Page Element, press the **Cancel** button in the lower right of the in the Page Element Window. The data already present in the form will be present and visible in the Page Window. You may wish to remove this element, which can be done in the Page Window.

## 7.4 Creating a Personal Schedule Element

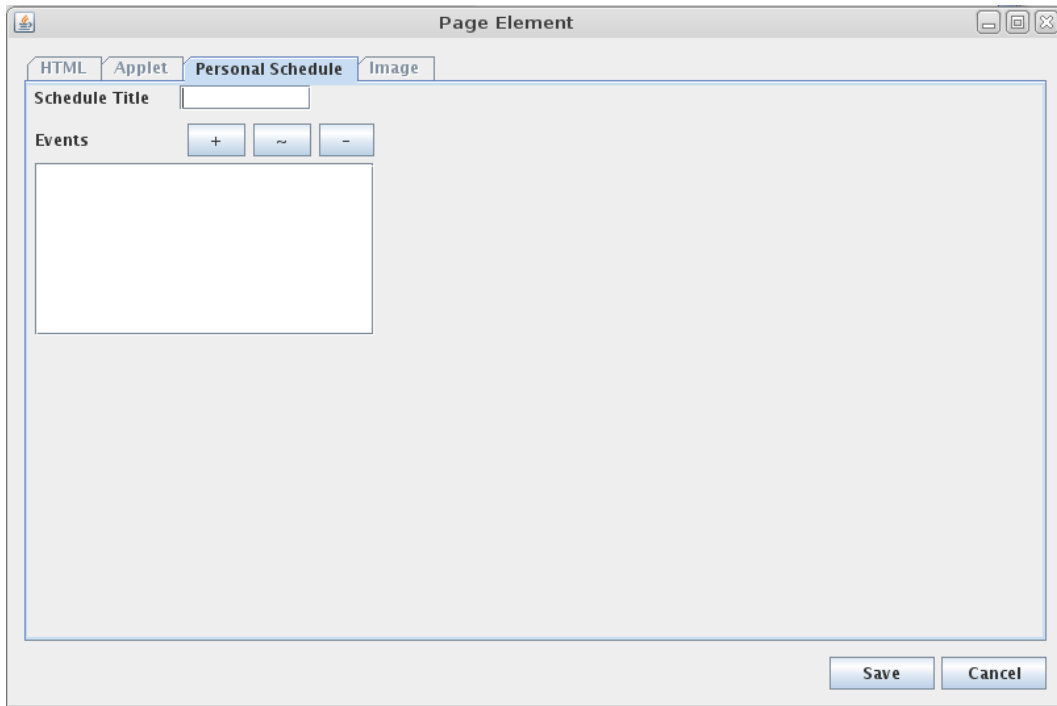


Figure 7: Personal Schedule Page Element Interface

### 7.4.a Entering the Title

To enter the title type in the title field.

### 7.4.b Adding an Event

To add an event click the + button. This will open the Schedule Event Windows, see section 7.5.

### 7.4.c Modifying an Event

To modify a page element, select a page element from the list and click the ~ button. This will open the Schedule Event Windows, see section 7.5.

### 7.4.d Removing an Event

To remove an event, select an event from the list and click the - button.

### 7.4.e Save Data

To save the data, press the **Save** button on the lower edge of the in the Page Element Window

#### 7.4.f Cancel Entry

If you wish to cancel the entry or updating of the Personal Schedule Page Element, press the **Cancel** button in the lower right of the in the Page Element Window. The data already present in the form will be present and visible in the Page Window. You may wish to remove this element, which can be done in the Page Window.

#### 7.5 Schedule Event Window

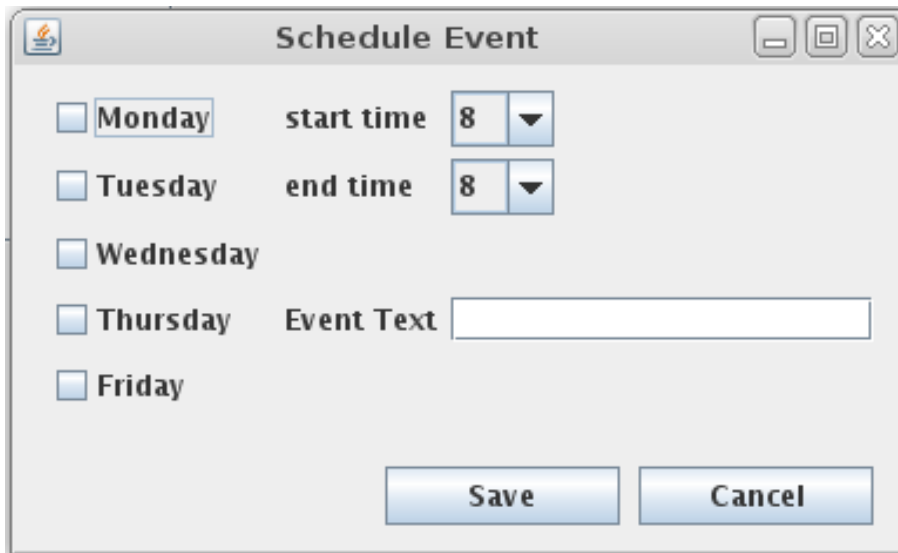


Figure 8: Personal Schedule Event Interface

##### 7.5.a Setting up the Schedule for the Event

To set the schedule for a single event click the checkboxes for the days on which the event is occurring. Also select the start and end time of the event from the dropdown menus provided with the labels "start date" and "end date".

##### 7.5.b Entering the Text of the Event

Enter the text of the event being scheduled in the field to the right of the "Event Text" label.

##### 7.5.c Save Data

To save the data, press the **Save** button on the lower edge of the in the Schedule Event Window

### 7.5.d Cancel Entry

If you wish to cancel the entry or updating of the Lab Document, press the **Cancel** button in the lower right of the in the Lab Document Window. No data will be saved.

## 8.0 Menu Window

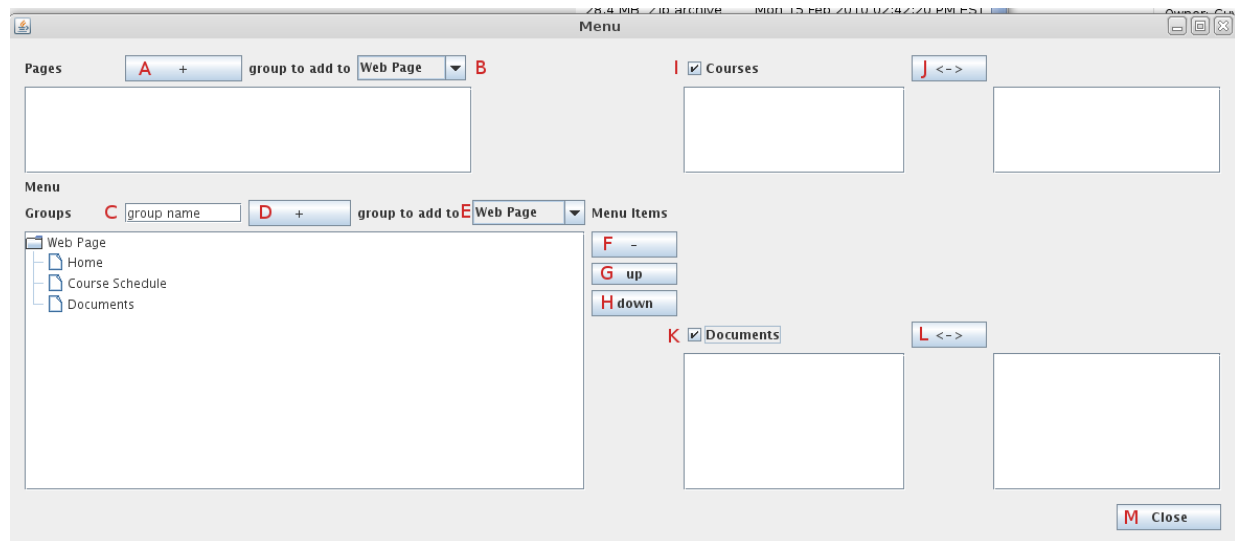


Figure 9: Menu Interface

### 8.1 Adding a Page to the Menu

To add a page to the menu select the page from the list of pages below the "Pages" label. Select a group from the drop down menu labeled **B** in Figure 9, this will determine what group/level of the menu to which the page will be added. Once the page and the group have been selected, press the + button labeled **A** in Figure 9.

### 8.2 Adding a Group to the Menu

To add a group to the menu type the name of the new group in the group\_name field, labeled **C** in Figure 9. Select a group from the drop down menu labeled **E** in Figure 9, this will determine the group/level of the menu to which the group will be added. Once the group name has been entered and the parent group has been selected from the drop down, press the + button labelled **D** in Figure 9.

### 8.3 Removing a Menu Item

To remove an item from the menu select the menu item to be removed from the menu tree below the Groups label in the bottom left corner of the form. Once the menu item is selected press the - button, labeled **F** in Figure 9.



## **8.4 Changing the Order of Menu Items**

To change the order of a item within its group, select the menu item from the menu tree below the Groups label in the bottom left corner of the form. Once the menu item is selected press the "up" or "down" buttons, labeled **G** and **H** in Figure 9, to move the menu item up or down in the menu group it is contained by.

## **8.5 Enabling Courses**

To enable the creation of courses in your website check the courses checkbox, labeled **I** in Figure 9. When you have checked the box the course selection interface will be enabled.

## **8.6 Adding and Removing Courses**

To add documents to your website select a document from the document list on the right side of the page and click the button labeled **L** in Figure 9. This will cause the selected document to be included when the website is generated.

To remove documents from your website select a document from the document list under the course title and checkbox and click the button labeled **L** in Figure 9. This will cause the selected document to no longer be included when the website is generated.

## **8.7 Enabling Document**

To enable the creation of the documents page in your website check the documents checkbox, labeled **K** in Figure 9. When you have checked the box the document selection interface will be enabled.

## **8.8 Adding and Removing Documents**

To add courses to your website select a course from the course list on the right side of the page and click the button labeled **J** in Figure 9. This will cause the selected course to be included when the website is generated.

To remove courses from your website select a course from the course list under the course title and checkbox and click the button labeled **J** in Figure 9. This will cause the selected course to no longer be included when the website is generated.

## **8.9 Closing the form**

To close the form click the "Close" button in the lower right corner of the form, labeled **M** in Figure 9.

## 9.0 Document Window

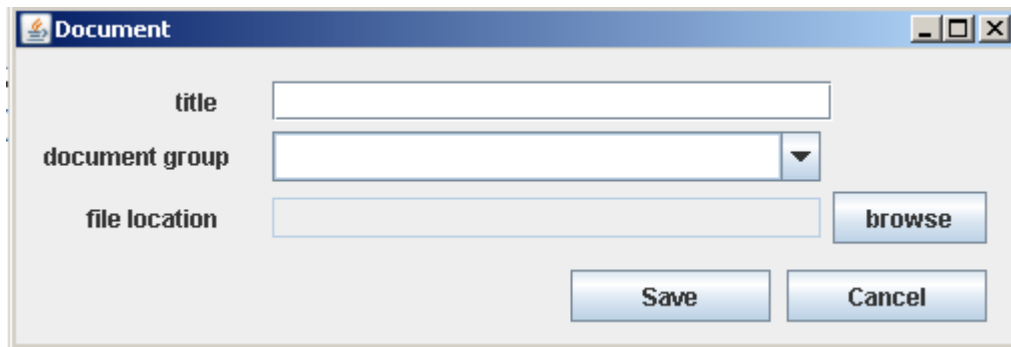


Figure 10: Document Interface

### 9.1 Entering the Title

Type the title of the document in the field labeled title in the Lab Document Window.

### 9.2 Selecting the Group

Documents are grouped on the documents page based on their group. To choose a pre-existing group select it from the drop down menu next to the document group title. If no groups have been entered or none fit your needs then create your own group by typing the group name into the drop down field.

### 9.3 Selecting the File

To select a file press the browse button and then navigate to the file you wish to include and press the open button in the browser window. The document's name will be displayed next to the file location title.

### 9.4 Save Data

To save the data, press the **Save** button on the lower edge of the in the Document Window.

### 9.5 Cancel Entry

If you wish to cancel the entry or updating of the Lab Document, press the **Cancel** button in the lower right of the in the Lab Document Window. No data will be saved.

## 10.0 Course Window

The screenshot shows a window titled "Course" with the following fields and controls:

- Title:** A text input field containing the word "Course".
- Course Number:** An empty text input field.
- Department Code (PH):** A text input field containing "PH".
- Academic Year (YY):** A text input field containing "YY".
- Term:** A dropdown menu currently showing "A".
- start date (YYYY-MM-DD):** A text input field containing "YYYY-MM-DD".
- end date (YYYY-MM-DD):** A text input field containing "YYYY-MM-DD".
- Lecturer:** A text input field with buttons "A Select", "B +", "C ~", and "D -" below it.
- Coordinator:** A text input field with buttons "E Select", "F +", "G ~", and "H -" below it.
- Lab Coordinator:** A text input field with buttons "I Select", "J +", "K ~", and "L -" below it.
- Labs:** A large empty text area with buttons "M +", "N ~", and "O -" above it.
- Sections:** A large empty text area with buttons "P +", "Q ~", and "R -" above it.
- Bottom Right:** "S Save" and "T Cancel" buttons.

Figure 11: Course Interface

### 10.1 Entering the Title

To enter the title for the course, type the title in the field labeled "Title" at the top of the form and press Enter. Pressing Enter will enable the course elements in the form. Be sure to press Enter

### 10.2 Entering the Course Number

To enter the course number type the course number in the field labeled "Course Number" at the top of the form.

### 10.3 Entering Academic Year

To enter the academic year type the last 2 digits of the year in the field labeled "Academic Year" at the top of the form

### 10.4 Selecting the Term

To select the term the course will be offer in, select A, B, C, D or E from the drop down menu labeled "Term" at the top of the form.

### 10.5 Entering the Start Date

To enter the start date of the course, type the date in the field labeled "start date" in the format YYYY-MM-DD. This format is important so be sure to stick to this date format.

## **10.6 Entering the End Date**

To enter the end date of the course, type the date in the field labeled "end date" in the format YYYY-MM-DD. This format is important so be sure to stick to this date format.

## **10.7 Selecting a Lecturer**

To select the lecturer, choose a lecturer from the list on the left side of the form, under the Lectuer label, and press the button labelled "Select" and labeled **A** in Figure 11. If no lecturers are listed enter one by following the instructions in section 10.8.

## **10.8 Adding a Lecturer**

To add a lecturer click the + button labeled **B** in Figure 11. This will bring up the Lecturer Window. For instructions on using the form which you are brought to go to section 18.

## **10.9 Modifying a Lecturer**

To modify a lecturer click the ~ button labeled **C** in Figure 11. This will bring up the Lecturer Window. For instructions on using the form which you are brought to go to section 18.

## **10.10 Removing a Lecturer**

To remove a lecturer, choose a lecturer from the list on the left side of the form, under the Lectuer label, and click the - button labeled **D** in Figure 11.

## **10.11 Selecting a Coordinator**

To select the coordinator, choose a coordinator from the list in the middle of the form, under the Coordinator label, and press the button labelled "Select" and labeled **E** in Figure 11. If no coordinators are listed enter one by following the instructions in section 10.12.

## **10.12 Adding a Coordinator**

To add a coordinator click the + button labeled **F** in Figure 11. This will bring up the Coordinator Window. For instructions on using the form which you are brought to go to section 17.

## **10.13 Modifying a Coordinator**

To modify a coordinator click the ~ button labeled **G** in Figure 11. This will bring up the Coordinator Window. For instructions on using the form which you are brought to go to section 17.

### **10.14 Removing a Coordinator**

To remove a coordinator, choose a coordinator from the list in the middle of the form, under the Coordinator label, and click the - button labeled **H** in Figure 11.

### **10.15 Selecting a Lab Coordinator**

To select the lab coordinator, choose a lab coordinator from the list on the right side of the form, under the Lab Coordinator label, and press the button labelled "Select" and labeled **I** in Figure 11. If no lab coordinators are listed enter one by following the instructions in section 10.16.

### **10.16 Adding a Lab Coordinator**

To add a lab coordinator click the + button labeled **J** in Figure 11. This will bring up the Lab Coordinator Window. For instructions on using the form which you are brought to go to section 16.

### **10.17 Modifying a Lab Coordinator**

To modify a lab coordinator click the ~ button labeled **K** in Figure 11. This will bring up the Lab Coordinator Window. For instructions on using the form which you are brought to go to section 17.

### **10.18 Removing a Lab Coordinator**

To remove a lab coordinator, choose a lab coordinator from the list on the left side of the form, under the Lab Coordinator label, and click the - button labeled **L** in Figure 11.

### **10.19 Adding a Lab**

To add a lab click the + button labeled **M** in Figure 11. This will bring up the Lab Window. For instructions on using the form which you are brought to go to section 13.

### **10.20 Modifying a Lab**

To modify a lab, select the lab from the list in the lower left hand corner of the form. Then click the ~ button labeled **N** in Figure 11. This will bring up the Lab Window. For instructions on using the form which you are brought to go to section 13.

### **10.21 Removing a Lab**

To remove a lab, choose a lab from the list in the lower left hand corner of the form, under the "Lab" label, and click the - button labeled **O** in Figure 11.

### **10.22 Adding a Section**

To add a section click the + button labeled **P** in Figure 11. This will bring up the Section Window. For instructions on using the form which you are brought to go to section 11.

### **10.23 Modifying a Section**

To modify a section, select the section from the list in the lower right hand corner of the form. Then click the ~ button labeled **Q** in Figure 11. This will bring up the Section Window. For instructions on using the form which you are brought to go to section 11.

### **10.24 Removing a Section**

To remove a section, choose a section from the list in the lower right hand corner of the form, under the "Lab" label, and click the - button labeled **R** in Figure 11.

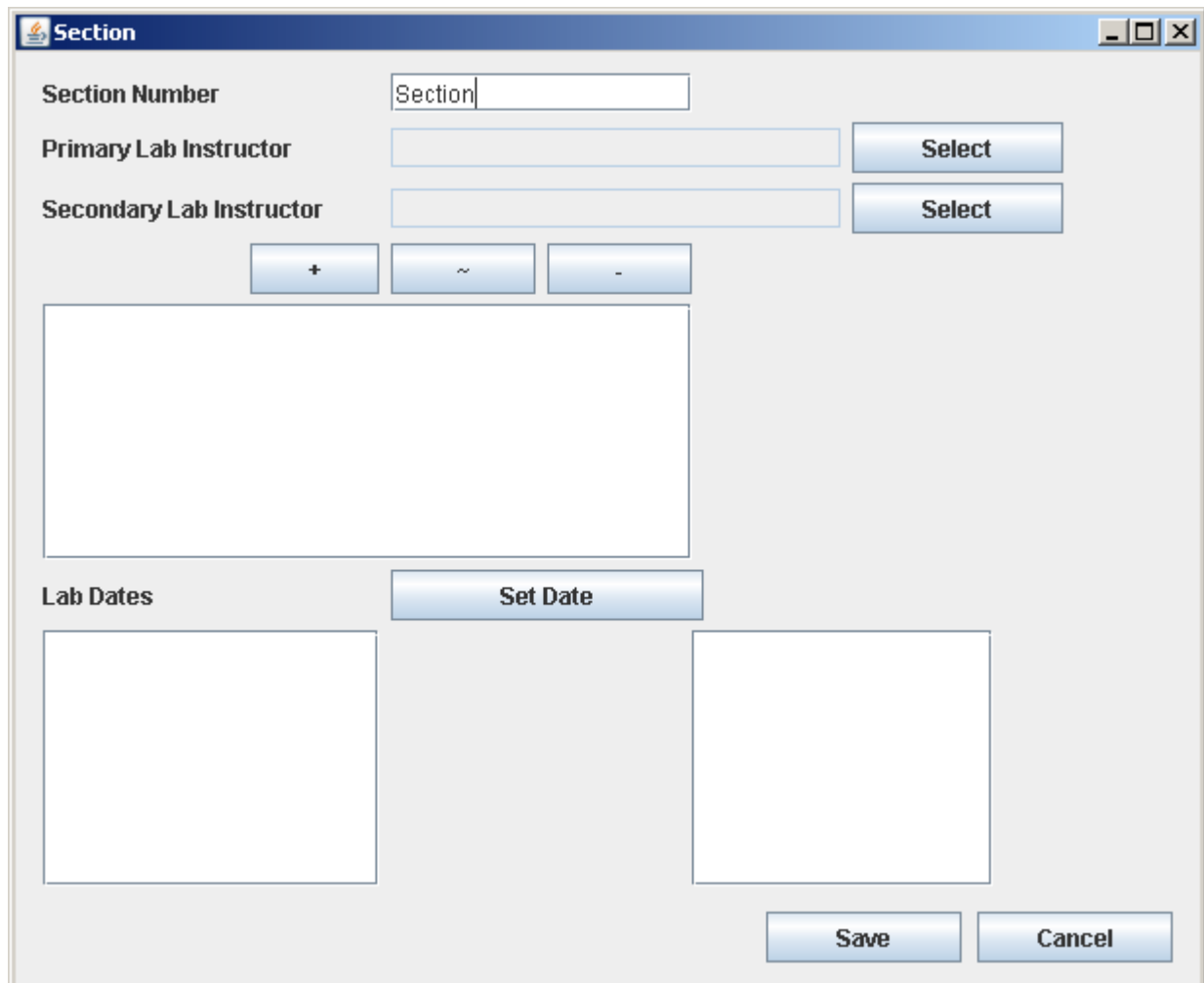
### **10.25 Save Data**

To save the data, press the **Save** button on the lower edge of the in the Course Window, labeled as **S** in Figure 11.

### **10.26 Cancel Entry**

If you wish to cancel the entry or updating of the Course, press the **Cancel** button in the lower right of the in the Course Window, labeled as **T** in Figure 11. No data will be saved.

## 11.0 Section Window



The screenshot shows a software window titled "Section". At the top left is a small icon and the title "Section". The window contains several input fields and buttons. The "Section Number" field has the text "Section" inside. Below it are two "Primary Lab Instructor" and "Secondary Lab Instructor" fields, each with a "Select" button to its right. Underneath these are three buttons: "+", "~", and "-". A large empty rectangular box is positioned below these buttons. The "Lab Dates" section features a "Set Date" button centered above two empty rectangular boxes. At the bottom right of the window are "Save" and "Cancel" buttons.

Figure 12: Section Interface

### 11.1 Entering the Section Number

Type the Section Number in the field labeled Section Number and press the *Enter* button. When the *Enter* button is pressed the Section controls are enabled for use and the section is created in the system. Therefore be certain to hit Enter.

### 11.2 Select Primary Lab Instructor

To select the primary lab instructor, choose a lab instructor from the list in the center of the form and press the upper of the two buttons labelled "Select". If no lab instructors are listed enter one by following the instructions in section 11.4.

### **11.3 Select Secondary Lab Instructor**

To select the secondary lab instructor, choose a lab instructor from the list in the center of the form and press the lower of the two buttons labelled "Select". If no lab instructors are listed enter one by following the instructions in section 11.4.

### **11.4 Adding a Lab Instructor**

To add a lab instructor click the + button below the Secondary Lab Instructor title. This will bring up the Lab Instructor Window. For instructions on using the form which you are brought to go to section 14.

### **11.5 Modifying a Lab Instructor**

To modify a lab instructor click the ~ button below the Secondary Lab Instructor title. This will bring up the Lab Instructor Window. For instructions on using the form which you are brought to go to section 14.

### **11.6 Removing a Lab Instructor**

To remove a lab instructor, select it from the list of existing lab instructors and click the - button below the Secondary Lab Instructor title.

### **11.7 Setting the Date of a Lab for the Section**

To set the date of a lab for the current section choose a lab from the list of labs on the bottom right of the form and press the button labelled "Set Date". This will open the Set Lab Date Window. For instructions on using this forms go to section 12.

### **11.8 Modifying the Date of a Lab for the Section**

To modify the date of a lab which has already been set choose a lab from the list of labs on the bottom left of the form and press the button labelled "Set Date". This will open the Set Lab Date Window and allow you to modify the data. For instructions on using this forms go to section 12.

### **11.9 Save Data**

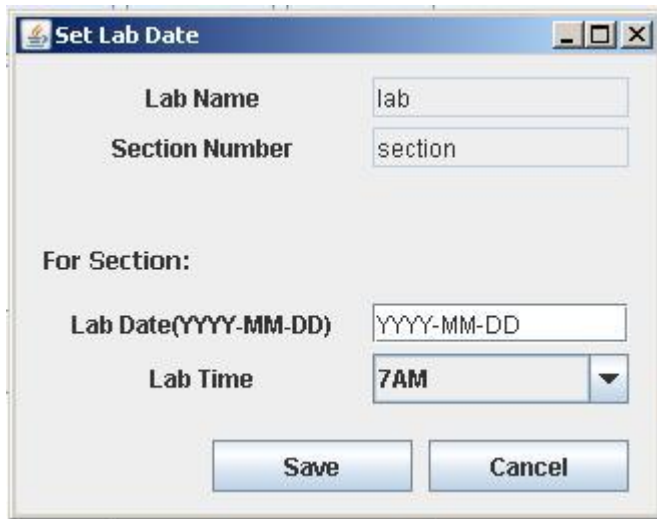
To save the data, press the Save button on the lower edge of the in the Lab Window.

### **11.10 Cancel Entry**

If you wish to cancel the entry or updating of the Lecturer, press the Cancel button in the lower right of the in the Lab Window. No data will be saved.



## 12.0 Set Lab Date Window



The image shows a dialog box titled "Set Lab Date". It has a standard Windows-style title bar with minimize, maximize, and close buttons. The dialog contains the following elements:

- Lab Name:** A text input field containing the text "lab".
- Section Number:** A text input field containing the text "section".
- For Section:** A label indicating the data is for a specific section.
- Lab Date(YYYY-MM-DD):** A text input field with a placeholder "YYYY-MM-DD".
- Lab Time:** A dropdown menu currently displaying "7AM".
- Buttons:** Two buttons at the bottom, "Save" and "Cancel".

Figure 13: Lab Interface

### 12.1 Enter Lab Date

To enter the lab date for the section, type the date in the field labeled "Lab Date" in the format YYYY-MM-DD. This format is important so be sure to stick to this date format.

### 12.2 Enter Lab Time

To Enter the lab time, select the time of the lab for this section from the drop down menu labeled Lab Time.

### 12.3 Save Data

To save the data, press the Save button on the lower edge of the in the Set Lab Date Window.

### 12.4 Cancel Entry

If you wish to cancel the entry or updating of the Lab Date, press the Cancel button in the lower right of the in the Set Lab DateWindow. No data will be saved.

## 13.0 Lab Window

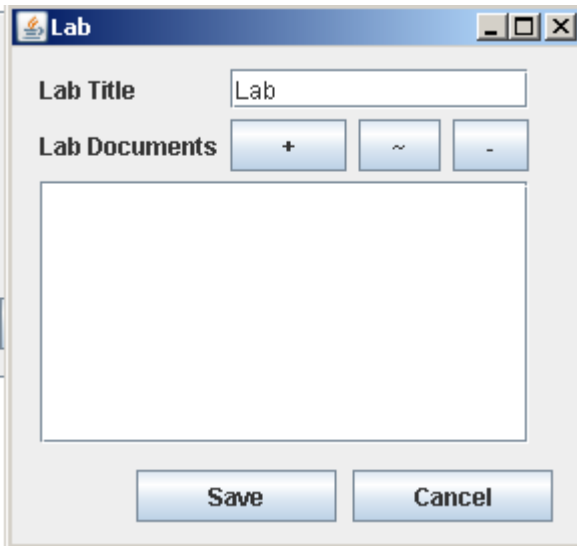


Figure 14: Lab Interface

### 13.1 Entering the Title

Type the title of the lab in the field labeled Lab Title and press the *Enter* button. When the *Enter* button is pressed the lab documents controls are enabled for use and the lab is created in the system. Therefore be certain to hit Enter.

### 13.2 Add Lab Documents

To add a lab document click the + button to the right of the Lab Documents title. This will bring up the Lab Document Window. For instructions on using the form which you are brought to go to section 13.

### 13.3 Modifying a Lab Document

To modify a lab document select it from the list of existing lab documents and click the ~ button to the right of the Lab Documents title. This will bring up the Lab Document Window. For instructions on using the form which you are brought to go to section 13.

### 13.4 Removing a Lab Document

To remove a lab document, select it from the list of existing lab documents and click the - button to the right of the Lab Documents title on the right side of the form.

### 13.5 Save Data

To save the data, press the **Save** button on the lower edge of the in the Lab Window.

### 13.6 Cancel Entry

If you wish to cancel the entry or updating of the Lecturer, press the **Cancel** button in the lower right of the in the Lab Window. No data will be saved.

### 14.0 Lab Document Window

The Lab Document Window is a tabbed window which allows access to the 2 possible lab document types. These 2 types are shown below with instructions.

#### 14.1 File

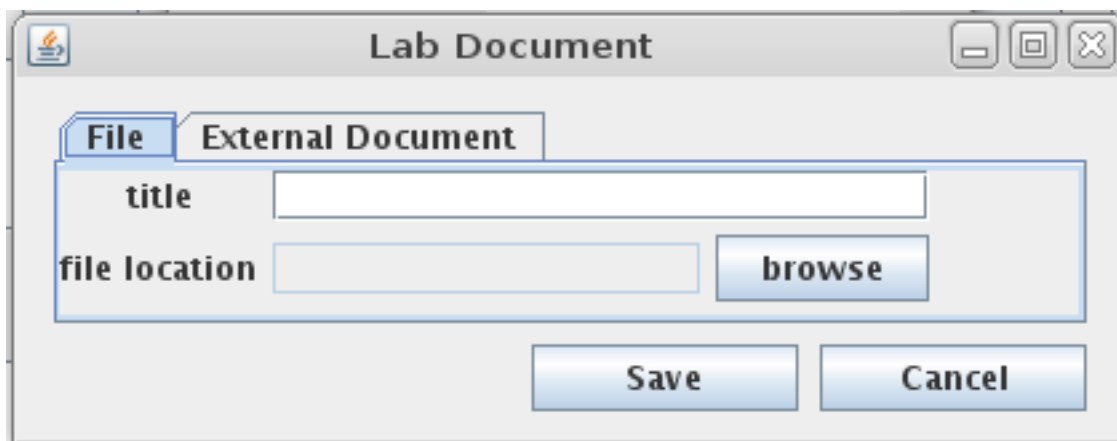


Figure 15: Lab Document Interface (File)

##### 14.1.a Entering the Title

Type the title of the document in the field labeled title in the Lab Document Window.

##### 14.1.b Selecting the File

To select a file press the browse button and then navigate to the file you wish to include and press the open button in the browser window. The documents name will be displayed next to the file location title.

##### 14.1.c Save Data

To save the data, press the **Save** button on the lower edge of the in the Lab Document Window.

##### 14.1.c Cancel Entry

If you wish to cancel the entry or updating of the Document, press the **Cancel** button in the lower right of the in the Lab Document Window. No data will be saved.

## 14.2 URL

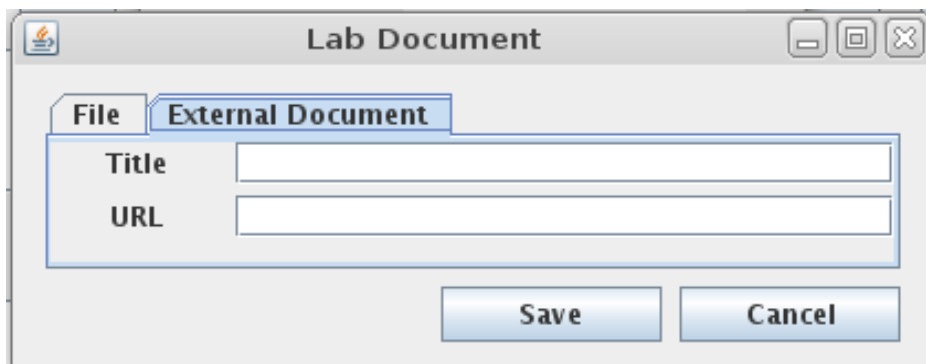


Figure 16: Lab Document Interface (URL)

### 14.2.a Entering the Title

Type the title of the document in the field labeled title in the Lab Document Window.

### 14.2.b Entering the URL

Type the URL of the document in the field labeled URL in the Lab Document Window.

### 14.2.c Save Data

To save the data, press the **Save** button on the lower edge of the in the Lab Document Window.

### 14.2.d Cancel Entry

If you wish to cancel the entry or updating of the Lab Document, press the **Cancel** button in the lower right of the in the Lab Document Window. No data will be saved.

## 15.0 Lab Instructor Window

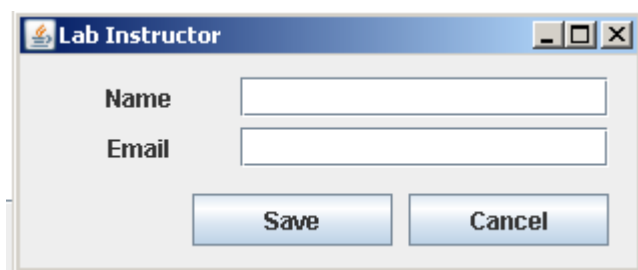


Figure 17: Lab Instructor Interface

### 15.1 Entering the Name

Type the Lab Instructor Name in the field labeled Name in the Lab Instructor Window.

### 15.2 Entering the Email Address

Type the Lab Instructor Email Address in the field labeled Email in the Lab Instructor Window.

### 15.4 Save Data

To save the data, press the **Save** button on the lower edge of the Lab Instructor Window.

### 15.5 Cancel Entry

If you wish to cancel the entry or updating of the Lab Instructor, press the **Cancel** button in the lower right of the Lab Instructor Window. No data will be saved.

## 16.0 Lab Coordinator Window

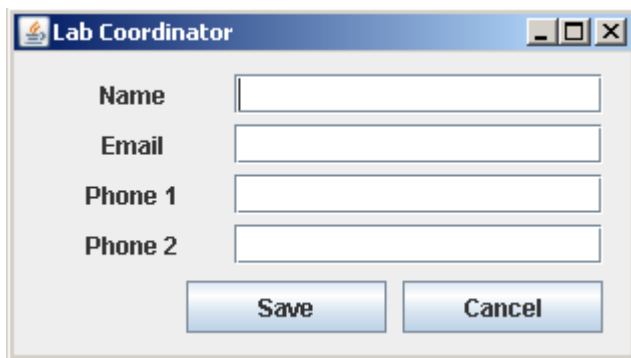
The image shows a screenshot of a software window titled "Lab Coordinator". The window has a standard Windows-style title bar with a blue background and a small icon on the left. Below the title bar, there are four text input fields stacked vertically. The labels for these fields are "Name", "Email", "Phone 1", and "Phone 2". At the bottom of the window, there are two buttons: "Save" on the left and "Cancel" on the right. The window is set against a light gray background.

Figure 18: Lab Coordinator Interface

### 16.1 Entering the Name

Type the Lab Coordinator Name in the field labeled Name in the Lab Coordinator Window.

### 16.2 Entering the Email Address

Type the Lab Coordinator Email Address in the field labeled Email in the Lab Coordinator Window.

### 16.3 Entering the Phone Number

Type the Lab Coordinator Phone Number in the field labeled Phone 1 in the Lab Coordinator Window. If you have a second phone number enter it in the Phone 2 field.

## 16.4 Save Data

To save the data, press the **Save** button on the lower edge of the Lab Coordinator Window.

## 16.5 Cancel Entry

If you wish to cancel the entry or updating of the Lab Coordinator, press the **Cancel** button in the lower right of the Lab Coordinator Window. No data will be saved.

## 17.0 Coordinator Window

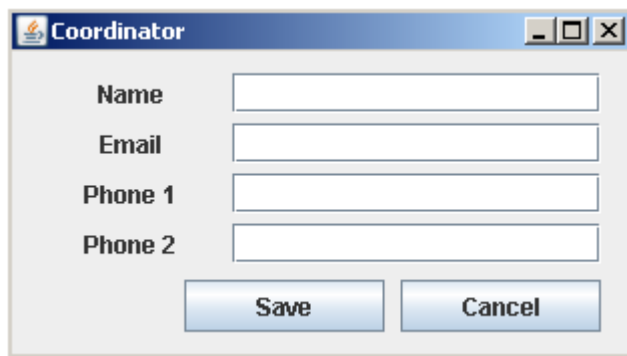


Figure 19: Coordinator Interface

### 17.1 Entering the Name

Type the Coordinator Name in the field labeled Name in the Coordinator Window.

### 17.2 Entering the Email Address

Type the Coordinator Email Address in the field labeled Email in the Coordinator Window.

### 17.3 Entering the Phone Number

Type the Coordinator Phone Number in the field labeled Phone 1 in the Coordinator Window. If you have a second phone number enter it in the Phone 2 field.

### 17.4 Save Data

To save the data, press the **Save** button on the lower edge of the Coordinator Window.

### 17.5 Cancel Entry

If you wish to cancel the entry or updating of the Coordinator, press the **Cancel** button in the lower right of the in the Coordinator Window. No data will be saved.

## 18.0 Lecturer Window

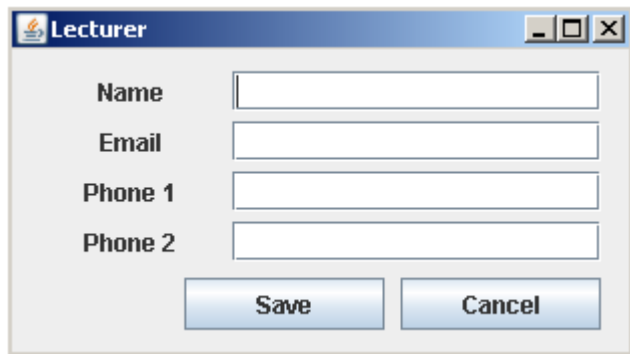
The image shows a software window titled "Lecturer". Inside the window, there are four text input fields stacked vertically, each with a label to its left: "Name", "Email", "Phone 1", and "Phone 2". At the bottom of the window, there are two buttons: "Save" on the left and "Cancel" on the right. The window has a standard Windows-style title bar with minimize, maximize, and close buttons.

Figure 20: Lecturer Interface

### 18.1 Entering the Name

Type the Lecturer Name in the field labeled Name in the Lecturer Window.

### 18.2 Entering the Email Address

Type the Lecturer Email Address in the field labeled Email in the Lecturer Window.

### 18.3 Entering the Phone Number

Type the Lecturer Phone Number in the field labeled Phone 1 in the Lecturer Window. If you have a second phone number enter it in the Phone 2 field.

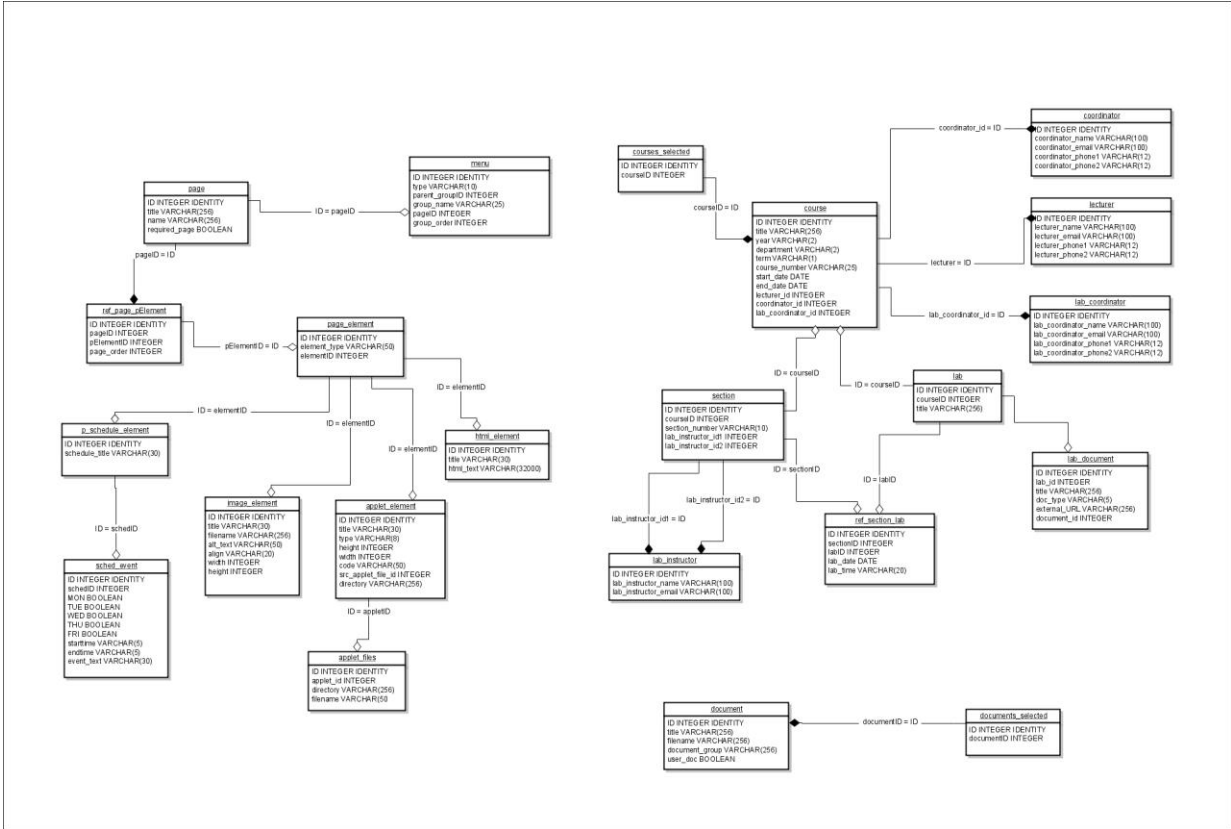
### 18.4 Save Data

To save the data, press the **Save** button on the lower edge of the in the Lecturer Window.

### 18.5 Cancel Entry

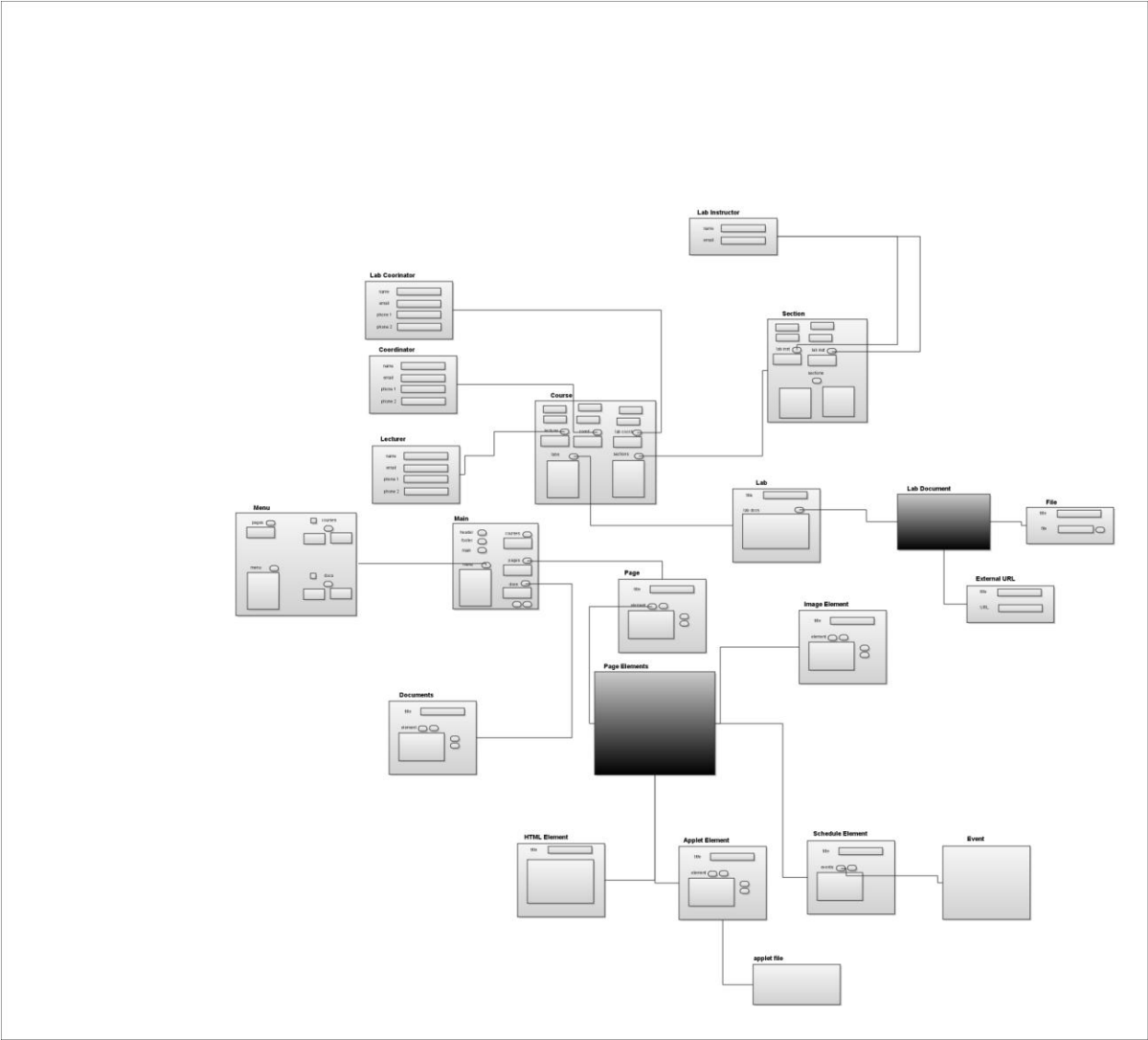
If you wish to cancel the entry or updating of the Lecturer, press the **Cancel** button in the lower right of the in the Lecturer Window. No data will be saved.

# Appendix I - Database Diagram





# Appendix J - UI Diagram



# Appendix K - Timeline

