Project Proposal: Android QuizApp

Project Description

Problem to be Solved

- Existing ITS tools are not targeted for mobile devices or on-the-go studying
- Ability to share student-created study tools such as flashcards or quiz questions and collaborate with other users in an engaging way
- Better recommendations about what to study through ML models

Proposed Solution

App Functionality

Currently, the mobile application delivers simple quiz questions to users. We display different formats: multiple choice (MC), vocab/concept flashcards, and short answer (SA).
Based on how the student does, it can recommend what they should study next. These recommendations will be determined using a machine learning model.

Features to work on:

- Setup application to handle communication with an ML model
    - Potential API development
- Generating quizzes tailored to a content area recommended by ML model
    - MongoDB data extraction
- Implement spaced repetition algorithms to increase memory recall in students.
- Collect data about time and how long it takes users to answer questions or take quizzes
- Add an existing dataset of questions and answers from the textbook to the MongoDB database and auto-generate QuizApp flashcards for each chapter based on that dataset.
- Students can add their own flashcard sets.
- Students can share flashcard sets with other students
- Tag questions by subtopics to filter

Benefits Of This App

- Convenience: Quick and easy for students to use since the majority of them are on their phone already.
- Utilizes proven methods to incentivize users to open the app and work on their comprehension of different concepts
- Provides a targeted learning plan for students who are struggling on a certain concept by prioritizing certain content

    - Many times when students are stuck, they are not sure where they are lacking in knowledge - the ML model would provide them questions that would be specifically catered to their skill set

Potential Problems and Pitfalls/Other Areas for Research

1. What is the content for the questions? Is it vocabulary from the textbook? Practice problems? Will it differ based on question format (how are certain concepts displayed the best for learning?)
    - Content from the (digital) textbook
    - Questions from ITS - vocabulary, fill in the blank, specific equations, etc.
    - Questions created by students
2. Setting data requirements on what type of data is needed to be collected for the ML models
3. Make sure we are designing and optimizing for a mobile environment.

Implementation

Software and Development Tools

- IDE
    - Android Studio
- Programming languages
    - Java
    - JavaScript
    - Python
- Database
    - Cloud-based MongoDB
- API for communication
    - NodeJS
    - ExpressJS
- Libraries/frameworks: Mongoose, Volley
- UI-UX wireframing:
    - Figma
- GitHub for File Management
    - Git Kanban Board for workflow management

Project Goal For This Semester

Last semester, we delivered a functioning mobile application that allowed users to create an account, sign in, and work through hard-coded questions. The goal for this semester is to set up the mobile application to handle activities that could be done with other users, allowing the user to collaborate or compete with other users. This will require standardizing what type of data we

want to send, what type of data we will receive, and how to process the data. We also want to refine the data extraction from our database to ensure that we are pulling the correct questions of the content area. We also want to develop and maintain an aesthetic look to the app, and refactor our code to streamline our development process.

Milestone	**Date Due** (Hard deadlines are in **bold**)	**Status**
Project Plan Draft	9/16/22	In progress
Final Project Plan	9/23/22	In progress
Fork last semesters github repo and initial project files + continue onboarding	9/30/22	Not Started
Finalize subgroups and what people want to work on	10/7/22	Not Started
Have the first few features done	10/14/22	Not Started
Continued work - depends on previous progress - see requirements section	TBD	Not Started
Continued work - depends on previous progress - see requirements section	TBD	Not Started
Continued work - depends on previous progress - see requirements section	TBD	Not Started
Continued work - depends on previous progress - see requirements section	TBD	Not Started
Prepare for Demo + Final Touches	TBD	Not Started
Demo	TBD	Not Started
Project Cleanup and Final Documentation	TBD	Not Started
Final peer review and presentation	12/8/22	Not Started
Project Submission	12/15/22	Not Started

Group Membership

Member Skill Sets

Member	Time Commitment and Credits	Skills and Interests
Rishi Nopany	6-8 hours/week (2 credits)	Java, JavaScript, React, Python, HTML/CSS, Android Studio
Abdulaziz Memesh	6-8 hours/week (2 credit)	Java, Python, Android Studio
Varun Patel	6-8 hours/week (2 credit)	Java, JavaScript, React, Python, HTML/CSS, Android Studio
Shreekrishna R Bhat	3-4 hours/week (1 credit)	JavaScript, React, NodeJS, Express, MongoDB, Java, Python, HTML/CSS
Sanjit Pingili	3-4 hours/week (1 credit)	React, Java, HTML/ CSS, JavaScript, NodeJS
Sahana Krishnan	6-8 hours/week (2 credit)	Java, JavaScript, React, Python, HTML/CSS, Android Studio, MongoDB
Samarth Parameswar	6-8 hours/week (2 credit)	Java, JavaScript, Python, HTML/CSS, Android Studio, MongoDB

Task Assignment and Participation

Responsibilities

- Abdulaziz Memesh
    - Project management, incorporating machine learning, UI/UX
- Varun Patel
    - Working on app design, possibly using React Native to design UI/UX
- Shreekrishna R Bhat
    - Working on adding API endpoints, and possibly working on a simple front end.
- Rishi Nopany
    - Working on implementing the Spaced Repetition algorithm (SuperMemo algorithm).
    - Working on adding existing questions to the MongoDB database and sectioning flashcard decks by Chapters.
- Sahana

    - Working on front end and also using MongoDB database for specific aspects of flashcard decks. Also working in React.
- Samarth
    - Work on backend with API endpoints and MongoDB database in order to streamline user experience
    - Work on the algorithms in the backend for repeating questions that were missed
- Sanjit
    - Will work on React and the front-end. Will attempt to integrate a timer function that stores data on the time it takes users to answer questions.

Work Division

- Front-End
    - Varun Patel
    - Samarth Parameswar
    - Sahana Krishnan
    - Sanjit Pingili
    - Abdulaziz Memesh
- Back-End
    - Shreekrishna Bhat
    - Rishi Nopany
- Data Analytics
    - Abdulaziz Memesh

Communication Resources

- Weekly meetings: TBD
- Google Drive: meeting notes, learning resources, and other shared documents
- Microsoft Teams for quick communications and meetings