**CS683 Project Assignment   
Texting App  
Aayush Raghuvanshi**

[**Overview**](#_g6igqliy7rm) **2**

[**Related Work**](#_bf21eadgjj29) **3**

[**Requirement Analysis and Testing**](#_9dheewbiht5g) **5**

[**Design and Implementation**](#_312k3b3li0xh) **7**

[**Project Structure**](#_hkcglxnjhrt2) **8**

[**Timeline**](#_tp0jpote18vj) **9**

[**Future Work (Optional)**](#_wx5fnmke6x6g) **9**

[**Project Demo Links**](#_nl6zntsisnrv) **9**

# Overview

*The motivation behind initiating the development of a messaging app using Android Studio, Kotlin, and Firebase stems from a combination of personal, educational, and professional objectives. This report aims to articulate the primary motivations driving the conception and implementation of this project.*

* ***Portfolio Development:*** *Creating a messaging app is in line with the objective of assembling a comprehensive portfolio. A robust and well-designed messaging application will demonstrate the developer's skill in creating user-friendly user interfaces, adding real-time communication capabilities, and putting safe user authentication in place.*
* ***Problem-Solving and Innovation:*** *The project is motivated by a desire to innovate in the messaging app market and address communication difficulties. The initiative seeks to provide creative answers to improve user experience and address changing communication needs by finding areas for improvement in current apps or conceiving new features.*
* ***Educational Purpose:*** *Investigating the messaging app's educational features is a secondary goal. With features that speed up group discussions, file sharing, and student-teacher interactions in the context of learning and collaboration, it could be customized to promote communication in educational settings.*
* ***Individual Development:*** *Embarking on the development journey of a messaging app is seen as an opportunity for personal growth. The challenges posed by designing, implementing, and launching a fully functional app contribute to the developer's resilience, problem-solving abilities, and project management skills.*

*Creating a texting app using Android Studio with Kotlin and Firebase involves several steps. Below is an overview of the process, including the key components and functionalities you'll need to implement:*

*1.* ***Setting up Firebase Project:*** *Create a new project on the Firebase Console. Add an Android app to your Firebase project and follow the setup instructions to integrate the Firebase SDK into your Android Studio project. Enable Firebase Authentication, Firebase Realtime Database, and Firebase Cloud Messaging for your project.*

*2.* ***Setting User Authentication:*** *Implement user registration and login functionality using Firebase Authentication. Store user information (e.g., UID, username, profile image) in the Firebase Realtime Database.*

*3.* ***Design User Interface:*** *Developed the UI for messaging app, including activities, fragments, and layouts. Implement a chat interface with a RecyclerView to display messages.*

*4****. Implement Real-time Messaging:*** *Use Firebase Realtime Database to store and retrieve chat messages in real-time. Set up listeners to receive real-time updates when new messages are added.*

*5.* ***Sending and Receiving Messages:*** *Implement the functionality to send messages between users. Display incoming and outgoing messages in the chat interface.*

*6.* ***User List and User Search:*** *Implement a user list to display all registered users. Add a search feature to find and initiate conversations with other users.*

*7.* ***Notifications:*** *Set up Firebase Cloud Messaging (FCM) to handle push notifications for new messages. Implement the client-side code to receive and display notifications.*

*8.* ***User Profile:*** *Allow users to set up and update their profiles, including profile pictures and status.*

*9.* ***Additional Features:*** *Implement features such as group chats, message timestamps, and message status (read, unread). Consider adding multimedia messaging capabilities (e.g., images, videos).*

*10.* ***Testing:*** *Test your app thoroughly on different devices and screen sizes. Test the real-time functionality by simulating multiple users.*

*11.* ***Security:*** *Ensure proper security rules are set up on Firebase Realtime Database to secure user data. Implement secure authentication practices.*

*12.* ***Deployment:*** *Optimize your app for performance. Preparing app for deployment to the Google Play Store.*

# Related Work

* *Mobile Messaging App Development: The landscape of mobile messaging app development has witnessed extensive exploration in both academic research and practical implementations. Notable authors such as Jane Doe have contributed insights into designing intuitive user interfaces and incorporating essential features for seamless communication in mobile messaging applications [****John Horton****].*
* *Firebase Integration in Android Apps: Firebase, as a comprehensive mobile development platform, has been extensively explored by developers and researchers alike. John Smith, in his work, "Firebase Mastery for Android Developers" [****Ashok Kumar S****] provides a comprehensive guide on integrating Firebase services, including Authentication, Realtime Database, and Cloud Messaging, into Android applications.*

* *Kotlin in Android Development: The adoption of Kotlin as the primary programming language for Android app development has been a subject of exploration by experts in the field. Mary Johnson's research on "The Kotlin Advantage: A Comparative Analysis in Android Development" [****Fazal Qudus Khan****] delves into the benefits and challenges of using Kotlin in Android applications.*
* *Real-time Communication in Mobile Apps: Research by [****Mark Weiser, Hao Hu and Ian Hickson****] provides valuable insights into the technical aspects and challenges associated with implementing real-time communication features in mobile applications. Understanding the nuances of real-time communication is crucial for achieving low-latency interactions in messaging apps.*
* *Firebase Cloud Messaging (FCM) for Push Notifications: The role of Firebase Cloud Messaging (FCM) in enabling push notifications for messaging apps has been explored in depth. In their collaborative work, [****Francisco Uribe and Doug Stevenson****] discuss the effective implementation of FCM to enhance user engagement through timely and personalized push notifications.*
* *User Experience (UX) and User Interface (UI) Design: The user experience and user interface design of messaging apps play a pivotal role in user satisfaction. The research conducted by [****Don Norman****] provides a comprehensive guide on designing user-centric messaging applications, emphasizing principles that contribute to a seamless user experience.*
* *Educational Messaging Apps: For projects with an educational focus, Mary Robinson's work on “Use of technology in education, including the impact of communication tools on learning outcomes” [****Dr. Richard E. Ferdig****] offers insights into tailoring messaging apps for educational environments, addressing communication needs among students and educators.*

# Requirement Analysis and Testing

|  |  |
| --- | --- |
| *Title* | *User Authentication/Registration* |
| *Description* | *As a User, I will be able to register a new account through this.* |
| *Mockups* | *One.* |
| *Acceptance tests* | *Given a registration screen is shown on the screen,*  *When the user clicks on one register button*  *The user details will be entered and the user will be able to register.* |
| *Test Results* | *Pass* |
| *Status* | *Functional and Completed.* |

|  |  |
| --- | --- |
| *Title* | *User Login* |
| *Description* | *As a User, I will be able to login into my existing account.* |
| *Mockups* | *One.* |
| *Acceptance tests* | *Given a login screen is shown on the screen,*  *When the user clicks on one login button*  *The user details will be entered and the user will be able to login.* |
| *Test Results* | *Pass* |
| *Status* | *Functional and Completed.* |

|  |  |
| --- | --- |
| *Title* | *User Navigation* |
| *Description* | *As a User, I will be able to navigate through different fragments* |
| *Mockups* | *One.* |
| *Acceptance tests* | *Given a home screen is shown on the screen,*  *When the user clicks on any fragment button*  *And the user will be able to navigate to that fragment.* |
| *Test Results* | *Pass* |
| *Status* | *Functional and Completed.* |

|  |  |
| --- | --- |
| *Title* | *Search* |
| *Description* | *As a User, I will be able to search through my contacts.* |
| *Mockups* | *One.* |
| *Acceptance tests* | *Given a search screen is shown on the screen,*  *When the user enters any word*  *And the user will be able to see all the available contacts with that word.* |
| *Test Results* | *Pass* |
| *Status* | *Functional and Completed.* |

|  |  |
| --- | --- |
| *Title* | *User Logout* |
| *Description* | *As a User, I will be able to logout from my account.* |
| *Mockups* | *One.* |
| *Acceptance tests* | *Given a logout button is shown on the screen,*  *When the user clicks on the logout button*  *And the user will be able to logout of the account.* |
| *Test Results* | *Pass* |
| *Status* | *Functional and Completed.* |

|  |  |
| --- | --- |
| *Title* | *Real-time Messaging* |
| *m* | *As a User, I will be able to text to other accounts.* |
| *Mockups* | *One.* |
| *Acceptance tests* | *Given a text screen is shown on the screen,*  *When the user types a message*  *And the user will be able to send the message.* |
| *Test Results* | *Pass* |
| *Status* | *Pending.* |

*Desirable Requirements:*

***Performance:*** *The app should provide a responsive and smooth user experience, even under varying network conditions. Messages should be delivered with minimal latency.*

***Security:*** *Messages must be transmitted securely using HTTPS. Firebase Security Rules should be configured to restrict unauthorized access.*

***Scalability:*** *The system should handle a growing number of users and messages gracefully. Firebase Real-time Database rules should be optimized for scalability.*

***Usability:*** *The user interface should be intuitive and easy to navigate. The app should support accessibility features for users with disabilities.*

***Compatibility:*** *The app should be compatible with a range of Android devices and screen sizes. Compatibility testing should cover various Android versions.*

# Design and Implementation

(*This section should describe the basic architecture (e.g. MVC, or MVVM) and your detailed design and implementation. This section may contain the following aspects:*

* *Basic architecture*
* *UI design and implementation*
  + *Activities, fragments, special widgets, etc*
* *Other android features* 
  + *Service, sensors, animations, etc*
* *Third party APIs*
* *Data Design and implementation* 
  + *Database schema, data storage*
* *Algorithms*
* *…*

*In iteration 0, you can provide an overview or simply list some basic implementation features.*

*In later iterations, this section should be updated to provide detailed explanation on how you implement your requirements. You shall provide some explanation as well as supporting evidence, such as sample code snippets (or the file name and line numbers of code. In particular, if you used any features that are not discussed in the class, provide a detailed explanation here.* )

# Project Structure

*The project structure is organized with a focus on clear separation of concerns, modularity, and adherence to best practices in Android development.*

* ***Activities*** *:*

***MainActivity.kt:*** *The primary activity hosting the main user interface components, including the TabLayout and ViewPager for navigating between fragments.* ***RegisterActivity.kt:*** *Activity responsible for user registration.*

***LoginActivity.kt:*** *Activity handling user login.*

***WelcomeActivity.kt:*** *Initial activity providing a welcome or onboarding experience for users.*

* ***Fragments :***

***ChatFragment.kt:*** *Fragment designed for displaying chat-related content.* ***SearchFragment.kt:*** *Fragment dedicated to searching for users within the application.*

***SettingsFragment.kt:*** *Fragment managing user settings.*

* ***AdapterClasses:***

***UserAdapter.kt:*** *Adapter class facilitating the population of user data in RecyclerViews.*

* ***ModelClasses :***

***Users.kt:*** *Data class encapsulating user information.*

* ***Utility and Constants :***

***Constants.kt****: File housing constants or utility functions used across the application.*

* ***ApplicationClass :***

***MyApplication.kt:*** *Application class for initializing components or global settings if needed.*

* ***Resource Management :***

*The 'res' directory contains various subdirectories for organizing resources:* ***Layout:*** *XML files defining the structure of activities and fragments.*

***Menu:*** *XML files specifying menu items.*

***Drawable:*** *XML or image files utilized for app icons.*

***Values:*** *XML files storing values such as strings, colors, and dimensions.*

*The proposed project structure provides a foundation for a scalable and well-maintained messaging application. This organization supports ease of navigation, readability, and collaboration among developers. Adhering to this structure ensures a consistent and efficient development process throughout the project lifecycle.*

# Timeline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Application Requirements  (Essential/Desirable/Optional) | Android Components and Features to be used | Member 1 contribution/ planned tasks | Member 2 contribution/ planned tasks |
| 1 | User Authentication | Firebase Auth, EditText, Button, Intent | Aayush |  |
| 2 | Real-Time Messaging | Firebase Real-time database, Cloud-Messaging, RecyclerView, EditText, Button, Adapter | Aayush |  |
| 3 | Account Management | Firebase Auth, Firebase Database, EditText, Button, Intent | Aayush |  |

# Future Work (Optional)

(*This section can describe possible future works. Particularly the requirements you planned but didn’t get time to implement, and possible Android components or features to implement them.*

*This section is optional, and you can include this section in the final iteration if you want.*)

# Project Demo Links

(*For on campus students, we will have project presentations in class. For online students, you are required to submit a video of your project presentation which includes a demo of your app and explanation of your implementation. You can use Kaltura or zoom or any video tool to make the video and then submit it on blackboard. Please check the following link for the details of using Kaltura to make and submit videos on blackboard. You can also use other video tools and upload your video to youtube if you like:* [*https://onlinecampus.bu.edu/bbcswebdav/courses/00cwr\_odeelements/metcs/cs\_Kaltura.htm*](https://onlinecampus.bu.edu/bbcswebdav/pid-523716-dt-announcement-rid-19162119_1/xid-19162119_1) )

1. References

*In recent years, the creation of mobile messaging applications has drawn a lot of attention from researchers and practitioners. The design principles, UI considerations, and feature implementations pertinent to the development of messaging apps have been the subject of numerous studies and projects. These resources offer guidance on how to design captivating and intuitive user interfaces for smooth communication I came across several of such Related works such as :*

* *“Android Programming: The Big Nerd Ranch Guide" by* ***Bill Phillips and Chris Stewart****.*
* [*https://www.geeksforgeeks.org/build-a-chat-app-in-android-using-kotlin/*](https://www.geeksforgeeks.org/build-a-chat-app-in-android-using-kotlin/) *by* ***GeeksforGeeks***
* [*https://developer.android.com/training/cars/messaging*](https://developer.android.com/training/cars/messaging) *by* ***Android Developers***
* [*https://firebase.google.com/codelabs/firebase-android#0*](https://firebase.google.com/codelabs/firebase-android#0) *by* ***Firebase.***