Assessment 2

Sprint 1: Project portfolio & Application Progress

SIT708 – Mobile Systems Development

Name: Viseth Kong

ID: 218416405

Expected Grade

I deserved a credit level because of my understanding of what I've studied and the high level of effort that I've put into the tasks. I used a lot of different coding techniques to make sure the program worked correctly, flawlessly, and consistently without crashing, and to make sure I did a good job on every assignment.

Throughout my time studying this course, I've run into compilation problems or bugs that have prevented me from constructing applications successfully. As a result, if I still have a problem or run into a compilation issue when debugging, I've looked over the material on the unit site to understand how to accomplish similar methods. To solve some of the difficulties, I used Google to research the issue or YouTube to look for related coding approaches. By reviewing the lecture and workshop videos, I was able to understand the mobile development resources step-by-step. The lecture and workshop videos are quite helpful in reminding me to complete all the weekly activities. These tasks introduced me to a variety of coding techniques, including the fundamentals of Kotlin code, implementing Android components, and focusing on many features and syntax.

I thought some of the coding activities were challenging, so I did some online research and went over the weekly lectures more than once to try to figure them out. Some tasks, such as RecyclerView and working with FCM tokens, I consider to be a little challenging for me. And when I progress from one activity to another, it becomes increasingly more difficult, forcing me to be resourceful in order to dedicate more time to it. But learning is enjoyable. In addition, I always completed the weekly activity after we had finished the class and went over some of the lessons we had learned.

I'm confident that all I've learned in this unit so far, as well as my contributions to the activity, is more than enough to earn me a credit, and I believe I deserve the grade for the individual effort I put into this unit.

Week 1: Mobile Apps Development and Market Research

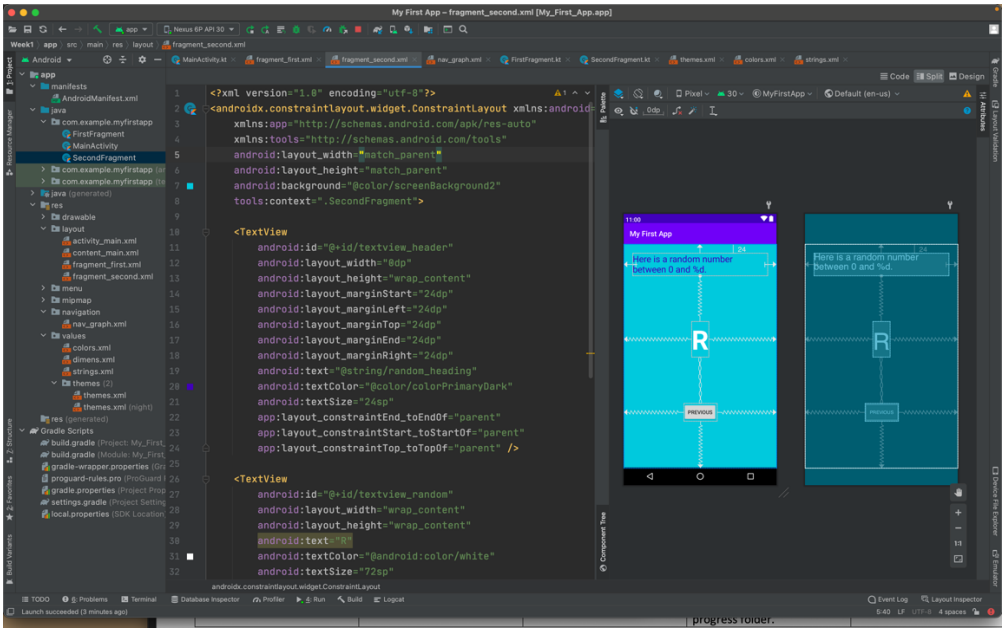
1. Notes:

- I did market research by reading market research articles from the weekly resources and other websites. I then used what I learned to write a project proposal for my SocialCollab application.
- Practicing and gaining an understanding of mobile app characteristics and development challenges is key.
- Set up Android Studio and learn how to get started with a new project in it.
- An in-depth look at Android frameworks like Kotlin and Java, as well as app components like UI design and development.

2. Exercise:

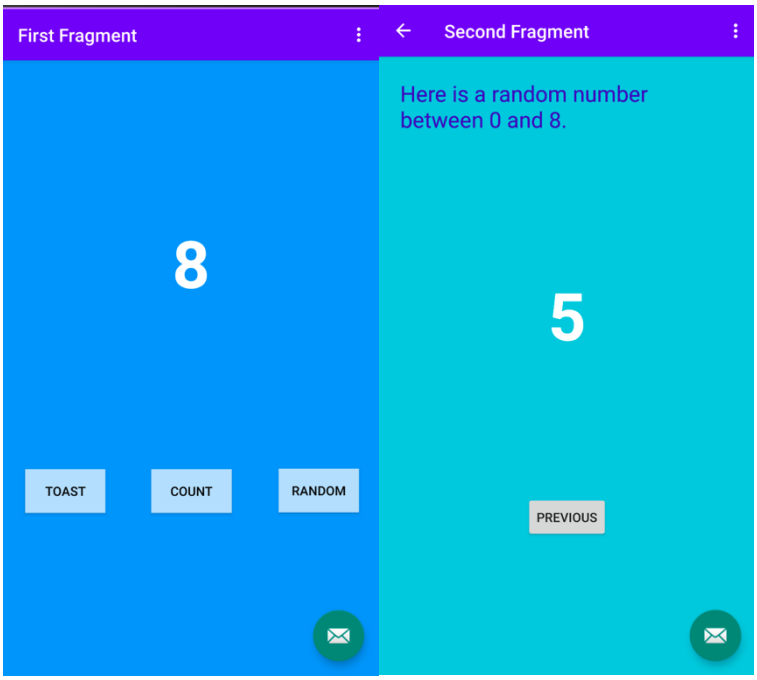
- **Functionality of app:** There are 2 fragments which have different functionality. The first fragment has 3 buttons, such as:
  - Toast button: pops up a message on the screen, which clicks on this button.
  - Count button: updates the number on the screen when click on this button.
  - When click on this button, it will display a random number between 0 and the current count on a second fragment when click on it.

- **Screenshots**: Installation of Android Studio and creating a first project with an output.



Screenshot Of Activity Layout

Outputs:



Screenshot Of First And Second Fragments

3. Usability in SocialCollab app:

- Setting and getting property values for resources such as colour and string resources is necessary.
- Designing the layout of the screen and navigating between screens is useful.
- It is essential to be able to navigate between screens and design the screen layout.
- Button design and on-click implementation also allow for passing information to other screens.

4. Summary:

- Create a new Kotlin or Java project and learn about the project's structure and layout.
- Modify the properties and attributes of the textboxes and buttons from the XML file and design views such as color, constraints, background, etc.
- button-click functionality and screen-to-screen navigation.
- Create toast on button click to display a message and the usability of random functions.
- Run the project on a virtual device (emulator).

Week 2: UI and UX Design Wireframes and Storyboards Proposal Draft

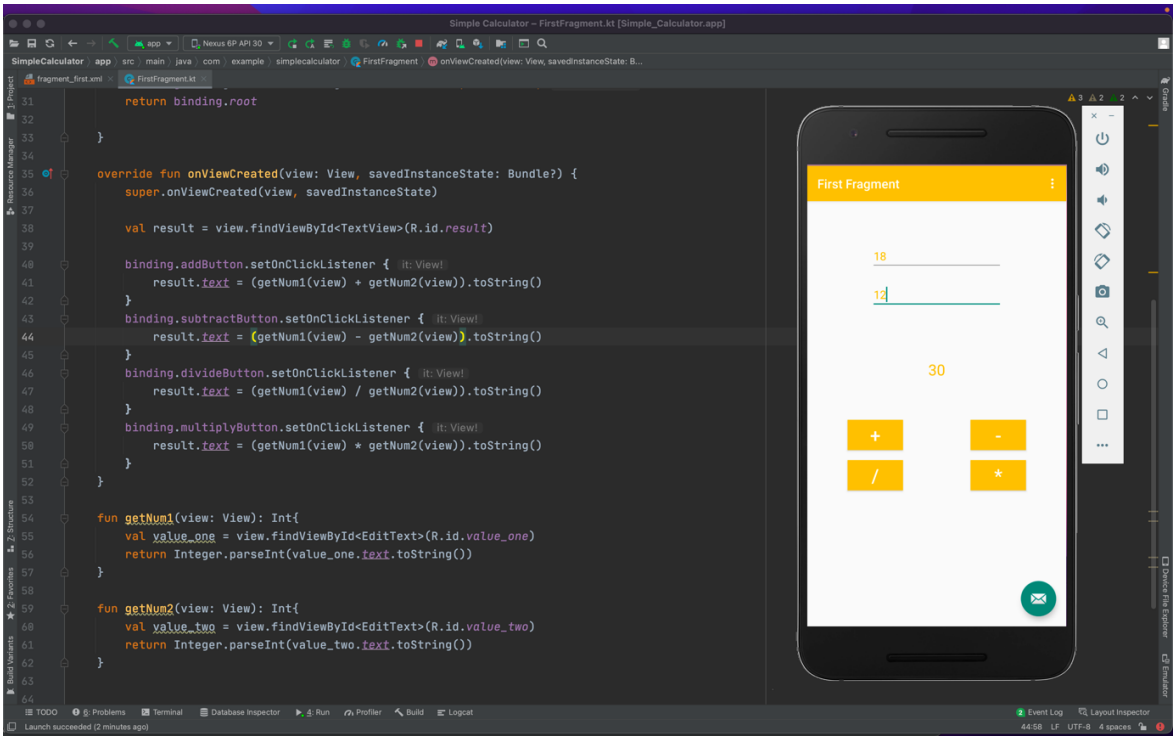
1. Notes

- Sprints in agile methodology development
- Responsive design with consideration of device dimensions
- Develop user stories with a sample.
- With a sample of a client proposal, it can help better understand the client document.
- Provide tools for UX/UI and wireframe design, such as Sketch on Paper, Figma, Adobe XD, Wireframe.cc, etc.
- There are 3 types of prototypes, including: Low Fidelity Prototype, Medium Fidelity Prototype, and High Fidelity Prototype.

2. Exercise

- **Functionality**: In this week's activity, Kotlin is used to create a simple calculator. The two integers should be used to perform basic arithmetic calculations, including addition, subtraction, multiplication, and division.

- **Screenshot**:



Week 2 Activity - Simple Calculator

3. Usability in SocialCollab App

- Identify and analyze the target audience.
- Create user stories for the SocialCollab app.
- Use FIgma to design a Low-Level Fidelity to High-Level Fidelity prototype.
- To combine multiple input controls on a screen into a chain.

4. Summary

- Wrote a user story and created a prototype using Figma.
- Ensure the responsiveness of the mobile application and adjust the input controls' constraints properly.
- Worked on an XML layout file, manually modifying the attributes and properties.
- Using the onSetClickListerner event, I was able to call several tasks using the arithmetic operation functions.

Week 3: Java/Kotlin Bootcamp and Portfolio of Project

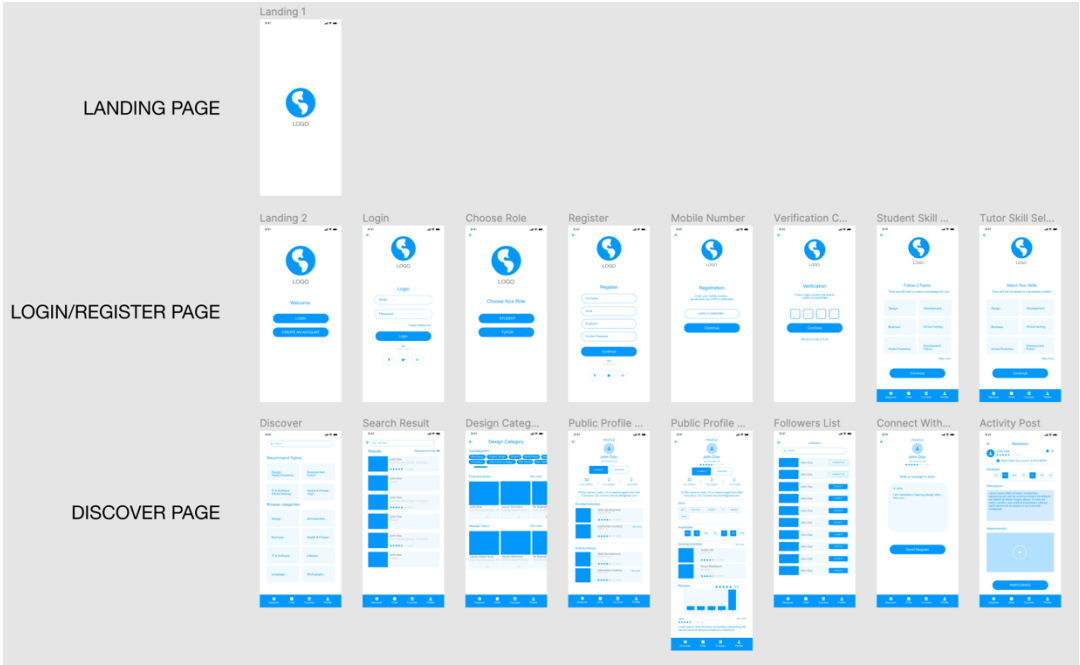
1. Notes

- An introduction to the open source Kotlin programming language, which is built on the JVM and is used by JetBrains internally.
- If compare Kotlin to Java, it's called "better Java" because of its compact and modern programming language and null safety.
- The syntax of Kotlin includes variable
    - and function names followed by data types.
    - There is no need for semicolons.
    - Functions start with the fun keyword
    - Default constructor in class signature
- Lambdas and data classes are a lot easier to build in Kotlin than in Java (a single line of code).
- Kotlin has a smaller runtime and less code than Scala, Groovy, and Ceylon, making it perfect for Android development.

2. Activity

- Using Figma to create a Low-fidelity prototype of the SocialCollab App.
- Link's to Low-Fidelity design: https://www.figma.com/file/H8TOyYK1tQ7O3iheo5fyGL/Low-Fidelity-SocialCollab?node-id=0%3A1

Screenshot:



Screenshot Of Low-Fidelity Design Of Socialcollab App

3. Usability in SocialCollab App

- Low-fidelity (lo-fi) prototyping is a quick and effective technique for transforming conceptual design ideas into tangible and testable products.
- The primary purpose of lo-fi prototypes is to evaluate and test functionality, not the product's visual appearance. So, during development, modifications may be made to improve the application's functionality.

4. Summary

- Preparation of the Project Proposal (Assignment 1)
- Learned how to design the UX and UI of an application using Figma.
- Get familiar with Kotlin's and Java's frameworks before developing apps.
- Make use of problem solving and critical thinking throughout the app design stage.

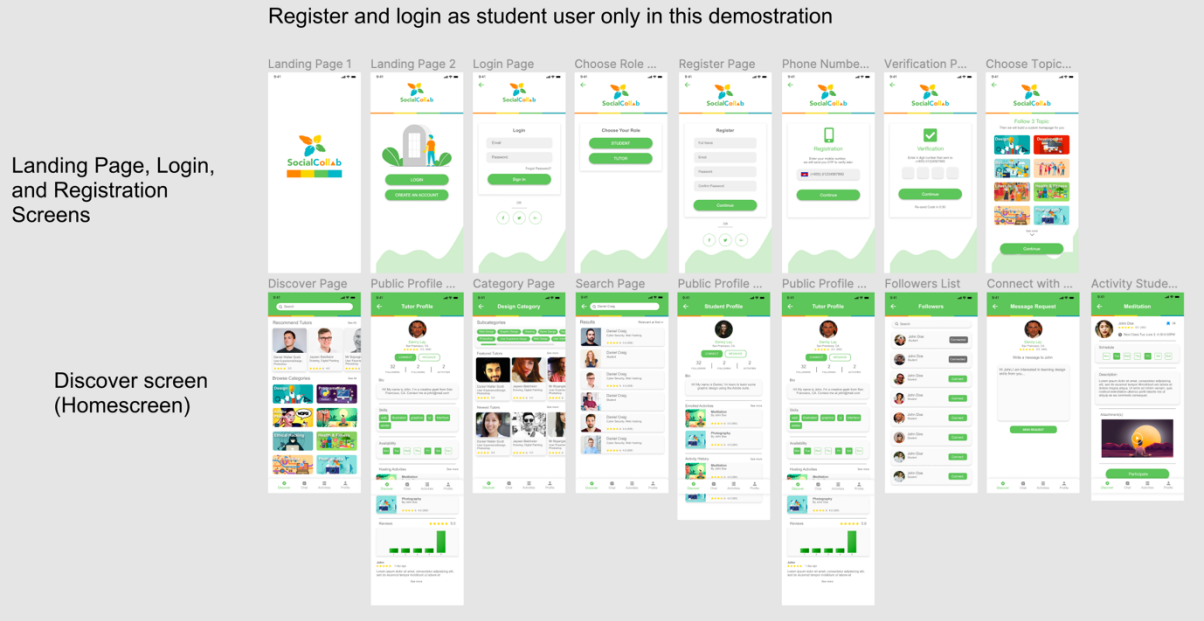
Week 4: Developing Android Apps, Layout and Navigation

1. Note

- The Android framework includes built-in services and add-ons such as a graphical user interface, operating system services, graphics, device access, Google Play Services, database support (SQLite), and WebKit/Chromium.
- It supports multiple tools:
    - IDE: Android Studio
    - Testing tools: JUnit, Espresso
    - Performance profiling tools: Android Profiler
    - Source code management: Git, Subversion, CVS, etc.
    - Software emulators: Android emulator, Intel HAXM, Genymotion
    - Sensor injection via emulator
- Android user interfaces are built from a hierarchy of View objects, including views, ViewGroup, and ViewGroup.LayoutParams
- There are many layout types, which are listed below:
    - Linear Layout
    - Absolute Layout
    - Table Layout
    - Frame Layout
    - Relative Layout
- Input controls are used to collect information from the user, such as
    - Text Field
    - Widgets: Button, Check box, RadioButton, Spinner
    - Dialogs
- An application always starts with a main activity, and that activity provides the user with a screen with which they can interact.
- Navigation is the act of going between screens of an app in order to perform tasks.
- There are three navigational directions:
    - Lateral navigation: moving between screens at the same level of hierarchy.
    - Forward navigation: moving between screens at consecutive levels of hierarchy, steps in a flow, or across an app.
    - Reverse navigation: moving backwards through screens either chronologically (within one app or across different apps) or hierarchically (within an app).

2. Activity

- Using Figma to create a High-fidelity prototype of the SocialCollab App.
- Link's to High-Fidelity design: https://www.figma.com/file/eCpQGCqm4VConE7FE75KQJ/High-Fidelity-SocialCollab?node-id=0%3A1



Screenshot Of High-Fidelity Design Of Socialcollab App

3. Usability in SocialCollab App

- High-fidelity designs are important to the app's development since it is both a creation of a concept and a demonstration of its functionality via a prototype.
- Input controls such as TextFields and Buttons on registration and login forms are essential in the SocialCollab app.

4. Summary

- Preparation of the Project Proposal (Assignment 1)
- Submit the Project Proposal

Week 5: Activity Fragment Life Cycle, App Architecture and UI layer

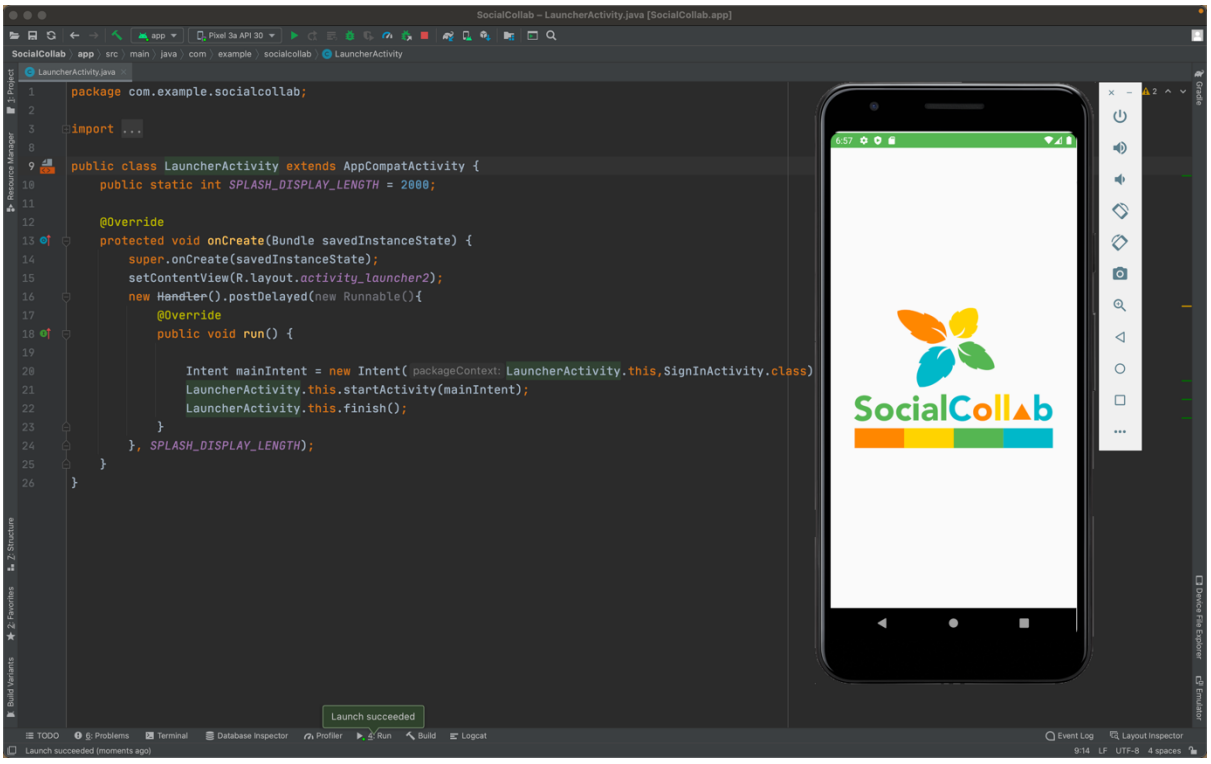
1. Notes:

- An Activity
  - represents a single screen with a user interface managed by the "Activity Manager".
  - An application can have one or more activities without any restrictions.
  - An activity class loads all the UI components using the XML file available in the *res/layout folder* of the project.
  - An activity can be created by the "Launcher" and also created by another activity or other SW components.
  - An activity may require permission to use resources of the device that are listed in the manifest file and granted at installation time.
  - **Process**: There is a process for each application to be executed.
  - **Tasks** are the set of activities that a user launches.
  - **Threads**: To execute the long-running operations.
  - **Back Stack**: When the current activity launches a new one, the new one is pushed to the top of the stack and takes control of the user's attention. In the back stack, the prior activity is moved to the bottom of the stack and stopped.
  - **State:** There are 3 states of an activity, such as Running, Pause, and Stopped.
- Lifecycle methods:
  - Describe the behavior of your activity when the user quits or re-enters the activity again.
  - Each application activity has its own lifecycle.
  - Lifecycle stages or callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy() ()
- Fragments:
  - A reusable portion of an application's user interface or behavior that may be used in an activity.
  - It defines and manages its own layout, has its own lifecycle, and is capable of handling its own input events.

2. Activity

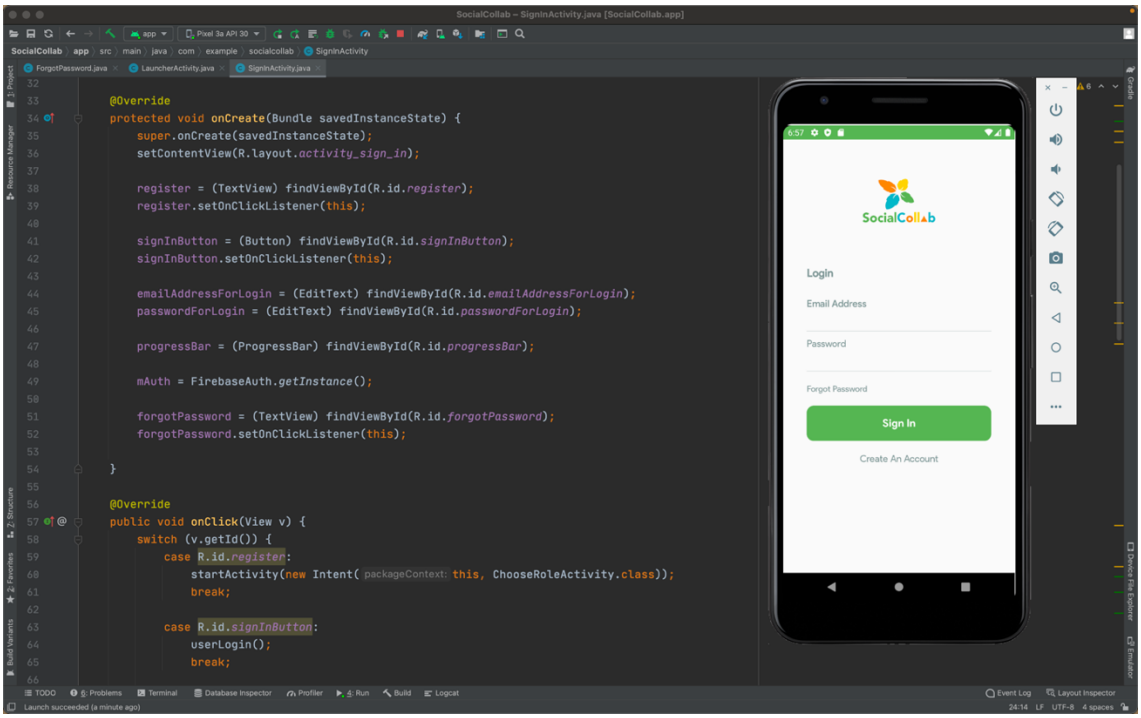
- **Functionality**:
  - All activities and fragments use a number of functions, including onCreate, onClick, and the layout XML file.
  - The bottom navigation menu has fragments of Discover, Chat, Activity, and Profile. The layout is defined in an XML file, and each event is represented by a fragment class.
  - Check the screenshots of the SocialCollab app below to view the launcher screen, login screen, sign up screen, mobile verification screen, and home fragment.

- **Screenshots:**
  - **Launcher Screen:** Display a SocialCollab Logo with an SPLASH_DISPLAY_LENGTH = 2000



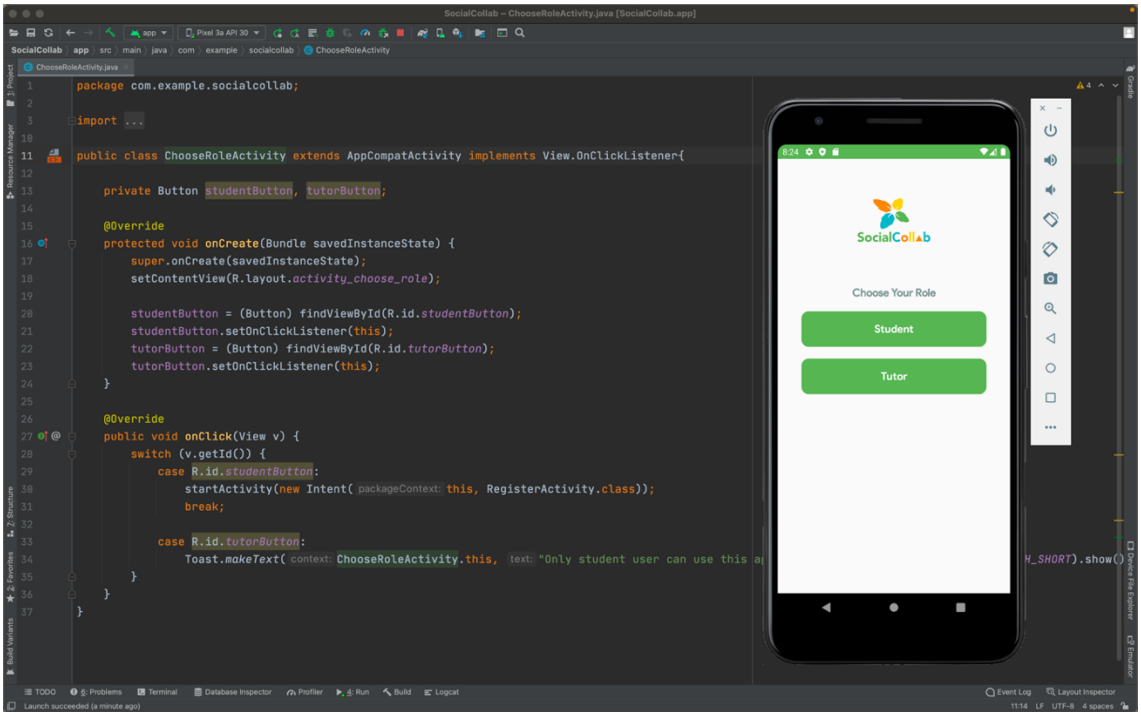
Screenshot 1: Launcher Screen

- **Login Screen:** User can sign in if they have registered already, or they can create a new account.

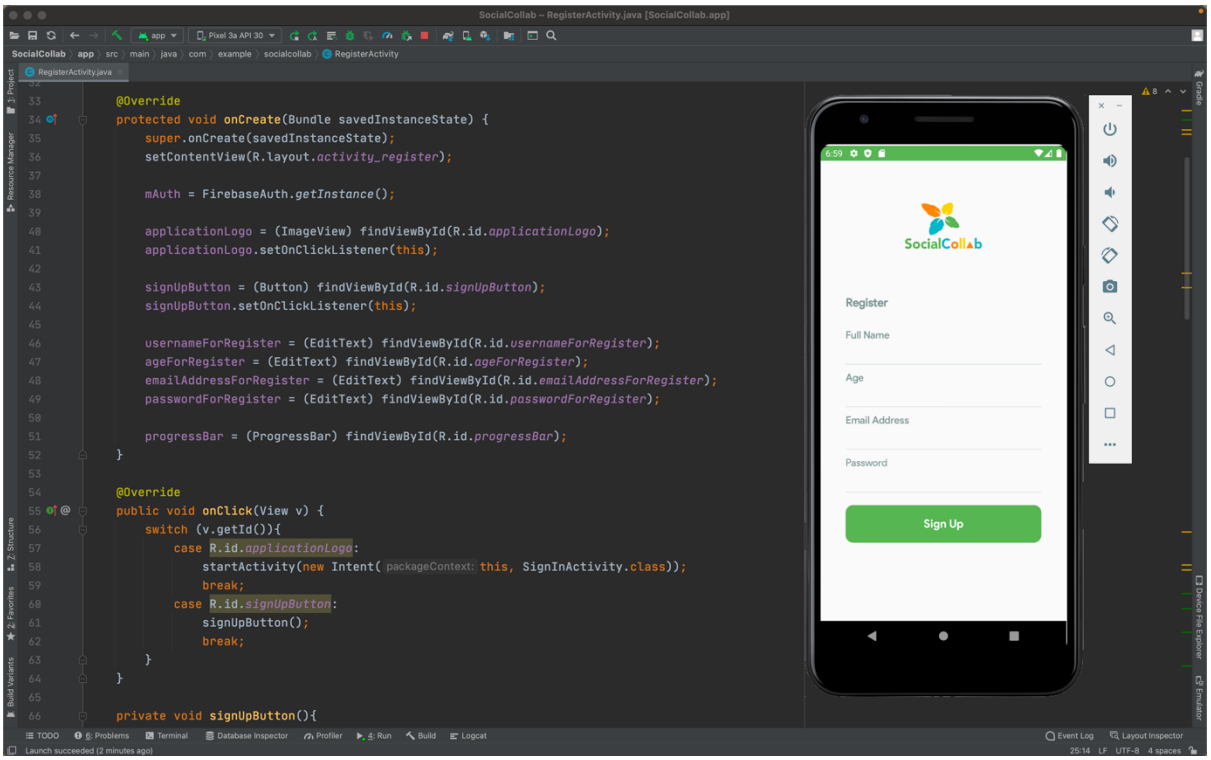


Screenshot 2: Login Screen

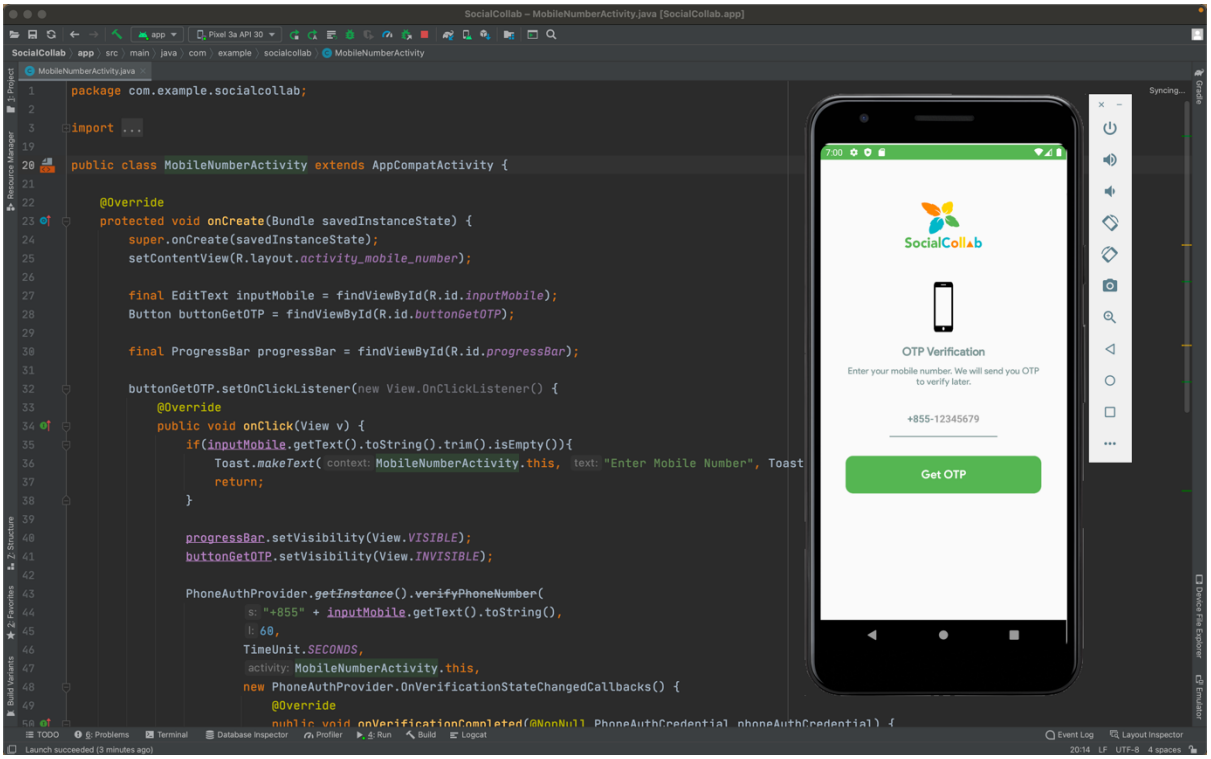
- **Register Screen:** A user must first choose between a student or tutor user and then fill out the registration form. After that, the user is required to verify their mobile number with an OTP code.



Screenshot 3: Choose Role Screen

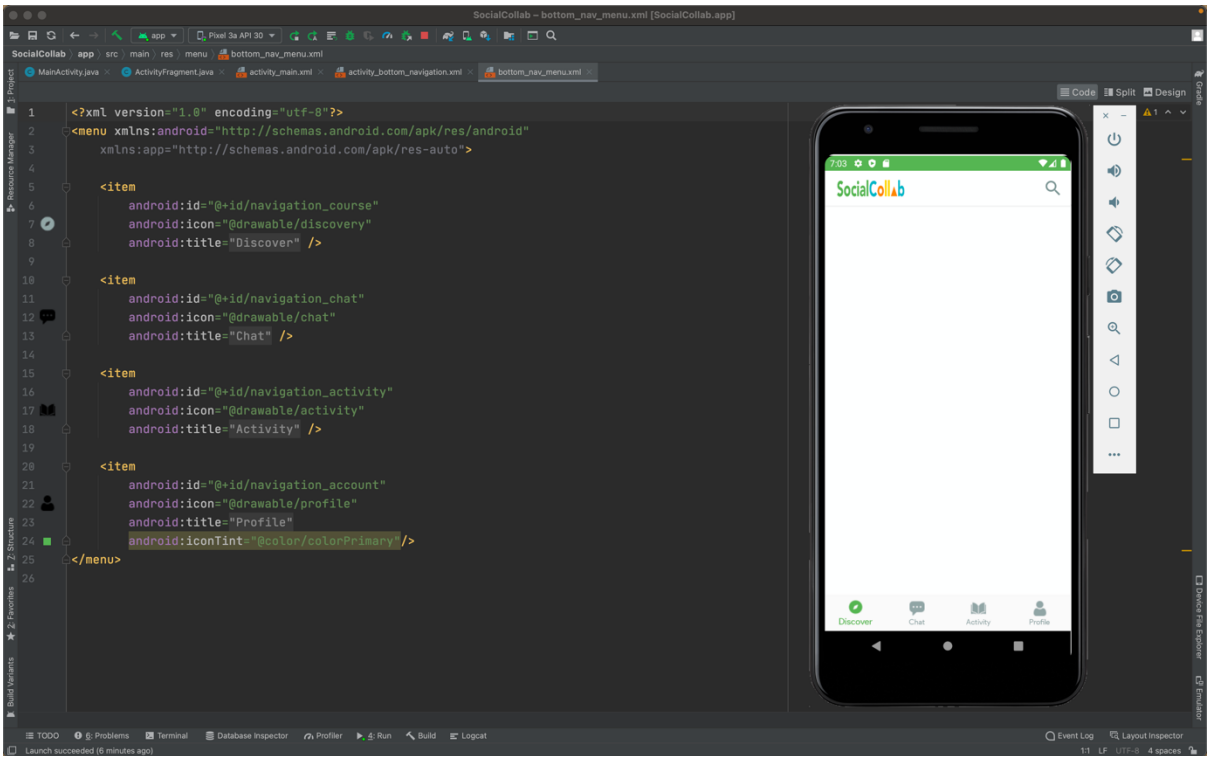


Screenshot 4: Register Screen



Screenshot 5: Mobile Verification Screen

- **Main Screen**: There are four fragments in the bottom navigation menu (Discover, Chat, Activity, and Profile)



Screenshot 6: Main Screen Or Discover Fragment

3. Usability in SocialCollab App

- Use activity to provide a screen for users to interact with the SocialCollap App.
- It is essential to use fragments on the bottom navigation menu screens. The project's structure will be well-defined and manageable if we use fragments.
- Some assets have been added to the SocialCollab App, including a logo, a logo banner, an icon, forms, and a background image.

4. Project Progress

- This week, I started to work on the SocialCollab App using the Android Studio in the Java programming language.
- At first, I created a Launcher Activity with a Welcome Screen that included a logo banner.
- After that, there is a login screen where a user can either log in (if they only have an account) or create an account (if they are a new user).

- When a user clicks on "Forgot Password", an email with a reset password link will be sent to the user's email address.
- When a user clicks on "Create An Account", the user must choose between a student or a tutor user. (Only student user available at the moment)
- In the Register screen, there is a form for users to fill in and they are required to verify their mobile number by receiving an OTP code.
- After completing all the above steps, the main screen (discover fragment) will display with a bottom navigation menu and a logo with a search icon in the tool bar.

5. Summary

- Learned the difference between activities and fragments.
- Began working on the SocialCollab application.
- Created a fragment for the bottom navigation menu.
- Designed layout in XML files and created several screens, such as launcher screen, login screen, register screen, etc.

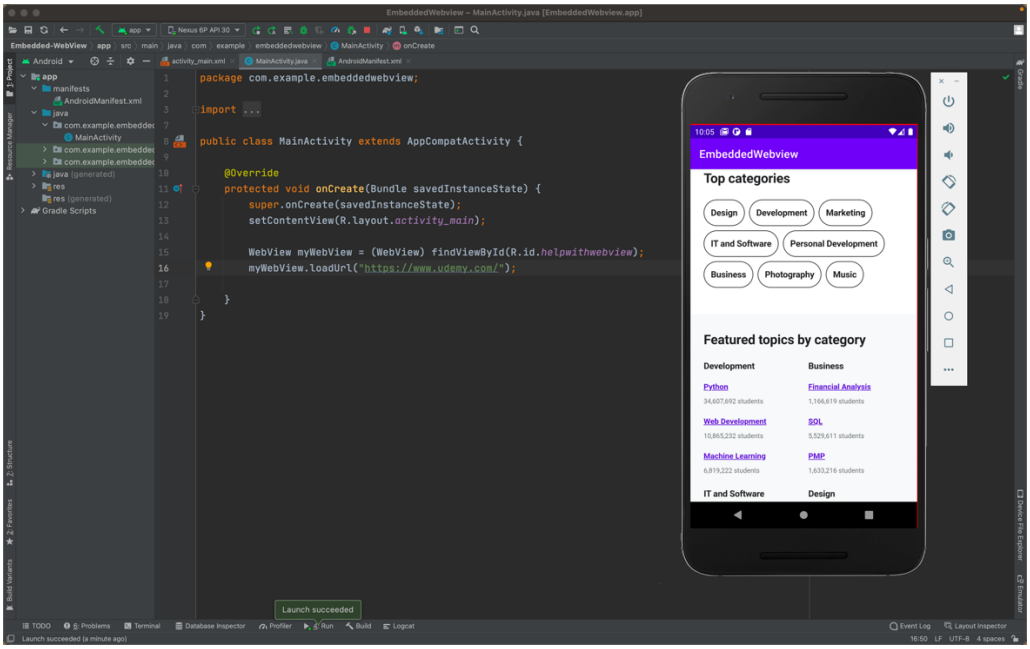
Week 6: RecyclerView and Internet Connection

1. Notes:

- Using Android WebView, Android applications may show content from the web directly inside an application. This is a system component of the Android OS.
- To embedded WebView to application:
    - Add a <WebView> element to activityís layout XML file or Add a WebView in OnCreate()
    - Use loadUrl() to load a web page
    - However, your app must be able to connect to the Internet in order for this to work. INTERNET permission must be requested in the manifest file to get internet access.
    - <uses-permission android:name="android.permission.INTERNET" />
- Using the Google Map API, we can access Google Maps servers, download data, display the map, and respond to map gestures are all handled automatically by the API. The API calls can be used to add markers, display a user's location, and offer interactive elements for the map.
- To integrate Google map into the application:
    - Register your app using an API key obtained from the Google API Console.
    - Embedded Google Map to the application.
    - Customize the map with markers, styling, and location tracking.

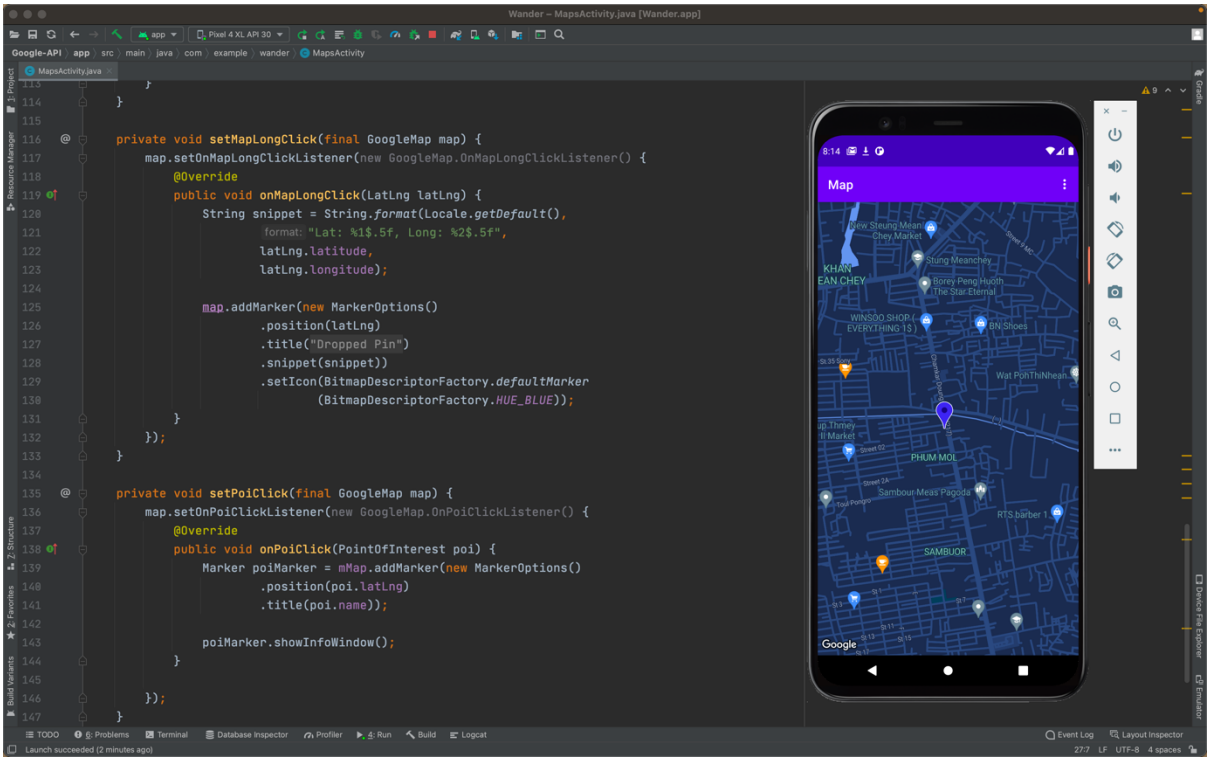
2. Activity:

- The list of categories available on Udemy's website is displayed below using a WebView that enables users to see all of the topics.



Week 6 Activity - Embedded WebView

- Move the default map location to the current location:
    - User permission is required to show the map on screen.
    - Create a new LatLng object to use the coordinates found on Google Maps.
    - Add map markers to show the marker at the current location.



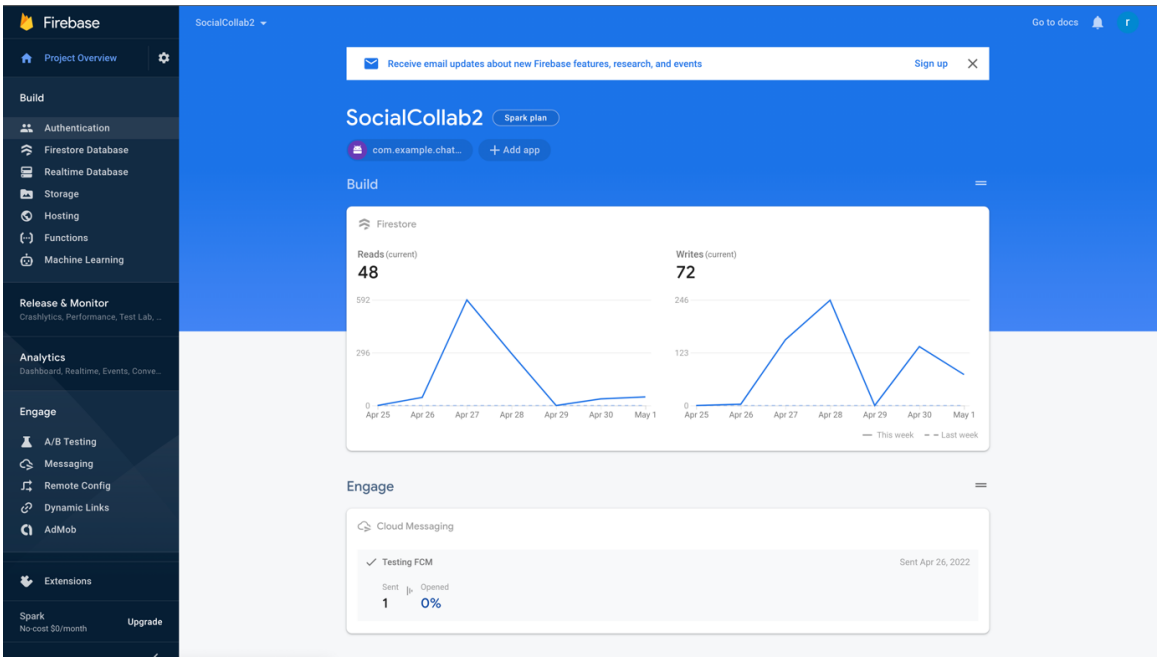
Week 6 Activity - Tracking Location Using Google Map API

3. Usability in SocialCollab App:

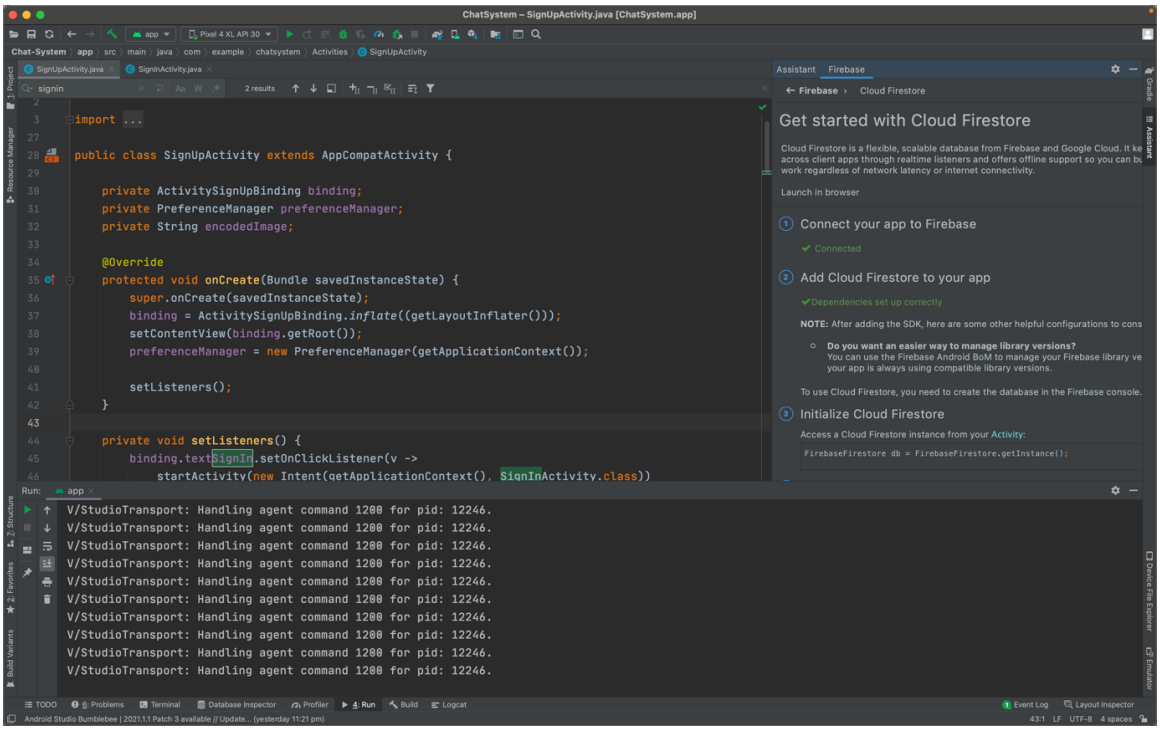
- Using WebView in SocialCollab to display a skill activity module such as videos, quizzes, etc., that has been given by the tutor.
- Using Google Map API, can be used in SocialCollab to show the users' locations.

4. Project Progress:

- This week, I have set up Firebase Cloud Messaging and Cloud Firestore Database in SocialCollab Application using firebase to store user information, make an OTP verification and implement chat feature.

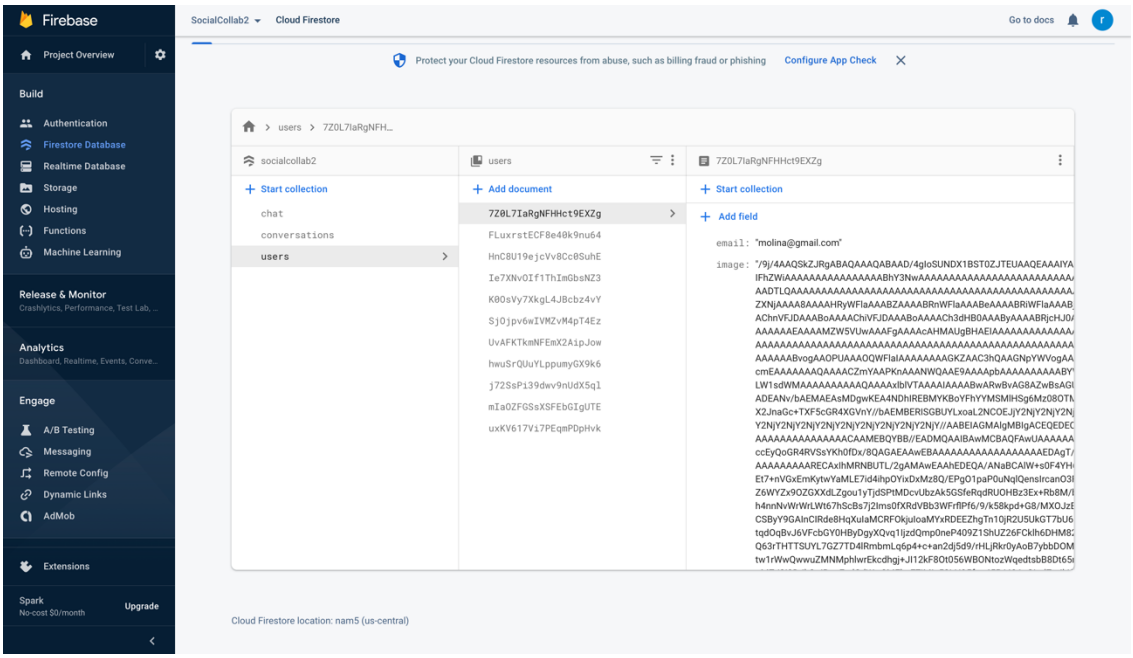


Firebase Overview



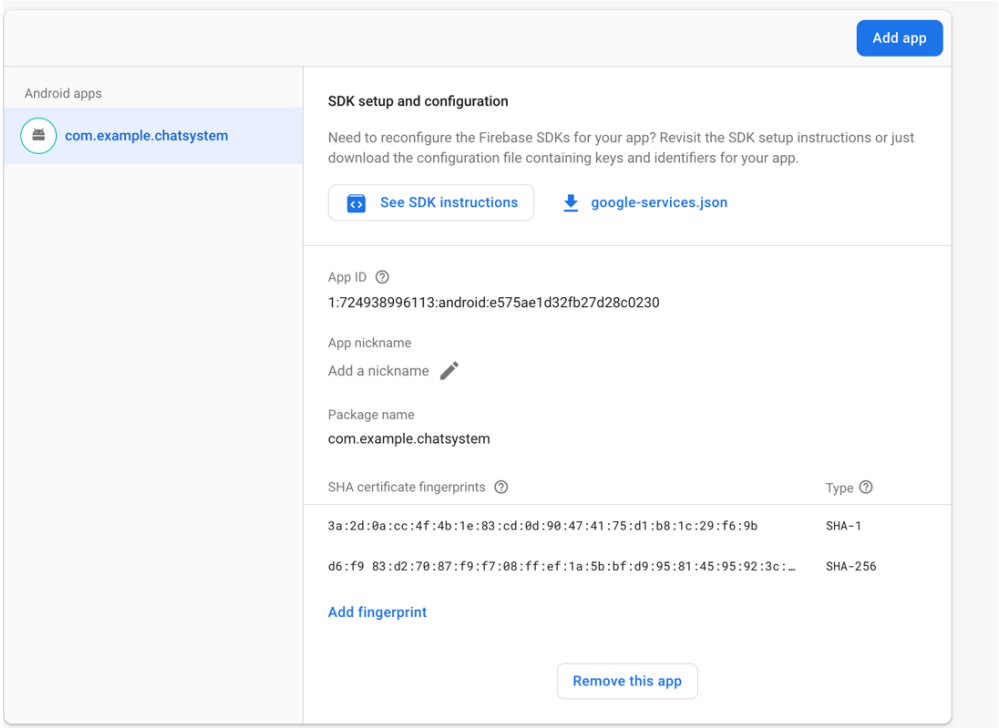
Connect The Project App With The Cloud Firestore

- Completed the sign-up and sign-in process using the Firestore Database, including:
    - Insert data in the Firestore Database from a device.
    - Input field validations
    - Use SharedPreferences to keep some local data.
    - Handle and perform queries in the Firestore Database.
    - Retrieve data from SharedPreferences for automatic sign-in.

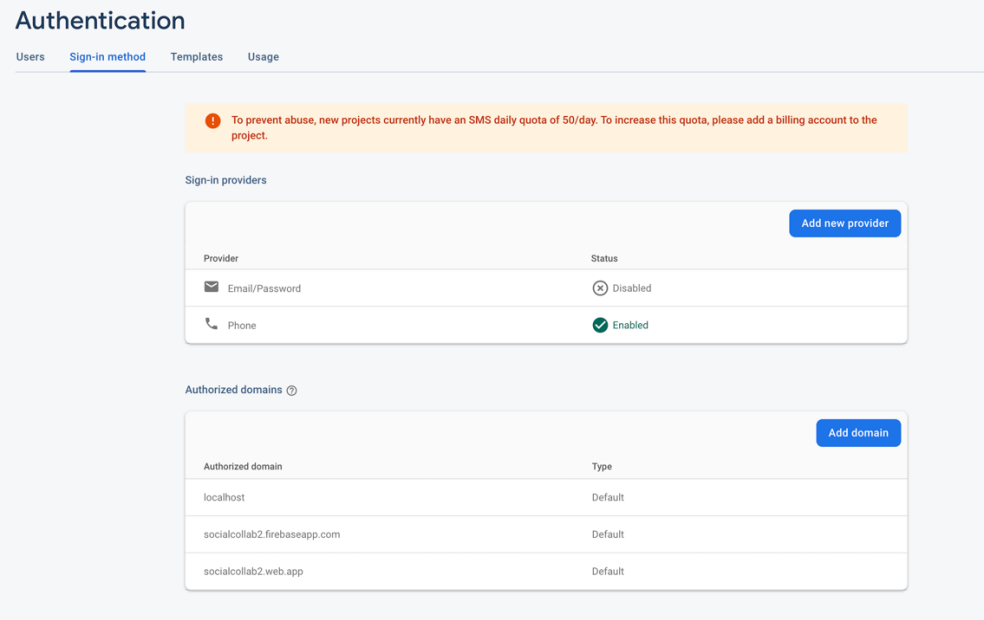


User Information Store On Firebase Database

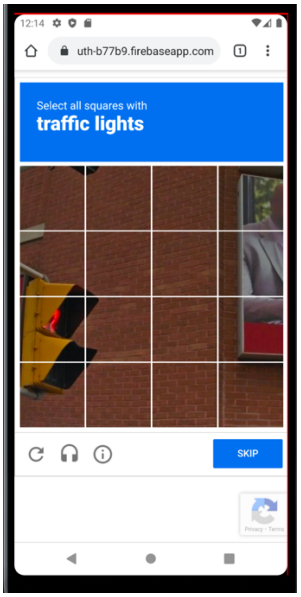
- Create an OTP verification feature using Firebase, including:
  - Add SHA Certificate Fingerprints
  - Enable the Phone sign-in method using Authentication
  - Since phone number sign-in requires a physical device, it won't work on an emulator.
  - Instead of phone verification, I switched to reCAPTCHA checks (verifying you are not a robot)



SHA Certificate Fingerprint



Enable Phone Sign-In Method Using Authentication



reCAPTCHA Verification

5. Summary:

- Implemented WebView and the Google Map API as part of weekly activities.
- Connected Firebase Firestore database with the SocialCollab project to allow the app to store user information when they have registered.
- Create an OTP Verification feature that allow the app to:
    - Send OTP
    - Verify OTP
    - Resend OTP

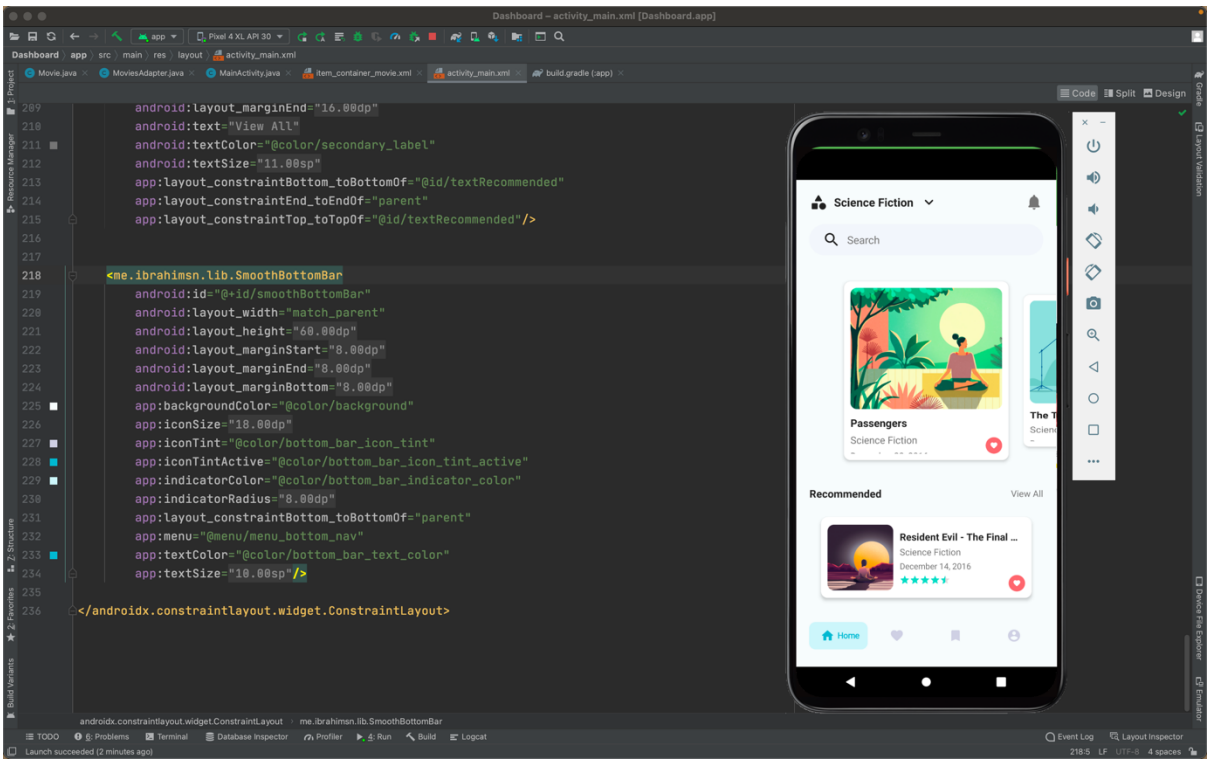
Week 7: Fragments and Libraries

1. Notes:

- Libraries can be used to implement functionality into an application with just a few lines of code, reducing the overall development time.
- There are several libraries for Android applications that can be created in a short period of time by following just a few steps by adding implementation link in dependencies in the gradle file and followed the instructions at readme file.

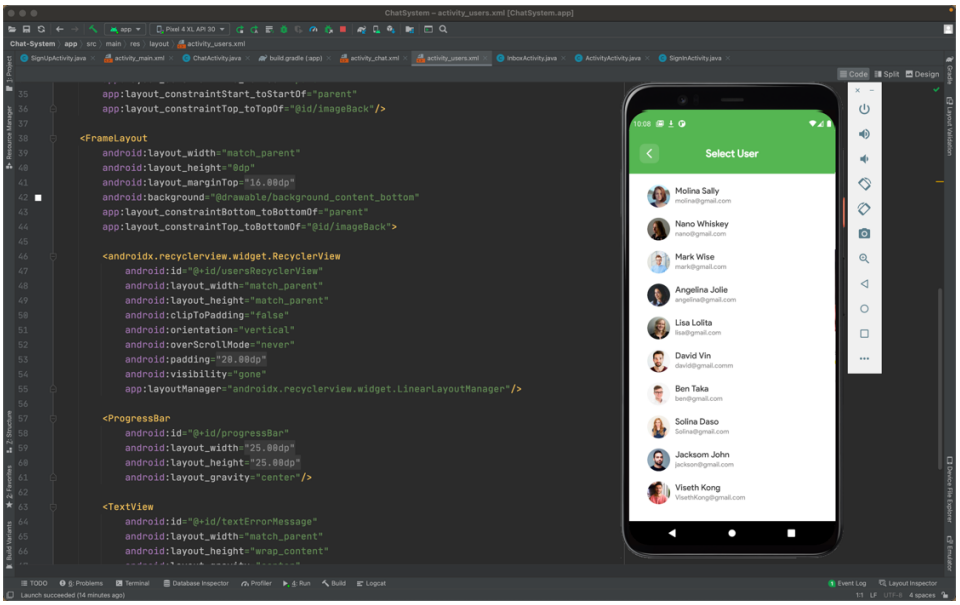
2. Activity:

- SmoothBottomBar:
    - Create a dashboard screen layout with the SmoothBottomBar which is described as “a lightweight Android material bottom navigation bar library.”



Week 7 Activity: Design Dashboard With Smoothbottombar Library

- RecyclerView:
    - Within SocialCollab app, display a list of user data from Firebase Database using RecyclerView and Adapter class.



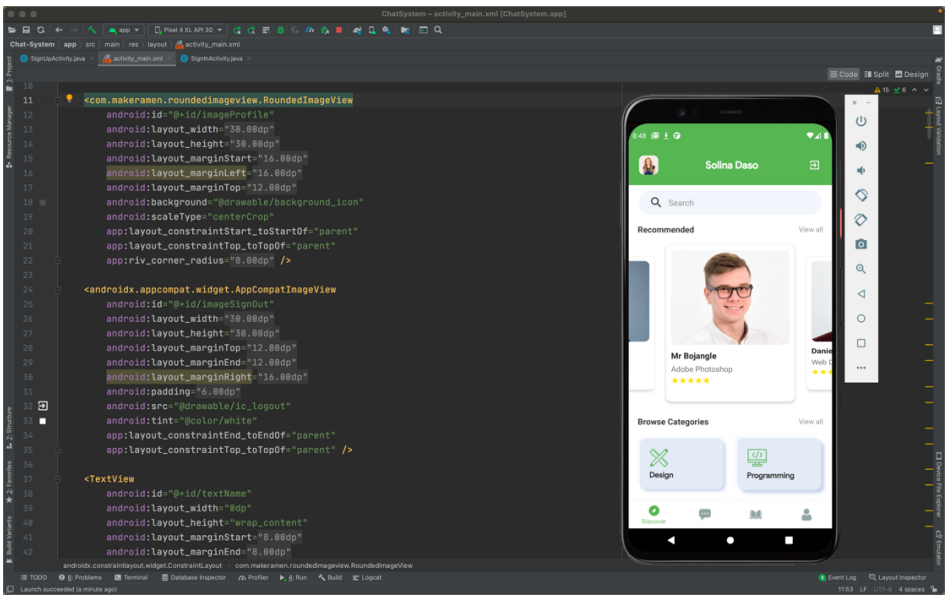
User Selection List In Socialcollab

3. Usability in SocialCollab App:

- SmoothBottomBar can be used in the SocialCollab app as it has a clean design which makes it easy for users to navigate through the app.
- RecyclerView is an important component for displaying a list of user information from the Firebase Database.

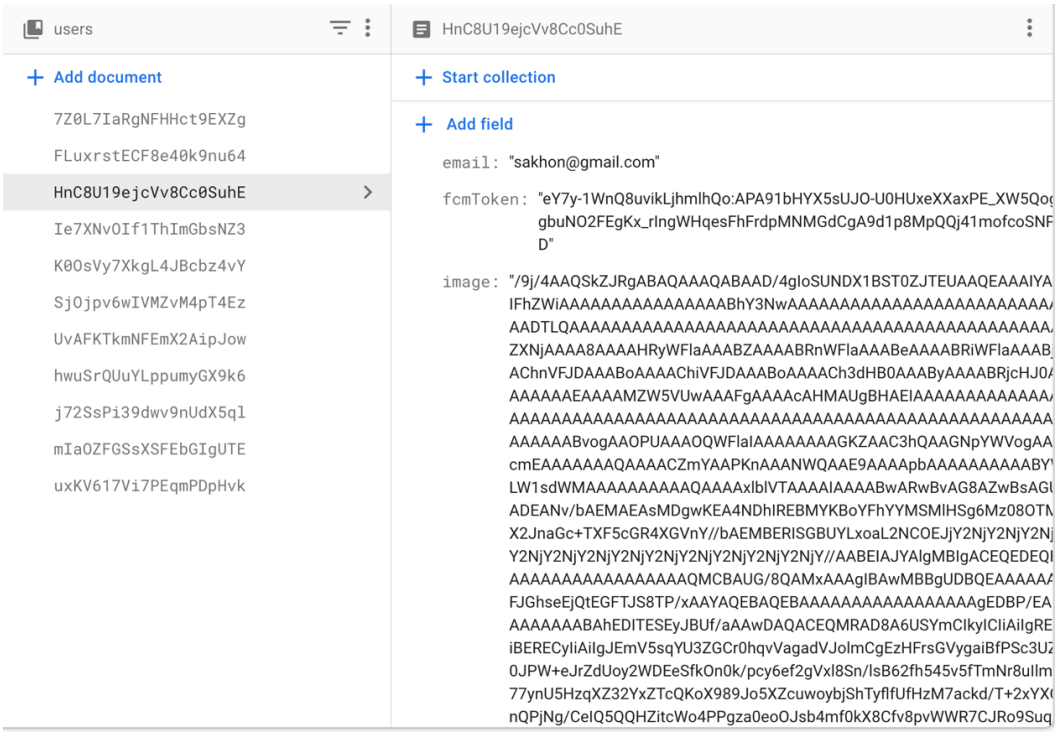
4. Project Progress:

- This week, I designed a Discover screen layout, worked with FCM (Firebase Cloud Messaging) Token and Sign-out Function, and implemented the chat system feature.
- Discover Screen Layout Design



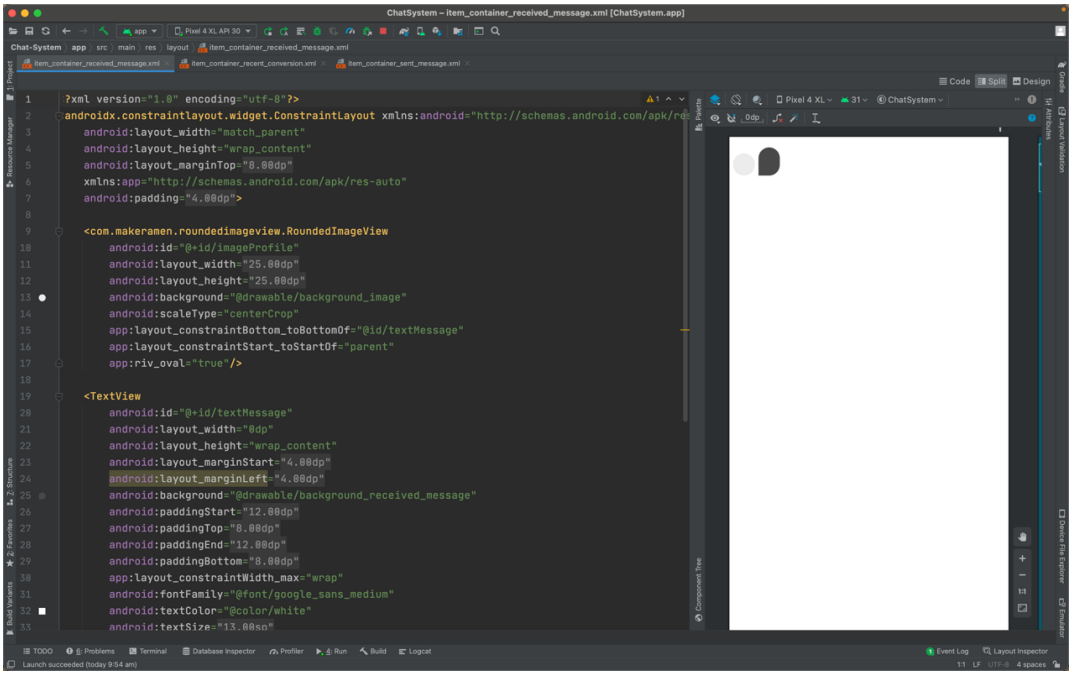
Dicover Screen Layout Design

- Update the FCM token of the user in the Firestore Database and remove it when the user clicks on the sign-out button:
    - Update or add new fields/data to existing records in the Firestore Database.
    - Delete field/data from a records in the Firestore Database.



FCM Updated When User Log In To The App

- Design Chat Screen Layout



Received And Sent Message Layout Design

5. Summary:

- Implemented a dashboard screen layout with Android Library such as SmoothBottomBar as part of the weekly activity.
- Designed Discover screen layout and used RecyclerView to display user list from Firestore Database.
- Worked on FCM token and sign-out function.
- Created layouts related to chat screen like chat screen, sent message layout, received message layout, etc.

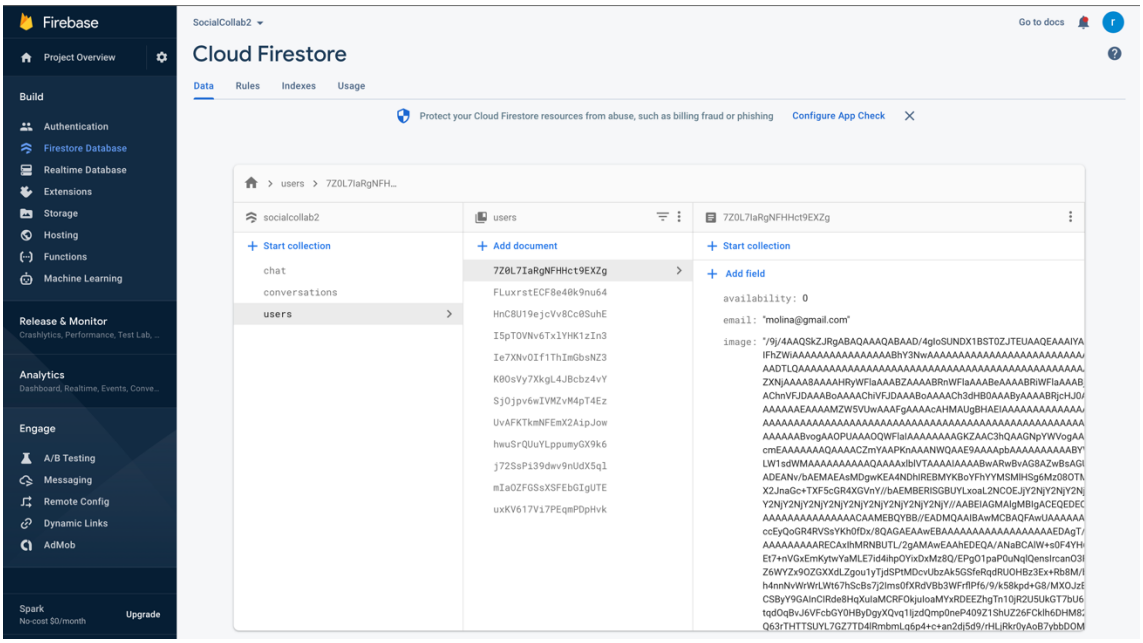
Week 8: Fire Cloud and Geofencing

1. Notes:

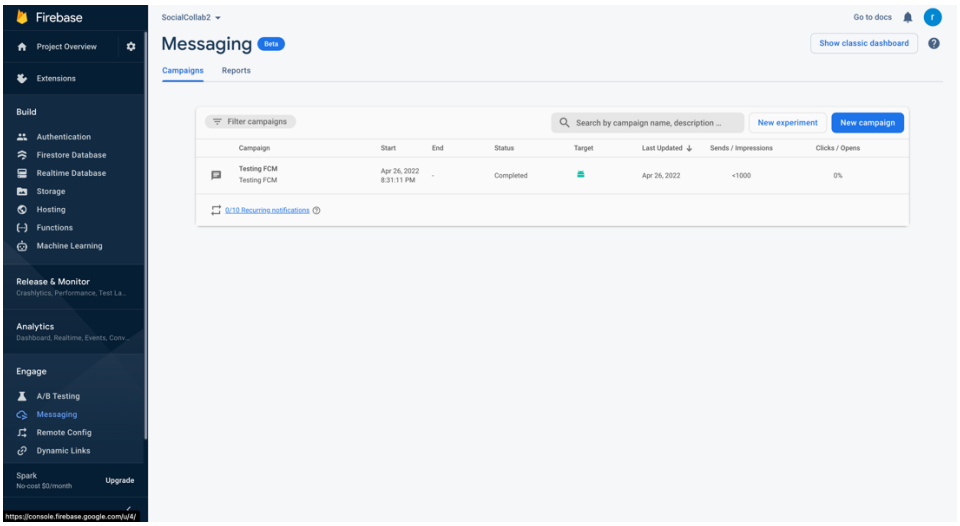
- Geofencing is a location-based service in which an application utilizes Wi-Fi or cellular data to perform a pre-programmed action. It represents the location's virtual area.
- It can be used in applications that require monitoring a certain area and getting alerts when people enter and leave that area.
- The Firebase Realtime Database is a NoSQL database that is hosted in the cloud and allows you to store and sync data between your users in real time.
- Firebase integration into an Android project is performed by first registering the application with Firebase and then adding the configuration file to the app's source code.

2. Activity:

- I have done some research on geofencing to apply it to the SocialCollab project, but it's not particularly necessary at this time.
- I have implemented and set up Firestore Database and Firebase Cloud Messaging to SocialCollab Android project since week 6.



Cloud Firestore Database



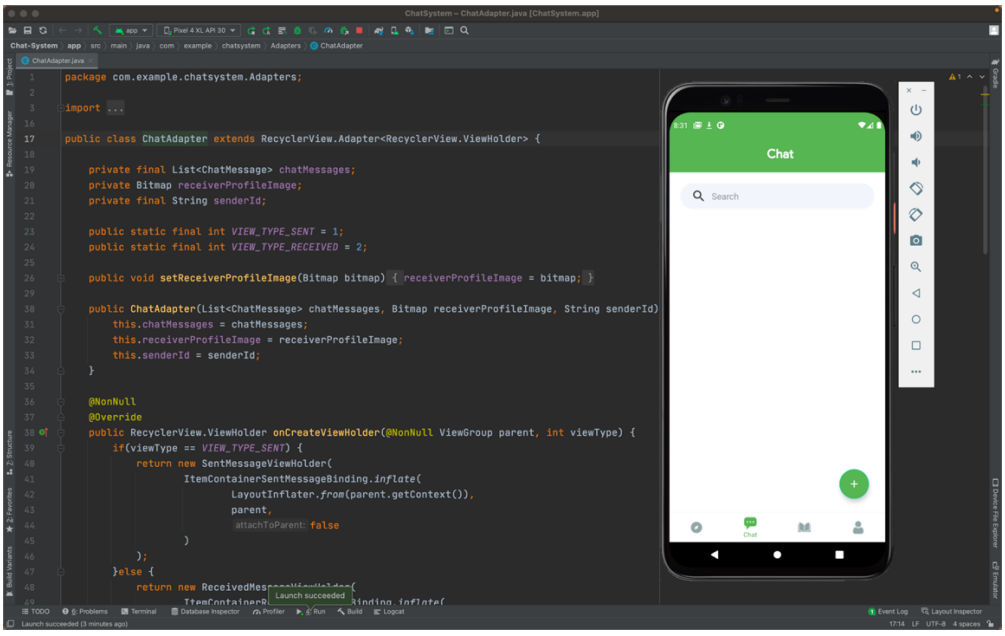
Testing FCM

3. Usability in SocialCollab App:

- For the SocialCollab project, Firebase is an ideal choice for storing user information, Real-time Database and Messaging, Cloud Functions, and Authentication.
- In the future, geofencing will be employed in the SocialCollab app to locate local tutors and students, allowing them to meet in person.

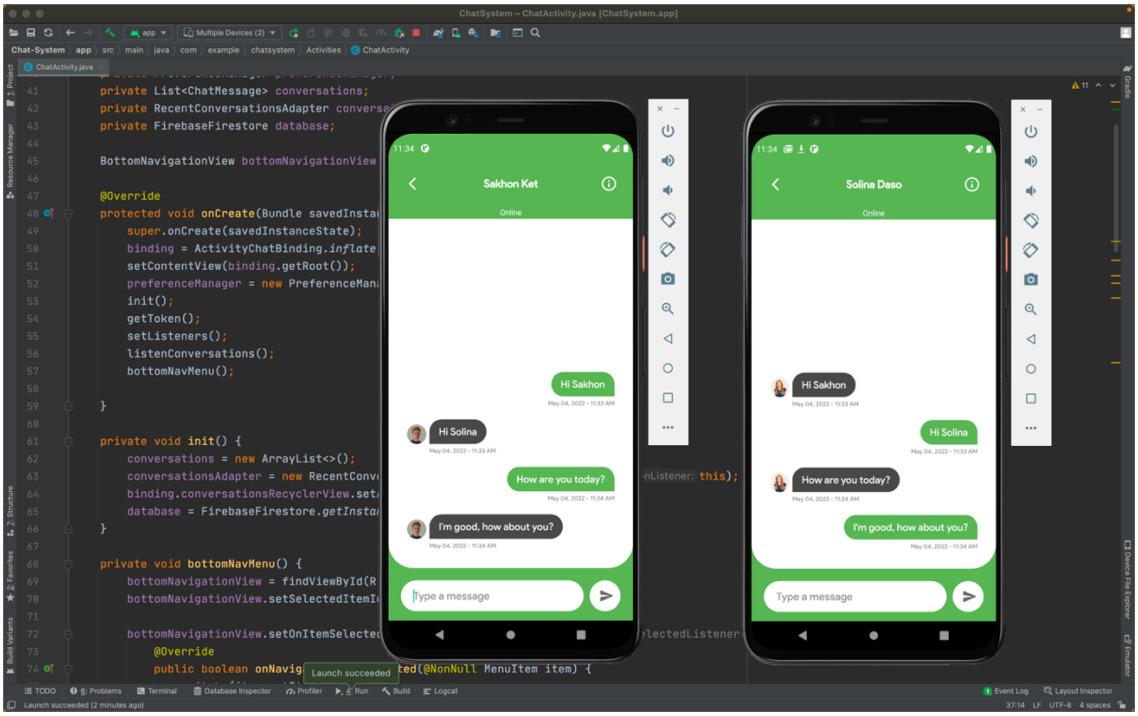
4. Project Progress:

- This week, I have continued working on the chat system features and modified some of the previous design layouts.
- Prepare data for real-time chat like models, layouts, chat adapters (RecyclerView's adapter), etc.



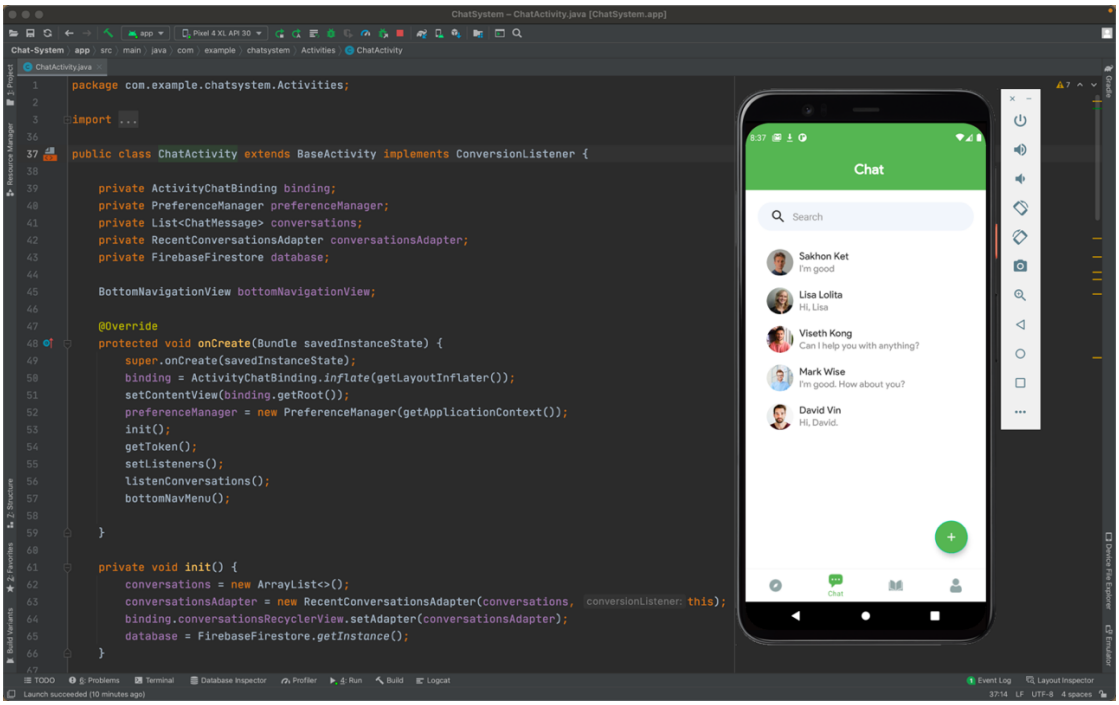
Chat Screen Layout Design

- Implement a real-time chat service that will have two primary features:
    - Send a message
    - Receive messages



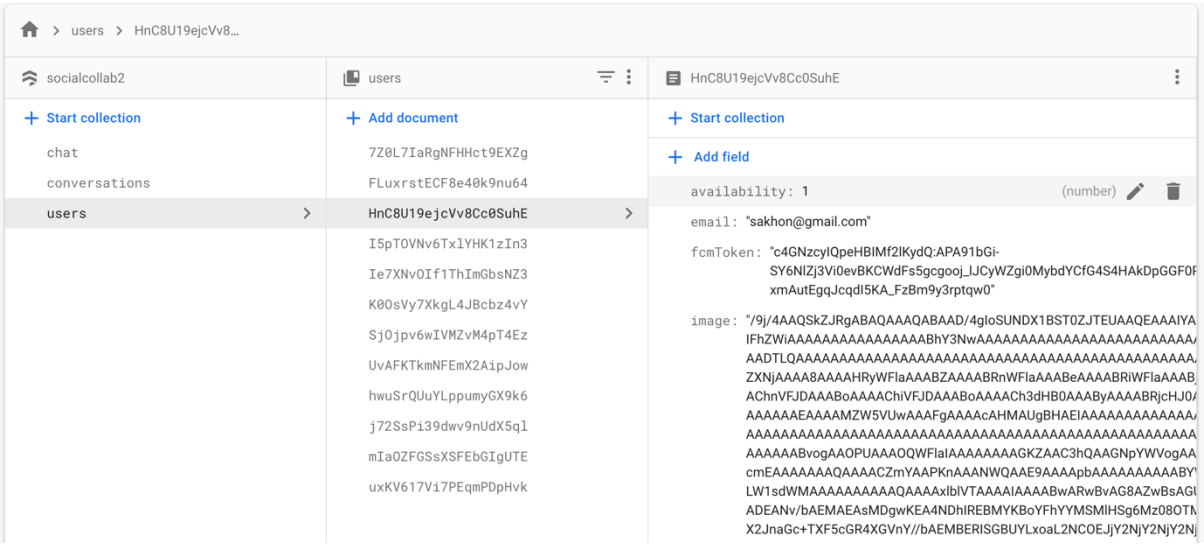
Real-time Chat Feature

- Implement a feature that allows users to see recent chats that has the following features:
    - Display the most recent message in a list of recent chats.
    - Real-time updating of recent conversations.
    - When a new message is received, the conversation will be moved to the top of the list.
    - Add a new conversation if there are no previous ones. Make sure the final message appears at the top of the list if the previous conversation is available.

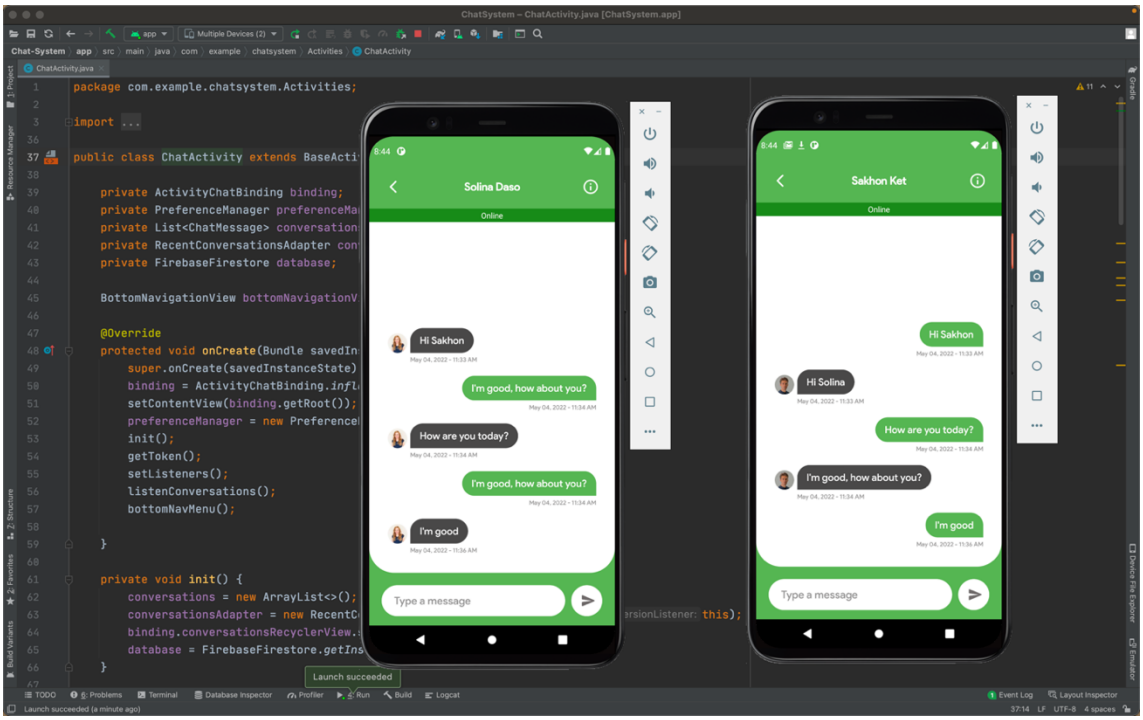


Show A List Of Recent Chat Message

- Add functionality to manage the availability of users, such as whether the person is online or not. It will send push alerts based on the availability status.

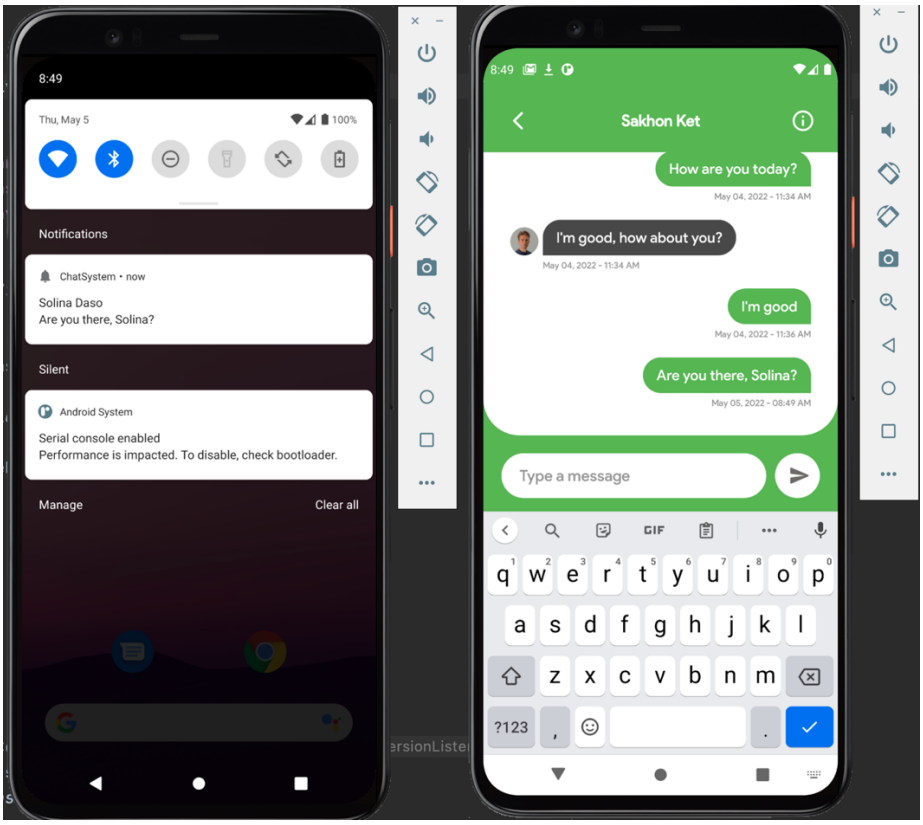


Availability Become 1 When User Is Online



Availability Status Of Online User

- Implement a system for delivering push notifications.
    - When the receiver is not online or reachable, it will send push notifications to him/her.



Push A Notification To Show Message When User Is Offline

5. Summary

- Researched about Geofencing and continued working on chat system features
- Prepared a chat data and implemented real-time chat features by showing a recent conversation, moving to the top when a new message arrives, etc.
- Added availability status when the user is online and pushed a notification when the receiver is not online/available.

Consider the following plans for the project to be completed in the next few weeks:

- According to SocialCollab's progress through the 8th week, several implementations have been completed, including the sign-up and sign-in processes, store user information on Firestore Database, and performed real-chat feature as mentioned above.
- I will be working on the following screens over the next several weeks:
    - Implement a video conferencing features
    - Create a list of instructors for each category
    - Add a list of Favorite Teachers to Bookmark
    - Add a public profile for instructors and students
    - Implement a search function and design a settings screen layout

Link to SocialCollab Android Project Folder:

Link to watch a video demonstration of SocialCollab App:

https://youtu.be/gYErv-WMU-E

Link to Github Folder which include all the weekly activities, assessments and other works that I have done in SIT708: