NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing and Cyber Systems

Software Design Document

Version 1.1

2/15/2024

Team IQ

Project Sponsor: Erika Konrad

Faculty Mentor: Igor Steinmacher

Team Members:

Logan Samstag (Team Lead)

Nicholas Persley

Kristiana Kirk

Elian Zamora-Rivera

Robin Pace

NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing and Cyber Systems

Table of Contents

Introduction

Courses on college campuses commonly have completing quizzes that do not help them learn the course content. For the product we are creating, we aim to solve this issue with courses that teach English skills in the field of professional journalism.

Our client for this product is a professor at NAU, and her name is Dr. Konrad. She currently uses another product in a similar field, the New York Times CopyEditThis quizzes, to teach her students. The current way that these quizzes are being used is just as a measure of how well the student understands the course materials, with the client having a way to see how the students in the course are doing as a whole. However, these quizzes have a few problems:

- The quizzes that are being offered by the New York Times have many problems, such as being locked behind a paywall.
- The quizzes made by the New York Times are made by an entity outside of the course, and thus may not reflect the course's contents.

However, how will our product, RedPen, help with the problems that our client is having? RedPen will:

- Will be free to all students on NAU's campus that desire to use RedPen.
- Allow for quizzes to be designed by our client so that the contents in the course match the quizzes that are being given much closer.
- Allow for the professor to look through results on a given quiz to see what issues students are having on quizzes and adjust content accordingly.

With these features, RedPen will improve the education of technological writing students greatly, as the client will be able to tailor quizzes to the problems that the students currently have.

During the requirements acquisition phase of this project, our team realized that there are a few key requirements that our team will need to follow during the development of RedPen. First, our team will need to consider some functional requirements, the main requirements being an administrator web page with all of the functionality of the pieces working, a student page that also has all smaller key portions of the web page working, and a quiz taker that has the interaction that is desired by the client. Second, our team will need to consider some non-functional requirements, such as making sure that the website is free for all NAU students as well as making sure that it is easy to use for our target demographic, that being English major students and professors at NAU. Finally, our team will need to consider some environmental constraints such as making the website feel like an NAU website while in use, while also needing to follow certain guidelines during development, such as the ADA guidelines for accessibility.

Implementation Overview

Our solution to the problem mentioned previously, is a custom web application that will resemble quiz functionality from an existing application created by the New York Times called *Copy Edit This!*.

The overall approach going forward begins with the architecture of the software. The chosen general architecture paradigm *Multi-tier architecture*, or more specifically, a *Three-tier architecture*. Where the program is divided into three layers – presentation, logic, and data. This means the user interface/visuals will be the presentation, data deals with the database side of the application, and the logic will be the interface between the two layers.

In regards to information closer to the implementation, the application will make use of a few languages such as

- HTML/CSS - With all existing websites, these provide the ability to design/customize the presentation of the website.
- SQL/MySQL - Provides the ability to store/manipulate/retrieve (SQL) various types of data, also known as a database (MySQL). For this specific website, these will store user, quiz, and question information.
- PHP - The translation from the data stored in a MySQL database to HTML and vice versa, so PHP is the interconnect between these, being able to generate HTML including data read from the database.
- JavaScript - A typical complement to HTML/CSS to make a web page dynamic. When a user interacts with the website, something presentation-wise may need to be updated. Or, using the framework jQuery, we are able to update data by

calling a PHP file from JavaScript without having to reload the whole page (which would typically be the only way to update a specific part of a page).

Architectural Overview

RedPen Educational Platform is made up of several distinct modules where each serves a specific purpose. These modules seamlessly interact, creating a robust framework that delivers the desired application functionality. Through thoughtful planning of these interactions, we've modularized the system, allowing for the efficient management of distinct components. In the diagram presented below, we provide a visual representation of the overarching architecture, illustrating how these modules collaborate to create a user-friendly and effective educational experience.

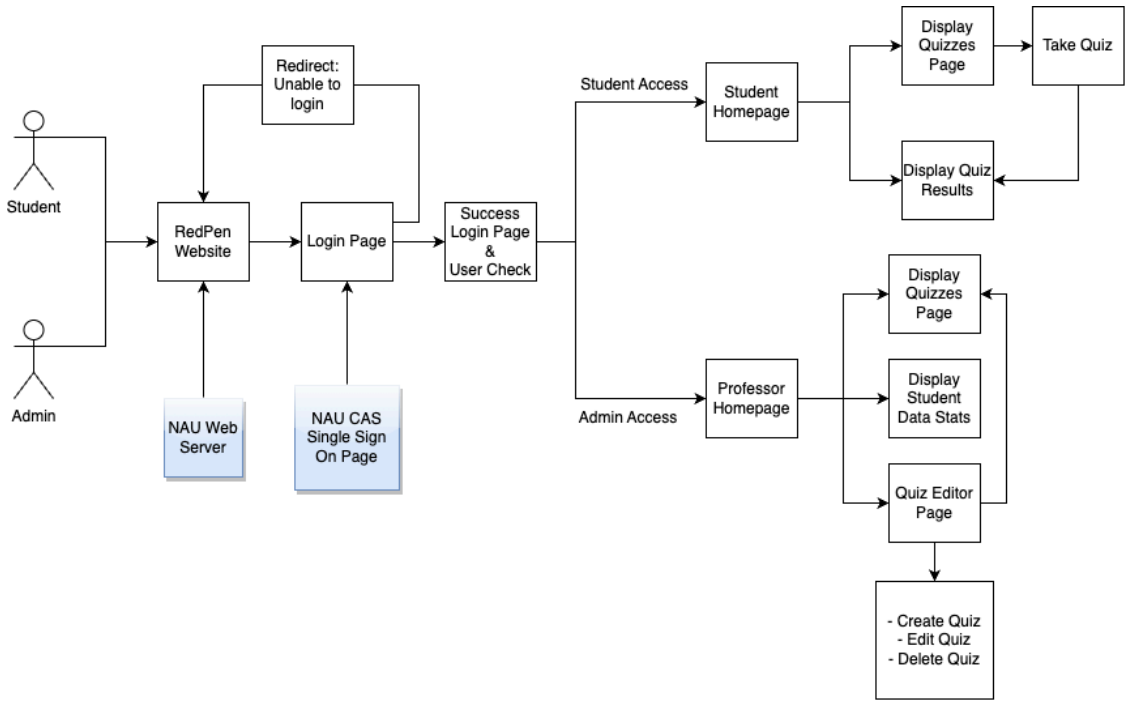


Figure 1: Diagram showing the high level architecture of the web application

As displayed in Figure 1, each module's individual functionality in the system interacts with our chosen frontend framework, HTML/CSS, and the database management system, SQL/MySQL and PHP. The RedPen Educational Platform, hosted by the NAU Web Server, boasts a modular architecture designed to cater to various user roles. The server's key responsibilities include hosting the website, managing web content, and handling user requests. It follows a client-server architecture, with the NAU Web Server serving as the central hub for processing requests and rendering responses. The login system integrates with NAU CAS for Single Sign-On, ensuring a secure authentication process. User credentials are submitted to the Login Page, which communicates with NAU CAS for authentication. Upon success, users are seamlessly redirected. This component aligns with Single Sign-On (SSO) architecture, enhancing security and user experience. In the event of unsuccessful login attempts, the system employs a redirection mechanism, sending users to a specific page while providing relevant error messages. This aligns with common error handling patterns.

Upon successful login, users are directed to the Success Login Page, where additional user checks may occur. The architecture reflects user authentication patterns, confirming successful login based on user credentials. For student access, the Student Homepage serves as a central hub, displaying relevant information and enabling navigation to student-related pages. Additionally, the Display Quizzes Page facilitates quiz interactions, emphasizing frontend-backend communication. Admin access encompasses the Professor Homepage, displaying information for professors and administrators, following MVC principles. The Display Student Data Stats page utilizes data visualization patterns to present statistics in a meaningful way. The Quiz Editor Page allows admins to perform CRUD operations, creating, editing, or deleting quizzes,

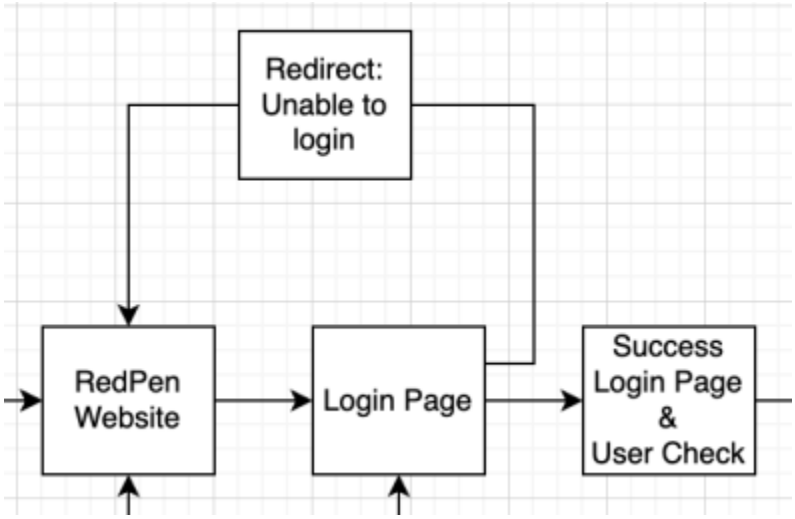
with the updated quiz list displayed. Overall, the RedPen architecture harmoniously integrates various components, ensuring a cohesive and efficient educational experience.

Module and Interface Descriptions

From the diagram above, each module has a written description in this section that details what it generally interfaces with and what its purpose is in the general functionality of RedPen.

RedPen Website Module:

The responsibilities for this component are simply directing the user that arrives at the page to log in. RedPen cannot work without the user being logged in to the NAU CAS system, so not having a page to land on in order to force the user to log in would not be very beneficial to the user experience.



The main service that this module provides is simply forcing the user to log in.

Login Page Module:

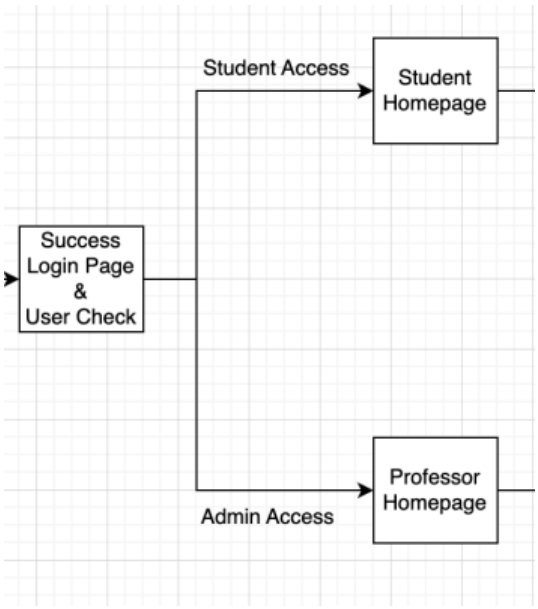
This is not a module that is implemented by our team, it instead is the NAU login page, which we are integrating with as we continue our development. The main responsibility of this module is to allow the user to log in via NAU, which will allow for our team to use their NAU username for certain purposes on RedPen.

As we are not designing this system ourselves, a diagram is not necessary.

The main service that this component provides is, as detailed above, populating some global PHP variables with information that we will need to operate RedPen, such as a username and password. Additionally, the NAU login will allow for us to implement privileges, and they will be much harder to bypass as the NAU login is more secure than any login we could make in a short development time frame.

Success Login Page & User Check Module:

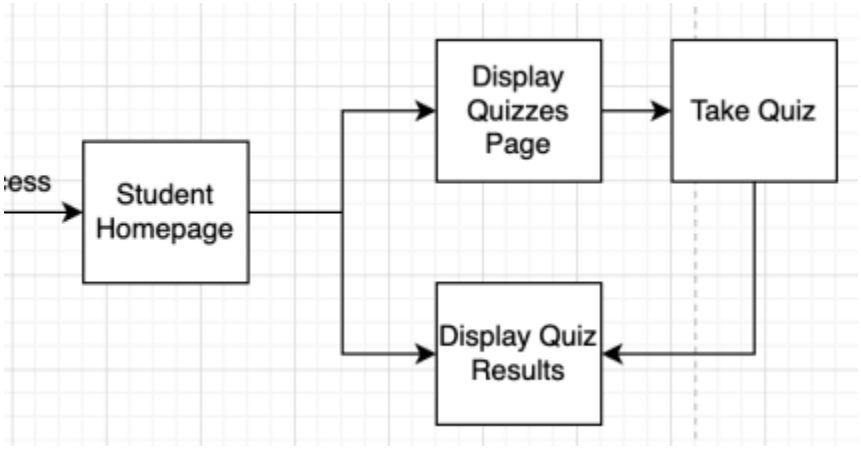
The main functionality that this page will provide is giving a guarantee to the user that they were successfully able to login. Additionally, this module will check that the user is either the professor or not the professor. If a student attempts to access pages that are professor exclusive (i.e. the quiz editor), then the student will be redirected back to the student homepage. This will allow for privileges to be in place, as students cannot access professor content.



The main public features that this module will provide is allowing for every professor page to be checked to see if a student is accessing the page. If a student is accessing the page, they will be redirected to the student homepage.

Student Homepage Module:

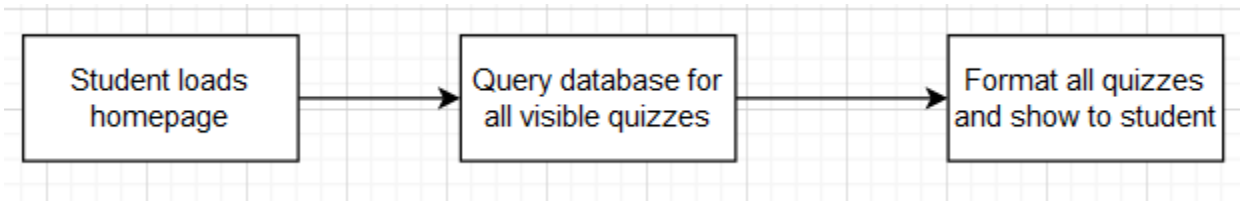
RedPen has two major homepages that need to be developed, with this being the main page for students who use RedPen. On this page, there will be two major sections that the student can access, those being the portion that lists all available quizzes for the student to take, and the other being a means for the student to display how they have done on previous quizzes.



The public features that this module will offer will be simply linking the students to a quiz via the portion of the page that displays the quizzes, or allowing the students to see how they have done on a certain quiz via the display quiz results page.

Display Quizzes Page Module:

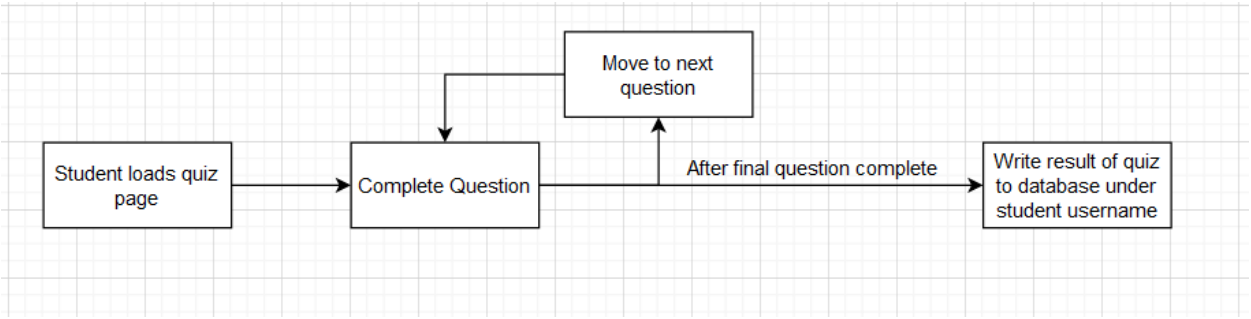
This module is responsible for getting the quizzes from the database, and formatting them for the student to see. All it does is query our database and grab the quizzes that Dr. Konrad has set via RedPen to be visible to the students.



The public interface that this module will provide is simply allowing for students to see quizzes that they are supposed to and allowing for them to go to the take quiz page in order to actually complete the quiz.

Take Quiz Module:

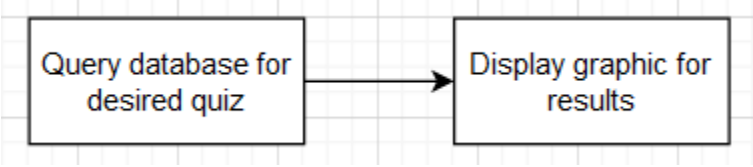
For this module, the main goal that it accomplishes is allowing for the student to take the quiz that they selected on their homepage. After the page is loaded, it allows for the student to progress through the quiz question by question until the end, where the result of the quiz is written into the database.



The public interface that this module provides allows the students' results to be written into the database, which allows for both the student and the professor to view the results any time after the quiz is complete.

Display Quiz Results Module:

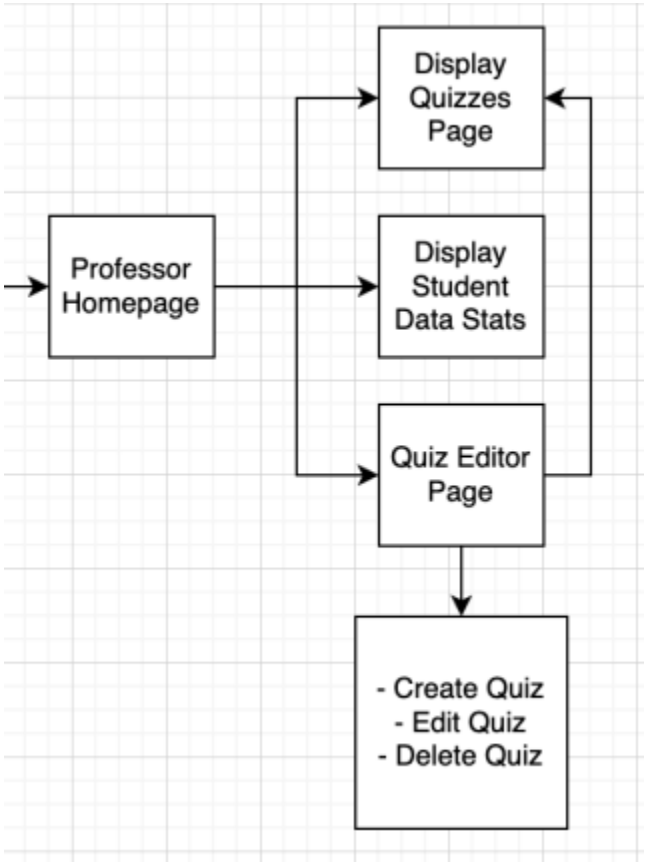
The main purpose of this module is simply to give the students a means of graphically looking at their quiz results after the quiz has been completed. They will load into the webpage, ask to look at the results for a certain quiz, then RedPen will display the results of the student in some graphical form.



There is no major public interface with this device, as it simply is a means of seeing data in a more user friendly way.

Professor Homepage Module:

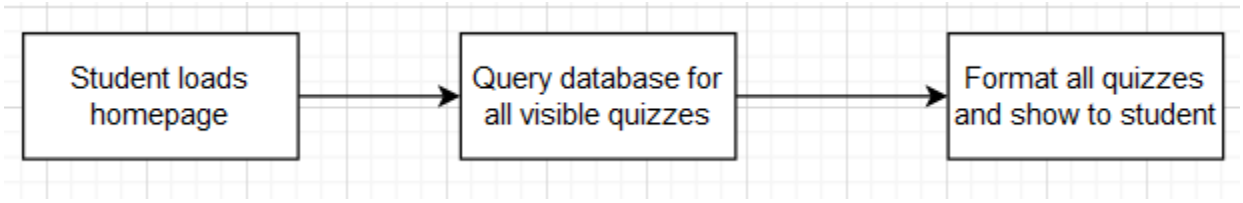
This module is the second of the major homepages that RedPen will have. On this page, the professor will be able to display the quiz results of students, see the quizzes that are available, and create/edit quizzes. All of these will be able to be accessed by the professor upon landing on their homepage for ease of access.



The public feature that this module provides is simply linking to other modules, such as the quiz editor.

Display Quizzes Page Module:

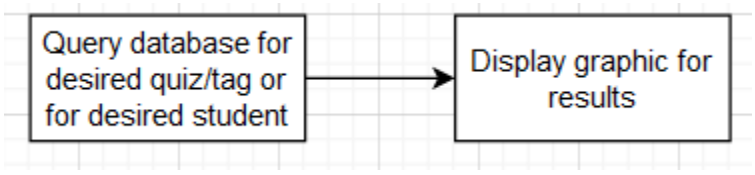
This module is essentially a student view for the professor, and thus they shall be treated as a student while looking at this portion of the page.



The main public interface that this module provides is allowing for the professor to see what the students see to make sure everything is clear on RedPen.

Display Student Data Stats Module:

This module is very similar to the module that displays graphics for the student, except it allows for the professor to additionally search for questions by tag and also search for a specific student to see their attempts on certain quizzes.

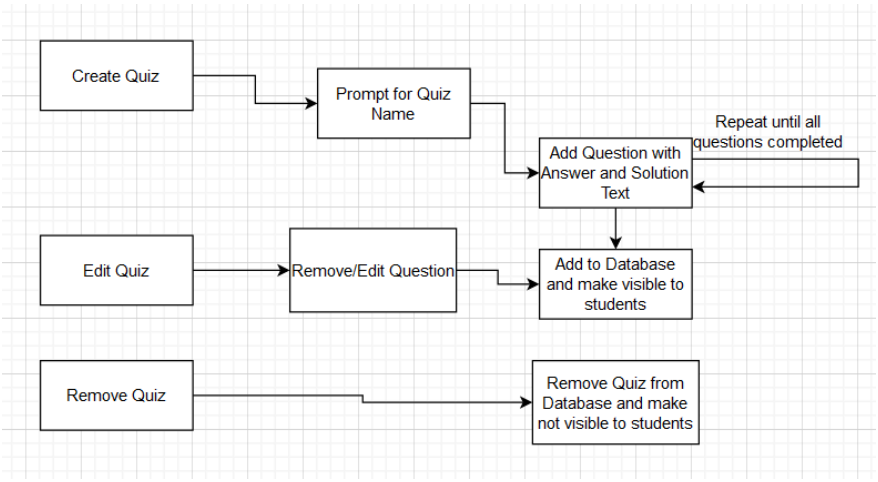


Similar to the student display model, this module provides no public benefit for other modules.

Quiz Editor Page Module:

For this module, there are three major sub-programs that must do a certain task. First, there is the ability to create an entirely new quiz, which requires a quiz name to be given

before creation, and then the ability to add as many questions as desired until the quiz is complete. Afterwards, it needs to be added to the database. Second, there needs to be the ability to edit the quizzes, which simply writes to the database after all edits are complete. Finally, there needs to be the ability to remove a quiz, which simply would remove it from the database, and by extension remove it from the student view.



For the public interface of this module, it will allow for quizzes to populate RedPen, which is necessary for the student Take Quiz module to function, as well as any modules that need a quiz to be taken.

Implementation Plan

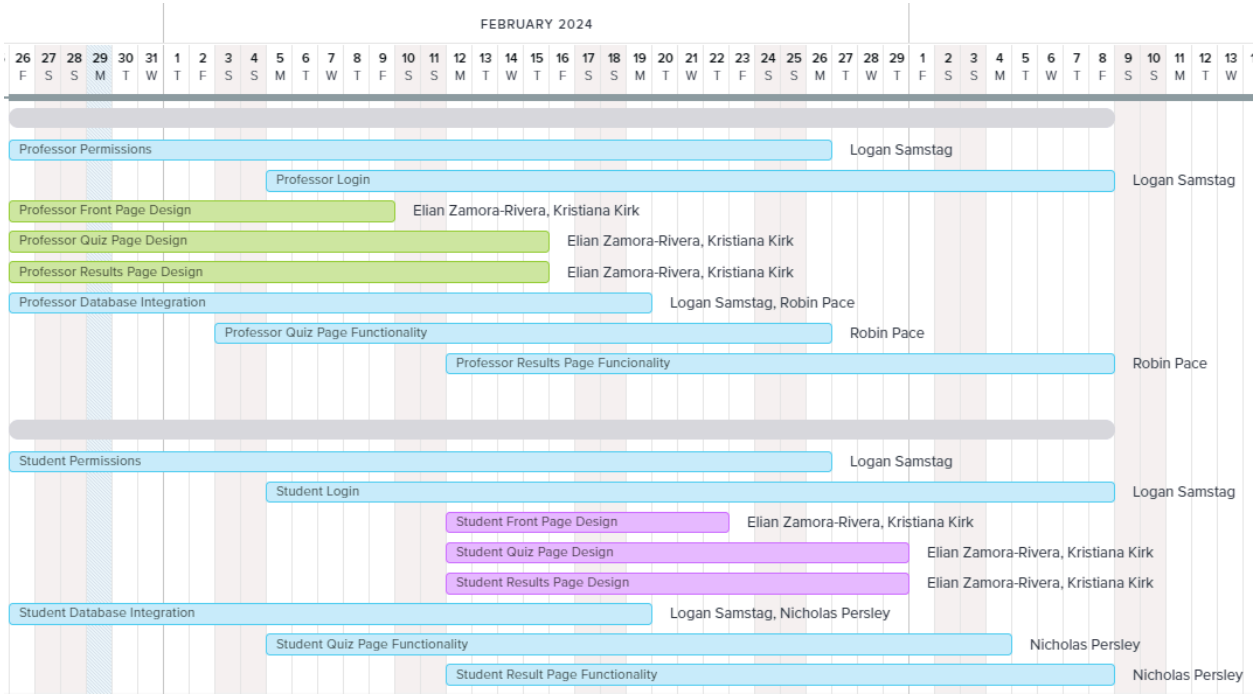


Diagram 1

As seen in diagram 1, TeamIQ's planned schedule for Red Pen is to have the main functionalities done by March 8th. This date was decided so we could be in a good place for the final stretch through April and May, with testing and corrections being handled by the end of March. We will dedicate two weeks to testing, where the team member testing a function will not be the same team member who created it. The professor pages will be handled by the front-end members, Kristiana Kirk and Elian Zamora-Rivera, and functionality will be handled by Robin Pace. This will be the starting focus for the front-end, so it has been projected that the design will be finished first. This was decided because there are more professor functionalities overall, and most design choices that will be made for the professors' end will be easily copied over and adjusted for the student side of things. The student pages will again be handled by the front-end, but the functionality will be handled by Nicholas Persley. While the students also have a

quiz page and a result page, they are designed for each individual student rather than the entire class, so the scope of the student page is a bit smaller and therefore has been assigned less time. The integration between those two systems will be handled by their respective back-end members along with Logan Samstag, who is handling the permissions and log-ins for both the professor and students. In the case that a decision cannot be reached between the two sides of the integration, Logan will be the one to step in and decide what is best for the overall system. The log-in page design has already been implemented, so all that is left is the functionality of the system. This means restricting who can access which system and ensuring that the right people have the right permissions so we can guarantee that unauthorized students have no access to the professors' end of things.

For both the student and professor page designs, the first goal is to get the front page done and designed, followed by the pages they will be accessing. The overlap given in both cases is to account for both early and late finishes, so we can ensure we are ready for any case. As for the functionality, the results pages rely on the quiz pages to be operational, so the quiz pages need to be completed before the result pages can be fully integrated. However, it's not impossible to work on the result pages without the quiz, as we can input fake data, so the major overlap is to again, ensure we're on track and prepared for anything. The quiz pages include creating, editing, and deleting quizzes for the professor, and attempting for the student. The result pages are where the results would be displayed, the professors' results being organized by quiz or question, and showing all students, while the student page tracks their individual progress between attempts of each question.

Conclusion

After diving into the architectural plan of RedPen, it is important to remember exactly what we are making. We are designing a web-based application named RedPen that will help our client, Dr. Konrad, with teaching her courses that cover the concepts of professional journalism. RedPen will allow our client to create custom quizzes that allow for her to ask the questions that she wants to ask to her students for them to learn the concepts that she is teaching in her course. Additionally, RedPen will also allow our client to view statistics on how students are doing in the course to see if certain concepts are being covered properly and if the students are understanding them.

Furthermore, throughout this document, our team has discussed how we plan to implement RedPen, along with a high-level and a lower level architectural overview of our system. Our team will proceed with development following these architectural guidelines to make RedPen as readable as possible for future developers if RedPen ever needs to be developed further. Additionally, we have provided a Gantt Chart that covers our plans for development progress, so that we can measure how close RedPen is to being completed.