VIP-ITS Spring 2023 Project Proposal - QuizApp iOS

Group Membership:

Member		
Nirjhar Deb	Skills	Java, Python, C++, HTML, SQL, CSS, JavaScript
	Credits	3
	Responsibility	● Read the previous code and learn Swift + MongoDB ● Understand and document how to use MacOS on Windows for iOS development ● Research and implement live, multiplayer quizzing ● Add functionality to the "next" button for flashcards
Alyssa Zhu	Skills	Java, Python, C, HTML, SQL, CSS, JavaScript
	Credits	1
	Responsibility	● Read the previous code ● Learn Swift, IOS Development, and MongoDB ● Research how multiplayer games function/how to increase learning ● Implement registration function ● Figure out frontend functionalities needed for multiplayer game
Alex Tang	Skills	Java, Python, CSS, HTML
	Credits	1
	Responsibility	● Read the previous code ● Learn Swift and any other related developer tools ● Research how live rooms work and ways to implement them ● Develop some of the frontend ideas for the live QuizApp (leaderboard, questioning) ● Add a feature to create practice tests from existing questions

Project Goals:

We are working on improving the QuizApp (iOS) by implementing a **live, multiplayer quiz feature**. We will start by reading through the previous code to understand what each section does and eventually build upon this to integrate an online component to it. There needs to be some research done to look into implementing a live server. Some front end work also needs to be done to create a finished look for the application itself. The focus of this project is to create the application for this multiplayer quiz idea itself. The algorithm for choosing the questions (in terms of relevance and difficulty) can be implemented either by the other sub-team or a future team. For now, we plan to randomly choose a certain number of questions based on units and subunits.

Individually, we all have another small feature we want to focus on too. We plan to implement a “next” button, create an algorithm to make sample tests, and connect the login page to the backend. By the end of this project, we hope to be proficient in Swift and gain more knowledge in networking and iOS development.

Project Timeline:

Week 1-3: Team formation, proposing project ideas, planning the project.
Week 4-5: Read into the previous QuizApp (iOS) code and research how live servers work and how we can apply it to our project.
Week 6-12: Build a live server room that allows for interaction between multiple people. Work on smaller side tasks on the backend such as login and "next" button. Also work on the frontend of the project itself.
Week 13-15: Refine the frontend work of the quiz app (integrating the old flashcards, creating a leaderboard). Refine the smaller side tasks.
Week 16-17: Final Presentation.

Project Description:

Problem and Research Ideas

The primary focus of this project is to create a room where users from different devices can interact with each other. Currently, the app functionality focuses on individual studying, so we wanted to give students the opportunity to learn together in a live setting. Our proposed solution is to create a function that would take the quizzing section from the previous iOS app and provide it to a group of students to each answer individually, where the number of points (determined by correctness and timeliness) will be tallied onto a leaderboard.

After we implement the multiplayer function, an interesting idea to be researched is how the system will decide which questions to ask, possibly personalized to the specific group of users.

Currently, we plan to take groups of questions from specific units to give a quiz focused onto certain subjects/chapters of the textbook. A potential idea could sort the questions in terms of relevancy or difficulty. Similar algorithms were already developed in previous projects, but some work would be needed to implement them into this specific project. Though it will not be a main focus of our project, if time permits, it could be a potential change in our project.

Foreseeable Problems and Potential Pitfalls

Since none of us have a background in Swift, we will need to spend some time getting comfortable with Swift and eventually using other libraries to implement the live feature. Also, most of our team does not have experience with networks, so we will have to invest time in learning the fundamentals, which could potentially delay our development timeline.

Another problem that we can foresee is that, since Nirjhar is planning to use the MacOS on Windows via a virtual machine, he might experience issues when testing multiplayer, such as higher latency (i.e., lag). This could negatively impact his development experience, and thus we will have to work through such problems together. Also, this means that for the demo, we will not be able to use his machine and will have to rely on Alyssa and Alex's Apple computers.

Implementation, Tools, Resources, and Collaboration

This project is mainly developed in Swift with some backend in Python/using MongoDB, which we will continue to use as well as any additional technologies we need for the multiplayer function. We will fork the ITS-Swift-Team-Fall-2022 repository on GitHub to use as the starting template that we can add our multiplayer game to. Each member will create their own branches and commit their individual contributions to their respective branches. We will primarily communicate on Microsoft Teams and will meet weekly on Thursday 6:30 PM to ask questions, solve problems, and coordinate tasks. These meetings should last around 30 minutes.