

# **SimpleTutors**

## Log-in & User system

- Log-in page
  - Create account page
- Tutor/Student listings
  - Tutor perspective:
    - Create tutor profile (list courses, etc)
    - View tutoring requests (accept/deny)
  - Student perspective:
    - Browse tutors (by course)
    - View tutor profile (request to be tutored)
- Student & Tutor Communication
  - Messages homepage (shows the tutors/students chats you are in, sorted by most recent message sent)
  - Chat view
    - Able to send images, maybe upload/download pdfs if possible
- Student Feedback
  - In the chat with a tutor, student has an option that brings them to a feedback page
    - Prompts student to reflect on how the tutoring session went, what topics they feel they understood, and what they need more help on
  - Student feedback is sent as a special chat message, that when the tutor clicks on it brings up a similar, read only version of the form the student filled out (? feel free to discuss how else to incorporate this if don't like that)

## **Descriptions of classes and methods**

### **Umar**

- ❖ LoginIn Activity (Java class and XML)
  - Allows a user to either sign in or create a new student or tutor account.
- ❖ NewUserPage Activity (Java class and XML)
  - New users can create a username and a password which is stored in the database afterwards, alongside their user type, either a student or tutor
- ❖ DBHelper
  - Created Tables Users, Students, and Tutors where original student data is captured when a student logs in, then updates that table with profile details when user logs in for the first time/chooses to update their profile.
  - InsertData: When a new user is created, this method is called and initially stores the users username, password, and usertype.

- InsertProfileDetails: Updates the users table with relevant information when they update their profile or log in for the first time
- getUserProfileDetails: Gets all user profile details
- createStudent: Creates a new student in student table
- createTutor: Creates a new tutor in tutor table
- checkusername: checks to see if username already exists in DB so it is not repeated
- Checkusernamepassword: Validating for login
- Checkusertype: returns user type, either a student or tutor
- ❖ StudentLandingPage Activity (Java class)
  - Created the java class for the student landing activity, along with the action bar that allows the student to navigate to other activities or log out.
- ❖ TutorLandingPage Activity (Java Class)
  - Created the java class for the tutor landing activity, along with the action bar that allows the tutor to navigate to other activities or log out.
- ❖ CreateStudentProfile (Java Class)
  - Java class used to retrieve users profile information and insert into database for use in StudentLandingPage
- ❖ CreateTutorProfile (Java Class)
  - Java class used to retrieve users profile information and insert into database for use in TutorLandingPage

## Tyler

- ❖ ChatList Activity (Java Class)
  - ChatList runs the logic for the activity\_chat\_list layout, which displays a list of each of the user's chat partners and allows them to continue a chat or start a new chat. ChatList populates the list of chats by polling the database represented with the ChatDatabaseHelper.
- ❖ ChatWindow Activity (Java Class)
  - ChatWindow runs the logic for the activity\_chat\_window layout, which displays the history of chat messages between the current user and another user. ChatWindow populates the list of messages by polling the database represented with the ChatDatabaseHelper
- ❖ ChatDatabaseHelper (Java Class)
  - ChatDatabaseHelper is an extension of the SQLiteOpenHelper class, and it implements the database which stores every message sent between any given pair of users.
- ❖ Activity\_chat\_window.xml
  - Layout for the chat window activity.
- ❖ Activity\_chat\_list.xml
  - Layout for the chat list activity.

## Hafsah

- ❖ StudentLandingPage
  - When a student logs into their account they are taken to the student landing page

- Developed XML file, including logo and buttons accordingly
- ❖ TutorLandingPage
  - When a tutor logs into their account they are taken to the tutor landing page
  - Developed XML file, including logo and buttons accordingly
- ❖ Presentation
  - Created the ppt presentation, added images, theme, etc. accordingly
- ❖ Strings
  - Created Strings xml file in the values folder for both english and british english
- ❖ Adjusted all XML layout files to include company logo and the correct formatting (font colour, button colour, button shape, etc.)
- ❖ CreateStudent layout
  - When a user creates a new account they hit student and the create student layout allows them to build their student profile
  - Developed the XML file for this
- ❖ TutorProfile layout files
  - When a user creates a new account and hit create new tutor profile, the create tutor layout allows them to insert their tutor information to be displayed
  - Developed the XML file for this

## Jessie

- BrowseTutorsActivity
  - One of the main activities for students: browse the tutors available using a RecyclerView. Clicking on a profile brings you to that profile loaded into ViewTutorProfileActivity. Search allows to search for courses (except that courses was not stored the way I originally expected and so does not work now unfortunately)
- TutorProfile
  - An object-oriented class for holding tutor profile information as objects, and various methods used to populate the profiles, including avatar loading that did not end up being implemented in other parts of the app
- TutorRecyclerViewAdapter
  - RecyclerView adapter for the BrowseTutorsActivity
- TutorsSQLiteHelper
  - An SQLite helper for the tutors database I originally had, but ended up not getting used due to miscommunication with team members as to who was storing what where
- TutorViewHolder
  - View holder for tutor profiles in the BrowseTutorsActivity recyclerview
- ViewTutorProfileActivity
  - When a profile in BrowseTutorsActivity is clicked, that profile's information is loaded in more detail here. A Request Tutor button starts a chat with that Tutor by sending an Intent to ChatWindow

## **Filip**

- ❖ student\_feedback\_final
  - Java class that allows user to input all the required information for their feedback form
  - Implementations in the DBHelper class to store all feedback that a user provides their tutor into the database
  - Implementations in the ChatWindow class to allow a student to have the option of submitting a feedback form (checks to see if one has already been submitted)
- ❖ Tutor\_view\_feedback
  - Java class that generates all the required information that will be displayed in the activity\_tutor\_view\_feedback activity.
  - Implementations in the ChatWindow class to allow access to view the feedback form as well as checking if there is a form available
  - Implementations in the DBHelper class to view all stored feedback forms for this tutor and to see check if they received feedback from a specific user
- ❖ Feedback
  - Java class that will be use to initialize a feedback object
- ❖ activity\_tutor\_view\_feedback
  - View only version of the feedback form that was submitted to the tutor by the student
- ❖ activity\_student\_feedback\_final
  - The feedback form that is generated when the user wishes to submit feedback to their tutor

## **Class Diagram**

Found in the Project, App -> Java -> com.example.simpletutors -> classdiagram.png

## **UML Diagram**

Found in the Project, App -> Java -> com.example.simpletutors -> umlfile.puml

## **Use Case Diagram**

