Building a Platform for Data Visualization Literacy

A Major Qualifying Project (MQP) Report Submitted to the Faculty of WORCESTER POLYTECHNIC INSTITUTE in partial fulfillment of the requirements for the Degree of Bachelor of Science

Prepared by: Joseph Doebbelaar, Luke Foley, Ryker Germain, Matthew McAlarney

Advisor: Professor Lane Harrison

April 2024

Abstract

The ability to interpret complex data visualizations is a crucial skill that appeals to professional and personal environments. Currently, the ability to understand visualizations on a mobile device is one that is hyper-critical yet underexplored. This project addresses the gap in accessible data visualization literacy (DVL) education by developing a mobile application that enables flexible learning through engaging, brief lessons on both common visualization types and complex, emerging charts used in academia. Unlike existing platforms, which are web-based and require extended periods of engagement, our app introduces a modern, interactive approach that integrates seamlessly into daily life, making learning both effective and convenient.

This platform uses interactive modules, gamification, and immediate feedback to teach users to critically understand and evaluate visual data, building off of the teaching strategies of platforms like Khan Academy and Duolingo. The app's design is optimized for mobile use, with clear, legible visualizations tailored for small screens and intuitive navigation facilitated by icon-based menus. Through our app, users engage with varied data visualizations, answer related questions, and receive instant feedback, which aids in building their data visualization literacy.

This report outlines the design, development, and potential future enhancements of the app, including an expanded content library and features for personalized learning. It emphasizes the application's role in empowering users to navigate the increasingly data-rich world more effectively, thereby contributing to a well-informed society. The development process, led by a cross-functional student team, highlights collaborative efforts in software engineering, user interface design, and educational content creation, marking a significant step toward making DVL more accessible and impactful.

Acknowledgements

We would like to thank Professor Lane Harrison and PhD student Yiren Ding for helping us guide our Data Visualization Literacy MQP and providing meaningful feedback on our work. Through the app development process, design, and content creation phases, Professor Lane Harrison and Yiren Ding fully supported our efforts to build a platform for data visualization learning.

Table of Contents

Chapter 1: Introduction

In today's information rich world, data visualizations have become increasingly common, serving as critical tools across a diverse range of domains: education, business, news media, health care, government, and beyond. These visual representations—graphs, charts, maps, etc.— translate complex data sets into intuitive visual formats, leveraging human strengths in recognizing patterns, colors, shapes, relative positions, and trends. Visualizations are powerful tools for making informed decisions and understanding the world more deeply. However, as these visualizations get more complex, interpreting them effectively is non-trivial; it is a skill that must be developed, necessitating better DVL among the general public. This skill enables individuals to not only understand the data presented in a visualization, but to engage with it critically, evaluate the data's validity, and foster a more informed and discerning society.

Despite the prevalence of data visualizations in daily life, there remains a significant gap in the availability of educational resources specifically designed to teach the general public how to effectively understand them. As the visualizations we see every day incorporate increasingly sophisticated data and statistical concepts, the need for accessible, user-friendly educational platforms becomes even more critical. Existing educational platforms, while valuable, often require prolonged engagement and are confined to desktop environments. In contrast, a mobile application would address this limitation by allowing users to quickly engage with lessons through their smartphones. Putting DVL education in an individual's pocket allows them to learn on-the-got, fitting brief, impactful lessons into their busy schedules—whether they're preparing for a meeting on short notice, wanting to interpret a graph they saw on the news, or making sure that they understand the statistic that their doctor is communicating to them. By making DVL accessible in moments and places where traditional tools cannot reach, the introduction of a mobile platform for assessing and strengthening data visualization skills will enhance learning efficiency and convenience, as well as encouraging regular interaction and sustained learning engagement.

Fortunately, in recent years, we have witnessed significant advancements in online learning platforms that have revolutionized education. Tools like Khan Academy have demonstrated the power of interactive, user-friendly educational technologies to engage users and enhance understanding through intuitive user interfaces, quick-to-complete lessons, and personalized curricula. It also shows how "gamifying" their visualizations leads to an increase in user interactiveness [1]. These platforms leverage adaptive learning technologies, gamification, and immediate feedback to cater to the needs of learners, proving that complex subjects can be mastered, with the right tools and approaches, outside of the classroom.

Building on this foundation, mobile learning applications like Duolingo have taken these advancements a step further by bringing the convenience of learning to users' pockets. The ubiquity of smartphones has enabled apps to deliver educational content in a more accessible format, allowing for spontaneous and flexible learning. Our project sought to create a mobile platform for assessing and strengthening DVL, providing lessons that are not only engaging and informative, but also readily available whenever and wherever they are needed.

Building a DVL tool for mobile platforms presented one primary hurdle: maintaining clarity and effectiveness of visualizations when scaled down to accommodate the limited screen real estate available on smartphones. Modeled after existing mobile education platforms, we created a thoughtful interface that keeps visualization legible on small screens. A secondary concern was that, since screen size is limited, users may be forced to scroll away from the visuals to interact with the data. By creating a set of button-based answer submission methods, we were able to circumvent this issue as well.

The development of our mobile application is the result of a collaborative effort. Luke Foley took charge of the user interface design, crafting Fogma mockups and applying styles to ensure the app's functionality, aesthetic appeal, and responsiveness on mobile devices. Ryker Germain focused on research and documentation, teaming up with Matthew McAlarney—who's contributions also included bringing much of the research to the front-end— to craft lessons and questions that are engaging and effective in teaching users. Joe Dobbelaar, as the software lead, constructed the backbone of our application in React Native, integrated the Google Firebase backend, and hosted the project on a personal web server for the duration of development.

This project sought to create a mobile platform for assessing and strengthening data visualization skills through practice, teaching techniques, and constructive feedback. Existing tools are much more academically-focused and are developed for web-based environments, so our initiative fills the gap through creating a mobile-first solution, allowing DVL exercises to fit seamlessly into the daily routines of the average person, encouraging regular interaction and sustained learning engagement. Over the course of this project, we built a mobile DVL tool with a focus on accessibility and convenience; all charts are readable on a small screen, users can interact with data without the need to scroll away from the visuals, and the navigation between lessons is intuitive.

In this paper, we will discuss the landscape of existing tools and studies related to DVL, highlighting the gaps that our mobile application seeks to fill. We will outline the methodologies that contributed to the design and development of our application, as well as the rich educational content that is offered in our app. Finally, we will explore potential paths for the future growth of our app, including its expansion into business and educational settings through a comprehensive organization suite.

Chapter 2: Background

2.1 Data Visualization vs Data Visualization Literacy (DVL)

The difference between data visualization and DVL was a major distinction of our project. Data visualization is the act of taking data and creating a visualization to convey a message, while DVL is geared towards teaching the reader how to decipher an already-existing visualization. Essentially, it boiled down to “How much can we expect our users to know, and what are we trying to teach with our app?” By establishing the goal of the platform, we were able to target our user base as well as tailor our content to create a more specific, cohesive experience.

DVL is becoming increasingly important in the age of technology, especially as society further evolves [2]. However, this skill is currently being neglected and isn’t widely tested, according to a survey of interactive visualization for education [3].

Data analytics have improved to the point where all major corporations track an extensive amount of data, whether that be sales metrics, employee performance, or user engagement. Because of this increased capability, organizations have dedicated data specialists to create visualizations to communicate ideas and influence business decisions [4]. However, no matter how impressive the visualization, the audience needs to understand the visual. If the audience doesn't understand what they're looking at and what it's trying to communicate, then it's all for naught. A study on people with low literacy levels has also shown that this impacts their ability to absorb information, leading to increased time collecting data as well as less informed decisions [5].

2.2 Data Visualization Literacy Research

One of the goals of this project was to create an accessible, centralized source of information about DVL. There are plenty of informative and comprehensive research papers that have already been made about DVL, ranging from broader topics to niche areas of study. However, accessing and deciphering these papers requires prerequisite knowledge about data visualization and experiment structure, which are not necessarily areas of study that a student is frequently exposed to. Therefore, we strived to create educational content influenced by aspects of the experiment setups and data visualizations in the research we reviewed.

We first learned about the increasing significance of DVL in a research study named Interactive visualization literacy; The state-of-the-art. The authors used this study to investigate and categorize previous research on DVL that evaluates user comprehension of visualizations [6]. In addition to providing a survey of DVL research, the authors also conduct comparisons between previous studies and expose both established and new directions for future work in the field. In particular, the authors noted that more work is needed to evaluate learning barriers and enhance software solutions that teach DVL. For example, studies that expose a larger and more diverse pool of participants from non-computer science fields to DVL and software that allows users to have influence over featured datasets and visualizations are two suggested approaches to this work. Based on these two findings for future studies in the field, our team gained an understanding that the learning experience of our app needs to be accessible to a well-educated audience that is also familiar with fields outside of computer and data science.

After reviewing the findings of Interactive visualization literacy; The state-of-the-art, our team narrowed the research focus to two papers featured in the survey study that best informed the types of data visualizations and corresponding questions we placed in our app, as well as the structure and presentation of these items. We also drew influence from other papers to inform the content creation process.

We first honed in on the findings from a study of 273 museum visitors [7]. The authors held this study in three US science museums with an aim to gauge adult and youth visitors' knowledge of different kinds of visualizations. The museum-goers observed five out of twenty different visualizations, which accounted for two charts, five maps, eight graphs, and five network layouts. The study asked each participant about their familiarity with a visualization, the visualization name, details about how to interpret it, where the visualization is commonly found, and what kind of data could serve as the source. The authors found that a considerable number of study participants had difficulty with identifying and reading data visualizations. Furthermore, the study noted that the museum participants see these visualizations at school, online, and through different media, and that they overall recognized the features of charts, graphs, and maps rather than aspects of network layouts. The authors also deduced that the participants generally had substantial barriers to naming and comprehending the visualizations that were shown to them. Based on these findings, our team determined that in order for our target users to have a meaningful experience using a DVL platform, we needed to prioritize creating data visualizations that individuals most commonly encounter in media such as various chart types.

After gaining insight into the types of visualizations that are most suitable for a DVL platform, we looked further into our research to understand the construction of a reliable framework for presenting visualizations with corresponding questions and answers. Our team examined a testing framework used in the Visualization Literacy Test Assessment [8]. The Visualization Literacy Test Assessment (VLAT) is a tool geared towards measuring the visualization literacy of non-expert users, and follows a well-founded approach to test development. The assessment contains twelve data visualizations, and fifty-three multiple-choice questions that account for eight data visualization tasks. The twelve included data visualizations are line chart, bar chart, stacked bar chart, 100% stacked bar chart, pie chart, histogram, scatter plot, area chart, stacked area chart, bubble chart, choropleth map, and treemap. The test content in the VLAT is essential according to the input of five subject experts in Information Visualization and Visual Analytics (average content validity ratio = 0.66), and demonstrated high reliability when administered to a sample of 191 participants (reliability coefficient omega = 0.76) [8]. Based on these statistics on the VLAT and additional recommendations from Professor Harrison to look into the research on the assessment, our team determined that the VLAT not only provided a trustworthy foundation for the testing framework for a DVL platform, but also a reasonable collection of available test content. A table illustrating the organization of test items that the VLAT uses accounts for both the framework and testing content. This test item organization consists of a number of fields and their associations; for each data visualization type, there are a series of stems (questions), and each stem is connected to an item ID, visualization task, content validity ratio, item difficulty index, and item discrimination index. Most notably, each stem asks for an answer such that the route to finding that answer is aligned with the associated visualization task. Through the use of these fields and associations to organize the test items, the VLAT contains a variety of questions that are appropriate for the relevant tasks needed to interpret the corresponding data visualizations. Thus, our team concluded from the VLAT that a critical step to successful content creation involves mapping each visualization type to the user tasks necessary for proper interpretation and comprehension.

In addition to utilizing ideas from these papers, we also considered the findings of a study focusing on "cheat sheets" for data visualization [9]. The authors present the idea of a cheat sheet illustrating data visualization techniques, which are visual representations of data patterns that are essential to understanding and engaging with data visualizations. The authors iteratively designed the cheat sheet with the assistance of students and teachers involved in data science and visualization. They also emphasized a key distinction of their cheat sheet from other data visualization resources; while the majority of educational materials about data visualization discuss common aspects of the field such as presentation, design and user interaction tasks, there is an overall shortage of structured, consistent teaching on decoding visualizations and analyzing patterns in the data in a way that is connected to foundational design principles. Although the educational experience of our app does not overlap with the majority of the data visualization techniques demonstrated in the authors' cheat sheet, the study still provided our team with an understanding that standardized, well-organized instructional materials are necessary and crucial to educating audiences about DVL.

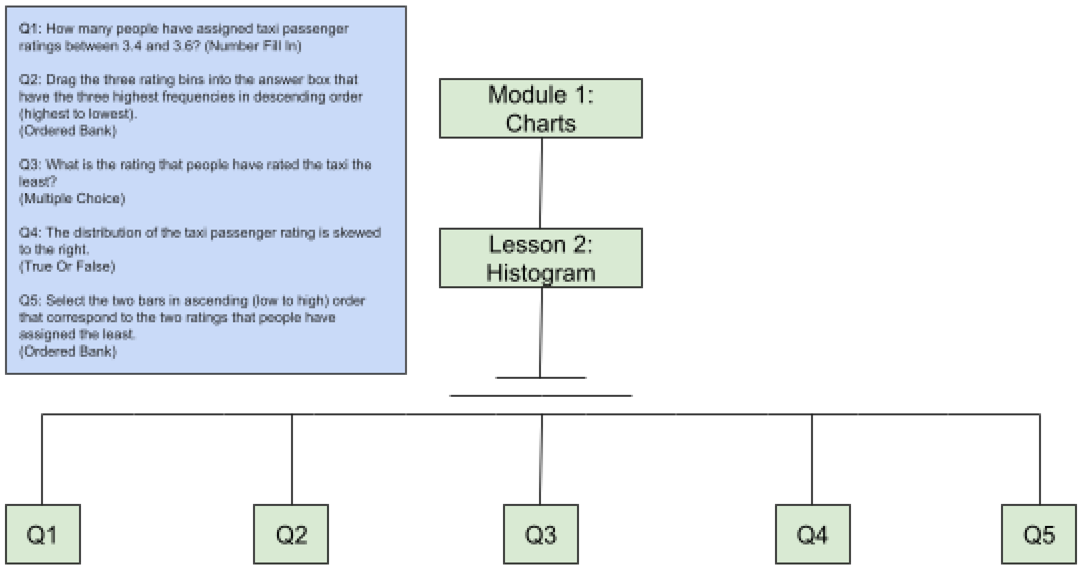
Chapter 3: Methods

3.1 Content Creation

Our team created the content of our app based on key aspects of the test item organization used to structure the papers in our research. In addition, we organized the content using a module-lesson structure to distinguish our app as an educational platform rather than a single assessment. This division of data visualizations has been shown to increase user comprehension, allowing them to focus on each visual piece by piece [10].

We created visualizations, questions, and answers for two modules; the first module is implemented in our app and covers various chart types, and the second module in progress covers the network visualization type. Within the first module, we established a number of lessons where each one is dedicated to a specific chart type. These particular lessons cover the majority of the chart types that are both featured in the VLAT and deemed most relevant to a general student audience. Specifically, bar chart, histogram, line chart, area chart, stacked area chart, scatter plot, bubble chart, and pie chart are the eight total lessons our team included in the first module.

For each lesson in the first module, we borrowed the data types, chart title, and axes titles of the corresponding chart featured in the VLAT to design a modified recreation; we ensured that the data values in our charts are different from the numbers shown in the VLAT charts. Within a lesson, there are five questions, with each pertaining to an instance of the current chart type holding a unique set of data values. The structure of a lesson can be seen in figure 1 below. Furthermore, our team borrowed components of the test item organization in the VLAT to characterize our questions. Each question in our app has an ID containing the current module number, lesson number, and question number, which mirrors the use of item IDs in the VLAT. In addition, every question has one or more task categories, and we only included tasks that are also tested in the VLAT and shown in its test item table. Each category identifies a way that a user should be able to interpret the visualization to provide an answer or approach an answer. Beyond the scope of the VLAT test item organization, we also included an answer type for each question, which is either multiple choice, bank, ordered bank, true or false, or number fill in. We specified one correct answer for each question except for questions that are the bank type, which accept multiple correct orderings of choices.



Q1: How many people have assigned taxi passenger ratings between 3.4 and 3.6? (Number Fill In)

Q2: Drag the three rating bins into the answer box that have the three highest frequencies in descending order (highest to lowest).
(Ordered Bank)

Q3: What is the rating that people have rated the taxi the least?
(Multiple Choice)

Q4: The distribution of the taxi passenger rating is skewed to the right.
(True Or False)

Q5: Select the two bars in ascending (low to high) order that correspond to the two ratings that people have assigned the least.
(Ordered Bank)

Figure 1: A diagram illustrating the layout of Lesson 2 in Module 1. Lesson 2 covers the Histogram chart type, and contains five questions. This shows the breakdown of the module/lesson/question structure of the app. This is how a user would go through the app.

In addition to the planning and creation of modules, lessons, and questions, we also established an efficient approach for generating the data visualizations featured in the app. Initially, we created our first few replications of the VLAT data visualizations for the first module in Microsoft Excel. However, after investigating other tools for creating high-quality data visualizations, we realized that Microsoft Excel is not an ideal software for generating app-caliber charts. Although Microsoft Excel offers a variety of chart types that align with our data visualization needs for the first module, the features available to customize the appearance of charts are quite limited compared to other software tailored towards data visualization creation. We determined that we needed to pursue an alternative software that would allow our team to generate data visualizations briskly while also providing a wide variety of features for designing the charts in the first module. Our team discovered a web-based software called DataWrapper, which enables a user to create and customize data visualizations in four major steps with relative ease. DataWrapper offers all of the chart types that we established in the first module, and allows for visualization configurations to be reused across an unlimited number of datasets. The first step to generating a chart in DataWrapper is uploading the data in the form of rows and columns. The second step then entails defining the axes and labeling each one. The third step is the most involved in the process as it requires the user to select the chart type and apply customization settings; there are a number of options available to refine the visualization, which include providing ranges and tick marks for both x and y axes, toggling grid lines on and off, defining the color, opacity and interpolation of the line(s), providing the line with a label, setting the margin of the line label, filling the area below the line, and changing the height and width of the plot. In addition, our team added an appropriate title and description of the relationship between the horizontal and vertical axes for each of the charts we produced. Once we finished the customization of a given chart, we exported it outside of DataWrapper to be featured in the corresponding module, lesson, and question of our app. An example of a DataWrapper visualization can be seen in figure 2 below. Our use of DataWrapper greatly benefitted the creation of each data visualization; the key features that we needed to design high-quality, properly sized charts are all placed in one of the four steps of the DataWrapper process, which made it easier for us to implement clear design decisions compared to the less organized presentation of Microsoft Excel chart capabilities.

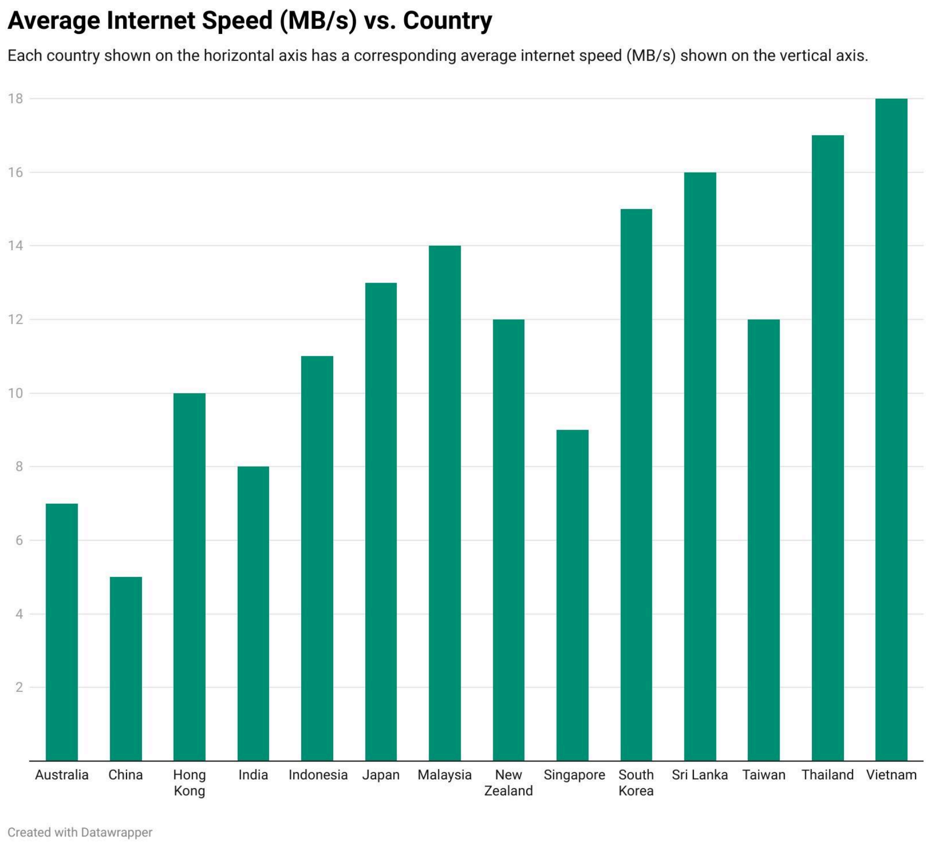


Figure 2: The visualization for module 1 lesson 1, featuring bar charts. This chart is an example of what a user would see with a corresponding question.

3.2 App Design

Our app went through multiple iterations over the course of the year. While we established the goal of "Duolingo for Data Visualization Learning" early on, it took a bit before we fully settled on what we wanted that to look like. One of our early Figma designs established a relatively minimalistic style, with the emphasis being on the educational content in the app. This can be seen in figure 3 below. We also wanted to ensure that the user had some way to track their progress of the modules.

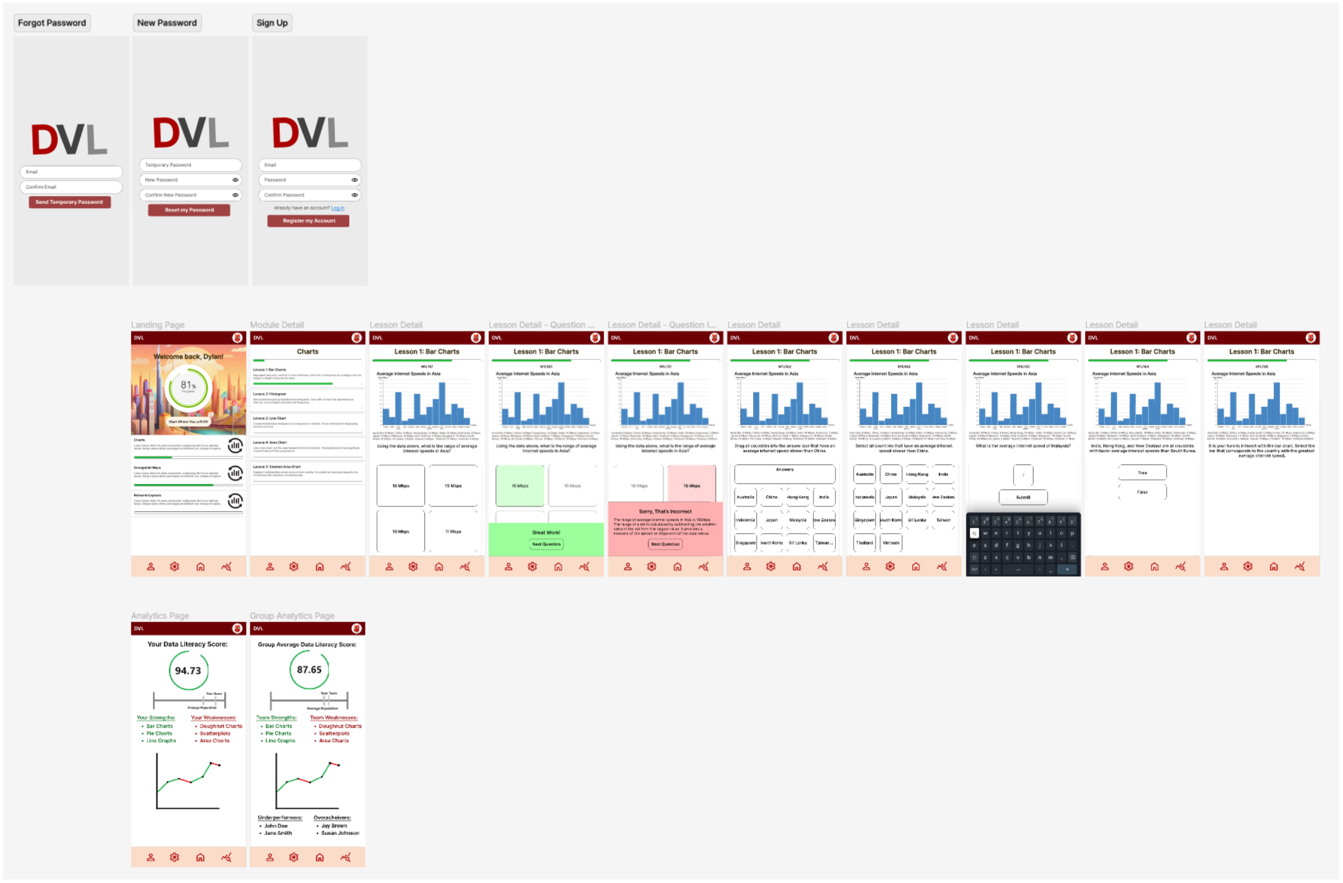


Figure 3: The first full mockup of the potential UI. By contrasting this with the current design, there is a trend towards more simplistic and more spacious screens.

In the development of the application, we chose to prioritize a mobile-first design, numerous decisions were made that are considered sensible within this context. The application is characterized by its responsive design, which is crafted to adapt seamlessly to the varying screen sizes encountered by mobile users. A center-first development strategy has been employed, aiming to position the content centrally on the screen for user convenience.

Positioned at the bottom of the screens, an icon-based navbar facilitates screen selection for easy user navigation. This type of navigation bar is highly effective in mobile applications because it optimizes the user interface for thumb-driven interactions, making it easy for users to reach options with one hand. Each icon in the navbar usually represents a core function or feature of the app, allowing for quick and intuitive access. This setup benefits user experience by reducing cognitive load, as users can recognize icons faster than text. Moreover, maintaining a consistent icon design throughout the app enhances memorability and user familiarity, which can lead to increased engagement and retention rates. Icon-based navbars are also space-efficient, providing more screen real estate for content while keeping essential navigation readily accessible. Research supports that attributes like efficiency, satisfaction, and learnability are critical for the effectiveness of mobile interfaces, which align with the advantages offered by icon-based navbars [11].

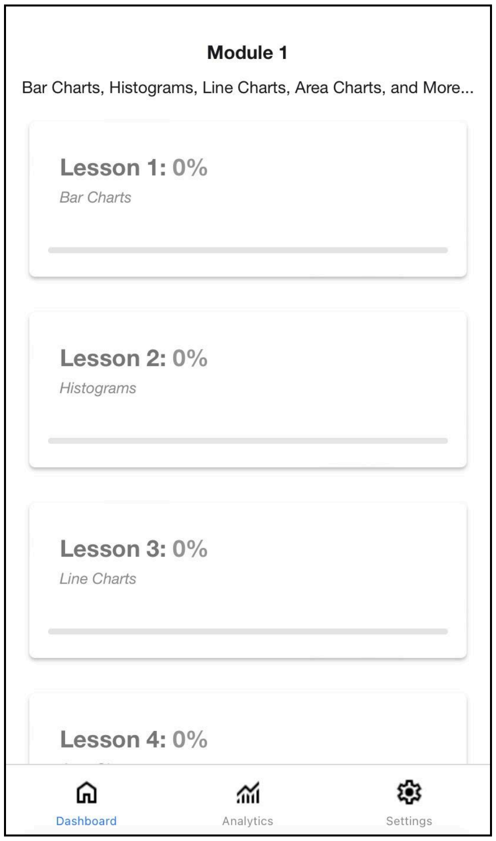


Figure 4: The dashboard implementation and bottom navigation bar. The user is shown information about their completed progress for each lesson in the first module and is given the ability to move to the analytics and settings pages.

On the dashboard, users' progress in their DVL journey is represented through the use of a circular progress chart, adding a visual and interactive element to track advancement. Modules and lessons are presented in clickable components that extend across the width of the screen, providing accessible information for users. These features leverage React's ability to reuse components and create a cohesive, uniform display. The application as a whole extensively incorporates reusable elements, such as buttons and information containers, contributing to a cohesive and efficient design framework.

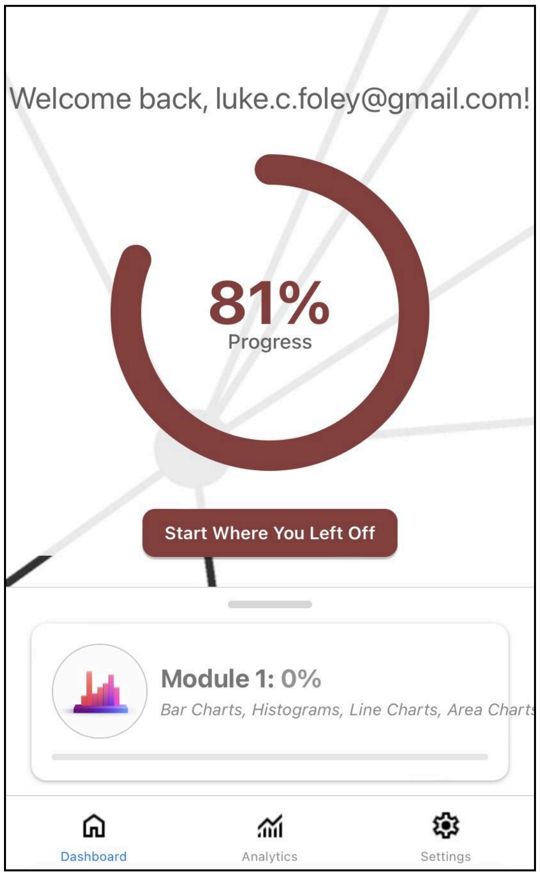

Welcome back, luke.c.foley@gmail.com!
81%
Progress
Start Where You Left Off
Module 1: 0%
Bar Charts, Histograms, Line Charts, Area Charts
Dashboard
Analytics
Settings

Figure 5: The final dashboard implementation. This minimal design draws the users attention to the listed modules where lessons and questions are located. A user can continue working through a module from where they previously left off.

The modules are organized within a draggable subcomponent, allowing for the information to be compactly stored yet easily expanded upon interaction, optimizing screen space utilization. Scholarly research has highlighted that draggable components significantly enhance mobile user interfaces by facilitating dynamic and flexible content management. This flexibility is key in applications that require frequent updates or customization, enhancing both user engagement and personalization of the interface

[12]. Studies emphasize that such interactive features improve usability by allowing users to tailor the interface to their specific needs, thus enhancing their overall satisfaction and efficiency in using the application. Notably, the PACMAD usability model specifically acknowledges the importance of efficiency and personalization in mobile applications, pointing out that these factors significantly impact user satisfaction and the effectiveness of the application [11].

The design of the login and register screens has been intentionally simplified to expedite the user access process, enabling swift entry into their accounts. This approach not only streamlines user experience but also adheres to the principles of clean aesthetic design through the reuse of components. Efficient login pages significantly enhance the user experience by reducing friction and simplifying the process of accessing accounts. Research suggests that streamlined login processes, such as eliminating the need for password confirmation and offering visible password requirements, as implemented in our service, improves conversion rates and user satisfaction by minimizing entry barriers and errors [13]. This leads to a quicker and more satisfying user experience, as users can easily understand and navigate the login process without unnecessary delays or confusion. For instance, allowing users to view passwords as they type can decrease the likelihood of errors and the need for multiple login attempts, enhancing overall efficiency and user satisfaction. We implement password viewing through togglable eye and eye-slash icons.

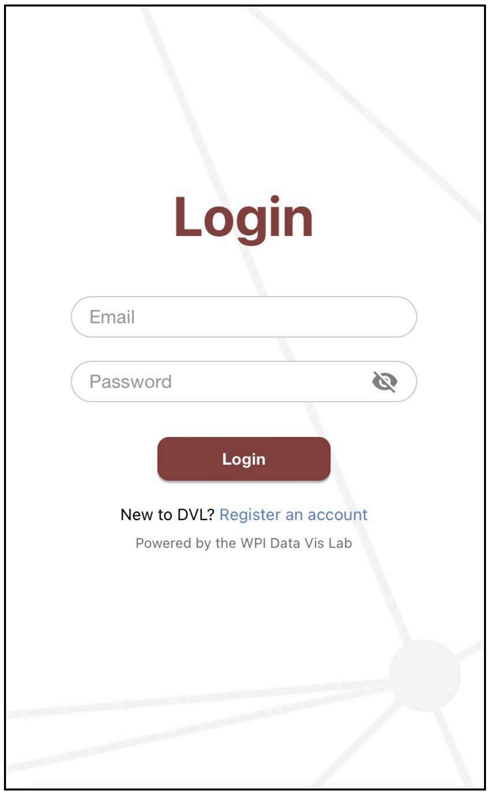

Login
Email
Password
Login
New to DVL? Register an account
Powered by the WPI Data Vis Lab

Figure 6: The login page implementation. This page showcases our consistent visual theme throughout the app, and provides reasonable margins and space between components for an enjoyable user experience.

Additionally, we integrated React Native Paper elements liberally, aligning with the responsive design objectives and contributing to the application's overall aesthetic appeal. Within the modules, Vega-Lite is utilized for creating visualizations, selected for its user-friendly JSON syntax, flexibility, and compatibility with mobile projects. This choice underscores the application's commitment to providing an engaging and accessible user experience, particularly in the visualization of complex data.

3.3 Front-End App Creation

We developed the front-end of our application using Expo, a platform for building Android, iOS, and web apps in React Native, allowing our team to maintain consistency across different operating systems effortlessly. Additionally, we chose to develop using TypeScript over JavaScript to make use of its strong typing system, which simplified code readability and clarity, particularly in a team setting. By enforcing type safety, TypeScript ensured that all of us had a clear understanding of the data structures and expected behaviors of classes.

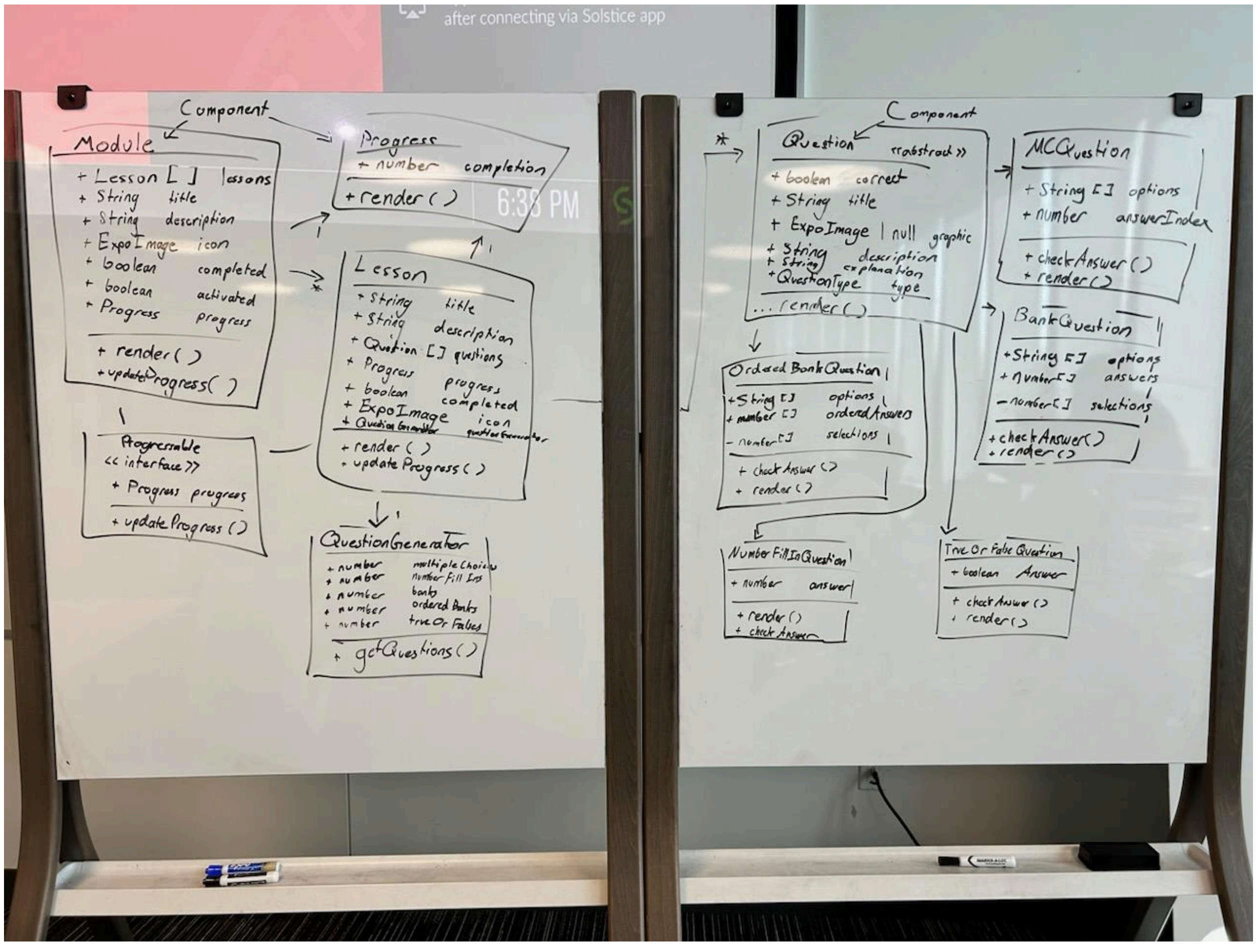


Figure 7: Our initial class diagram layout. From the beginning of the planning phase of our app, we followed a hierarchical and connected design for components, fields, and functions.

The initial phase of our development began with a planning session where we gathered to draw class diagrams on a whiteboard. Spending a few hours collecting our thoughts not only set a strong foundation for our app’s architecture, but also encouraged full-team collaboration from the start. It ensured that everyone on the team was aligned with the project structure and had a clear understanding of the components and their interactions.

In terms of content organization, we settled on a three-dimensional structure for module building; the app has a single set of Modules, each Module contains a single set of Lessons, and each Lesson contains a single set of Questions. This hierarchical organization allows for easy navigation and content delivery, where each Module focuses on a specific set of related visualizations—such as bar charts, pie charts, and scatterplots—and each visualization type is explored in depth in individual lessons.

We implemented a set of navigation buttons that allow users to move seamlessly between screens. For instance, when in a Lesson, users can navigate between Questions using “previous” and “next” buttons, or return to the Module overview with a “back to Module” button. These buttons are strategically placed to be accessible yet minimally invasive, as to not take up precious screen real estate from the question being asked. Future iterations of the app should aim to automate the transition between Questions as they are answered correctly.

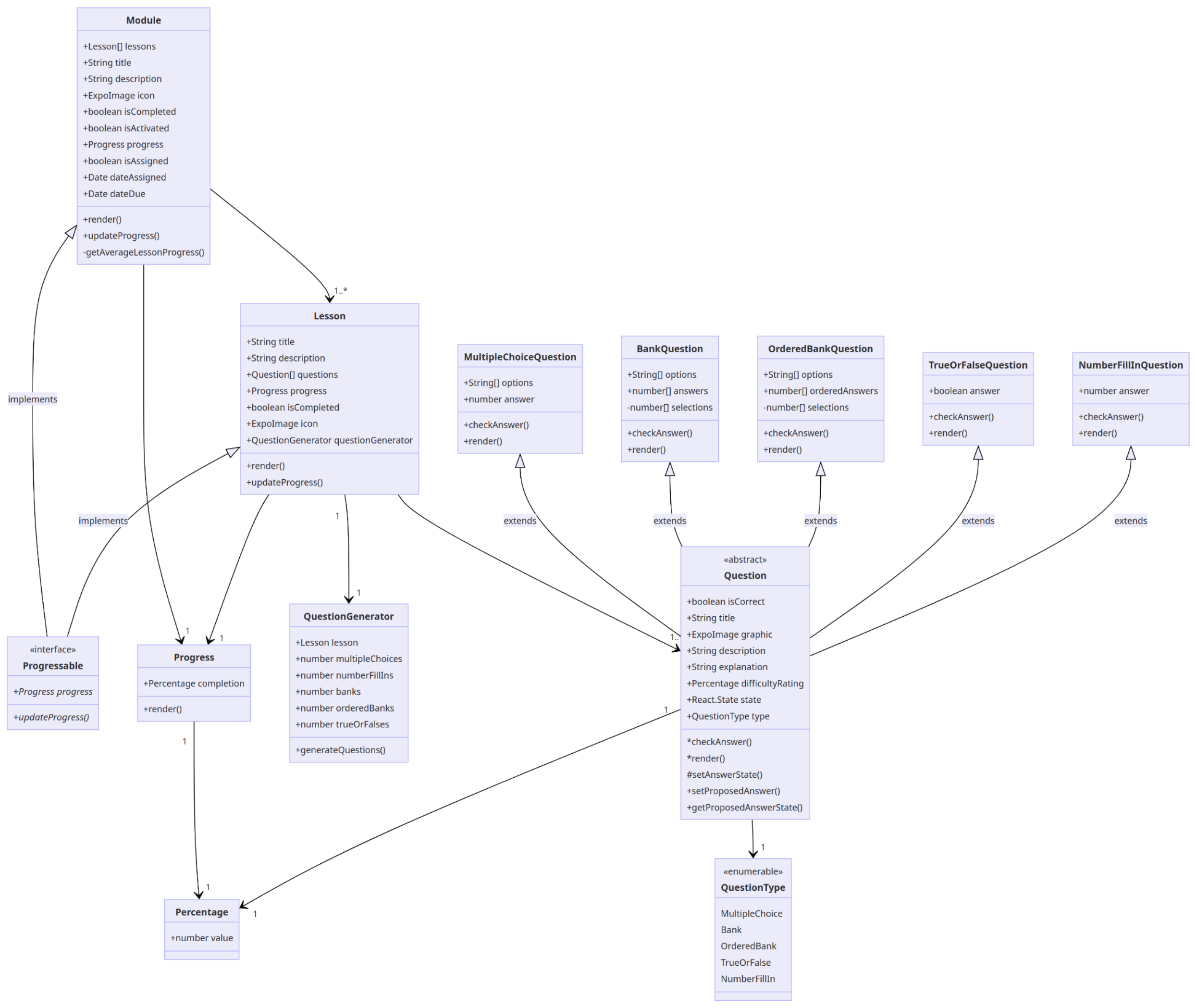


Figure 8: The front-end class diagrams in digital form. This diagram demonstrates the logic we followed to solidify the front-end architecture of our app.

The Dashboard is the main component of our front-end design. It serves as the central hub where users can view and select Modules. Upon selecting a Module, the screen transitions to a list of Lessons within that Module. This Dashboard design allows for intuitive selection of the desired Lesson and allows the user to more easily track their progress.

On the Dashboard's main screen, there is a circular progress bar indicating the user's total progress through the app. There are also straight progress bars on each Module and Lesson, showing the user their overall app completion. Implementing these bars necessitated the creation of a Percentage class, which can only store a value between 0 and 1. Classes that implement IProgressable, ensuring that there is a Percentage stored in each instance, are Modules and Lessons.

The QuestionGenerator class is, as of now, unused. The idea was to allow for procedurally generated content within defined contexts. See section 4.2 to better understand how this would have worked.

3.4 Back-End App Creation

We selected Google Firebase as our backend platform due to its numerous, secure tools that would allow us to focus on the user's experience rather than building our own back-end services from the ground up. While Firebase offers a wide array of features, we only needed to use Firebase Authentication and Cloud Firestore. Firestore's ability to "listen" for data changes and update the user interface in real-time is a valuable feature, ensuring that any database changes are immediately reflected on the front-end without the need of a reload button.

In our current implementation, Firebase primarily handles user authentication on the login screen through a username and password scheme. User credentials are validated by Firebase, and upon successful authentication, the user's record is retrieved and sent to the front-end. Choosing Firebase Authentication allowed us to bypass the complexities associated with developing our own security protocols for user management, saving time and eliminating the need for us to directly handle sensitive user data like password encryption.

Our application's data should be managed in Cloud Firestore under a "users" collection, with documents keyed by user IDs returned from Firebase Authentication after a successful login attempt. This collection allows for management of user-specific data such as app progress, email, login frequency, feedback submissions, location, and other metadata, which stores the user's progress and assists us in analyzing usage patterns. In the future, it would be nice to be able to see who is using our app, where they're using it, when they're using it, and so on.

For hosting static images and the web build, we used an Express.js server. This server is currently hosted on a student-owned virtual machine, however, it will soon be transitioned to a WPI-owned VM to ease the handover to future development teams.

Finally, one reason for choosing Firebase as our backend was its cost-efficiency and scalability. It was vital that this app did not cost us money to develop, but it must be able to support whatever growth it experiences in the future. Firebase offers a tiered pricing model that allows us to start with a free plan and scale up as our user base grows. The "Spark" plan, which is what we currently use, provides sufficient resources for our needs without incurring any costs. This plan includes significant quotas for Cloud Firestore and Firebase Authentication. The Firestore database allows up to 1 GB of stored data and 50,000 document reads, 20,000 document writes, and 20,000 document deletes before additional costs apply. Authentication charges are applied per monthly active user only after 50,000 users.

3.5 App Use Cases

In one common use case example, our DVL app can be used to assist high school students preparing for the ACT science section, which heavily emphasizes the ability to interpret and analyze scientific data presented in various visual formats. The app features a targeted module that focuses on the diverse types of graphs and tables typically encountered in the exam, such as line graphs, bar charts, scatter plots, and data tables.

To enhance students' proficiency in DVL, the app employs a variety of question types including multiple choice, bank, and ordered bank questions, each tailored to a different type of question that the ACT could ask. For instance, multiple choice questions may be used for identifying trends in line graphs, allowing students to select the correct interpretation among several options. This helps each student build their pattern recognition ability, which is a crucial skill under the time constraints of the ACT.

Bank questions, which present multiple correct answers for a single prompt, are particularly useful when dealing with complex scatter plots and tables. These questions encourage students to evaluate multiple aspects of the data simultaneously, enhancing their ability to interpret multifaceted scientific data effectively.

Ordered bank questions require students to sequence events or numerical data correctly, which is particularly useful when dealing with bar charts that compare quantities. This type of question reinforces the student's ability to order data logically, helping them better analyze comparative data visualizations.

By integrating these question types in different visual formats, our app not only aids in developing a deeper understanding of each graph type but also equips students with the agility to switch between different modes of data interpretation. This methodical approach to learning is designed to increase both the speed and accuracy of the students' responses, boosting their confidence and competence in handling the kind of data visualization tasks that are critical to success in the ACT Science section.

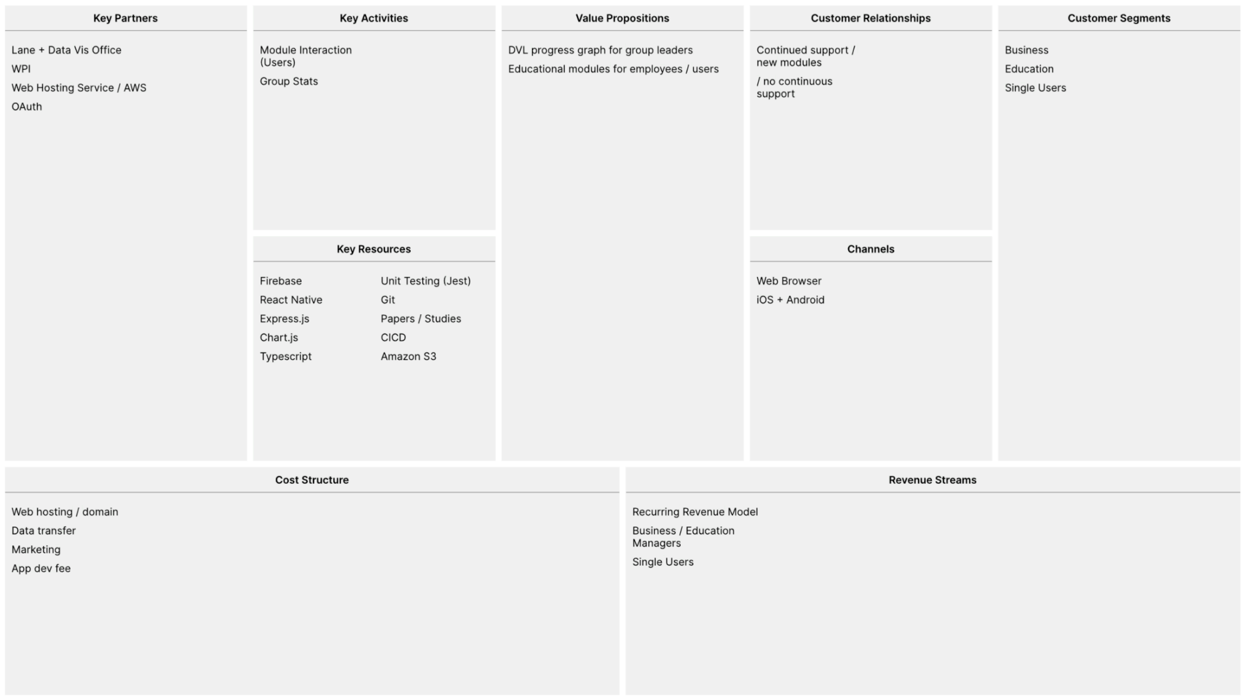
Moreover, the app's structure allows students to integrate these practice sessions into their broader ACT preparation plans. Features like progress tracking and customizable reminders help ensure that students engage with the material regularly, making consistent gains in their DVL. This ongoing practice is key to mastering the skills needed to excel in the exam and ultimately enhances the overall learning experience by making students more adept at interpreting complex scientific information.

As a second use case example, our app can also be used as a tool to reinforce student learning in math and science classes at the high school and college level. In particular, math and science classes that require students to frequently read and analyze data visualizations benefit from assigning exercises that promote the practice of these tasks. For instance, in a STEM class, an instructor may teach about a series of data-driven topics where a student's understanding of the information relies on their interpretation of different data visualizations. Upon completion of teaching these topics, the instructor would be able to promote students' comprehension of the topics through assigning certain modules in our DVL app as homework. The modules and lessons assigned as homework to students would depend on the types of data visualizations discussed in class. An instructor would also be able to specify whether students should complete modules and lessons in a certain order as well. When students work through the questions of assigned modules and lessons on their own, they will be able to improve their ability to interpret information and identify trends in the relevant data visualization types. As a result of completing this work through our app, students will gain a fuller understanding of the broader STEM material that is taught in class. Thus, instructors can assign work in our app as a way to help students comprehend larger topics that are featured in STEM curriculums.

Chapter 4: Discussion and Future Potential

4.1 Cut Content

Initially, we framed a business model to structure our development process. We used the business model canvas to identify the 9 key points in the potential business, as seen in the figure below [14]. This business approach could evolve into a full-fledged enterprise, appealing to corporate sectors and educational institutions. Employers and educators alike would find significant value in a tool that ensures participants not only understand the data visualizations presented in meetings or classrooms, but also engage critically with them to identify trends that may not be immediately apparent. In educational settings, such as high schools and universities, a mobile-friendly DVL learning tool would be effective in teaching students critical thinking skills through data analysis. The expanded business model would include functionalities where users could be grouped by their organization or educational institution, offering detailed statistics on individual and group progress. An administrator portal, likely web-based, would allow educators and employers to view these statistics, manage user groups, and access a leaderboard that highlights top performers and areas needing improvement. This portal could also be integrated into the mobile app as a separate section.



Key Partners
Lane + Data Vis Office
WPI
Web Hosting Service / AWS
OAuth

Key Activities
Module Interaction (Users)
Group Stats

Key Resources

Firebase	Unit Testing (Jest)
React Native	Git
Express.js	Papers / Studies
Chart.js	CICD
Typescript	Amazon S3

Value Propositions
DVL progress graph for group leaders
Educational modules for employees / users

Customer Relationships
Continued support / new modules
/ no continuous support

Channels
Web Browser
iOS + Android

Customer Segments
Business
Education
Single Users

Cost Structure
Web hosting / domain
Data transfer
Marketing
App dev fee

Revenue Streams
Recurring Revenue Model
Business / Education Managers
Single Users

Figure 9: Our business model canvas diagram. We created a high-level business structure to help develop use cases for our app, which then informed our decisions for the app's core functionality.

Dedicated organizational accounts would allow leaders to log into an administrator portal. Here, they could create and manage groups, add or remove members, and assign specific lessons or modules with optional deadlines, catering to the needs of businesses and educational entities that require flexibility in training and assessments. Organizations could set up multiple groups with customized names and structures, enabling them to organize their training programs efficiently and view progress in varied contexts. This level of customization would make the platform adaptable to organizations of all sizes, from small teams to large enterprises.

Lastly, to increase the content's variability and reduce the reliance on manually created questions, we considered implementing algorithmically-generated questions. These would be created using random number generators with predetermined constraints to ensure that data is relevant and appropriate to the question's context. A React Native-friendly charting library—or possibly a custom-developed one—would then dynamically generate visualizations based on this data. This approach would allow for an exponential increase in the number of questions available, while maintaining that data is realistic. The focus of content writers could then, also, shift to authoring contexts for questions rather than hard-coding numbers. By automating question generation, we could provide a much richer, more diverse, and therefore more engaging platform.

4.2 Future Plans

There are some initially planned features that our team did not have time to implement. Additionally, we explored several possibilities for how we could turn our demo application into a full DVL suite.

Firstly, the overview portal for educators and employers will be a milestone feature in expanding our application. This portal will feature a comprehensive dashboard that displays graphs of individual and group performance, broken down by modules and lessons. Organizations could view completion rates, average scores, time spent per lesson, and improvement metrics over time. Such data would allow group administrators to identify areas where students or employees are excelling or struggling, as well as how users' scores compare to one another.

Of the intended features that were not fully implemented, the one of the more notable is saving user progress. A future team could implement persistent tracking of user progress using the already existing, global, React Context that is available to all components in the app. The context would update each time the user answers a question, and this updated information would be sent to and stored in the Firestore database under the user's record. Additionally, changes in the database can be "listened" to and reflected back in the Reach Context, updating the UI as changes are detected on the back-end. To track progress, one would utilize the already existing ModuleAddress, LessonAddress, and QuestionAddress. These classes denote the specific location of each content within the curriculum, such as "M1L3Q5" for Module 1, Lesson 3, Question 5, and just "M1" for Module 1. The current progress indicator is a circular progress bar which displays a placeholder value— a demo. Each address class has a static method for parsing an address string, so implementing loading of progress data into this component will be easy; one could pull a list of QuestionAddresses that have been answered correctly, and working backwards will allow a developer to reconstruct their path through the curriculum.

To make the learning process more engaging, we experimented with various methods for answer selection that go beyond the monotonous multiple choice, true or false, number fill in, and answer bank format. One approach under consideration was allowing users to interact directly with the visualization to submit an answer. We imagined scenarios where a user would tap on the correct bar or point. We also thought of interactive exercises where users might drag elements of a chart into place, or move points to align with a trend. Maybe the user could draw a shape around points that form a cluster. These methods for answering would have been great to explore in greater depth, however, several challenges prevented the implementation of these features within our initial time frame. The primary barrier was the technical complexity involved in adapting our current architecture to support these interactions, especially since our graphs are just static images. Since our app was originally designed around basic button presses, transitioning to a more complex system would require significant changes. Additionally, we made a strategic decision to focus on teaching users how to interpret visualizations rather than creating them from scratch. This decision was based on our distinction between data visualization (the creation of visual data representations) and DVL, which is more about reading and understanding these representations. While teaching users to draw charts could improve their overall data visualization skills, it falls outside the primary scope of DVL, which was our focus.

At present, we only have one module: a basic charts compendium. To allow for more content to be added in the future and appeal to more specific educational applications, we imagine a collection of hyper-specific modules tailored to distinct fields such as chemistry, biology, statistical analysis, geology, or any other discipline with unique ways of visualizing data. These modules could focus on visualizations that are most relevant and frequently used within these disciplines, providing users with context-specific examples that improve their ability to understand and analyze data within their specialized areas. By adding these modules, we would greatly expand our user base by connecting with different academic and professional fields, thus helping a more diverse set of users develop a deeper, practical understanding of data visualization in their specific areas of interest or work.

Recognizing a major limitation of mobile devices in displaying visual data, the screen size, we recommend the development of a magnification feature. This tool would allow users to enlarge specific sections of a visualization without having to rely on their browser's zoom functionality, which would not be available in native app versions. Modeled on interactive magnification tools like that of Amazon's product pages, the magnifier should enable users to zoom in on details without their finger obstructing the focused area. It's important that this magnifier be enabled via a tastefully placed button, making it clear to the user that the developers understand the challenges associated with reading data visualizations on a small screen, but not forcing them to use it.

Chapter 5: Conclusion

In this project, we successfully created a platform for teaching Data Visualization Literacy (DVL), targeted towards an audience of varying experience levels, creating an effective learning experience and allowing users to increase their understanding of data visualizations. By reviewing existing research and identifying knowledge gaps among our potential user base, we authored many modules of educational content that use common visualization types, as well as a structured method of creating content that effectively increases one's DVL.

By referencing some of the most powerful papers in the field of data visualization, specifically on networks and varying chart types, we ensured that our users learned from some of the most commonly encountered visualization types, as well as the most important, novel graphs emerging in the field. Through compartmentalizing our app into a series of modules and lessons, and incorporating techniques from the Visualization Literacy Assessment Test (VLAT), our app provides an encompassing and impactful user experience. The architecture of our application allows users to progressively build upon their current knowledge of data visualizations, as well as track their progress.

The finalized design of the app implements a responsive, user friendly design that uses mobile-first styling and progress tracking visualizations to engage the user in an easy-to-use and aesthetically pleasing manner. Through many rounds of iteration and feedback, we polished the functionality and enhanced the engagement and educational value for the user. Building off of our application in the future, there are possibilities to include AI-generation in the lesson building process to create variability in the user experience, and there are avenues to track and save user progress. This application has the potential to be extended beyond individual users, but to corporations as well, targeting the increase in DVL in a group of workers, monitoring each employee's progress and recommending individualized areas of growth.

References

[1] Morrison, Briana & Disalvo, Betsy. (2014). Khan Academy gamifies computer science. SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education. 39-44. 10.1145/2538862.2538946.

[2] Chevalier F, Riche NH, Alper B et al. Observations and reflections on visualization literacy in elementary school. IEEE Computer Graphics and Applications 2018; 38(3): 21– 29.

[3] Firat, Elif E. and Robert S. Laramee. "Towards a Survey of Interactive Visualization for Education." Computer Graphics and Visual Computing (2018).

[4] Arruda, Darlan, and Nazim H. Madhavji. "The Role of Big Data Analytics in Corporate Decision-making." DATA. 2017.

[5] Kodagoda, Neesha & Wong, B.L. & Rooney, Chris & Khan, Nawaz. (2012). Interactive visualization for low literacy users: From lessons learnt to design. Conference on Human Factors in Computing Systems - Proceedings. 10.1145/2207676.2208565.

[6] Firat, E. E., Joshi, A., & Laramee, R. S. (2022). Interactive visualization literacy: The state-of-the-art. *Information Visualization*, *21*(3), 285–310. https://doi.org/10.1177/14738716221081831

[7] Börner, Katy, et al. "Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors." *Information Visualization*, vol. 15, no. 3, 6 Aug. 2015, pp. 198–213, https://doi.org/10.1177/1473871615594652.

[8] Lee, Sukwon, et al. "vlat: Development of a visualization literacy assessment test." *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, Jan. 2017, pp. 551–560, https://doi.org/10.1109/tvcg.2016.2598920.

[9] Wang, Z., Sundin, L., Murray-Rust, D., & Bach, B. (2020). Cheat sheets for data visualization techniques. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3313831.3376271

[10] Peppler, K.A., Keune, A., & Han, A. (2021). Cultivating data visualization literacy in museums. *Information and Learning Sciences*.

[11] Harrison, Rachel, Derek Flood, and David Duce. "Usability of mobile applications: literature review and rationale for a new usability model." Journal of Interaction Science 1 (2013): 1-16.

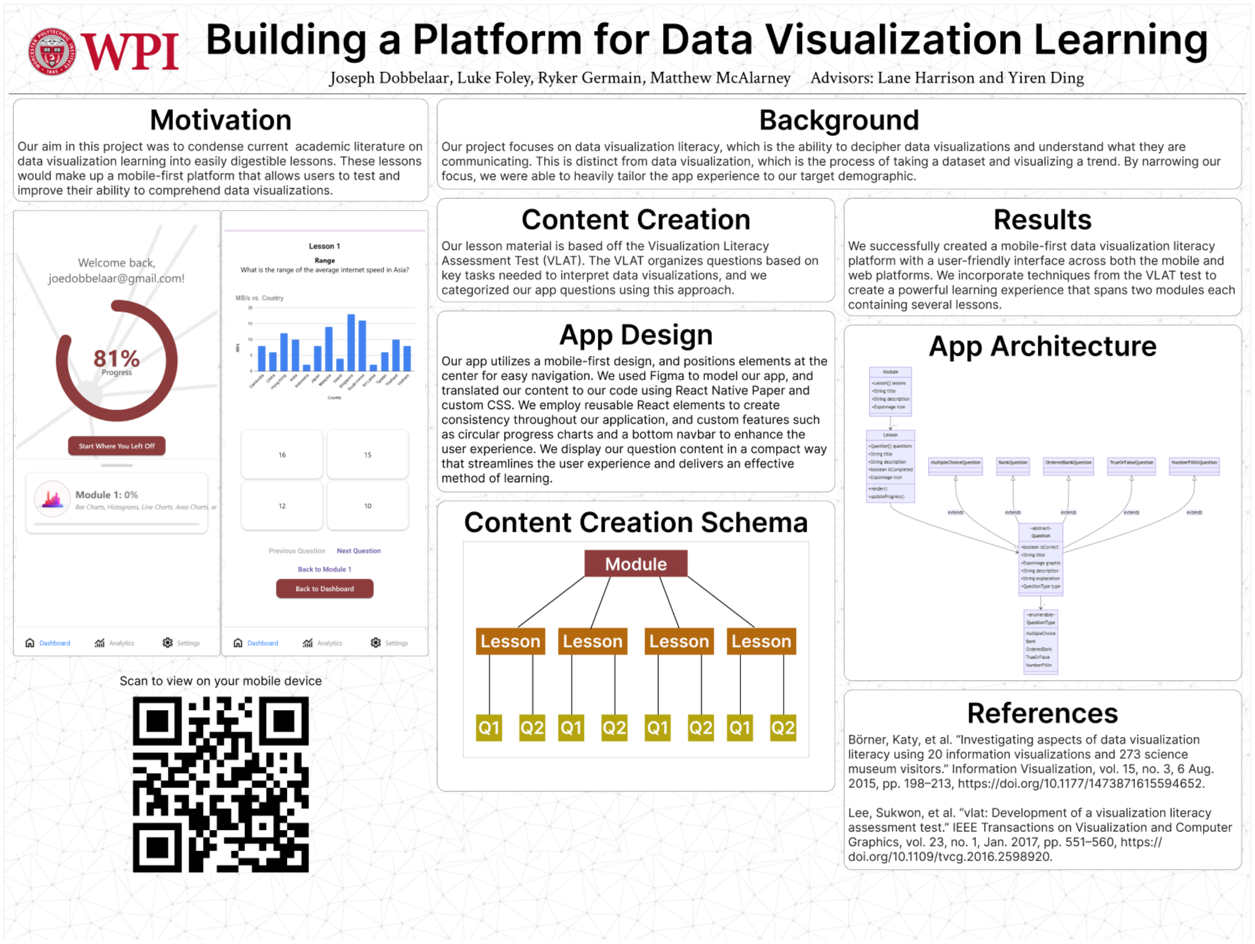
[12] Alshamrani, Mazen, et al. "Usability Studies on Mobile User Interface Design Patterns: A Systematic Literature Review." *Advances in Human-Computer Interaction*, vol. 2017, 2017, Article ID 6787504, www.hindawi.com/journals/ahci/2017/6787504/.

[13] Yang, Jie, and Junhui Gao. "Experimental Study on Seismic Behavior of Recycled Concrete Hollow Column." *Journal of Civil, Architectural & Environmental Engineering*, vol. 9, no. 3, 2017, pp. 245-253, www.scirp.org/journal/paperinformation.aspx?paperid=84810.

[14] Fritscher, Boris, and Pigneur, Yves. "Visualizing Business Model Evolution with the Business Model Canvas: Concept and Tool," 2014 IEEE 16th Conference on Business Informatics, Geneva, Switzerland, 2014, pp. 151-158, doi: 10.1109/CBI.2014.9.

Appendix

A. Project Presentation Poster



WPI **Building a Platform for Data Visualization Learning**

Joseph Dobbelaar, Luke Foley, Ryker Germain, Matthew McAlarney   Advisors: Lane Harrison and Yiren Ding

Motivation

Our aim in this project was to condense current academic literature on data visualization learning into easily digestible lessons. These lessons would make up a mobile-first platform that allows users to test and improve their ability to comprehend data visualizations.

Background

Our project focuses on data visualization literacy, which is the ability to decipher data visualizations and understand what they are communicating. This is distinct from data visualization, which is the process of taking a dataset and visualizing a trend. By narrowing our focus, we were able to heavily tailor the app experience to our target demographic.

Content Creation

Our lesson material is based off the Visualization Literacy Assessment Test (VLAT). The VLAT organizes questions based on key tasks needed to interpret data visualizations, and we categorized our app questions using this approach.

Results

We successfully created a mobile-first data visualization literacy platform with a user-friendly interface across both the mobile and web platforms. We incorporate techniques from the VLAT test to create a powerful learning experience that spans two modules each containing several lessons.

App Design

Our app utilizes a mobile-first design, and positions elements at the center for easy navigation. We used Figma to model our app, and translated our content to our code using React Native Paper and custom CSS. We employ reusable React elements to create consistency throughout our application, and custom features such as circular progress charts and a bottom navbar to enhance the user experience. We display our question content in a compact way that streamlines the user experience and delivers an effective method of learning.

Content Creation Schema

App Architecture

Scan to view on your mobile device

References

Börner, Katy, et al. "Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors." Information Visualization, vol. 15, no. 3, 6 Aug. 2015, pp. 198–213, https://doi.org/10.1177/1473871615594652.

Lee, Sukwon, et al. "vlat: Development of a visualization literacy assessment test." IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 1, Jan. 2017, pp. 551–560, https://doi.org/10.1109/tvcg.2016.2598920.

Figure 10: Our team's project presentation poster for the WPI MQP Presentation Day