Final Report: Java “Online” Quiz System

Margaret Nixon, Jordan Garnett, Xuchu Ma, Myles St. Pierre

2020-12-14

GitHub: <https://github.com/MaybeeLater/CS2042-Final-Project-MN-JG-XM-MP>

**1 Team Profile**

**1.1 Margaret Nixon**

We all have close to the same amount of time programming, but I believe I have the most experience working with JavaFX. Working with interfaces is one of my favourite things to do when programming, and so I would say it is one of my strengths. My role in this project was decided to be a front-end programmer.

**1.2 Jordan Garnett**

My role in this project was to contribute to the back-end section of the project. I believe I am also somewhat decent with JavaFX, but that doesn’t mean I like it. I’d say my best strength that helps with this project is being able to work out all the odd little problems that come up when working with Java, which in the end ties a good amount of the project together.

**1.3 Xuchu Ma**

I believe my strength is in back-end logic programming and testing. I contributed a lot to both the back-end of our program, I also did most of the testing for this project, by carefully testing each version of the program.

**1.4 Myles St. Pierre**

I was also to contribute to the back end of this project, and to help with testing. I do not have any preference and was put where I was needed.

**2 Overview of Project**

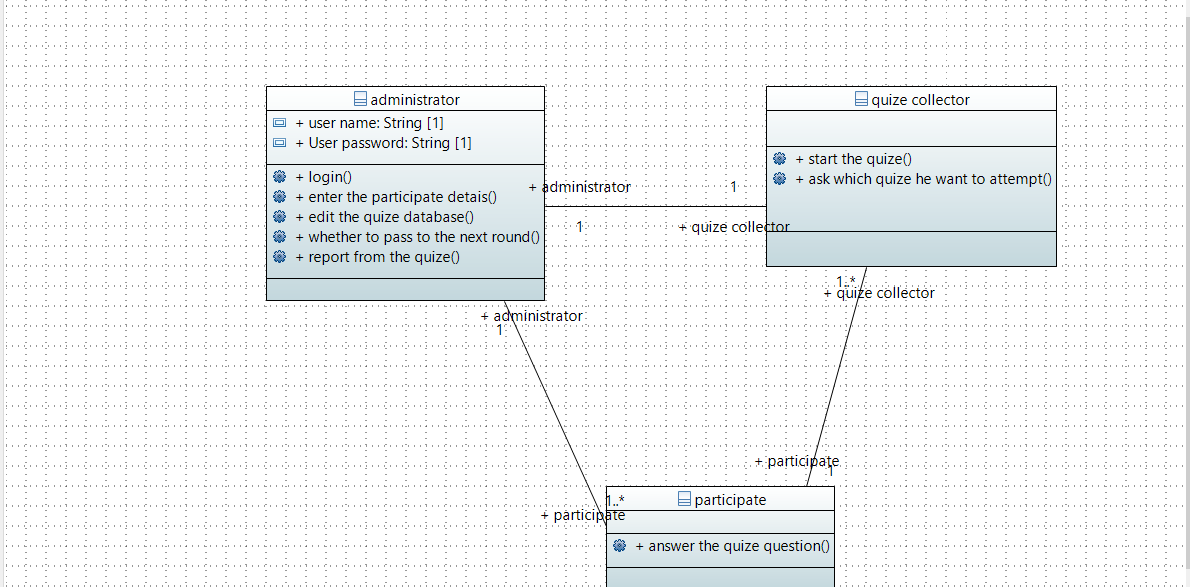
This program performs a local version of a quiz system. As this is a local version, all windows for both admin and user occur from one system, if this were to instead be implemented online as originally intended, this would not be the case.

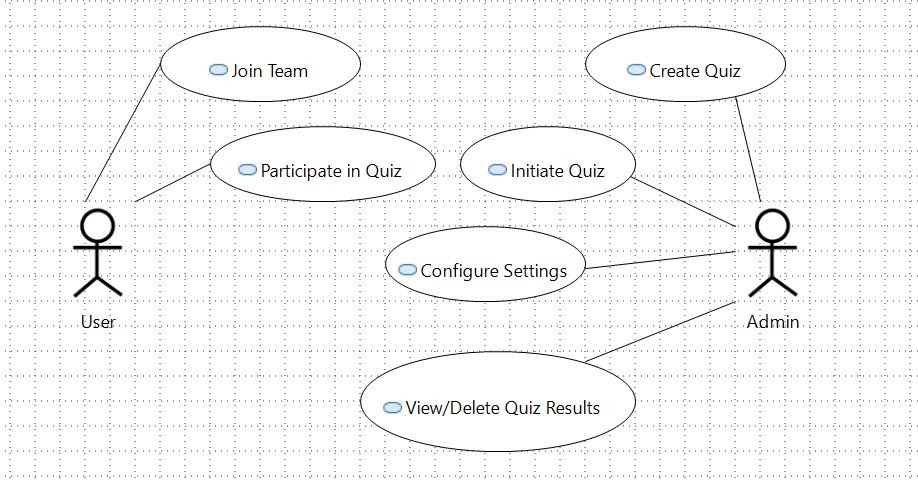
Coded entirely in Java, this system uses 2 text files for data storage, one for the storage of quiz objects (which include quiz settings and questions), and another to store quiz results. This was decided, rather than using a database such as MySQL, in order to keep the system simplistic, as we felt using MySQL was out of the scope of this project.

System interaction involves 2 types of users, an admin, and quiz participants. An admin can create, delete, and edit quizzes. This includes changing quiz settings such as: number of teams, number of questions, setting the quiz name, and number of correct answers required to win. Quiz participants can answer questions during a quiz. As this is a local version of an online quiz system, the admin starts the quiz, and from the same screen participants answer questions. If this were online, participants would be able to select which quiz they wanted to participate in, and select a team as well, and would then be able to answer only their team's questions.

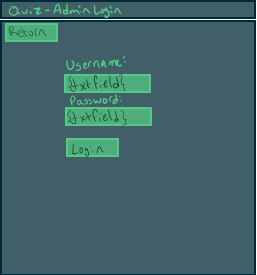
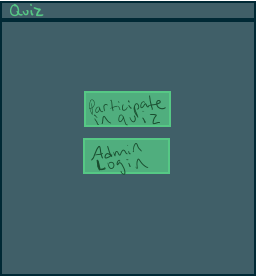
**3 Architecture Draft**

Using an Agile approach, we first coded a basic interface, and programmed the backend functions by sections. The following is the order we followed regarding the creation of our quiz system: Admin login, creating quizzes, implementing quiz settings, then how users taking the quiz can interact with the system.

**3.1 Class Diagram**Administrator has to login, but can then edit quizzes, view report of results from quiz, and we originally intended to have participants need to be listed by the Admin, in order to be able to participate, although that was not a part of our final program. Participants can answer quiz questions. Quiz Collector was supposed to work as an in-between and be where the quiz starts from. The user can also choose which quiz they want to attempt.

**3.2 Use-Case Diagram**Users can join a team and participate in a quiz. Admin can create, initiate, configure settings, and view/delete results from a quiz.

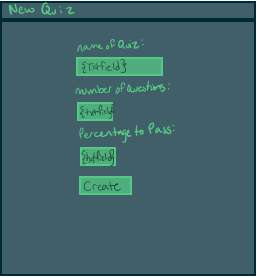
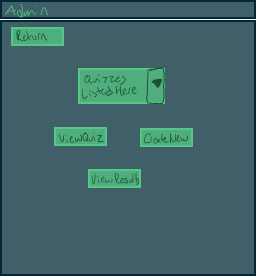
**3.3 Interface Design Diagrams**



Above are the Main screen, and Admin Login screen.

The main screen includes 2 buttons, one takes you to participate in a quiz, and the other takes you to the Admin Login interface.

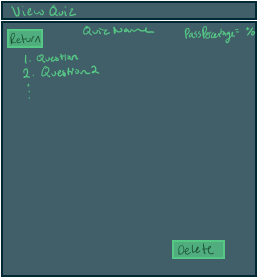
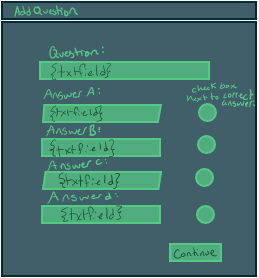
The Admin Login screen lets you return to the previous screen. You can also enter a username and password here, and if they are valid, will be taken to the Admin Main screen.



Above is the admin main screen and quiz creator screen.

The Admin Main screen has a button to return to the previous screen. Admin Main screen lets you select a quiz to: View Quiz or View Results, in which you are redirected to the corresponding screens. You can also be redirected to the quiz creator screen.

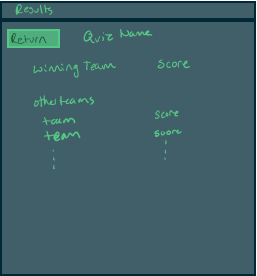
The Quiz Creator screen lets you name your quiz, set a number of questions, and a mark to pass. The create button will then bring up the correct number of Add Question screens before returning you to the Admin Main screen.



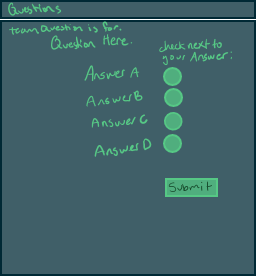
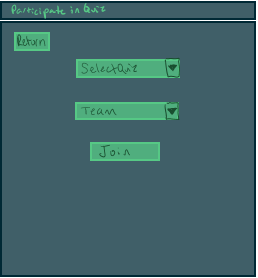
Above are the Add Question screen and the View Quiz screen.

The Add Question screen lets you enter a question with 4 multiple choice answers. There is a check box next to the correct answer, and a continue button to either continue to enter the next question, or to return you to the Admin Main Screen again.

The View Quiz Screen has a return button to go back to the Admin Main screen, and will list the quiz name, percentage required to pass, and all the questions included in the quiz. There is also a delete button to delete this quiz.



Above is the View Results screen. Here will be a return button to return to the Admin Main Screen. This screen displays the quiz name, the winning team and their score, and all the other teams and their scores.

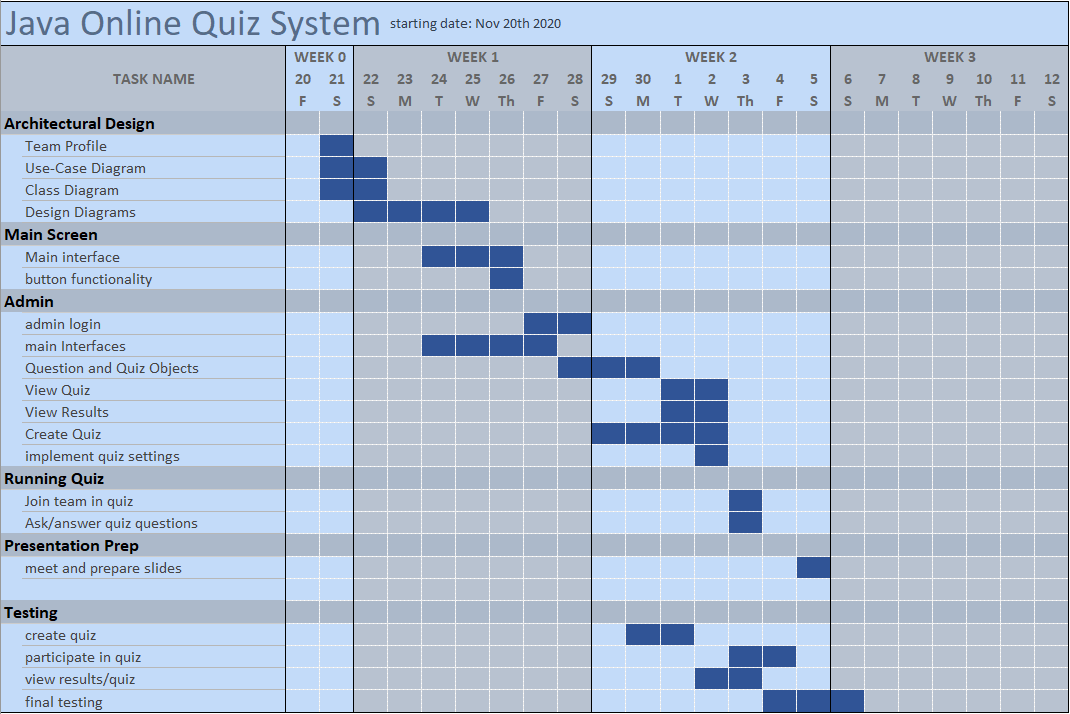


Above is the Participate in Quiz screen, and the Quiz Questions screen.

The Participate screen has a return button to the Main screen, lets you select a quiz and what team to be a part of, and has a join button, which brings you to either a temporary waiting screen that says please wait, or to the Quiz Questions screen.

The Quiz Questions screen displays which team the question is for, the question, the 4 answers, and after checking the box next to the answer, you can submit your answer, and then carry on to other questions.

**4 Estimated Schedule**



We planned to have a working prototype by December 2nd, in which we had a working interface (which was ready November 26th) and admin login/quiz set up (which was for the most part complete December 1st). We got this prototype working well by December 3rd, after having some problems with JavaFX. Another section which was delayed was our final testing and adjustment time. This was supposed to be done for December 4th, but was instead completed December 5th, because of our earlier delays.

**5 Design**

**5.1 Synopsis**

The general architecture our group decided on was the agile development method, intending to get our working prototype done as quickly as we could in order to continually improve it. Due to some of the delays mentioned in the previous section, we did not utilize this style to the fullest of its potential. However, this was still the basis behind our decisions when making up for these delays. To maintain order and ensure we knew what our priority was, we set our milestones to finish one part of the prototype before moving on to the next. For example, once the admin pages were made and could create and save tests, the next natural section to work on was how these tests would be given to the user. Given the overall short time working on this project along with other course work, this allowed us to maintain constant testing throughout the entire process, always having at least some working features in every iteration. This is also the benefit of having all the user requirements given to us by the project file, the required features always had the highest priority being added.

**5.2 User Requirements**

A summarized list of the user requirements given to us would be:

* Admin Login System
* User-ID System
* Having Numerous Teams
* Automatic Scoring
* A Given Number of Rounds
* Quiz Details Given by Admin
* Viewable/Deleting Quiz Results

We have met all these requirements in some form or another, aside from the user-id system. Due to the lack of online features in our prototype, we had decided to go in a different direction with that idea. We instead decided to give the names of the team members before the quiz starts, and those names would be printed in the results. Speaking of the results, our implementation of that also came out simpler than intended due to the lack of a database system. We had opted to store the results in a local file instead, which could be managed outside of the program in more depth than our program would currently allow.

**5.3 Limitations of Design**

The biggest limitation of our design needs no introduction, the lack of online features. Our program is currently designed in a way that favours our local system instead of database connections through systems such as JDBC. Some features such as the login system would require adjustments to implement this, such as calling a database of users before determining if the user has permission to enter admin pages. The storing of quizzes would also need a readjustment, but once the quizzes are loaded onto the user's local copy of the program it would still function as it does now. Therefore, given more time and experience, these limitations to the program are not detrimental enough to compromise the overall design.

**5.4 Problems**

A couple of problems had been introduced when working on the project, either with the program or outside of it. Our first major problem had come when one of our group members, the one designated to work on the frontend (Margaret Nixon), had issues with getting JavaFX to run on her computer. Following that, due to a misunderstanding on Jordan’s part, the other members of the team could not push to GitHub. While these issues were quickly resolved when they arose, there were more general issues that came up during development. One such issue was a misunderstanding of the user requirements and the format of the actual quiz portion. After a meeting with the team, a plan was eventually formed after some discussion on how it should work. Time management was another reoccurring issue with some members working later hours in order to meet certain established milestones. Overall, the problems found during development were typically resolved by coming together as a team to solve it.

**6 Testing**

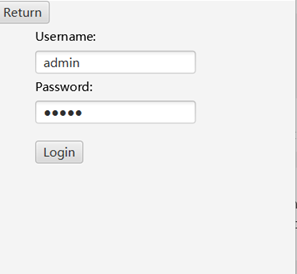
The first prototype with the basic interface GUI is created on November 26th by Margaret. It has the first page which user should see when they login into that website.



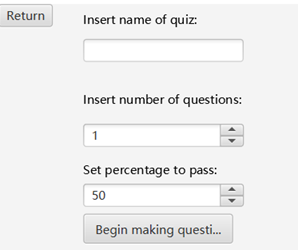
Nothing can be test on this case and this is the first look of our program.

On December 2nd Jordan has posted the second version of our program which has a lot of functions such as login interface, create quiz, view quiz and delete quiz. However, the quiz master for team members to finish the quiz is not yet finished.

**Login interface**

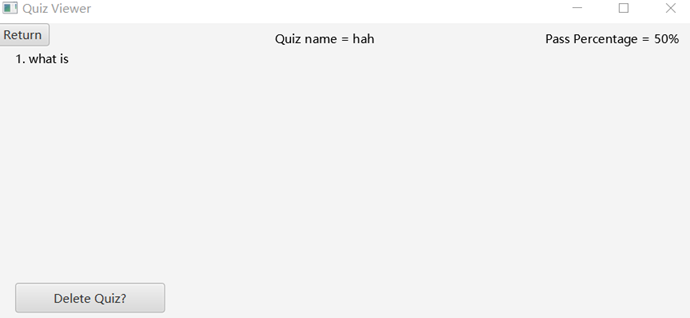


**Making quiz**



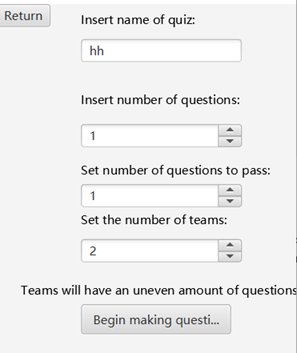
Bugs occur making pass percentage unnecessary; it could simply be the number of questions needed to pass.

**View quiz and delete quiz**

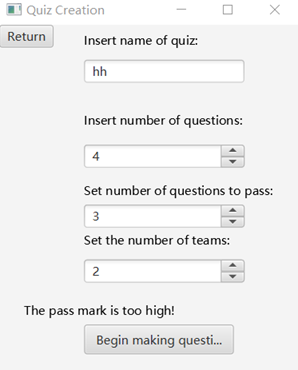


On December 5th, we already have the final version of the prototype. I found some bugs like the number of questions like uneven question, pass mark will be two high.

**Uneven question**

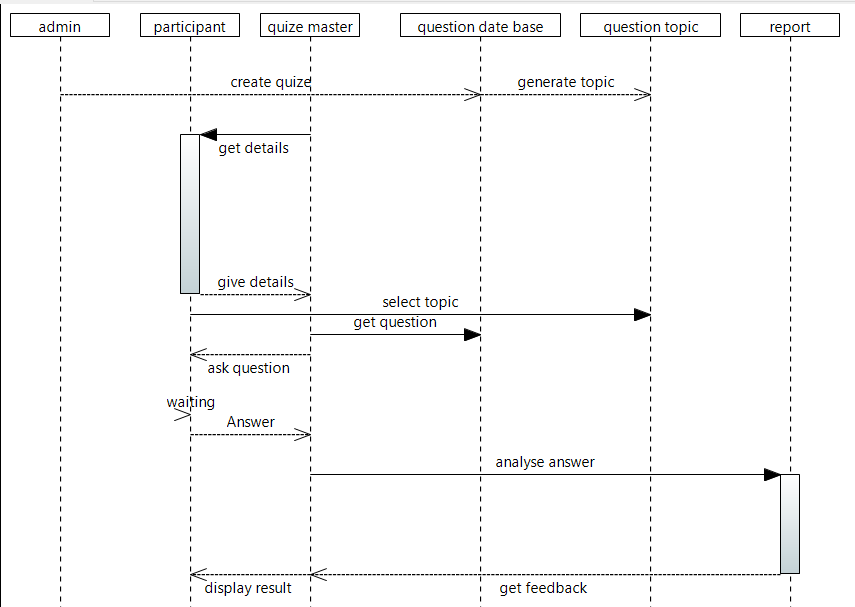


**Pass mark too high**



Finally, when we finished the presentation, we realized we still have a lot of issues related to this program. We cannot perform this quiz with multiple users using different devices, this system is not secure enough that everybody could get the password of admin to modify the quiz and the team member could peek at the other team’s result when they are doing the quiz. Also, I do an extra sequential diagram for the quiz that work online.

**Sequential diagram**



**7 Resulting Impressions**

**7.1 What We Learned**

We did not know much about GitHub and how it works, and so we learned a fair bit about how to use it by the end of this project.

**7.2 Different Next Time**

This time around we did not have the time to investigate and complete the online portion of this project but wish we could have. If we were to complete a similar project in the future, we would also try to manage our time and communicate amongst ourselves better. If we planned exact times to work on the project, and what parts to work on when, our work could have gone smoother. Because of our problems with GitHub working for all members, we did not use it as a resource much, when it has a lot to offer. Using GitHub will greatly improve the flow of work for any future projects similar to this that we work on.

**Contents**

**1 Team Profile** **1**

1.1 Margaret Nixon....................................................................................................1

1.2 Jordan Garnett......................................................................................................1

1.3 Xuchu Ma ............................................................................................................1

1.4 Myles St. Pierre ...................................................................................................1

**2 Overview of Project** **1**

**3 Architecture Draft**  **2**

3.1 Use-Case Diagram...............................................................................................2

1.3 Class Diagram......................................................................................................3

1.3 Design Diagrams..................................................................................................3

**4 Estimated Schedule** **6**

**5 Design** **7**

5.1 Synopsis...............................................................................................................7

5.2 User Requirements...............................................................................................7

5.3 Limitations of Design ..........................................................................................7

5.4 Problems ..............................................................................................................8

**6 Testing** **8**

**7 Resulting Impressions** **11**

7.1 What We Learned ..............................................................................................11

7.2 Different Next Time ..........................................................................................11