

ANDROID APPLICATION DEVELOPMENT BY KOTLIN
SMART BRIDGE INTERNSHIP
PROJECT REPORT

submitted by

WASIQ MAJEED MIRAB

ANKIT MAHAJAN

ARNAV GUPTA

TABLE OF CONTENTS

	TITLE	Pg No.
1	INTRODUCTION	
	1.1 Overview A brief description about your project	
	1.2 Purpose The use of this project. What can be achieved using this.	
2	LITERATURE SURVEY	
	2.1 Existing problem Existing approaches or method to solve this problem	
	2.2 Proposed solution What is the method or solution suggested by you?	
3	THEORITICAL ANALYSIS	
	3.1 Block diagram Diagrammatic overview of the project.	
	3.2 Hardware / Software designing Hardware and software requirements of the project	
4	EXPERIMENTAL INVESTIGATIONS	
	Analysis or the investigation made while working on the solution.	
5	FLOWCHART	
	Diagram showing the control flow of the solution	

6	RESULT	
	Final findings (Output) of the project along with screenshots.	
7	ADVANTAGES & DISADVANTAGES	
	List of advantages and disadvantages of the proposed solution	
8	APPLICATIONS	
	The areas where this solution can be applied	
9	CONCLUSION	
	Conclusion summarizing the entire work and findings.	
10	FUTURE SCOPE	
	Enhancements that can be made in the future.	
11	BIBLIOGRAPHY	
	References of previous works or websites visited/books referred for analysis about the project, solution previous findings etc.	
	APPENDIX A. Source Code Attach the code for the solution built.	

Introduction:

1.1. overview:

Project Overview:

The project aims to develop a chat application for Android that allows users to engage in real-time conversations with other users. The application will provide a platform for seamless communication, enabling users to exchange messages, and other media.

Project Scope:

The key features and functionalities of the chat application may include:

- User registration and authentication: Users can create accounts and log in securely.
- Real-time messaging: Users can send and receive messages instantly, creating individual or group chats.
- Multimedia support: Users can share images, videos, and other media files within conversations.
- Push notifications: Users receive notifications for new messages and chat updates.
- Online status: Users can view the online status of their contacts.
- User profiles: Users can view and update their profiles, including profile pictures and status messages.
- Search and add contacts: Users can search for other users and send friend requests or invite others to join the app.
- Security and privacy: Implementation of encryption and privacy settings to protect user data.

1.2. Purpose.

The chat application project you're working on can have several practical uses and provide various benefits. Here are some potential use cases and achievements that can be realized using a chat application:

1. **Real-time Communication:** The primary purpose of the chat application is to enable real-time communication between users. This can be used for personal or professional purposes, allowing users to connect with friends, family, colleagues, or clients instantly.
2. **Social Networking:** The chat application can serve as a platform for social networking, where users can meet new people, join interest-based groups or communities, and expand their social connections.
3. **Collaboration and Team Communication:** The chat application can facilitate efficient collaboration among team members. It allows individuals or groups to exchange information, discuss ideas, share files, and coordinate activities in real time, regardless of their physical locations. This is particularly useful for remote teams or distributed projects.
4. **Customer Support and Service:** The chat application can be utilized by businesses to offer customer support and service. Users can reach out to businesses directly through the chat application to ask questions, request assistance, or seek information.
5. **Community Engagement:** The chat application can foster community engagement by providing a platform for users to interact, share experiences, seek advice, and participate in discussions related to specific topics or interests.
6. **Personal Messaging:** Users can utilize the chat application for one-on-one personal messaging, connecting with friends, family, or acquaintances. It allows for private and

direct communication that is more immediate and convenient than traditional methods like SMS or email.

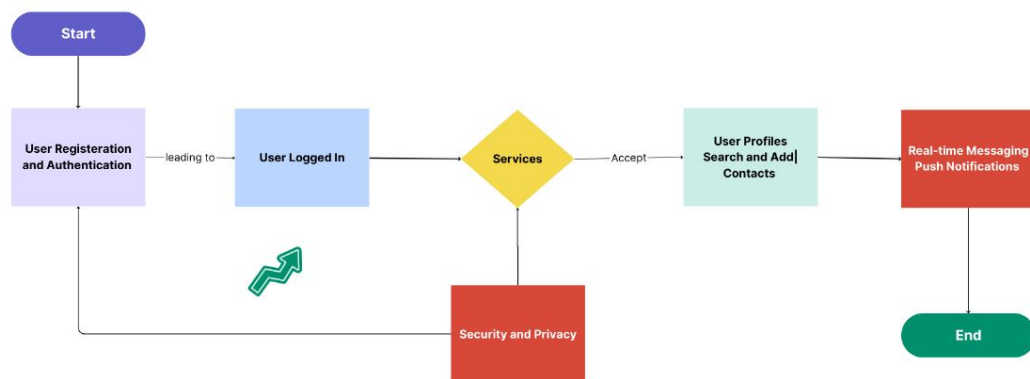
2. LITERATURE SURVEY

As of now there is a lot of advancement in the chatapp area but the USP of our app is that if the number exists in the database then the user can just enter his number and the chat facility will be available he doesn't need to save the number first or name everything will be fetched from database only the number should be registered.

3. THEORITICAL ANALYSIS

3.1 Block Diagram

Workflow Diagram



3.2 Hardware / Software designing

Software Requirements

- Android Studio: Android Studio is used for Android app development.

- Kotlin Plugin: To write the code and run the code in kotlin we need to install kotlin plugins.
- Android SDK (Software Development Kit): The Android SDK provides a set of tools, libraries, and APIs necessary for developing Android applications.
- Java Development Kit (JDK): To create the java code for the chat application.

4. EXPERIMENTAL INVESTIGATIONS

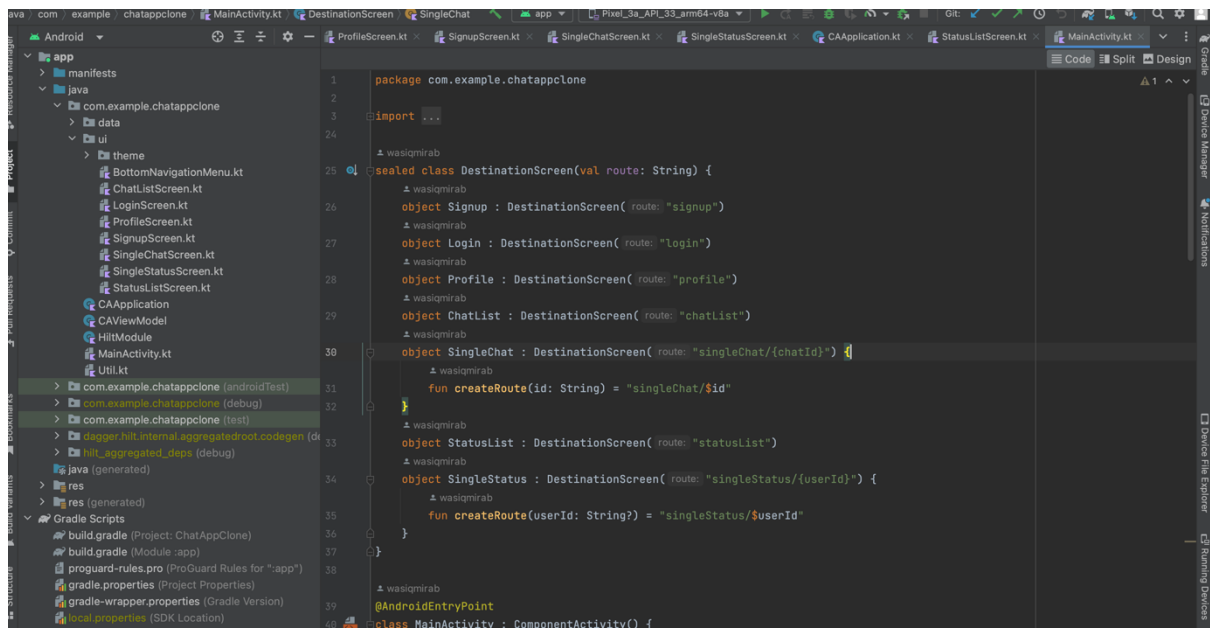
Various analysis were made while working on the chat application like the

- User interface design was correctly aligned to the phone screen and the login page was clear and easy to access for the user.
- Test the functionality of the application to ensure that users can perform essential tasks, such as sending and receiving messages.
- Evaluate the app's performance in terms of speed, responsiveness, and resource consumption. Check for any lags or slowdowns, especially when the chat list contains a large number of messages or contacts.

- Security analysis like during login taking a password set by the user to enter the application so that all the content is safe and secure.
- Sign up page being secure and private to the user as using personal and unique mobile number and password to login in the application.
- Status option which can help to know about the status of the chat of the user given access with their mobile number.

6.Result.

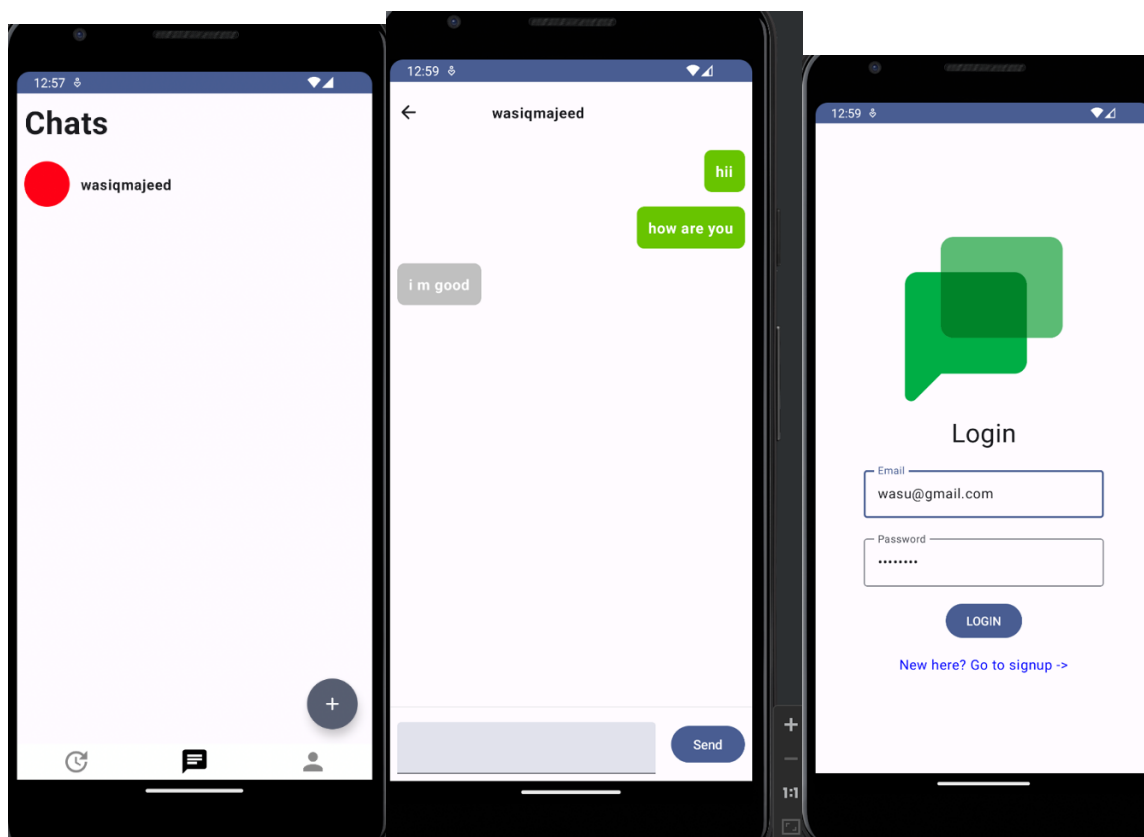
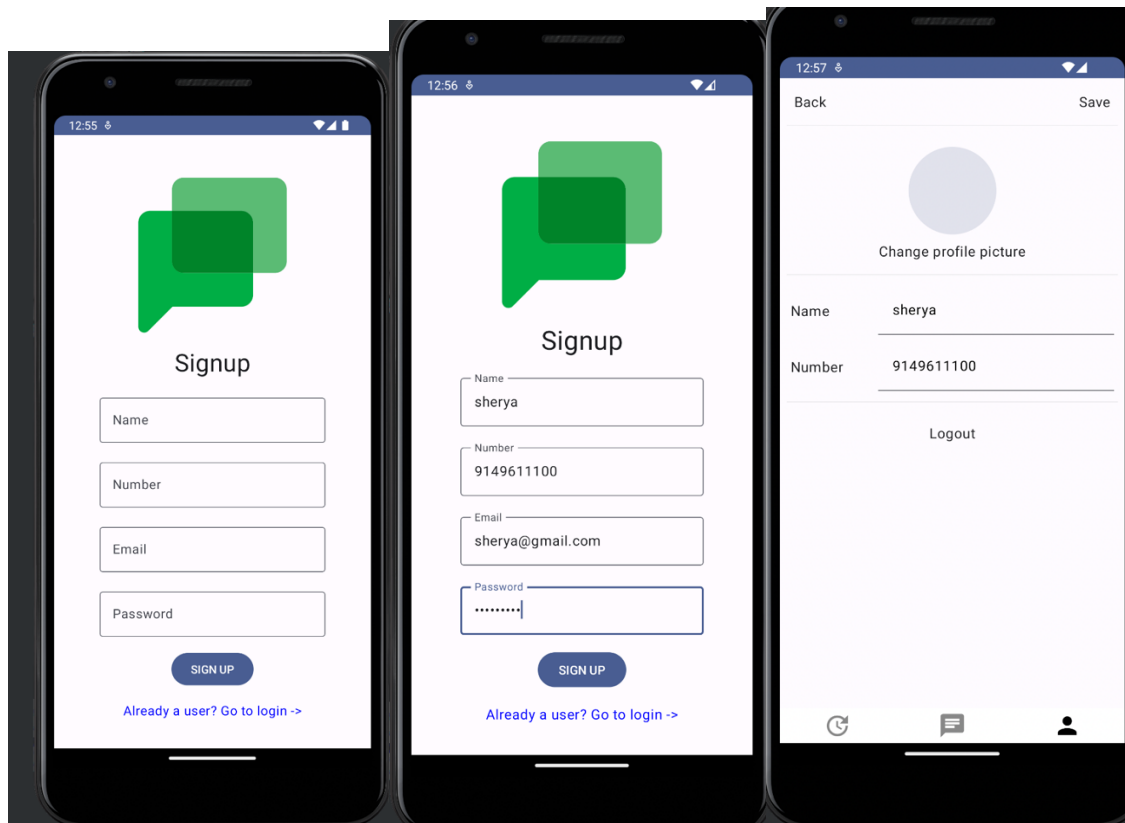
Output screenshots.

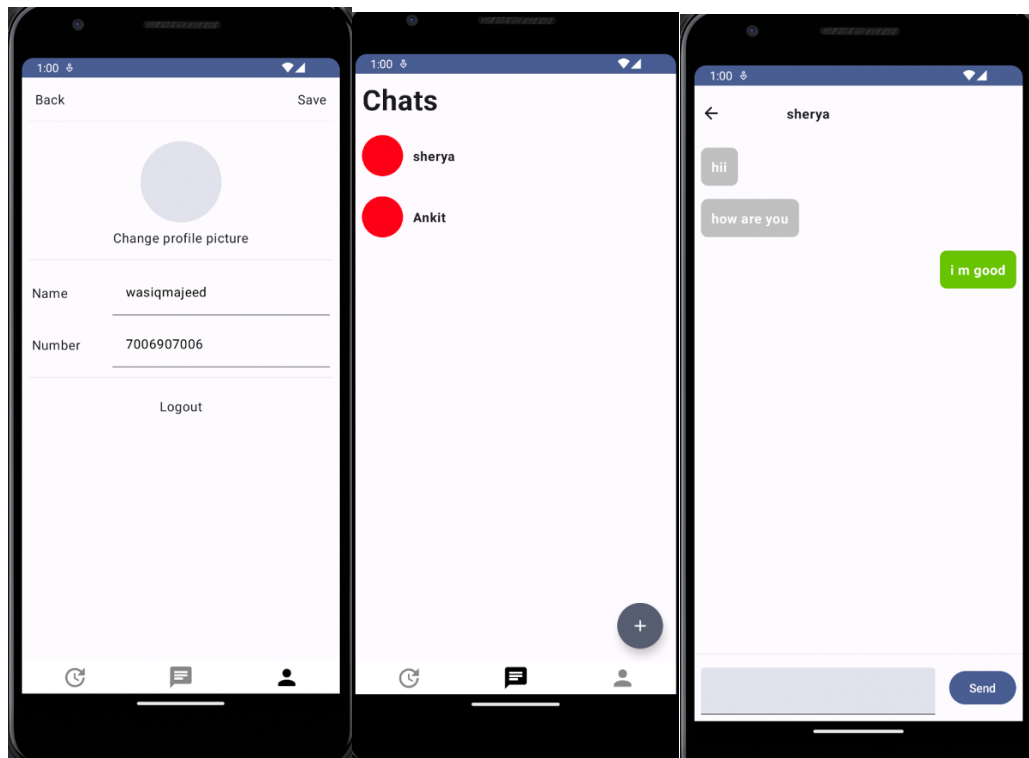


```

1 package com.example.chatappclone
2
3 import androidx.navigation.NavController
4
5 sealed class DestinationScreen(val route: String) {
6     object Signup : DestinationScreen(route = "signup")
7     object Login : DestinationScreen(route = "login")
8     object Profile : DestinationScreen(route = "profile")
9     object ChatList : DestinationScreen(route = "chatList")
10    object SingleChat : DestinationScreen(route = "singleChat/{chatId}") {
11        fun createRoute(id: String) = "singleChat/$id"
12    }
13    object StatusList : DestinationScreen(route = "statusList")
14    object SingleStatus : DestinationScreen(route = "singleStatus/{userId}") {
15        fun createRoute(userId: String?) = "singleStatus/$userId"
16    }
17 }
18
19 @AndroidEntryPoint
20 class MainActivity : AppCompatActivity() {
21
22     private lateinit var navigationController: NavController
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_main)
27         navigationController = NavController.fromMain(this)
28     }
29
30     override fun onStart() {
31         super.onStart()
32         navigationController.navigate(DestinationScreen.Login.route)
33     }
34
35     override fun onResume() {
36         super.onResume()
37         navigationController.navigate(DestinationScreen.Profile.route)
38     }
39
40     override fun onPause() {
41         super.onPause()
42         navigationController.navigate(DestinationScreen.ChatList.route)
43     }
44
45     override fun onStop() {
46         super.onStop()
47         navigationController.navigate(DestinationScreen.SingleChat.route)
48     }
49
50     override fun onDestroy() {
51         super.onDestroy()
52         navigationController.navigate(DestinationScreen.StatusList.route)
53     }
54
55     override fun onBackPressed() {
56         navigationController.navigate(DestinationScreen.SingleStatus.route)
57     }
58 }

```



7. Advantages and Disadvantages:

Advantages:

1. **Real-time Communication:** Chat applications enable instant messaging, allowing users to have quick and direct conversations with others, facilitating efficient and timely communication.
2. **Convenience and Accessibility:** Users can access the chat application from their mobile devices, making it convenient to communicate on-the-go. It provides accessibility and flexibility in connecting with others whenever and wherever they are.
3. **Multimedia Support:** Chat applications often support multimedia file sharing, including images, videos, and documents, enhancing the communication experience and enabling richer content sharing.

Disadvantages:

1. **Dependency on Internet Connectivity:** Chat applications require a stable internet connection to function properly. Poor internet connectivity can lead to delays in message delivery or make the application unusable until connectivity is restored.
2. **Privacy and Security Concerns:** Chat applications may be vulnerable to privacy breaches or security threats. User data, including conversations and personal information, should be protected through proper encryption, authentication, and security measures.
3. **Platform Compatibility:** Chat applications need to be developed for specific platforms, such as Android, iOS, or web. This can result in the need for multiple versions or adaptations to cater to different devices and operating systems.

8. Applications.

Chat applications have various applications and are widely used in different contexts. Here are some common applications of chat applications:

Personal Communication: Chat applications are widely used for personal communication between individuals, allowing friends, family members, and acquaintances to connect and have private conversations in real time. It provides a convenient and immediate way to stay in touch and share updates.

Business Communication: Chat applications are increasingly used for business communication, both internally within organizations and externally with clients, partners, or customers. Teams can use chat applications for

collaboration, project discussions, sharing updates, and quick communication.

Customer Support: Many businesses utilize chat applications as part of their customer support systems. It allows customers to reach out to businesses directly, ask questions, seek assistance, or provide feedback in real time. Chat applications provide a more interactive and efficient customer support experience compared to traditional methods like phone calls or emails.

Social Networking: Chat applications often integrate social networking features, enabling users to connect with people who share similar interests, join groups or communities, and participate in discussions. It provides a platform for social interaction, making new connections, and sharing ideas.

Remote Collaboration: Chat applications play a crucial role in facilitating remote collaboration and teamwork. With the ability to communicate in real time and share files, remote team members can collaborate effectively on projects, discuss ideas, and coordinate tasks regardless of their physical locations.

9.Conclusion:

In conclusion, the development of a chat application for Android offers numerous advantages and opportunities for users in various contexts. Chat applications enable real-time communication, making it convenient and accessible for individuals to connect with others. Whether for personal, professional, or social purposes, chat applications provide a platform for instant messaging, multimedia sharing, group communication, and collaboration.

The advantages of chat applications include convenience, accessibility, multimedia support, group communication capabilities, push notifications, and cost-effective communication. However, it is important to consider potential challenges such as dependency on internet connectivity, privacy and security concerns, platform compatibility, user adoption and competition, user interface and user experience, and the need for maintenance and updates.

Despite these challenges, the wide range of applications for chat applications makes them highly valuable. They facilitate personal communication, enhance business communication and customer support, enable social networking and community building, support remote collaboration, aid in education and e-learning, and assist in event coordination.

By understanding the advantages, disadvantages, and applications of chat applications, developers can create robust and user-friendly solutions that meet the communication needs of individuals, businesses, and communities. With the right approach, a well-designed chat application has the potential to significantly improve real-time communication and foster connections in our increasingly digital world.

10 Future scope:

The future scope of chat applications is promising, as they continue to evolve and adapt to changing technologies and user needs. Here are some potential future advancements and trends in the field of chat applications:

Enhanced User Experience: Chat applications will likely focus on providing seamless and intuitive user experiences. This includes features such as voice and video calling, augmented reality (AR) integration, chatbots for

automated interactions, and improved personalization based on user preferences.

AI-Powered Chatbots: Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies will play a significant role in chat applications. AI-powered chatbots can handle customer queries, provide instant assistance, and even perform tasks such as scheduling appointments or making recommendations.

Integration with IoT Devices: As the Internet of Things (IoT) expands, chat applications may integrate with IoT devices for enhanced communication and control. Users could interact with their smart home devices, appliances, or even vehicles through chat applications, providing a centralized hub for managing connected devices.

End-to-End Encryption: Privacy and security will continue to be a key focus. Chat applications may implement robust end-to-end encryption to ensure that user conversations and data remain private and secure from unauthorized access.

Integration with Other Platforms: Chat applications may integrate with other popular platforms and services, such as social media networks, e-commerce platforms, or productivity tools. This integration would enable users to access a wider range of functionalities within a single application, streamlining their digital interactions.

References:

We used udemy, android classes and some internet links for this project.

<https://www.jetpackcompose.net>

<https://developer.android.com>

SourceCode:

<https://github.com/WasiqMajeedMirab/ChatAppAndroid>