**Student Name : Aayush Raghuvanshi**

**Student ID : U61625730**

**Student Email : aayushr@bu.edu**

**Date : 11/26/2023**

**Professor Name : Michael Tizio**

**Department : Computer Science**

**University Name : Boston University**

**Github link :-** <https://github.com/CS683/project-aayush-raghuvanshi>

**CS683 Project Assignment**

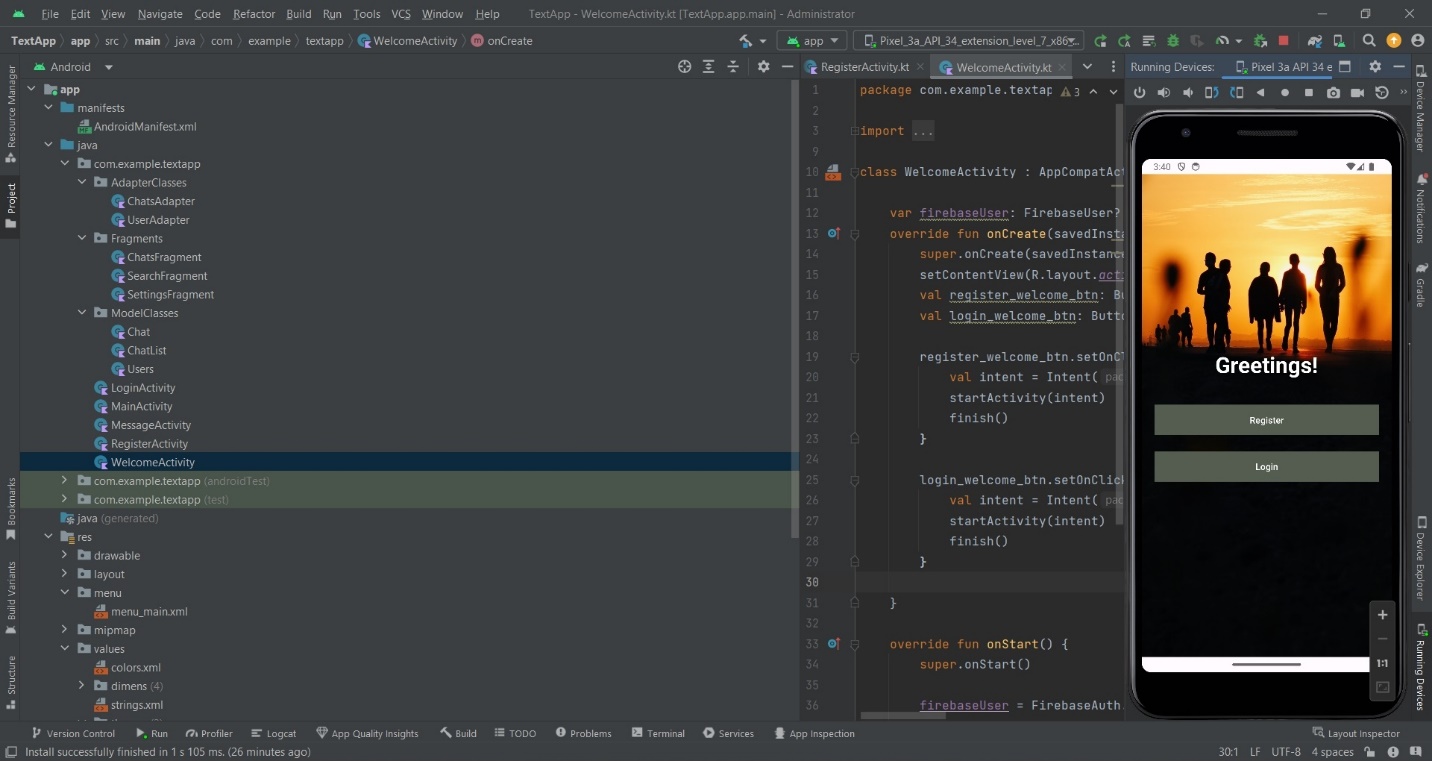
**Progress Report   
Texting App  
Aayush Raghuvanshi**

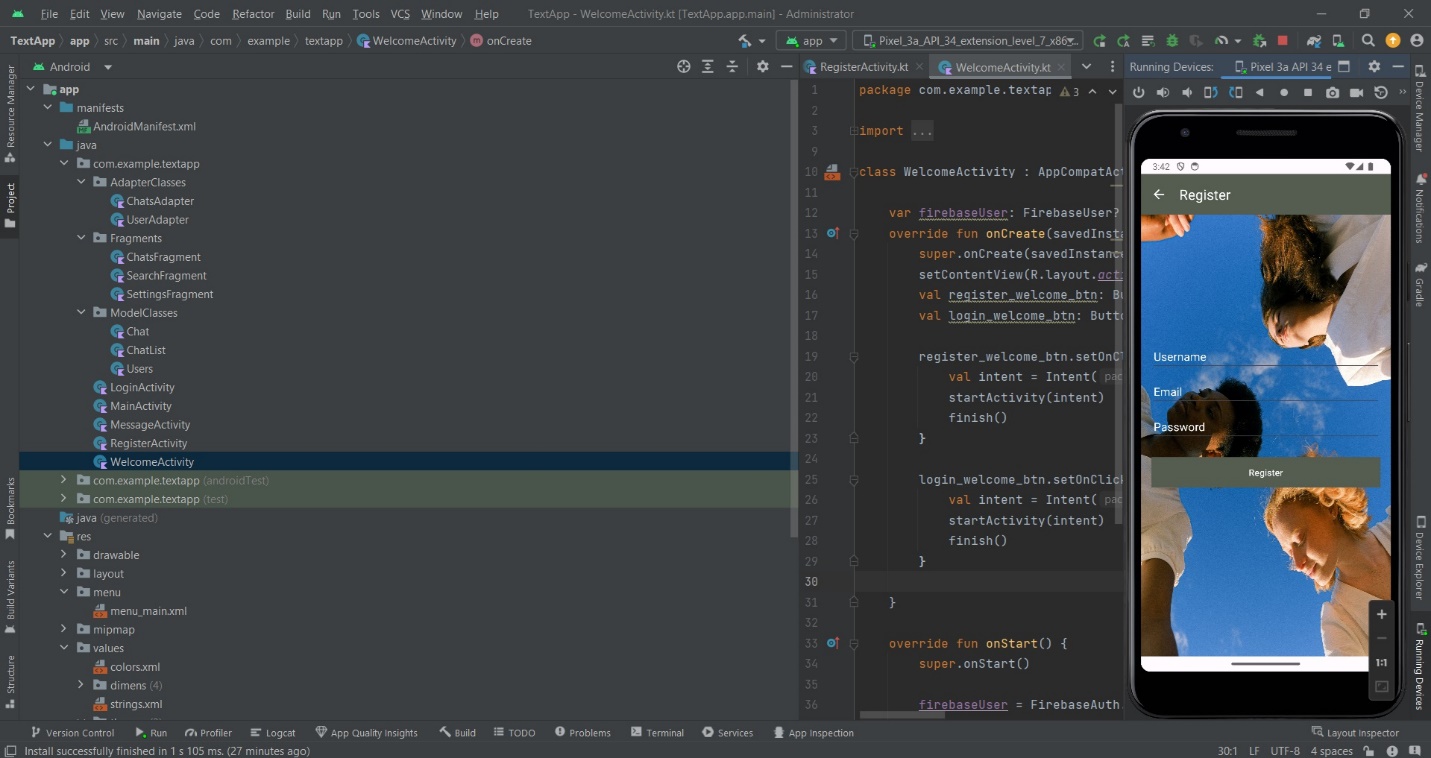
**Summary :**

* I, conclude my project development in this report. The Texting App project, an Android-based, feature-rich real-time messaging software. Users can communicate with one other on a smooth and user-friendly platform thanks to the application. Texting App, which was created with Kotlin and linked with Firebase services, lets users sign up, log in, and have private conversations. A wide range of capabilities are included in the project, such as image uploading, real-time database integration, user authentication, and profile management.
* **Key Features:**
* **User Authentication:**

Implemented user registration and login functionalities using Firebase Authentication.

Firebase Realtime Database utilized for storing and retrieving user data.

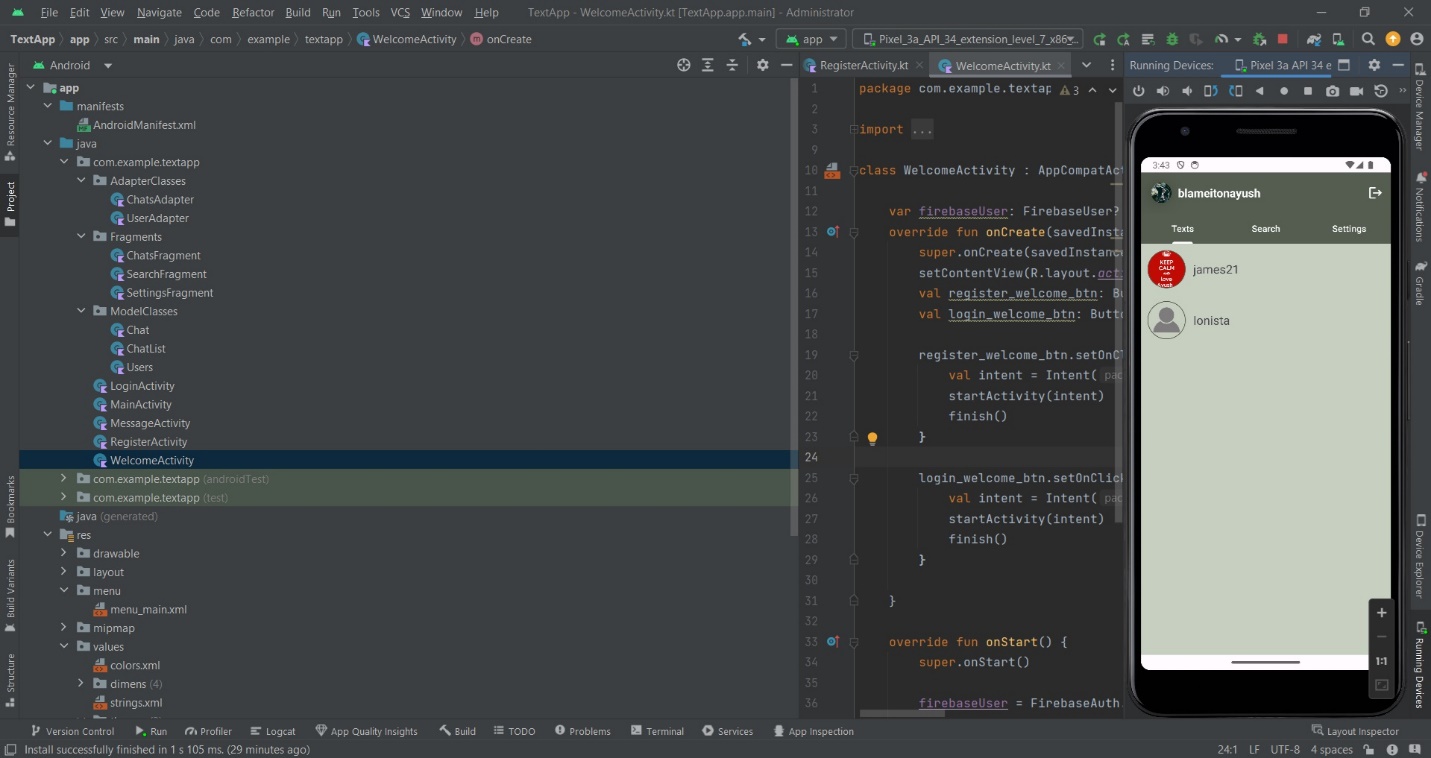


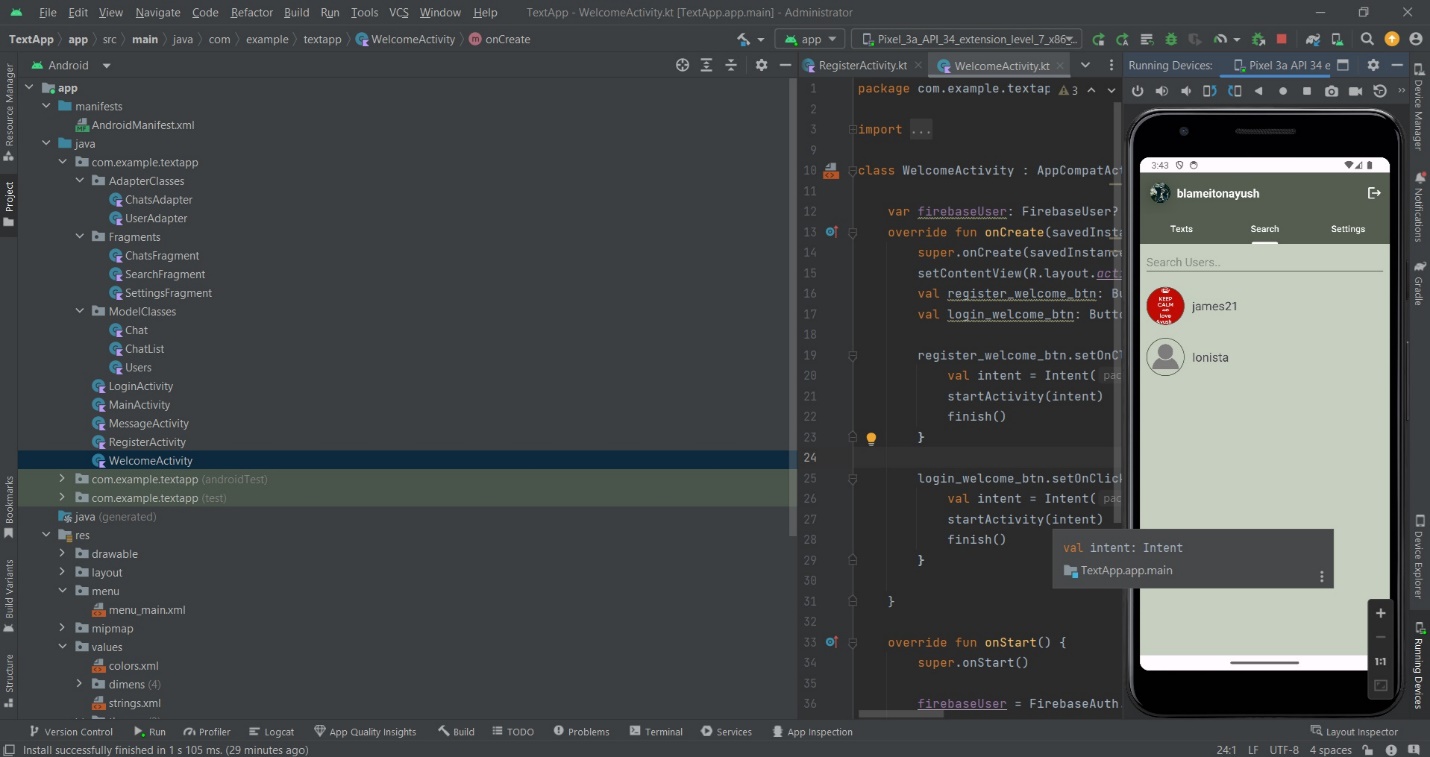


* **Main Activity:**

Core interface presenting a user-friendly dashboard with tabs for Chats, Search, and Settings.

Integrated ViewPager and FragmentPagerAdapter for smooth navigation.

****

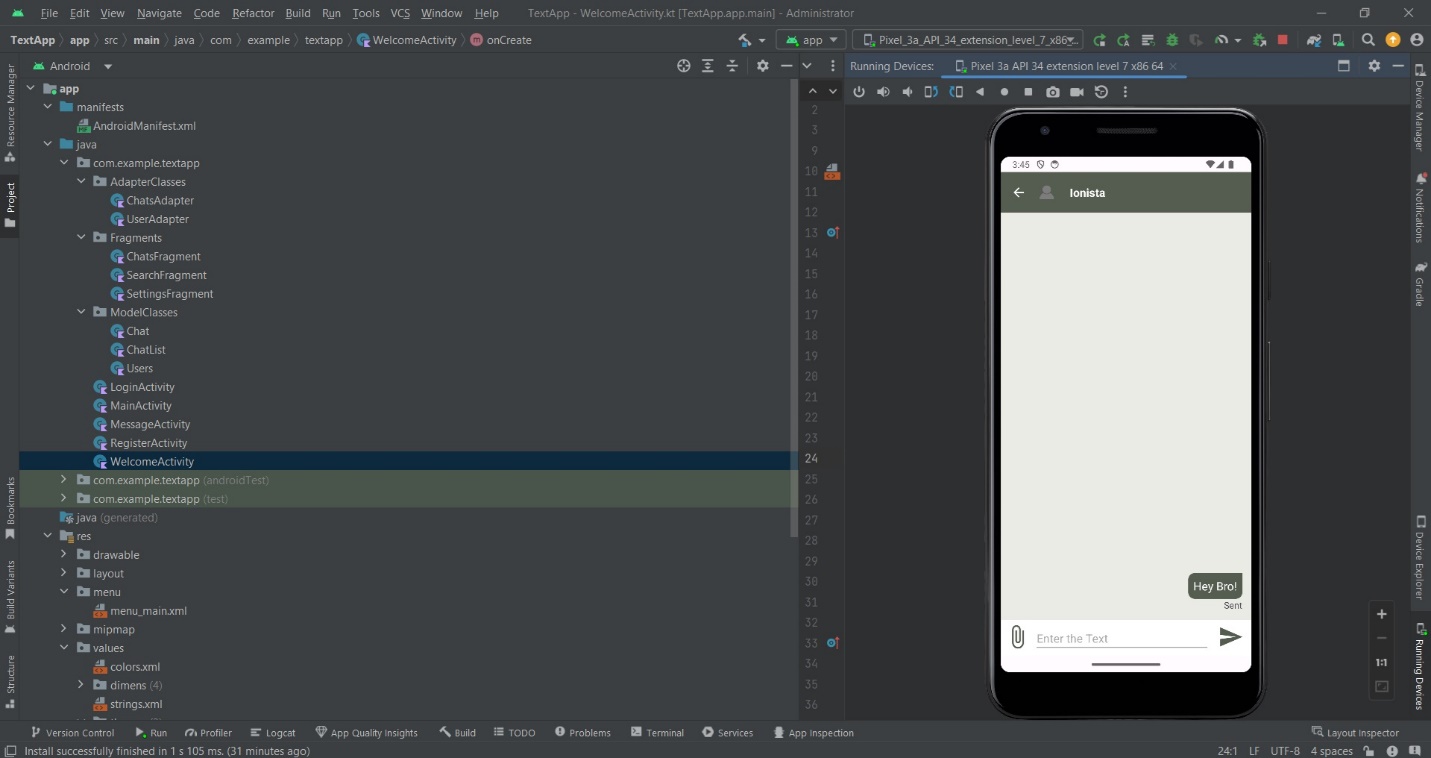
****

* **Chats Fragment:**

Real-time display of user chats.

Firebase Realtime Database used for fetching and displaying chat messages.

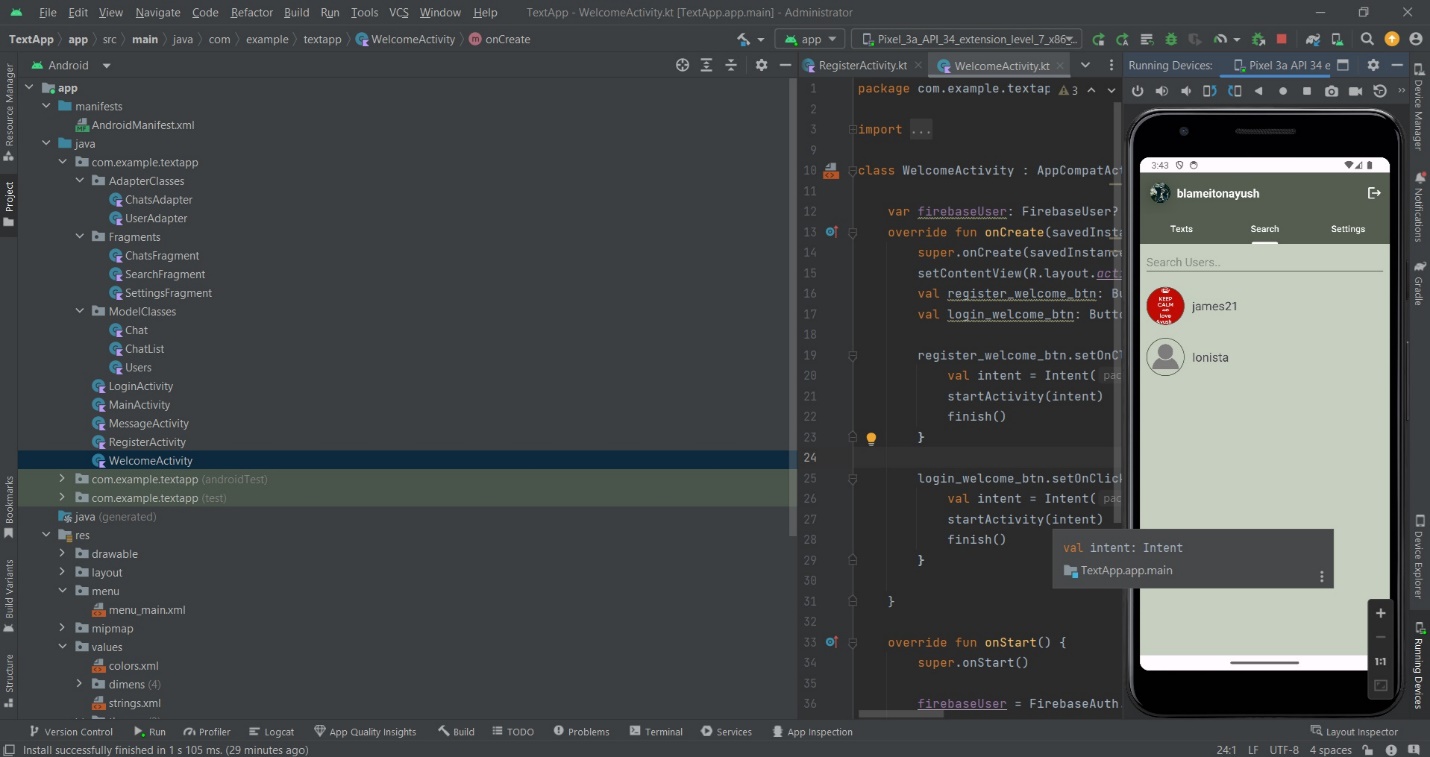
Seen/unseen functionality for messages.

****

* **Search Fragment :**

Enables users to search for other users.

Utilizes Firebase Realtime Database for efficient user search.

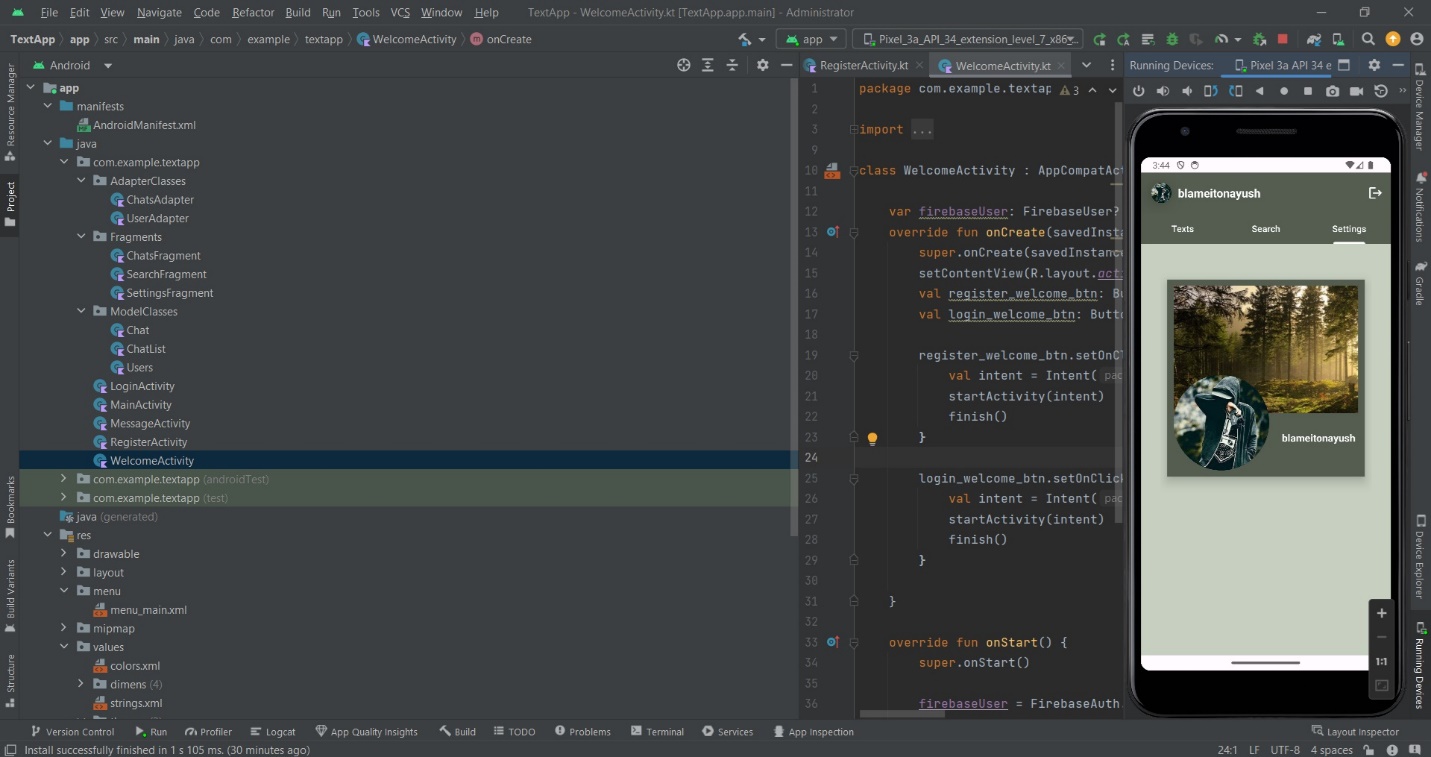


* **Settings Fragment:**

Manages user profile and application settings.

Allows users to set or change profile images and cover photos.

Updates user details in Firebase Realtime Database.

****

* **Message Activity :**

Facilitates one-on-one messaging.

Real-time exchange of messages.

Supports image uploads, displaying images within the chat.

* **Conclusion :**

The Texting App project offers a reliable and easy-to-use messaging platform, fulfilling its goals. Effective real-time communication and user data management are ensured by the integration of Firebase services. The project provides a seamless and interesting user experience while adhering to the specifications specified in the original design.

I, thank the Kotlin programming language for facilitating effective Android application development as well as Firebase for offering a robust backend solution. The Texting App project's successful completion shows how cutting-edge technologies may be integrated effectively to produce a contemporary and useful chat tool.

* **Error Handling and Validation:**

**User Registration :**

Challenge: Invalid or incomplete user data during registration.

Solution: Implement validation checks for username, email, and password. Display error messages for users to correct input.

**User Login:**

Challenge: Incorrect email or password during login.

Solution: Use Firebase Authentication callbacks to handle login errors. Provide feedback to users about the login failure.

**Empty Fields:**

Challenge: Users submit forms with empty fields.

Solution: Add checks to ensure that essential fields are not left blank. Display appropriate messages prompting users to complete required fields.

**Profile Image Upload :**

Challenge: Failure to upload profile images.

Solution: Implement error handling for image upload tasks using Firebase Storage. Notify users of upload failures and guide them on possible solutions.

**Real-time Database Read/Write:**

Challenge: Errors during data read or write operations.

Solution: Implement try-catch blocks around database operations. Provide users with meaningful error messages and prompt them to retry.

**Network Connectivity:**

Challenge: No or slow internet connection.

Solution: Check network connectivity before making network requests. Display messages to users indicating network issues and advising them to check their internet connection.

**Input Data Validation:**

Challenge: Users entering invalid data in search or chat messages.

Solution: Validate user input using regular expressions or custom validation rules. Inform users of invalid inputs and guide them on the correct format.

**References :**

* <https://developer.android.com/kotlin>
* <https://www.pluralsight.com/paths/android-development-with-kotlin-fundamentals>
* <https://firebase.google.com/>
* <https://frontendmasters.com/courses/firebase/>