**Side project: “ClassLink”**

1. A Kahoot-style program that connects the lecturer with the large number of students in a lecture hall. In such an environment it can be difficult to make sure everyone is understanding the material.
2. The program aims to remove interruptions to the lecturer by eliminating the old-school “raise your hand to ask a question” by connecting the students and the lecturer to a live display (or feed) of questions from the students.
   1. Questions can be ‘upvoted’ so the professor knows what the most pressing issues are
   2. Questions can be answered by other students. These answers can also be upvoted.
      1. These answers should provide justification, for example a picture of the steps taken or the reasoning that led to the answer.
      2. Click on an answer to expand and show the justification
   3. The professor can mark answers as ‘correct’ rather than take time out of the lecture and answer themselves.
   4. If there is no correct answer then the professor will know that something was very unclear and must improve the lecture. And also answer the question.

The end game:

* An app (Android and iOS,) a desktop application (Windows, Mac OS) and a web based version
  + Kind of like discord

<https://developer.android.com/training/basics/data-storage/databases.html>

<https://www.tutorialspoint.com/android/android_sqlite_database.htm>

mySQL

SQLite

Firebase

Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Rahul | Jay | Deliverables |
| 0 | Skeleton structure for activities  +lecture layout UI using illustrator | Illustrator make activity layouts | Working user interface |
| 1 | Code/UI for login, settings, signup. Test button stretching. Make sure button pressed/unpressed animation is onClick and the function is onRelease | Code/UI for main menu, user profile  Continue design of activity layouts in Illustrator. Buttons (pressed/unpressed) | Functional & polished UI |
| 2 |  | Get adding and reading questions from firebase working | Classes and database structure I |
| 3 |  |  | Classes and database structure II |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |

**May:**

Finish front end. Get started on back end.

**June:**

Develop back end.

**July:**

Ideally get the back end working by mid July. Hook everything up towards the end of July.

**August:**

Touch everything up. Get a working app by mid-August. Test.

* Back end needs two classes that need to work together:
  + User
    - Includes login, preferences (stored locally,) profile (stored in database)
  + Posts
    - includes questions, answers, etc.
* Front end includes:
  + Menus, transitions (according to UML), graphics, layouts, display
  + Front end is linked to back end very simply
    - Only through fetching user data and post data

NOTE: CONSIDER RESTART FEATURES THAT ACTIVATE AT THE END OF THE SEMESTER

Front-End:

Modules:

1. Activities:
   1. Login Screen
      1. Sign up UI
   2. Main App Menu
      1. Settings
      2. Profile (provide user information) (potentially on database or locally)
      3. Study Group Menu
         1. Create Group
            1. Main Group UI
         2. Join Group
            1. Main Group UI
      4. Class Lecture Menu
         1. Create Group
            1. Main Class UI
         2. Join Group
            1. Main Class UI

Back-End:

Classes:

1. User
2. Question/Answer
3. Notifications

Class Tree:

(Classes ordered from super classes (left) to child classes (right)

(regular case = concrete class)

(italic case = abstract class)

(bold case = interface class)