**Project Charter - StudyBank**

Group #8 - Danni Shang, Tevin Omoruan, Sizhao Lin, Riyaz Khan, Jessa Sundberg

Created Dec. 09 2020

Login Credentials

|  |  |
| --- | --- |
| Username | Password |
| [myaccount@mylaurier.ca](mailto:myaccount@mylaurier.ca) | password |
| estone@mylaurier.ca | 11223344 |

**Project Purpose**

With the recent necessity for online schooling, students have found themselves turning to digital resources to help them with their studies more than ever before. Self-testing has always been a popular study method, and flashcards are an excellent self-testing approach that allow students to learn and retain new information even through the process of creating the flashcards. Flashcard quizzes also make use of the spaced repetition technique that helps students figure out what information they remember and what they need more practice at. Finally, another advantage of flashcards is that they can be used in short bursts of time throughout the week, creating small and informal study sessions. This would work well with a mobile application that can be accessed quickly and easily on-the-go. The purpose of StudyBank is to provide an easy-to-use, convenient, and portable way to study that doesn’t limit the user to sitting in front of a computer screen for hours on end. The app aims to help students be more productive and effective with their studying, and to take away the stress that a normal study session can bring.

**Project Scope**

The scope of our project is as follows:

* Provide an application where students can create their own flashcards, either through manually inputting the questions and answers or through importing questions from their device storage
* Students can create quizzes out of various flashcards and keep track of their quiz grades and overall user score generated from these grades
* Students create a user profile through which they can upload information about themselves, post a profile picture, and keep track of their flashcard library, quiz results, and user score
* The application includes the ability to input and export flashcard questions and answers to and from the device storage
* The focus is on ease-of-use and convenience, so the functionality of the app should be straightforward and should feature a simple UI

**Product Perspective**

**Front End:** The user interface consists of a navigation drawer and toolbar that can be used to access various activities and app sections. Buttons, dialog boxes, and progress bars are some of components that will be used to make the app more user friendly and easy to understand. The app will support a variety of screen sizes.

**Back End:** The backend will store data such as user login and profile information, flashcards, and quiz scores. Log in will be done through Firebase Email and Password Authentication, and profile information will be stored on Firebase Realtime Database to allow changes to be reflected in the app immediately. Flashcards and quiz scores will be stored either through Firebase Cloud Storage or locally in the device’s internal storage.

**Stakeholders**

The primary stakeholders will be the application’s users. The main target for this user base is students who are tech-savvy, currently receiving an online education, and self-motivated to succeed in their studies. Students can be from any age range, although the primary focus is on secondary and undergraduate students since these education levels tend to require the most rote memorization for a wide range of topics, which fits well with the functionality of the app. Other targeted users include tutors who are interested in creating and sharing flashcards and quizzes with their pupils, and teachers and professors who are looking for a platform to quiz their students on or a way to provide additional course resources. Users must speak English or French, as these are the app’s two supported languages.

**Constraints**

* The app must be able to store user information, quiz grades, and overall user score in a database
* Must be able to store a large number of flashcards and separate flashcards by category
* Must meet project requirements (ex. Have progress bars in each section)
* Users cannot access other user’s accounts and authentication is required when logging in
* Must be able to be accessed by at least 70% of Android devices (minimum SDK of API 24 or lower)

**Specific Requirements**

**Functional Requirements**

|  |  |
| --- | --- |
| **Requirement No.** | **Description** |
| **1.1** | **Register user account** |
| **1.2** | **Log in** |
| **1.3** | **View and update user profile** |
| **1.4** | **Create flashcards** |
| **1.5** | **Write quiz and view grades** |
| **1.6** | **Import and export questions** |

**Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement No.** | **Type** | **Description** |
| **2.1** | **Ease of Use** | The application should be user-friendly and feature a simple, intuitive UI. Each section of the app will have a straightforward functionality. Components such as progress bars, toasts, and navigation buttons will be included to improve usability. A help menu is included to give users more information if needed. |
| **2.2** | **Performance** | Flashcards and quiz results should load quickly onto the screen. The application should respond quickly to user inputs and each activity should open smoothly. |
| **2.3** | **Security** | User information such as password is only visible in the Firebase database, which is only accessible by the application’s developers. Only the user can change their profile information, modify and delete their flashcards, and view their quiz results. |

**SIGNED BY:**

Danni Shang

Tevin Omoruan

Sizhao Lin

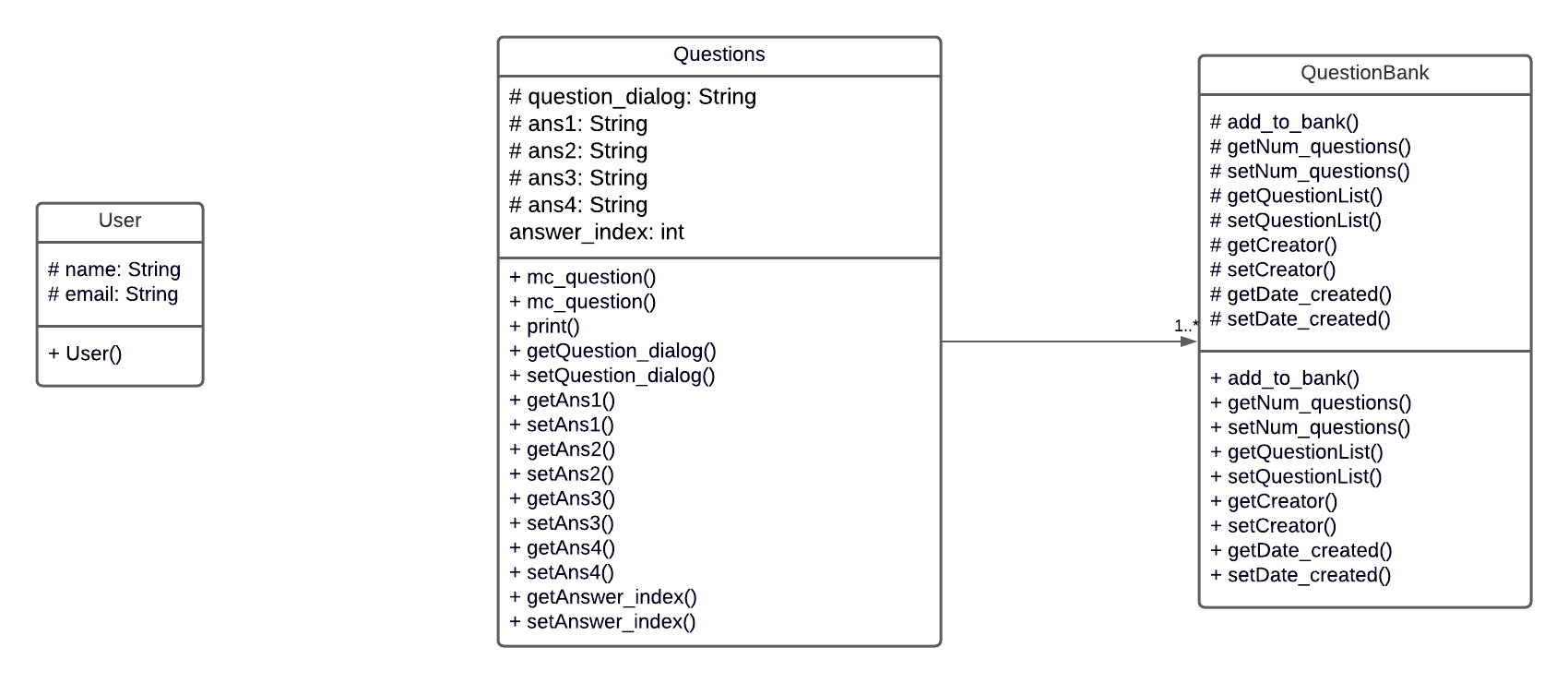
Riyaz Khan

Jessa Sundberg

**Description of class and method**

|  |  |
| --- | --- |
| Class and methods | Description |
| **User**  Values  name: String  email: String  Methods  User(String name, String email) | The User class holds the user’s login information.  Creates the user object |
| **Questions**  Values  question\_dialog: String  ans1: String  ans2: String  ans3: String  ans4: String  answer\_index: int  Methods  mc\_question()  mc\_question(String dialog, String ans1, String ans2, String ans3, String ans4, int i)  print()  parse\_line\_to\_mc(String line)  getQuestion\_dialog()  setQuestion\_dialog(String question\_dialog)  getAns1()  setAns1(String ans1)  getAns2()  setAns2(String ans2)  getAns3()  setAns3(String ans3)  getrAns4()  setAns4(String ans4)  getAnswer\_index()  setAnswer\_index(int answer\_index) | The Question class holds the questions and options created by the user.  creates the question object  Returns the question  returns the question dialog  sets the question dialog  returns option 1  sets option 1  returns option 2  sets option 2  returns option 3  sets option 3  returns option 4  sets option 4  returns the index of the answer  sets the index of the answer |
| **QuestionBank**  Values  num\_questions: int  questionList: ArrayList<Questions.mc\_question>  creator: String  date\_created: String  Methods  add\_to\_bank(Questions.mc\_question e)  getNum\_questions  setNum\_questions(int num\_questions)  getQuestionList()  setQuestionList(ArrayList<Questions.mc\_question> questionList)  getCreator()  setCreator(String creator)  getDate\_created | The QuestionBank class will hold the objects created by the Question class  Adds the question object to the question bank  returns the number of questions  sets the number of questions  returns the list of questions  sets the question list  returns the creator  sets the creator  returns the date in which the bank is created |

**Class Diagram**

****

**Use case description**

User logs into application

|  |  |
| --- | --- |
| Use Case | Log in |
| Actors | Students and professors |
| Trigger | User clicks “LOGIN” button |
| Precondition | User is on the application’s main page |
| Postcondition | User logs in and lands on application’s home page |
| Basic Path | This page is accessed by opening the application |
| Exception Paths | If the user does not enter email address, an error symbol will show up indicating to the user that email address was not entered  If the user does enter an email address, but enters a wrong password, a toast message will show up telling the user that the combination of password and email is incorrect, and will ask user to try again |

User registers an account in the application

|  |  |
| --- | --- |
| Use Case | Register an account |
| Actors | Students and professors |
| Trigger | User clicks on “Don’t have an account? Sign up” button |
| Precondition | User is on the application’s main page |
| Postcondition | User lands on the application’s sign up page |
| Basic Path | This page is accessed by opening the application and clicks on “Don’t have an account? Sign up” button |
| Exception Paths | If user does not fill out all or any of the entries, the text fields will be highlighted in red, and a toast message will pop up asking user to fill in the highlighted areas    If user enters something other than an email for the email text field, an error will show up indicating to user that invalid email was entered    If user cannot confirm the password (password and confirm password fields do not match), an error will show up indicating to user that password confirmation does not match    If user registers with an existing email, a toast will show up indicating that account was registered. User will not be able log in using that account. |

User views their user profile

|  |  |
| --- | --- |
| Use Case | View user profile |
| Actors | Students and professors |
| Trigger | User clicks on “Profile” button |
| Precondition | User is on the application’s home page after logging in |
| Postcondition | User lands on the application’s profile page |
| Basic Path | From the application's main page, the user enters their login information properly and heads towards the application’s home page. They then open the menu bar with the 3 dash symbol, then click on “Profile” button to visit the user profile |
| Exception Paths | None |

User edits their user profile

|  |  |
| --- | --- |
| Use Case | Edit user profile |
| Actors | Students and professors |
| Trigger | User clicks on “Edit” button on the profile page |
| Precondition | User is on the profile page |
| Postcondition | The user lands on a page where they can edit the information that would be displayed on their profile page. The information can be saved by clicking on “CONFIRM” button |
| Basic Path | From the application's main page, the user enters their login information properly and heads towards the application’s home page. They then open the menu bar with the 3 dash symbol, and access the profile page by click on “Profile” button, They will be able to edit the user profile by scrolling down and click on “EDIT” button |
| Exception paths | None |

User logs out from the application

|  |  |
| --- | --- |
| Use Case | Log out |
| Actors | Students and professors |
| Trigger | User clicks on the button with logo on top right |
| Precondition | User is on the home page after logging in |
| Postcondition | User logs out; leaving the home page and lands on the application’s main page |
| Basic Path | From application’s main page, user enters their login information and heads towards the application’s home page, where they then click on the button with door logo to log out and land on the main page |
| Exception paths | None |

User creates a flash card, with either a multiple choice or a true/false question

|  |  |
| --- | --- |
| Use Case | Create a flash card |
| Actors | Students and professors |
| Trigger | User clicks on the “Create Quiz” button. Then, the user will have the option to create a flash card with a true/false question or a flash card with a multiple choice question, by selecting the option on the custom dialog |
| Precondition | User is on the home page after logging in |
| Postcondition | User lands on the page where they can create their own question in the form of multiple choice or true/false, and determine the correct answer for that question |
| Basic Path | From the application's main page, the user enters their login information properly and heads towards the application’s home page. They then open the menu bar with the 3 dash symbol, and then click on the “Create Quiz” button. They will then click on the button “CREATE FLASHCARDS” and be met with a custom dialog asking if the user wants to create a flash card with a true/false question or a multiple-choice question. They will then be sent to a page that allows them to create a flash card with their own multiple-choice or true/false question |
| Exception paths | None |

User answers the quiz consisting of the questions created by the user

|  |  |
| --- | --- |
| Use Case | Quiz activities |
| Actors | Students and professors |
| Trigger | User clicks on the “Quiz” button |
| Precondition | User is on the home page after logging in |
| Postcondition | The user lands on the quiz page, where they are given randomized questions to answer. These questions are the ones that were created by the user. The scores will be visible to user as user answer the questions |
| Basic Path | From the application's main page, the user enters their login information properly and heads towards the application’s home page. They then open the menu bar with the 3 dash symbol, and then click on the “Quiz” button. They will then see a page where they are given either a true/false or a multiple-choice question |
| Exception paths | None |

User exports/imports flash cards

|  |  |
| --- | --- |
| Use Case | Import and export |
| Actors | Students and professors |
| Trigger | User clicks on “Import/Export Files” button |
| Precondition | User is on the home page after logging in |
| Postcondition | The user lands on the Import/Export page, and will have the option to either import flash card file or export flash card file |
| Basic Path | From the application's main page, the user enters their login information properly and heads towards the application’s home page. They then open the menu bar with the 3 dash symbol, and then click on the “Import/Export files” button. They will then see a page where they can either import a flash card file or export a flash card file |
| Exception paths | None |

Help

|  |  |
| --- | --- |
| Use Case | See help |
| Actors | Students and professors |
| Trigger | User clicks on “Help” button |
| Precondition | User is on the application’s home page after logging in |
| Postcondition | User sees the Help page popping up |
| Basic Path | From the application's main page, the user enters their login information properly and heads towards the application’s home page. They then open the menu bar with the 3 dash symbol, then click on “Help” button to see help |
| Exception Paths | None |

**Use case diagram**

