NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing and Cyber Systems

Final Project Report

5/6/2024

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

Every college student has to take courses that ask questions that do not test the students knowledge in the best way possible. For example, in an english course, if a student was to be asked what the incorrect word in a sentence is, if the format the question was presented in was multiple choice, either the answers are a small subset of the sentence, which does not let the student assume everything is wrong, or every word is an answer and the answer section becomes incredibly long. This is why we have developed RedPen, which allows for students to take quizzes made by professors in an interactive format.

RedPen has 2 groups of users that will use the platform, those users being Students and Professors. Students are exclusively NAU students, as they need to log in with their NAU ID through the standard NAU login, and they are able to take quizzes on RedPen that are interactive. What this means is that the student is presented with a sentence where every word and every sentence is a clickable button. The student then needs to find the word in the sentence that is incorrect and click on it, and when they do they will

be given an explanation of why that word was incorrect. Additionally, if the student has 3 attempts on a question and does not believe that they are able to understand the question, they are able to hit a "Give Up" button that shows them the explanation without them needing to answer the question. RedPen is not graded based on attempts that a student takes, but simply on completion, and when a student completes a quiz they are able to look at their results and report back to the prof saying that they completed the quiz.

The Professor has many more privileges than the students, as they are able to make quizzes, tags, and see all results. For context, a tag is an item put onto a question that identifies what the question is asking, i.e. “Run On Sentences”. The professor is able to make any number of these tags, and put them onto any number of questions. Additionally, the professor can create any number of quizzes for the students to take so that the professor can make a backlog of quizzes that can be opened at any time. Finally, the professor is able to look at any student's results in a few ways. The professor can query based on a student ID, a quiz name, or a tag. These queries can also be merged together to get a specific result for the professor.

The main issues that our client was having that led us to building RedPen was the quiz format, as detailed above, however, our client liked the New York Times Copy Edit This quizzes. In those quizzes, you are able to click on any word in a sentence to see if it is wrong. However, the main issue she had was the fact that the quizzes were not customizable, and we have fixed this issue with RedPen.

Process Overview

For the development of RedPen, we started by dividing our team into roles based on their past experience. We ended up with 2 members of the team on the front end and 3

on the back end, with each back end team member having a unique role that they fulfilled. One back end member focused on the student side of RedPen, while another focused on the professor side, and the last was for other tasks on RedPen that were either between the two categories or not related to either (i.e. user authentication). Other than those roles, we had specific standards that we needed to follow related to the code that we wrote, and if you wish to see those standards please refer to the Team Standards document.

In terms of how we organized our team, we had a task sheet that detailed all tasks that were outstanding that needed to be done. Then, after a task was made clear to the team, the person who was responsible for that section of RedPen had the job of getting it done. For example, a professor side task would be given to the back end member responsible for dealing with the professor side of RedPen. The only other major piece of technology that we used was a Github repository that we uploaded all of our files to, however, we did not use the issues feature as we felt it was unnecessary with our task sheet.

Requirements

For RedPen, we had a fairly large amount of Functional and Non-functional requirements that we needed to follow.

This is a list of all of our functional requirements:

- Need access privileges: Students need to only be able to access their resources and not the professor's resources, and the professor should be able to access all of their features without needing to worry about any students getting in to interfere with their quizzes.

- Administrator web page to view student quiz results: The professor wishes to view student quiz results so that they can determine what the students are struggling on and change the course material accordingly.
- An Editing Panel to create Quizzes on RedPen: The professor needs some place to create quizzes for the students.
- Data analysis for the professor between different mediums: The professor should be able to view the data of a student either in a bar graph on RedPen, or should be able to download the results and make their own graph.
- Allow the professor to tag quiz items: The professor needs to tag quiz questions so that they can see what concepts the students are struggling on.
- Allow storage of previous semester information and current semester information: The professor should be able to look back onto previous semesters in order to see what students struggled on in the past.
- A student page that shows what they have done wrong in a quiz: The student should be able to see their results in terms of how many attempts they took on a question in order to see what concepts they can improve on.
- Application to support all modern browsers: RedPen should work on all modern browsers so that no student needs to install a new browser.

This is a list of all our non-functional requirements:

- Ease of Use: RedPen should be easy to use for anyone, as the users of RedPen could be older and less familiar with technology.
- Price: RedPen should be free for every NAU student who wishes to access it.

- Easily Maintainable and Manageable: It should be easy for the professor on RedPen to go in and edit quizzes and tags without needing to touch any back-end code.
- Quiz Interaction Page should be instant: The interaction on the quiz page should be instant so that students are not confused and click on a word they already have clicked on.
- Database upload/download should be within 1 second: We wish for any result that the student gets to be written to the database in a short time, and quizzes should also be loaded in a very short time.

Those are all of the requirements that we had for RedPen before beginning. The way that we got these requirements was weekly meetings with our client and discussing what she would like out of RedPen, with mockups for certain pages and features being developed along the way.

Architecture and Implementation

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.

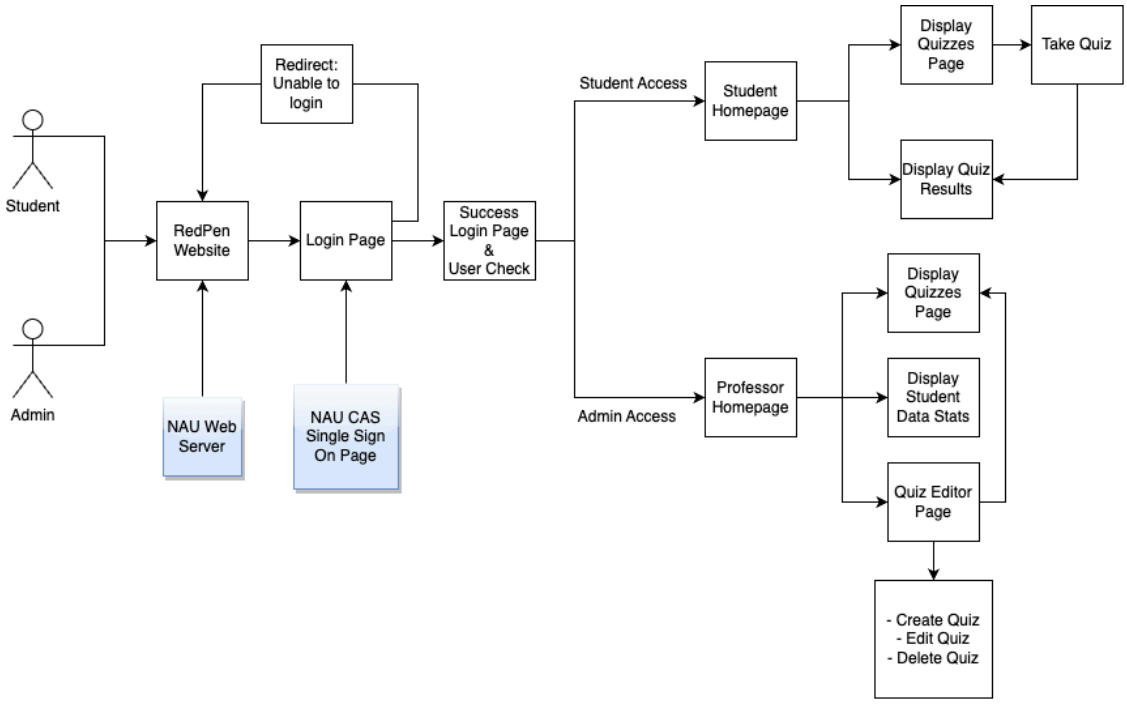


Figure 1: Diagram showing the high level architecture of the web application

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.

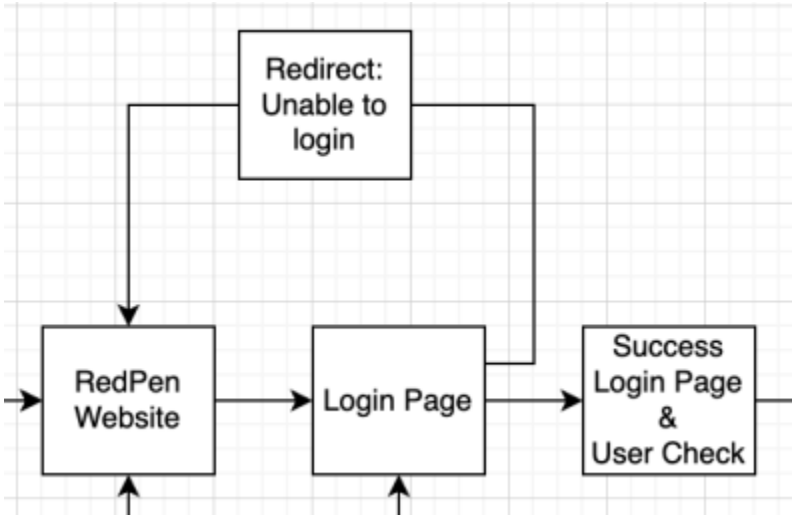
For an example of a user going through our architecture, they would start by logging in at the “Login Page” Module, and assuming they succeed, if they are a student they are directed to their homepage while an admin will be directed to the admin homepage. After this, the user can access any of the modules after their homepage for any of the features on RedPen.

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

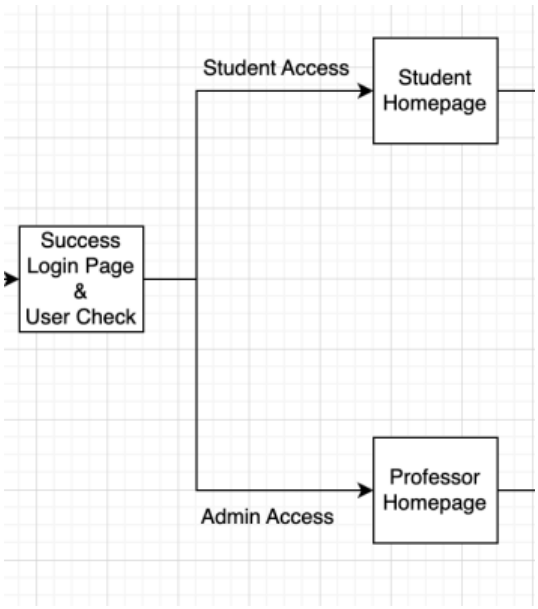
of this module is to allow the user to log in via NAU, which will allow for our team to use their NAU username for certain purposes on RedPen.

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:

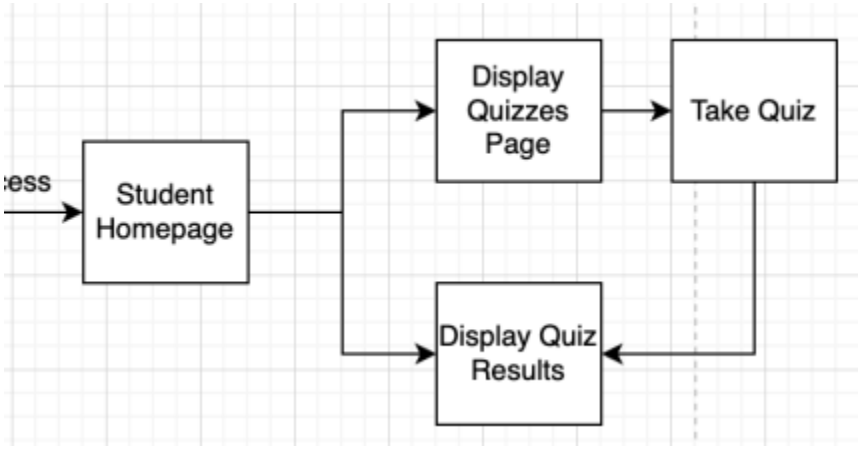
The main functionality that this page will provide is giving a guarantee to the user that they were successfully able to login. Additionally, this module will check that the user is either the professor or not the professor. If a student attempts to access pages that are professor exclusive (i.e. the 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:

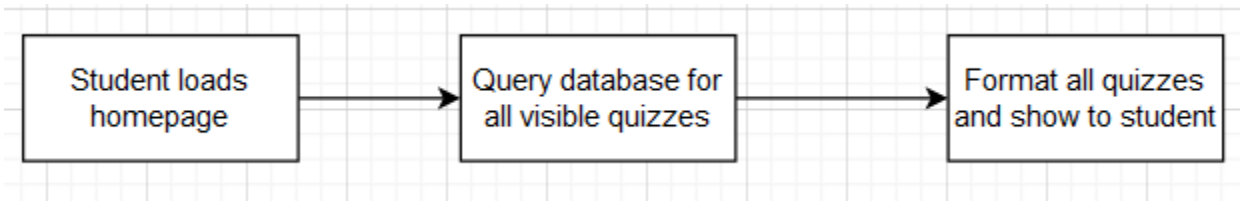
RedPen has two major homepages that need to be developed, with this being the main page for students who use RedPen. On this page, there will be two major sections that the student can access, those being the portion that lists all available quizzes for the student to take, and the other being a means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

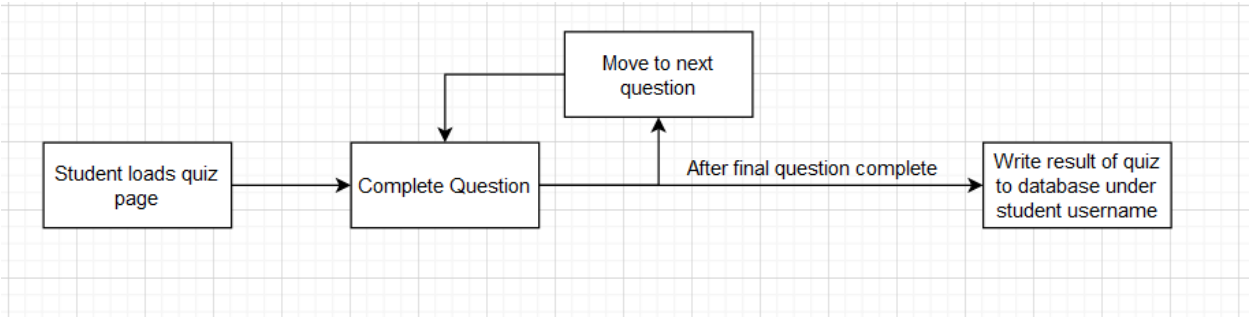
This module is responsible for getting the quizzes from the database, and formatting them for the student to see. All it does is query our database and grab the quizzes that Dr. Konrad has set via RedPen to be visible to the students.



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:

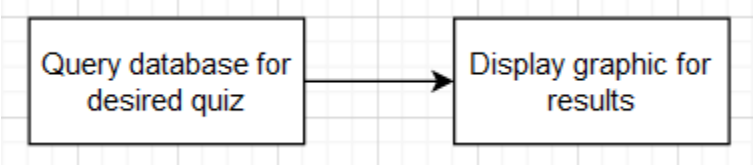
For this module, the main goal that it accomplishes is allowing for the student to take the quiz that they selected on their homepage. After the page is loaded, it allows for the student to progress through the quiz question by question until the end, where the result of the quiz is written into the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

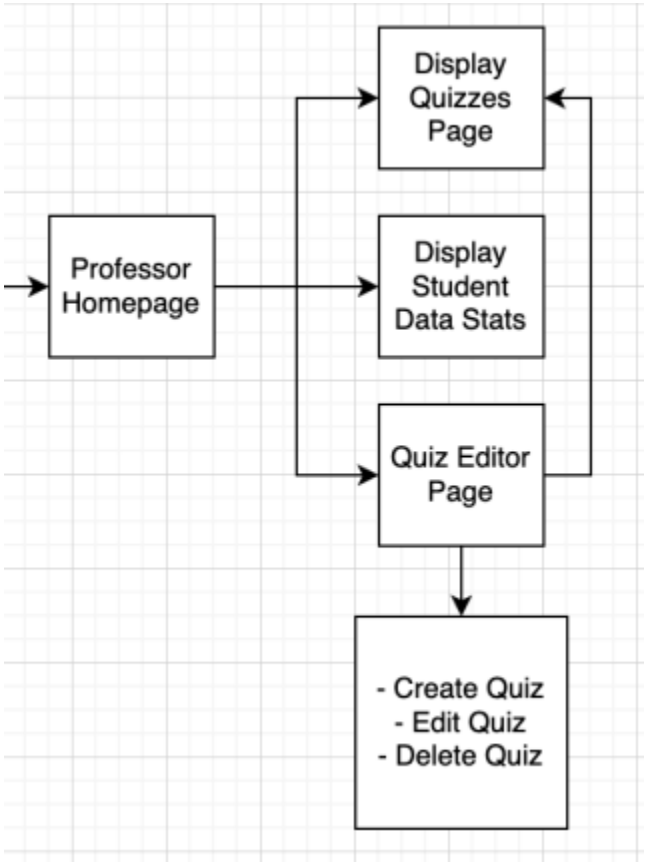
The main purpose of this module is simply to give the students a means of graphically looking at their quiz results after the quiz has been completed. They will load into the webpage, ask to look at the results for a certain quiz, then RedPen will display the results of the student in some graphical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

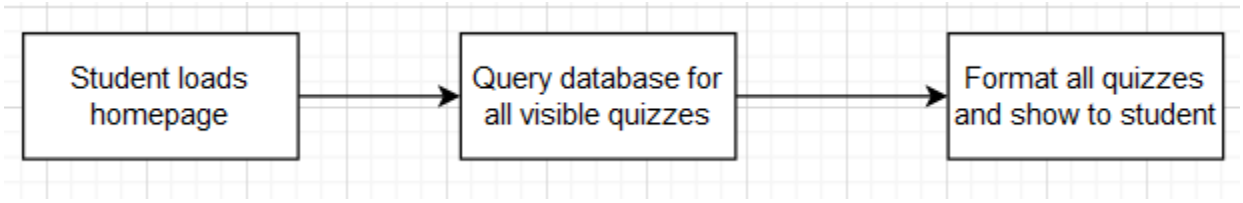
This module is the second of the major homepages that RedPen will have. On this page, the professor will be able to display the quiz results of students, see the quizzes that are available, and create/edit quizzes. All of these will be able to be accessed by the professor upon landing on their homepage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

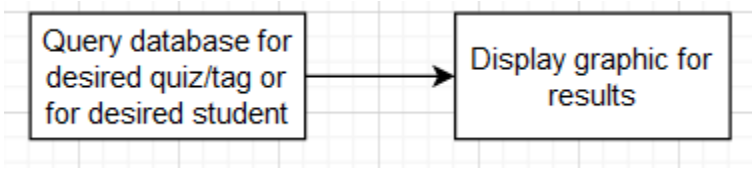
This module is essentially a student view for the professor, and thus they shall be treated as a student while looking at this portion of the page.



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.

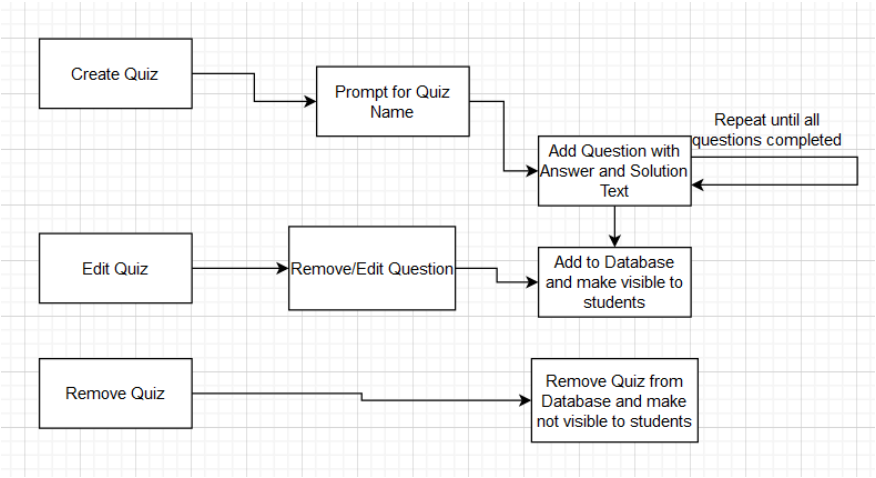


Similar to the student display model, this module provides no public benefit for other modules.

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.

Those are the modules that compose all of RedPen. These were made before we started majorly developing RedPen, however, nothing of note changed during development. All of the major modules stayed the same and no connections between them needed to be changed.

Testing

For the testing of RedPen, we had 3 major categories that we tested, those being Unit, Integration, and Usability. For Unit Testing, the main testing that we did was using

PHPUnit to test any function that returned something that we were able to test, and we used similar logic to test our Javascript with Jest. We ended up not being able to do much Unit Testing, as most of our functions in Javascript and PHP were for writing, reading, or rendering to the screen.

For Integration Testing, the approach that we took was to test from the bottom up, checking that the lowest level system is working and then testing the next level up. Our integration testing started right above where Unit Testing ended, and focused mainly on making sure that our reading and writing operations to the database were successful. We kept testing up and up, fixing bugs as we went, until we knew that each level of RedPen was bug free and works with the other modules.

For Usability Testing, we enlisted the help of those outside of the team in order to make sure that we were not missing anything that we could not see as members of the team developing RedPen. For the professor, we had our client go into RedPen and start messing around with all of the professor's side features, and this allowed for our client to give us feedback for us to implement before the end of the semester. For the student side, we asked our client to allow some students to access RedPen and attempt some quizzes in order to see if all quiz functionality was functional. There were some bugs but we got the bugs ironed out before delivery.

Project Timeline

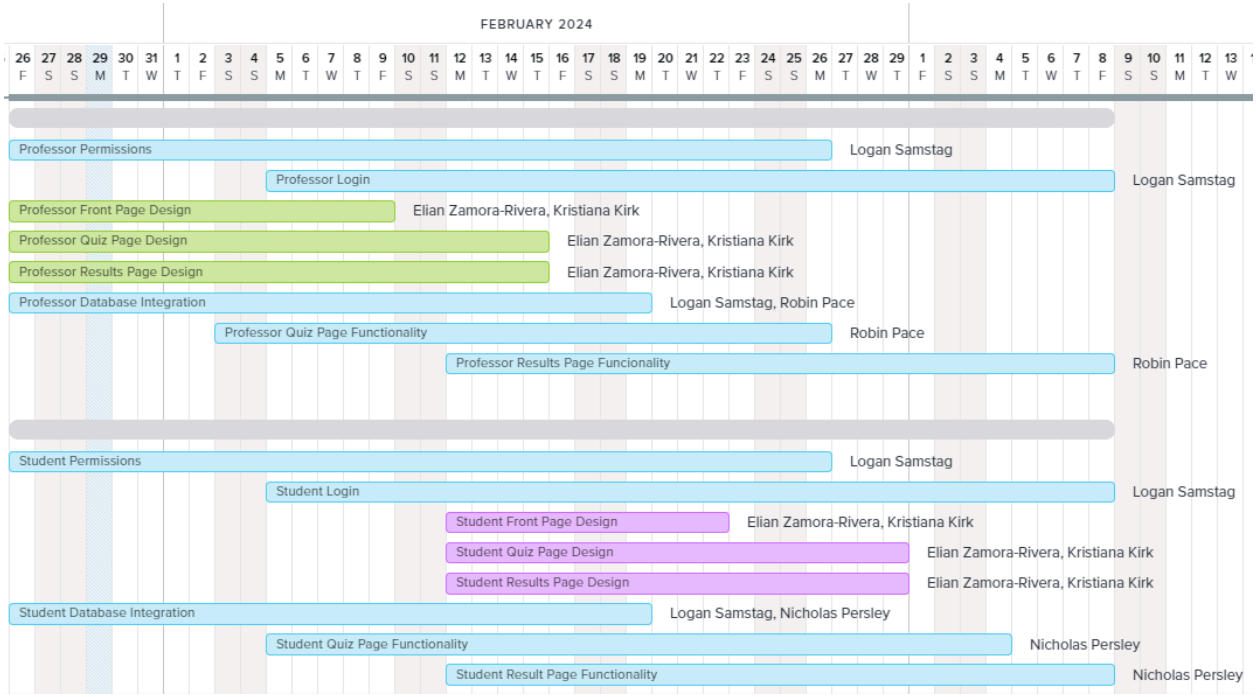


Figure 2: General Functionality Gantt Chart

This was our Gantt chart for the general functionalities of RedPen. Our first goal was to have the student side front end complete, as that was where we had a baseline from CS476 in Fall 2023. After that, we had the front end team go to the professor side to finish that front end development so that all portions of RedPen would have a user facing website complete. While all of the front end development was happening, both the student and professor back end were being developed in parallel by 2 different people. The user authentication was also being developed at this time. After this Gantt chart, we went over to a Gantt chart for testing:

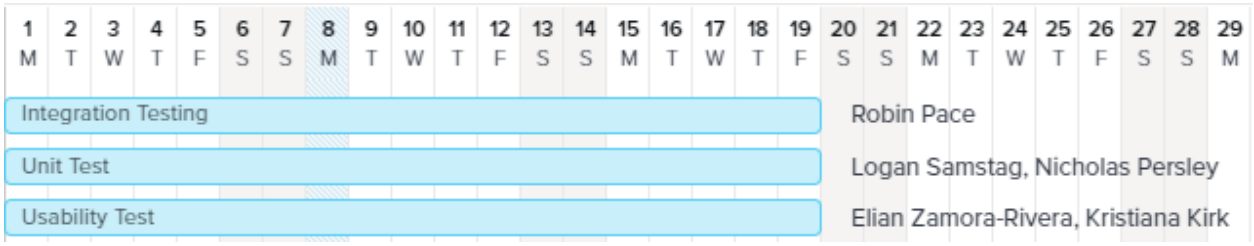


Figure 3: Testing Gantt Chart

This Gannt chart shows what our plan was for testing in terms of Integration, Unit, and Usability. We followed this Gantt chart fairly well, and now we are past all testing and simply are wrapping up the delivery of RedPen.

Future Work

For future work on RedPen, there are a few features that we considered that we did not implement as a result of time constraints. First, we wish to make a way for our client to assign a new admin should she wish to pass RedPen onto another person. This is possible in the back end of RedPen, but not possible for our client to do on RedPen itself. Second, we wish to make some means of messaging the professor on RedPen directly. As of now, our client will have to go through Canvas to get feedback from students about what they did not understand, and we wish to remedy that in the future. Finally, we wish to restrict the student access to RedPen to students that are in the course that Dr. Konrad is teaching. This will filter the results to exclusively be the students in her course.

Conclusion

Therefore, we have developed RedPen in order to help our client with a better means of asking questions to their students. Existing solutions either do not have the best means of asking questions to the students or ask students questions in a desired manner but

are not customizable for our client, i.e. the New York Times Copy Edit This quizzes. RedPen addresses these issues by:

- Using the format from the New York Times Copy Edit This quizzes and allowing our client to fully customize the quizzes on RedPen.
- Allowing any NAU student to log in to RedPen for free
- Giving our client the ability to see student results very easily.

This will help our client as she will not need to compromise on her quizzes anymore, and will be able to make the ideal quiz for the content that she is teaching. RedPen will not have any wide impact as it was designed specifically for a small group of students at NAU. Our team is grateful for the time we have spent on RedPen, as it has taught us a lot about how to design a product, including how to go from idea to an actual product.

Glossary

We do not have any specific terminology for RedPen.

Appendix A: Development Environment and Toolchain

- **Hardware:** Our team developed on Windows and Mac, as we developed in order to be deployed on a web-based environment. All of our computer specifications were very different, and any computer that is able to run an IDE will be able to be used to develop on RedPen.
- **Toolchain**: The only main tools that we used during the development of RedPen were VSCode as an IDE, Github in order to store files, and some tool to ssh into the file server. VSCode we used as it is the general IDE that was preferred by the team and made the code a lot easier to read, Github was used as the team was familiar with it, and we did not have a general tool for ssh as each team member

had a different computer and a different tool preference. Any tool that lets you ssh should work.

- **Setup**: Other than installing the above parts of the toolchain, the main issue is connecting to the file server that houses the files for RedPen. We are going to leave that to the reader as the means of connecting might change over the years, but the goal is to simply connect to the file server that has the files for RedPen.
- **Production Cycle**: For RedPen, we did not have the luxury of a test server, so we developed directly on the live website. So, open the file explorer and navigate to a file that you wish to edit. Edit the file, save, and refresh RedPen to see any changes that have been made. If they are visual changes, remember to hard refresh so that they become visible.