Design And Development Of A Real Time Messenger Application For Android

Amar Jyoti
SCSE
Galgotias
University Greater
Noida, India

Deepak Walia
SCSE
Galgotias
University Greater
Noida, India

Dr. Ramesh Kumar
SCSE
Galgotias
University Greater
Noida, India

Abstract - In today's digital age, while messaging apps have become a staple in our daily communications, many feel impersonal. The majority lack the flexibility for users to express their individuality, leaving a one- size-fits-all experience that often feels cold and generic. To address this, we've crafted an Android messenger that balances core messaging functionalities with a splash of personal flair. Users aren't just limited to sending and receiving messages; they're empowered to shape their app experience, from profile tweaks to chat window aesthetics, making each interaction genuinely personal. Leveraging the robustness of React Native CLI and the versatility of Firebase, we've created a responsive and secure platform, including sliders to change the look and feel of the chat environment and image pickers to help users customize it. Upon rollout, the app not only performed seamlessly but also resonated with users. Feedback highlighted the unique blend of efficient communication and the freedom to personalize, offering a refreshing twist to conventional messaging. As we look ahead, the possibilities are exciting—be it introducing multimedia sharing or voice chats, the goal remains: make communication personal, always.

Keywords - Digital Age, Messaging Apps Android Messenger, React Native CLI, Firebase, User Customization, Efficient Communication

I. INTRODUCTION

The pace at which communication has changed in our fast-paced digital environment is unprecedented. Instantaneous messaging has replaced the days of waiting for mail or making landline calls; this shift has been spearheaded by the ubiquitous smartphones. One of the most popular mobile operating systems in the world, Android, has been leading this revolution in communication. Consumers frequently express a need for a more intimate element in their everyday digital interactions. Let's take a look at the world of Android messaging applications.

These instant messaging apps have become an essential component of our digital toolbox since they allow for instantaneous text-based communication. With a few touches, they can even order food or hail a car, allowing us to stay in touch with loved ones on other continents and facilitating quick customer encounters for companies. Unfortunately, even while these applications have unquestionably made life easier, a lot of them feel quite same. There's a clear lack of coziness and human touch with the templated blue chat bubbles, the predictable notification bells, and the rigid user interfaces. Imagine a messaging software that combines the personal touch of a handwritten message with the convenience we've become accustomed to from premium applications. a platform that enables you to express yourself,personalize the app, and do more than simply interact. This is an experience, a digital version of oneself, not simply another communication software. Our Android messaging software seeks to close the distance between emotionless digital text and intelligent conversation by utilizing the strength and adaptability of React Native CLI. By working with Firebase, we've made sure that although the app oozes friendliness on the front end, the back end is safe, quick, and trustworthy. We've also included a ton of other features to give some extra flare, including sliders to change the look and feel of the chat environment and image pickers to help users customize it. Simply put, this isn't about sending and receiving messages. It's about breaking the mold, questioning what a messaging app should be. It's about redefining digital communication, making every "Hello" feel special, every "Goodnight" feel intimate, and every chat feel like two friends catching up in a warm, cozy room. As you read on, we invite you to experience the marriage of professionalism and personality— a new take on Android messaging that promises not only to inform but also to inspire. Welcome to the world of personal digital communication in the future.

II. FORMULATION OF PROBLEM

Messaging apps stand as towering giants in the vast landscape of digital communication, connecting millions and facilitating countless interactions on a daily basis. However, a closer examination reveals a widespread problem: a lack of personalization and individual expression. The market is flooded with apps that, while useful, provide a consistent experience, leaving users yearning for a platform that can be tailored to their specific needs. This uniformity frequently leads to a sense of detachment, with the communication tool feeling more like a sterile instrument than an extension of one's personality. The issue isn't just aesthetic. In the absence of customizing possibilities, consumers may feel confined, thereby impeding open and free conversation. When the setting feels generic, the conversations within it may unintentionally take the same tone. The aim is to create an Android messaging app that effortlessly incorporates key functionality while also allowing users to express themselves, resulting in a more intimate and engaging conversation experience.

III. TOOLS AND TECHNOLOGY USED

We used a combination of industry-leading techniques and technologies to solve the above problem:

React Native CLI: React Native CLI, chosen for its unparalleled flexibility and performance, enables the building of native Android apps using JavaScript. Its component-based architecture offers a consistent user experience and quick development cycles.

Firebase: Serving as the backbone of our app, Firebase offers a suite of cloud-based tools that handle authentication, database storage, and real-time data synchronization. Its integration ensures that our app remains scalable, secure, and responsive.

Supplementary Libraries: To enhance the app's feature set and provide a richer user experience, we integrated several additional libraries:

- **@miblanchard/react-native-slider**: Enables dynamic UI adjustments.
- **@react-native-async-storage/async-storage**: Provides local storage capabilities, allowing users tosave preferences.
- **react-native-image-picker:** Facilitates image selection, crucial for profile and wallpaper customization.
- **react-native-contacts**: Integrates with the device's contact list for easy access and communication.
- **react-native-gesture-handler & react-native-reanimated**: These libraries enrich the app's interactivity, ensuring fluid animations and responsive touch controls

IV. LITERATURE SURVEY

A. The Origins of SMS and the Rise of Text Messaging

In the early 1990s, the world of communication experienced a silent revolution: the introduction of the Short Message Service (SMS).

This seemingly basic device, which allowed communications of up to 160 characters, suddenly became a cornerstone of modern communication. It marked a new era in which brevity and speed were valued, profoundly altering our communication patterns from phone conversations to fast, text-based discussions.

The subsequent introduction of Multimedia Messaging Service (MMS) broadened this environment. We were no longer simply texting; we were now exchanging photographs, movies, and sounds. MMS, however, was not without its difficulties. Issues with network compatibility and varying phone capabilities frequently hampered its potential, making it a less ubiquitous answer than its predecessor, SMS

1. The Smartphone's First Wave of Messaging Apps

With the emergence of smartphones came the birth of a completely new breed of communication tools: the first generation of messaging applications. BlackBerry Messenger (BBM), which debuted in 2005, was a pioneer. It provided quick delivery and read receipts, both of which we now take for granted but were groundbreaking at the time . But it wasn't until messaging applications like WhatsApp and Viber hit the market that they truly came into their own. They dismantled the boundaries erected by network carriers and device makers, providing a cross platform method of staying connected. These applications took use of the increasing availability of internet connectivity to deliver a more comprehensive and adaptable messaging experience. Smartphones and Internet connectivity are the driving forces. This incredible journey of messaging applications would not have been feasible without two essential factors broad smartphone usage and increased internet access. These technical improvements not only made messaging applications more accessible to a wider audience, but they also fostered continuous innovation in terms of functionality and user interface design.

B. Messaging App User Interface and Experience Design Evolution in Messaging Apps

The growth of messaging app User Interface (UI) and User Experience (UX) has been a continual path of innovation, closely reflecting the fast improvements in smartphone technology.

1) Messaging UI/UX in its Early Stages:

Early messaging applications featured basic and effective UIs that focused mostly on text-based communication. The design was simple, with only a few input boxes and send buttons.With the introduction of touchscreen smartphones, messaging applications began to add more intuitive and visually appealing components. Chat bubbles, swiping motions, and customisable keyboards were all introduced as part of this.

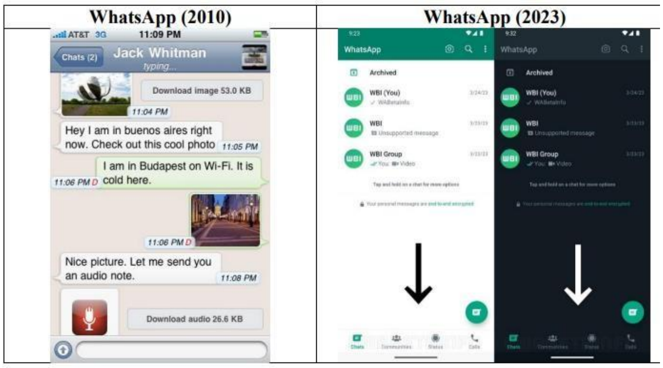
2) Multimedia Feature Integration:

Emojis and Stickers: The addition of emojis and stickers to communications added a new layer, allowing users to convey emotions more clearly. Media Sharing: The ability to exchange photographs, movies, and music files gradually became the norm, considerably improving the user engagement experience.

UI/UX of a Modern Messaging App:

Interfaces that are clean and intuitive: Modern applications promote simplicity and ease of navigation. Design components are intentionally placed to direct user attention and make the most commonly used functionality readily available.

Dark Mode and Customization: Recent trends reveal that dark mode for messaging applications is becoming increasingly popular, giving an eye-friendly alternative to standard bright screens. Furthermore, customization features such as chat themes and text sizes are becoming more widespread.



11:04 PM

Hey I am in buenos aires right now. Check out this cool photo 11:05 PM

I am in Budapest on Wi-Fi. It is cold here. 11:06 PM D

11:06 PM D

Nice picture. Let me send you an audio note. 11:08 PM

Fig:1 WhatsApp UI Comparison

Security Protocols in Messaging Applications

Encryption Technologies

End-to-End Encryption (E2EE)

Overview: E2EE is a method of secure communication that prevents third parties from accessing data while it's transferred from one end system to another. In messaging apps, this means only the communicating users can read the messages.

Implementation: Popular messaging apps like WhatsApp, Signal, and Telegram have implemented E2EE. WhatsApp uses the Signal Protocol, while Telegram uses its proprietary MT Protoprotocol.

Benefits: E2EE ensures that the content of conversations remains private and secure from interception or eavesdropping, even by the service providers themselves.

Challenges: While E2EE offers privacy, it also raises concerns, particularly in law enforcement,where it's seen as a hurdle in criminal investigations.

a) Transport Layer Security (TLS)

- **Overview:** TLS encrypts the channel of communication between the user and the server. It isused in combination with E2EE to provide an extra degree of security.
- **Implementation**: TLS is used by many messaging apps to secure communications in transit between the client and the server.
- **Benefits:** TLS prevents data during transmission from being intercepted. It works very well against man-in-the-middle attacks.

C. Case Study

a) Case Study 1: Security Vulnerabilities in WhatsApp (2019)

- **Incident**: In May 2019, WhatsApp discovered a flaw that allowed hackers to install commercial Israeli spyware on phones. A missed WhatsApp callwas enough to install the malware.
- **Impact**: The impact of this compromise was significant, raising major worries about the security of messaging applications, especially those with strong encryption algorithms.
- **Response**: WhatsApp swiftly patched the flaw and recommended its 1.5 billion users to upgrade their applications as a precaution

b) Case Study 2: The Encryption Controversy at Telegram (2016)

- **Incident: In 2016**, researchers discovered weaknesses in Telegram's encryption mechanism, MTProto. It was chastised for being homegrown, untested, and not as secure as advertised.
- **Impact:** The impact of this occurrence was that it triggered a debate regarding the dependability of proprietary encryption techniques and the need of utilizing well-established, peer-reviewed cryptographic methods.
- **Response**: Telegram responded by defending its protocol, but the criticism underlined the need of openness and standardization in encryption methodsused in messaging apps.

D. Technology Trends in Messaging Apps:

1) AI and Machine Learning in Messaging Apps

The use of AI and Machine Learning (ML) in messaging applications marks a significant advancement in how we engage with technology. These developments are not only changing user experiences, but also establishing new benchmarks for app usefulness and performance.

Predictive Text and Smart Replies Powered by AI Predictive text is one of the most apparent implementations of AI in messaging apps. As users type, machine learning algorithms assess their typing tendencies and linguistic preferences to propose words or sentences. This function simplifies communication, making it faster and more straightforward. Apps like Google's Gboard have taken this a step further by providing contextually appropriate responses based on the substance of the chat. These AI-powered capabilities are very valuable for increasing productivity, especially for users conversing on the move.

2) Chatbots are Automated Assistors.

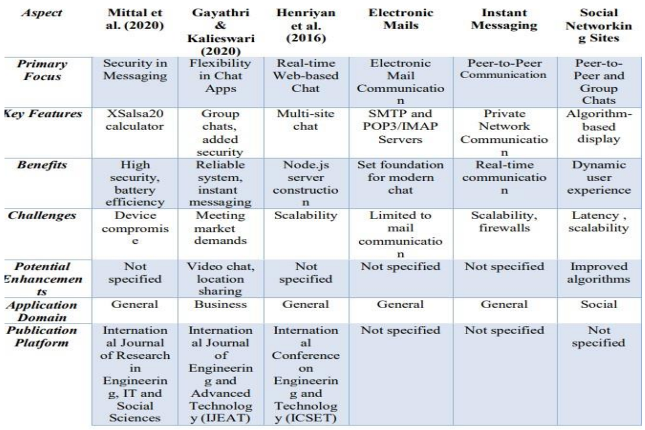
Another notable AI integration in messaging apps is chatbots. These virtual assistants may perform a range of functions within the chat interface, such as answering questions and providing customer service. Chatbots have been widely used for business reasons on platforms such as Facebook Messenger and WhatsApp, altering customer service paradigms. Chatbots employ natural language processing (NLP) and machine learning to interpret and answer to user questions, making the user experience more engaging and responsive.

3) Adoption and Impact Studies

Because of the real benefits these technologies provide, AI and ML have seen fast acceptance in messaging apps. According to Gartner, 70% of white-collar professionals will interact with conversational platforms on a regular basis by 2022. This statistic highlights the increasing dependence on AI-powered communication technologies in professional contexts.

A Comprehensive Survey of Messaging System Research The current rise in internet usage has radically altered the communication scene. As a result, numerous messaging and chatting systems have evolved, each attempting to bridge communication gaps and provide consumers with a unified experience.

In their study "Online Chat Request," published in the International Journal of Research in Engineering, IT, and Social Sciences in 2020, Mittal et al. addressed the crucial topic of security in messaging apps. Their emphasis on the employment of current tools to maintain safe communicationemphasizes the value of user privacy in today's interconnected environment. They demonstrated the XSalsa20 calculator, emphasizing its strong security and compatibility for mobile phones, which ensures that users' messages stay private even if devices are compromised. Simultaneously, Gayathri and Kalieswari's work in 2020 highlighted the need for flexibility in chat applications, with the goal of providing a stable and instant messaging system. Their study, published in the International Journal of Engineering and Advanced Technology (IJEAT), stressed thepotential of chat apps in the business arena, with capabilities such as group chats, improved security, and the ability to conduct video conversations and share location. On the technical side, Henriyan et al.'s 2016 paper, "Designing and deploying a real-time web-based chat server," presented at the International Conference on Engineering and Technology (ICSET), emphasized the importance of real-time multi-site chat apps, advocating for server construction using Node.js. Prior to these improvements, however, one of the oldest and most widely used communication methodswas electronic mail. The design of the e-mail system, which consists of SMTP servers for outgoing mail and POP3 or IMAP servers for incoming mail, has laid the groundwork for many modern chat programs.



Aspect	Mittal et al. (2020)	Gayathri & Kalieswari (2020)	Henriyan et al. (2016)	Electronic Mails	Instant Messaging	Social Networking Sites
Primary Focus	Security in Messaging	Flexibility in Chat Apps	Real-time Web-based Chat	Electronic Mail Communication	Peer-to-Peer Communication	Peer-to-Peer and Group Chats
Key Features	XSalsa20 calculator	Group chats, added security	Multi-site chat	SMTP and POP3/IMAP Servers	Private Network Communication	Algorithm-based display
Benefits	High security, battery efficiency	Reliable system, instant messaging	Node.js server construction	Set foundation for modern chat	Real-time communication	Dynamic user experience
Challenges	Device compromise	Meeting market demands	Scalability	Limited to mail communication	Scalability, firewalls	Latency , scalability
Potential Enhancements	Not specified	Video chat, location sharing	Not specified	Not specified	Not specified	Improved algorithms
Application Domain	General	Business	General	General	General	Social
Publication Platform	International Journal of Research in Engineering, IT and Social Sciences	International Journal of Engineering and Advanced Technology (IJEAT)	International Conference on Engineering and Technology (ICSET)	Not specified	Not specified	Not specified

Table 1: Comparative Study Analysis

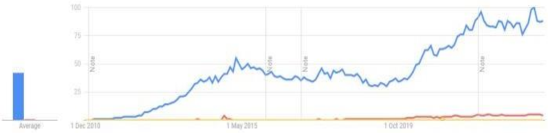
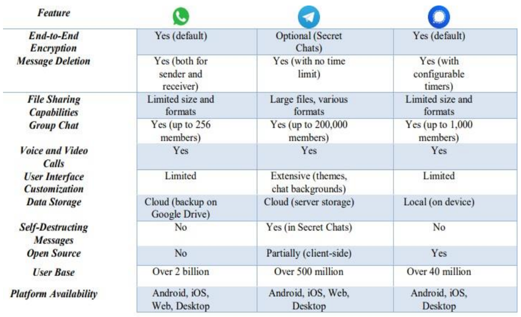


Fig:2 Usage Trends of Messaging Apps from 2010 to 2023

Instant Messaging (IM) has now grown as a real-   time extension of this, allowing for private conversation between two users as well as longer chat sessions with multiple participants. The difficulty with IM is its scalability, especially when millions of people are logged in at the same time. This scalability issue is compounded when clients are behind firewalls, making continuous and quick communication difficult. Furthermore, the rise of social networking sites has added another degree of complexity to digital communication. While these platforms provide peer-to-peer discussions, typically in groups, the algorithms used to display news and online friends have a significant impact on user experience. The scalability of these platforms' simultaneous discussions, combined with the dynamic nature of news updates and online presence, can occasionally induce latency in message delivery. Finally, as the digital communication landscape evolves, the problems and solutions given by many scholars highlight the importance of secure, scalable, and real-time messaging platforms. The balance between user experience, security, and technical resilience is at the center of this domain's continuous study

1. *Comparative Study Analysis of Literature Survey*

2. *Comparative Analysis of Messaging Platforms*



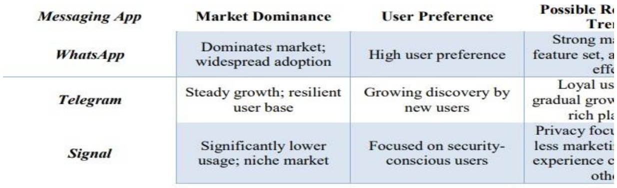
Feature	🟢	🔵	🔵
End-to-End Encryption	Yes (default)	Optional (Secret Chats)	Yes (default)
Message Deletion	Yes (both for sender and receiver)	Yes (with no time limit)	Yes (with configurable timers)
File Sharing Capabilities	Limited size and formats	Large files, various formats	Limited size and formats
Group Chat	Yes (up to 256 members)	Yes (up to 200,000 members)	Yes (up to 1,000 members)
Voice and Video Calls	Yes	Yes	Yes
User Interface Customization	Limited	Extensive (themes, chat backgrounds)	Limited
Data Storage	Cloud (backup on Google Drive)	Cloud (server storage)	Local (on device)
Self-Destructing Messages	No	Yes (in Secret Chats)	No
Open Source	No	Partially (client-side)	Yes
User Base	Over 2 billion	Over 500 million	Over 40 million
Platform Availability	Android, iOS, Web, Desktop	Android, iOS, Web, Desktop	Android, iOS, Desktop

Table 2: Comparative Analysis of Messaging Platforms

3. *Usage Trends of Messaging Apps from 2010 to 2023*

- **WhatsApp (Blue Line):** The graph demonstrates a substantial increase trend in WhatsApp usage beginning in 2010. There is a clear growth surge in early 2013, and usage steadily rises, peaking about 2019 before fluctuating. Nonetheless, the general trend for WhatsApp shows a dominant and growing usage over time, with a clear lead over the other two applications.
- **Telegram (Red Line):** Telegram's use line shows a gradual growth but remains pretty stable until around 2018, when there is a minor boost in usage. Although there are tiny gains and plateaus, the trend for Telegram reveals that it is growing steadily but modestly in comparison to WhatsApp, with no abrupt spikes or declines.
- **Signal (Yellow Line):** The yellow line depicts Signal use, which is continuously low over the time period, implying a considerably smaller user base. The line is practically flat, showing that the app's usage has not increased significantly over the years.

4. *Comparative Analysis on Usage Trends:*



Messaging App	Market Dominance	User Preference	Possible R Tre
WhatsApp	Dominates market; widespread adoption	High user preference	Strong m feature set, a eff
Telegram	Steady growth; resilient user base	Growing discovery by new users	Loyal us gradual grow rich pl
Signal	Significantly lower usage; niche market	Focused on security-conscious users	Privacy focu less marketi experience c oth

Table 3: Comparative Analysis on Usage Trends

5. *Market Share of WhatsApp, Facebook Messenger, Telegram Signal according to Statista*

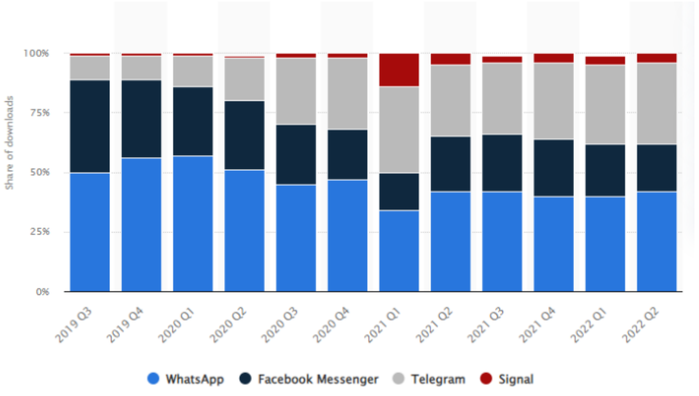


Fig 3: Market Share of WhatsApp, Facebook Messenger, Telegram and Signal

- **WhatsApp (Blue):** WhatsApp has continuously maintained a sizable market share throughout all quarters. It looks to be the dominating player, with the blue part taking up the majority of the space in each stacked bar.

- **Facebook Messenger (Grey):** As indicated by the grey regions, Facebook Messenger has a sizable market share. While it is somewhat lower than WhatsApp's share, it has been reasonably stable over the quarters, showing a solid and consistent user base

- **Telegram (Black):** Telegram, represented in black, has a lesser but noteworthy market share. There may be minor variations from quarter to quarter, but the platform remains active continuously.

- **Signal (Red):** Signal has the smallest market share, as indicated by the red sections. Despite being the smallest, there is a noticeable uptick in the later quarters, indicating growing user adoption, which could be driven by increased awareness of privacy issues.

6. *Monthly Active Users (MAU) in Millions according toStatista*

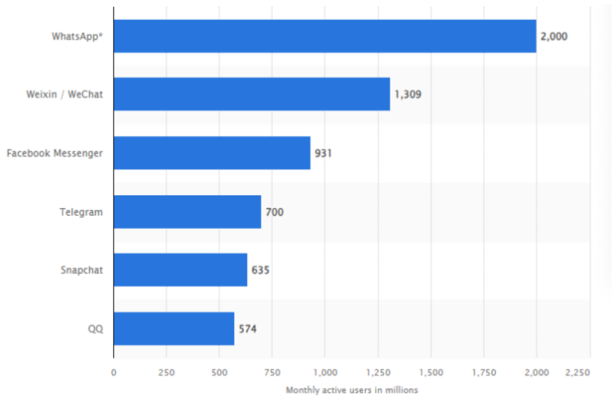


Fig3: Monthly Active Users (MAU) in Millions

This bar graph depicts the number of Monthly Active Users (MAU) for a selection of popular messaging apps, as provided by Statista:

- **WhatsApp**: WhatsApp is the most popular messaging app in the world, with an amazing 2,000 million MAU.

- **Weixin / WeChat:** Weixin (also known as WeChat outside of China) has 1,309 million MAU, suggesting its strong position, notably in the Chinese market.

- **Facebook Messenger:** With 931 million MAU, Facebook Messenger ranks third, demonstrating its importance in worldwide digital communication.

- Telegram is next on the list, with 700 million MAU, indicating sizable user base for the privacy- focused messaging app.

- **Snapchat:** Snapchat shows 635 million MAU, highlighting its popularity, especially among younger demographics.
- **QQ**: Lastly, QQ, another messaging service popular in China, shows 574 million MAU.

Analysis:

The data depicts a varied landscape of messaging applications, each with its own market sector. The popularity of WhatsApp and WeChat indicates their widespread adoption and incorporation into regular communication, particularly in their respective core markets. The high MAU of Facebook Messenger highlights the benefit of integrating with the Facebook social media network. Telegram's large user base reflects a rising concern for privacy and security among users, whereas Snapchat's MAU reflects its popularity among people who like multimedia messaging. Finally, QQ's presence highlights the company's localized dominance in the Chinese market

V. SYSTEM DESIGN

1. *App Architecture*

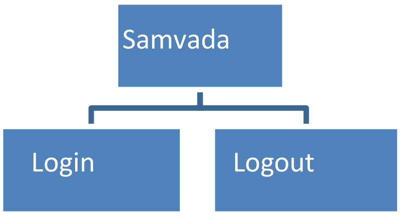


Fig. 5: App Architecture

" **Samvāda**" is the name of the app. This app is primarily divided into two main parts:

Login Stack: This section likely handles the functionalities and processes related to user login. It may include user authentication, sign-in procedures, and possibly account creation or password recovery options.

Logout Stack: This section presumably deals with the operations related to logging out of the app. It might include the process of safely ending a user's session and ensuring data security during logout.

A. Logout Stack

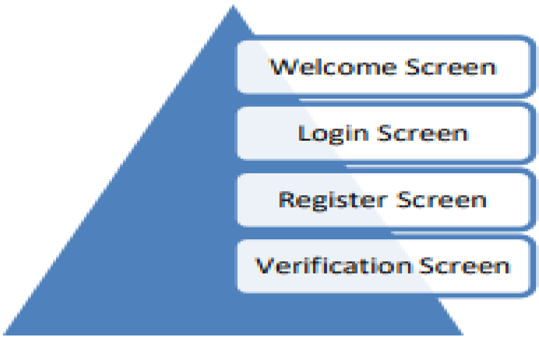


Fig. 6: Logout Stack

- **Welcome Screen**: The initial screenusers see upon opening the app.
- Login Screen: A dedicated screen where existing users can enter their credentials to access theiraccount.
- **Register Screen**: A screen for new users to create an account
- **Verification Screen**: After registration, users might be navigated here to verify their credentialsor account details

B. Login Stack:

- **Home Screen**: The main screen users see after successfully logging into the app.
- **Chat Screen**: A section dedicated to user chats or messaging.
- **All Contact Screen**: A list or display of all the user'scontacts within the app.
- **Account Edit Screen**: Allows users to edit or modify their account information.
- **Wallpaper Screen**: A screen where users can view orselect different wallpapers.
- **Wallpaper Edit Screen**: Users can likely customize or make changes to their selected wallpaper here.

II. User Data Structure

S.NO.	**FIELD**		**Data Type**
1.	UID		(String)
2.	phoneNumber		(String)
3.	timestamp		(timestamp)
4.	userImage		(String)
5.	username		(String)

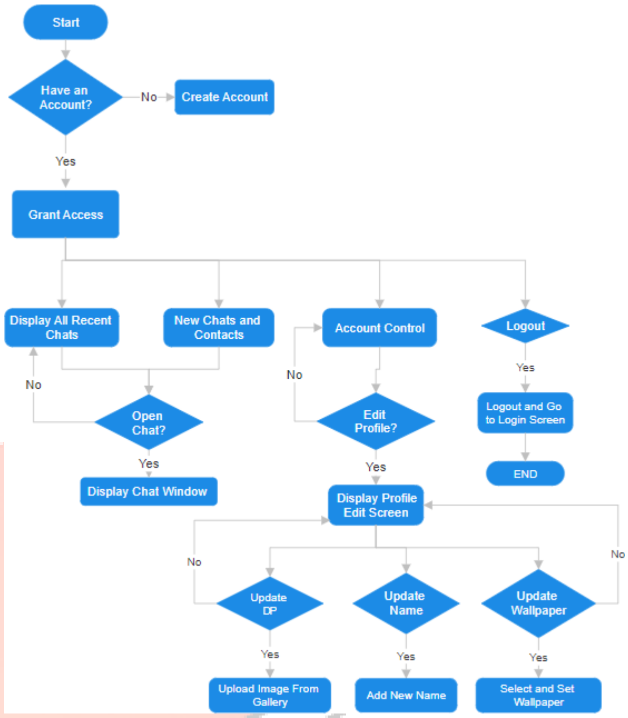
Table 4: User Data Structure

III. Chat Data Structure

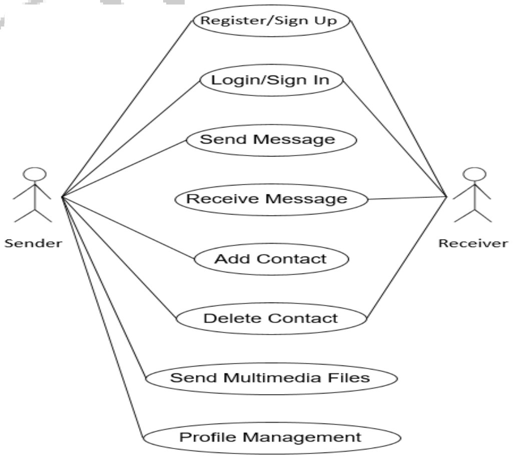
S. No.	**Field Name**	**Data Type**
1.	**sender**	(String)
2.	**receiver**	(String)
3.	**text**	(String)
4.	**timestamp**	(String)

Table 5: Chat Data Structure

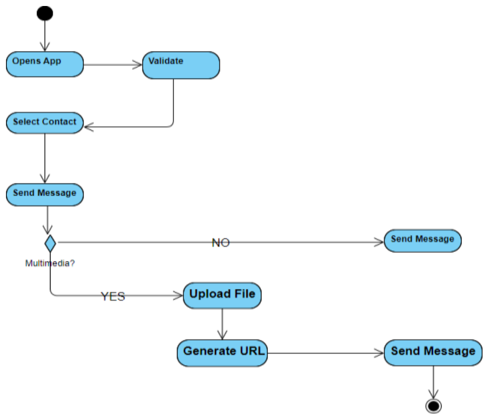
a. User Interface Design Flowchart



b. Use case Diagram



c. Activity Diagram



1. **Start**: The journey begins here.
2. **Have an Account?:** The application checks if the user has an account.
   - **No**: The user is directed to "Create Account." Yes: If the user has an account, the application will "Grant Access."
3. Once access is granted, the user is presented with multiple pathways:

a. Display All Recent Chats: Shows the user's recent chat conversations.

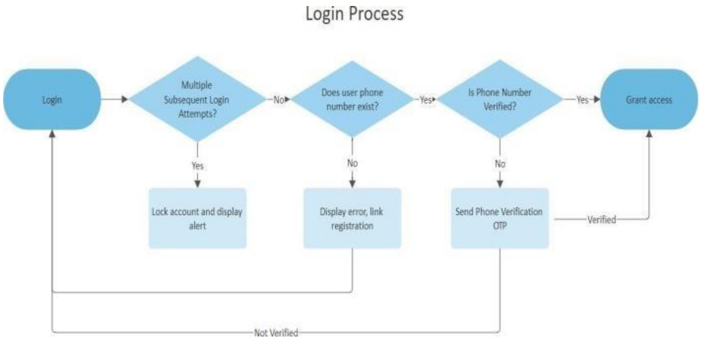
- *Open Chat?:*
- **Yes**: Opens the "Display Chat Window" where the user can view and continue a conversation
- **No**: Returns to the main chat screen

b. New Chats and Contacts: Allows the user to start new chats or add new contacts.

c. Account Control: Gives the user the ability to modify their profile settings.

- □ *Edit Profile?:*
- **Yes**: Takes the user to the "Display Profile Edit Screen" where they can make various changes:
- *Logout*
- **Yes**: The user will "Logout and Go to Login Screen," effectively ending their session.
- **No**: Returns to the main chat screen

Login Process Flow Chart


Login Process
Login
Multiple Subsequent Login Attempts?
No
Does user phone number exist?
Yes
Is Phone Number Verified?
Yes
Grant access
Yes
Lock account and display alert
No
Display error, link registration
No
Send Phone Verification OTP
Verified
Not Verified

Algorithms

For Sending a Message in a Real-time:

Step 1: Establish Database References

- Define the path in the database where chat messages will be stored.
- Create a reference to this path using the Firebase SDK.

Step 2: Prepare the Message

- Collect the message text and any other relevant information (sender ID, timestamp, etc.).
- Trim the message to remove any leading or trailing whitespace.

Step 3: Validate the Message Content

- Check if the message is not empty.
- Optionally, validate the length of the message or any other constraints you wish to enforce.

Step 4: Create a Unique Identifier for the Conversation

- **If** it's a one-on-one chat, you could combine the sender's and receiver's user IDs in a way thatthe combination is always consistent regardless of who sends the message.
- For group chats, use the group ID as the identifier.

Step 5: Send the Message

- Use the push() method provided by Firebase to add the message to the database at thedesignated path.
- Include the message text, sender ID, and a server-generated timestamp in the data to be pushed.

Step 6: Handle Acknowledgements

- Listen for a success response from Firebase to confirm that the message has been sent.
- Handle any errors that may occur during the send operation, such as permission issues or network failures.

Step 7: Update the UI

- After a successful send, update the chat UI to display the new message.
- Optionally, scroll the chat view to the bottom so the new message is visible.

Step 8: Clear the Message Input

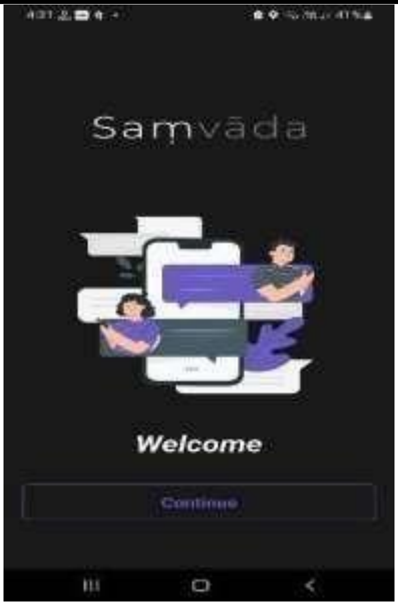
- Clear the input field in the chat UI to prepare for the next message.

Step 9: Maintain Chat Meta-data

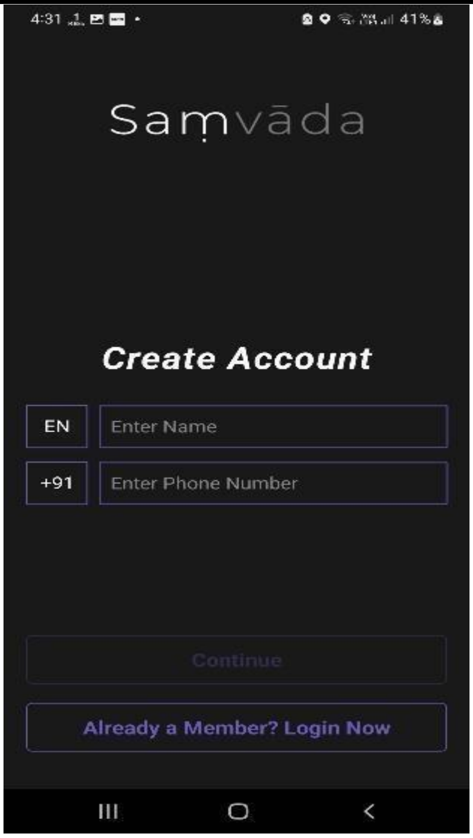
- Optionally, update the last message, time, and unread counts in a separate node in the databaseto make it easy to show recent chats and notifications.

VI. IMPLEMENTATION & RESULTS

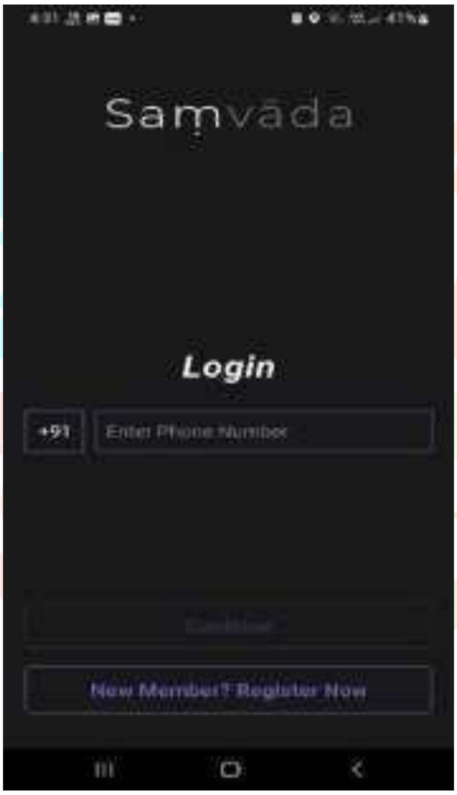
App Screen Shots:



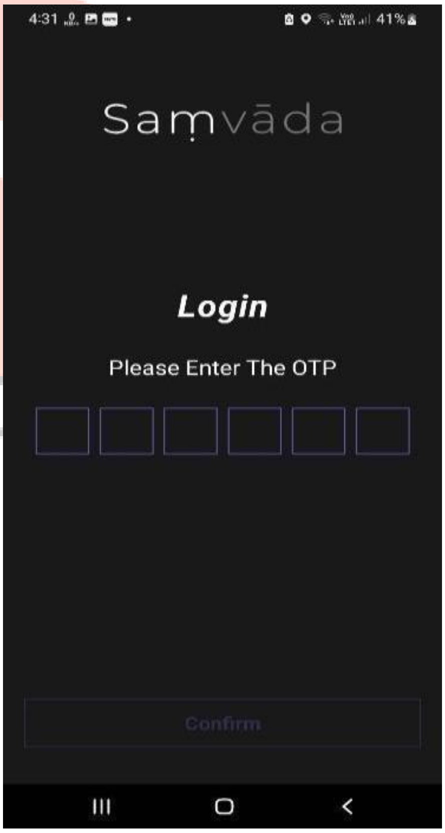
(WELCOME SCREEN)



(REGISTRATION SCREEN)



(LOGIN SCREEN)

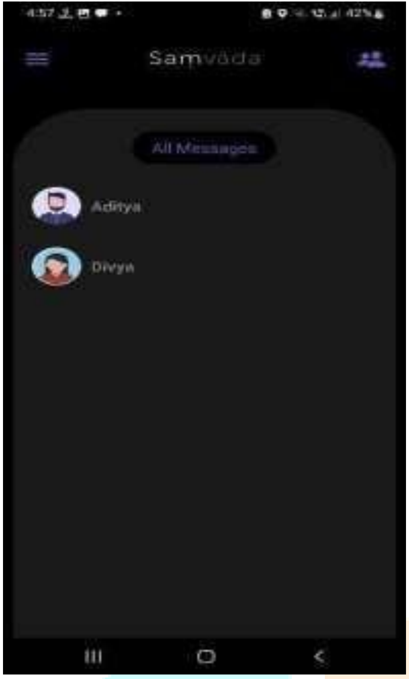


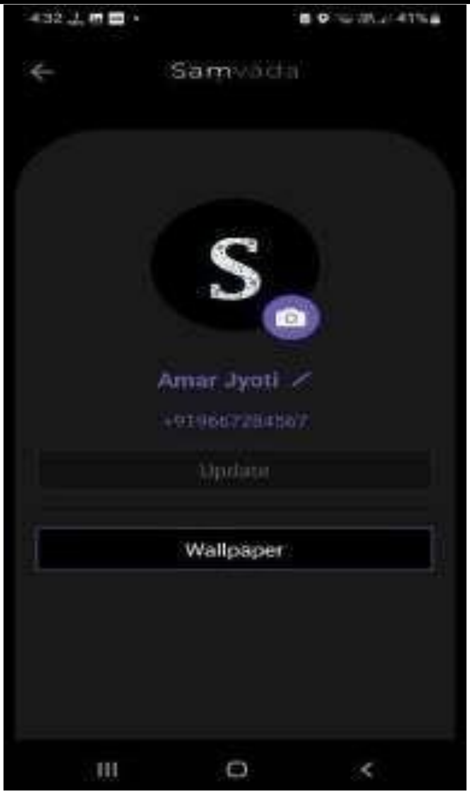
(VERIFICATION SCREEN)

The "Samvāda" app offers an intuitive user authentication process showcased in a contemporary dark mode design. Initiating with the "Welcome Screen," users are visually greeted and led to the subsequent "Login Screen" to input their phone number. For newcomers, the "Registration Screen" facilitates account creation by capturing their name and phone number. The journey concludes with the "Verification Screen," where an OTP ensures user authenticity. This systematic flow ensures a smooth and secure experience for all users.

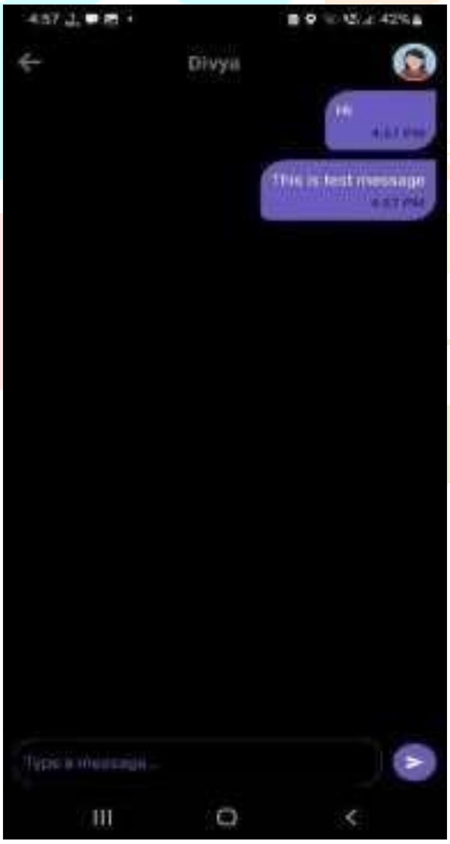
- Login Screen - (Dark Mode)

(HOME SCREEN)





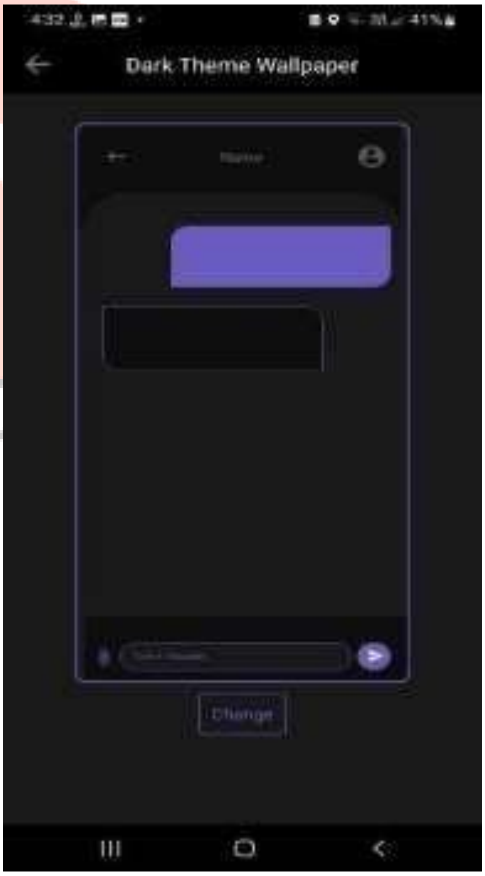
(PROFILE SCREEN)



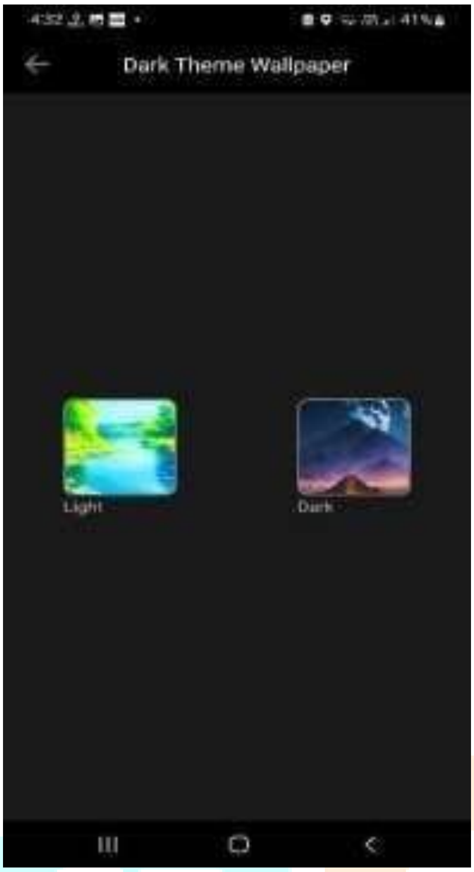
Hi
4:57 PM

This is test message
4:57 PM

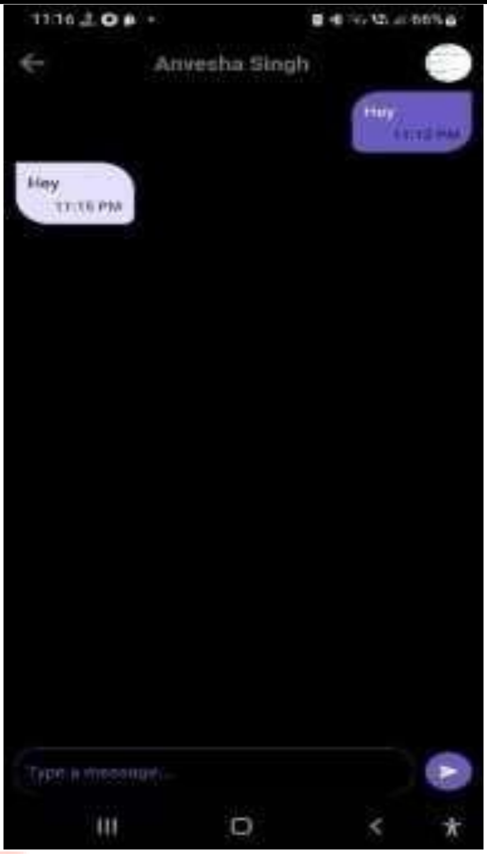
(CHAT SCREEN)



(WALLPAPER PREVIEW SCREEN)



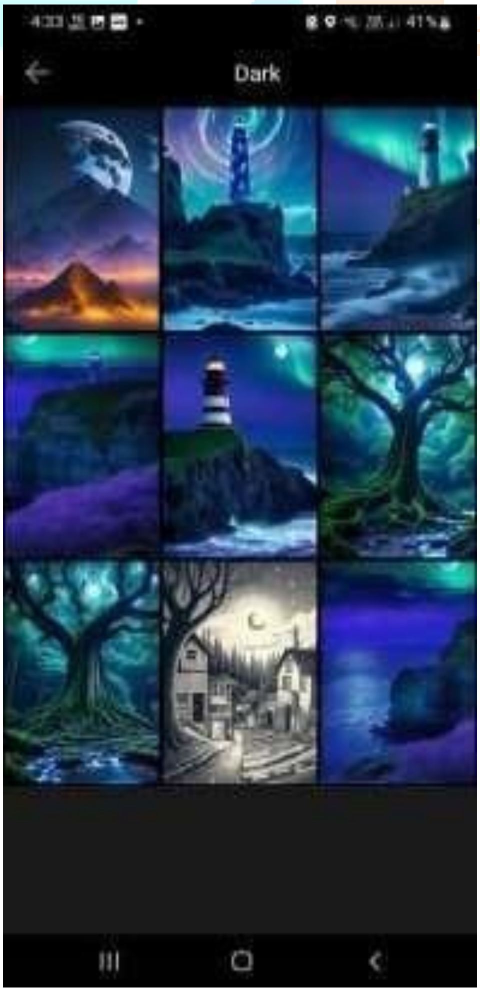
(WALLPAPER)



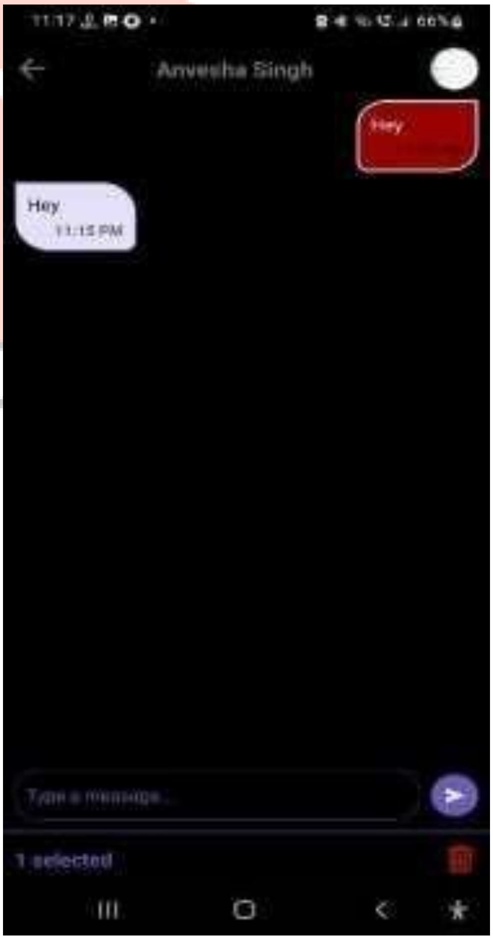
Hey 11:12 PM

Hey 11:15 PM

(CHAT SCREEN)



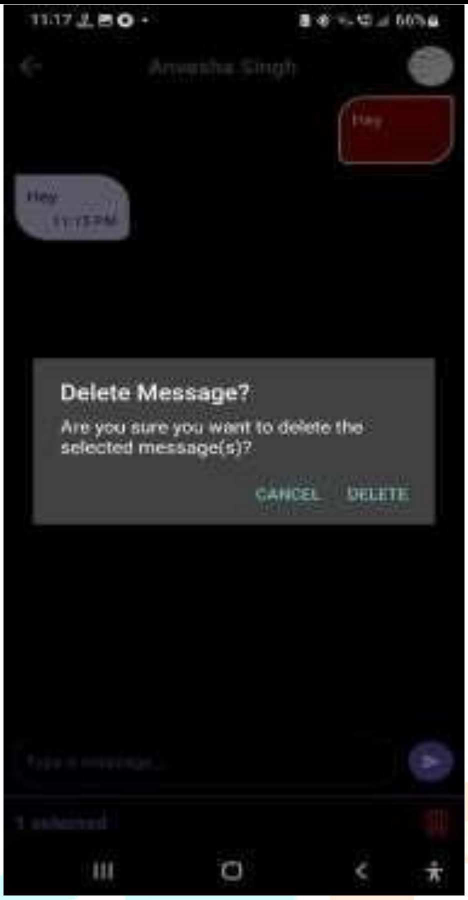
(WALLPAPER SELECTION SCREEN)



Hey

Hey 11:15 PM

(CHAT SCREEN MESSAGE SELECTION)



Hey

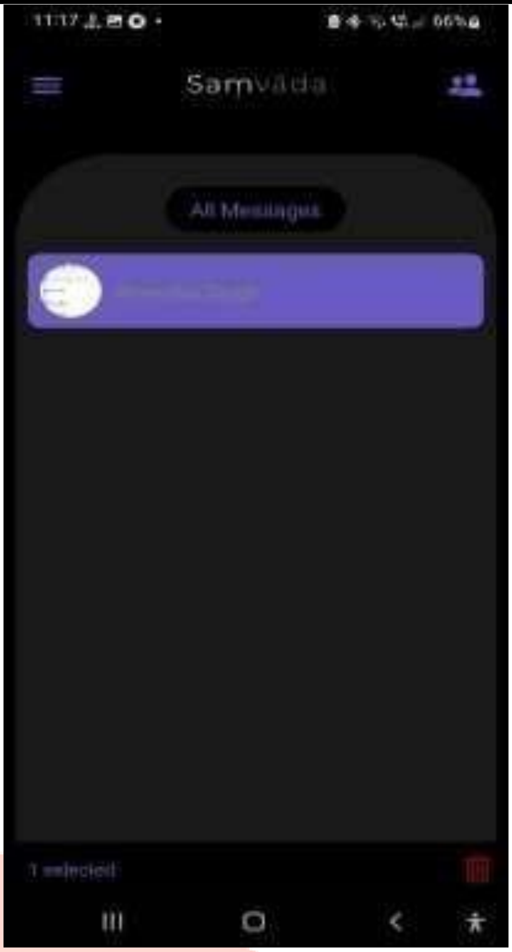
Hey
11:17 PM

Delete Message?

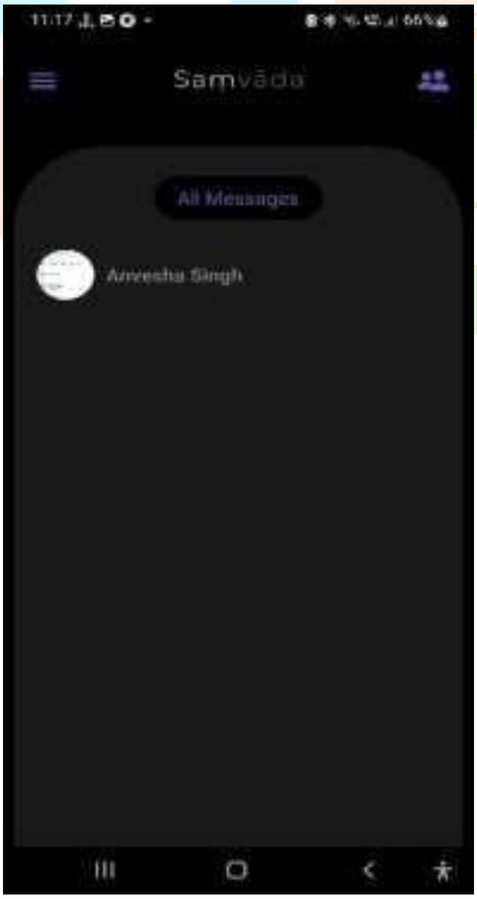
Are you sure you want to delete the selected message(s)?

CANCEL DELETE

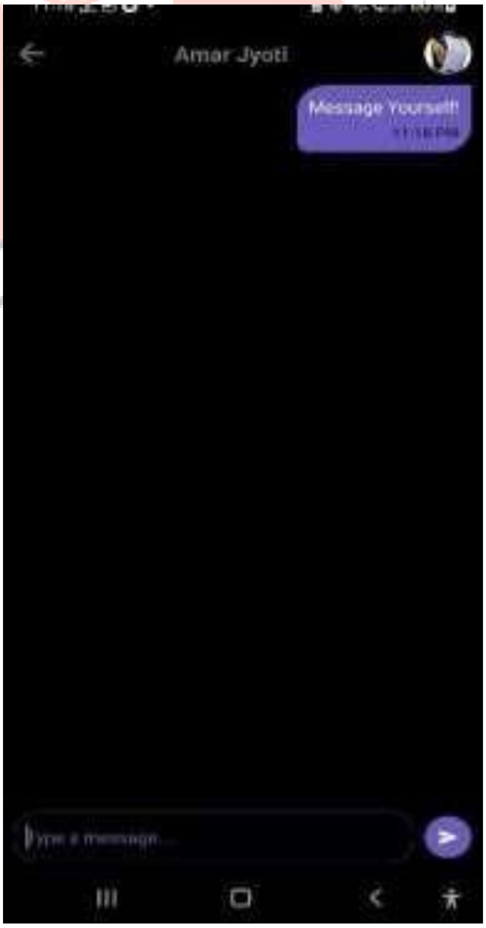
(MESSAGE DELETE)



(RECENT CHAT DELETE FUNCTION)

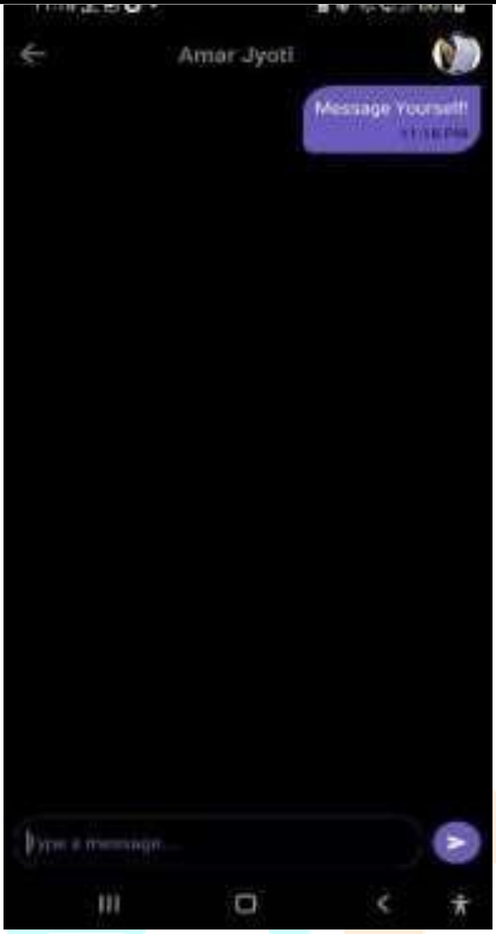


(RECENT CHATS)



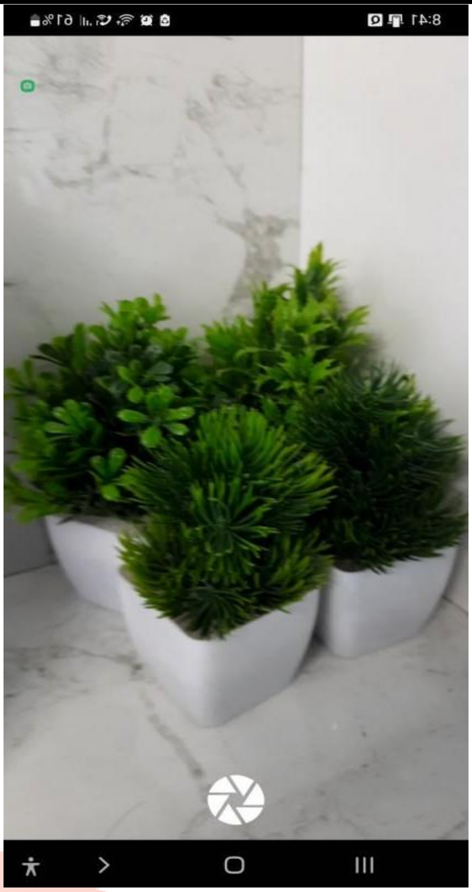
Message Yourself!
11:18 PM

(MESSAGE YOURSELF)

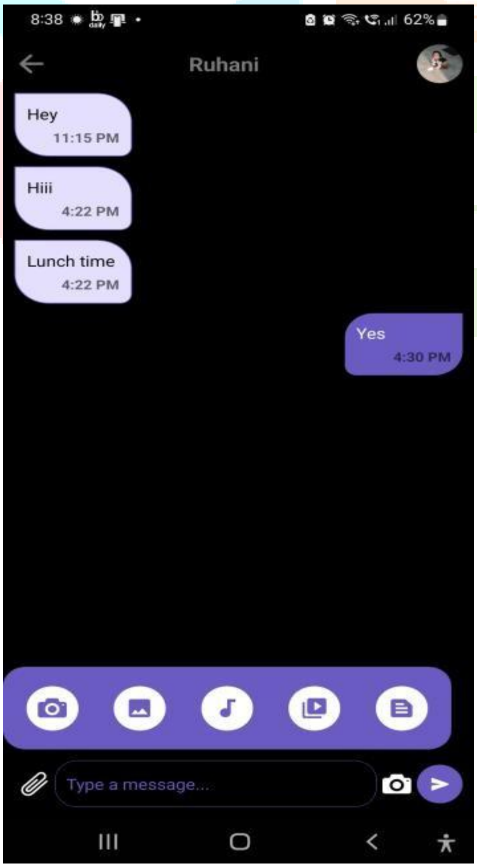


Message Yourself!
11:16 PM

(DELETE RECENT CHAT)



(CAMERA)



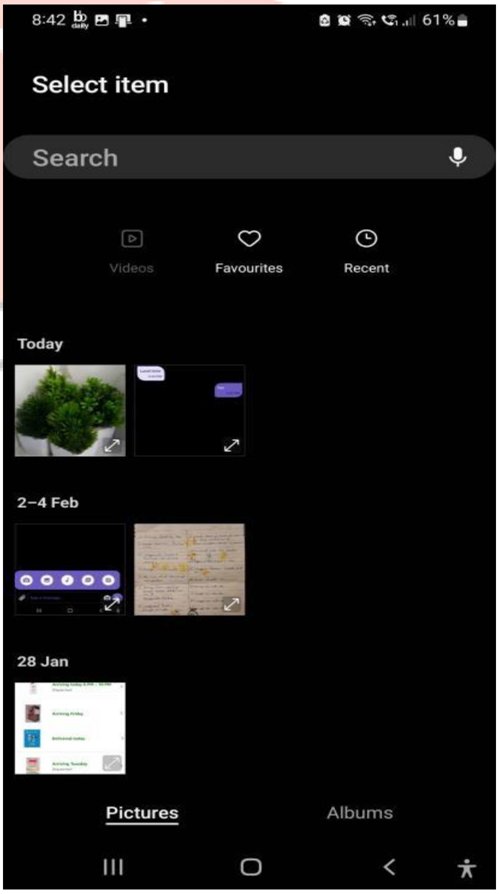
Hey
11:15 PM

Hiii
4:22 PM

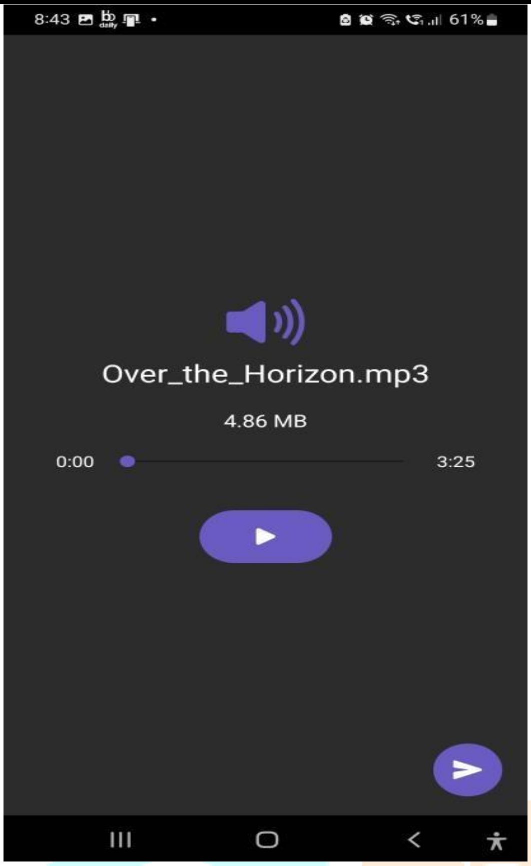
Lunch time
4:22 PM

Yes
4:30 PM

(MULTIMEDIA SHARE OPTION)



(OPEN GALLERY TO SELECT IMAGE)



(SELECT PREVIEW AND SEND)

Following authentication, the "Samvda" software improves user engagement by displaying its major features against a stylish dark mode backdrop. The "Home Screen" consolidates all messages and directs the user to specific chat sessions via the "Chat Screen." The "Profile Screen," where individual profiles can be managed, demonstrates a tailored user experience. The "Wallpaper Preview Screen" provides a preview of the background for customers wanting personalization, while the "Wallpaper Selection Screen" showcases a plethora of bright wallpapers for users to choose and elevate their chat environment. This organized layout offers a user-friendly, dynamic, and visually pleasing journey.

VII. RESULT ANALYSIS

Results:

1. *User Interface and Experience (UI/UX):*
The interface has been designed with a modern and intuitive layout, aiming to cater to the current trends and user preferences. The profile customization and visual adjustments have been integrated to enhance user experience and offer a personalized touch.

2. *User Authentication and Profile Management:*
With the integration of Firebase, the app offer robust and secure authentication system. The sign-up and login processes are streamlined, and users can easily manage and update their profiles.

3. *UI Customization:*
The app provides users with the flexibility to adjust and personalize various aspects of the interface. Such features aim to set the app apart in a market where customization options in messenger apps can sometimes be limited.
Chatting Capability:
Real-Time Messaging: Users mayinstantaneously send and receive messages, allowing for real-time conversation.
One-on-One interactions: The software allows two users to communicate directly, allowing for confidential and individualized interactions.
Message Deletion: Users may remove messages, giving them control over their chat history.
Recent Chat Display: For easy access, the most recent chats are prominently presented in a distinct'Recent' area.

Discussion:

User Interface and Experience (UI/UX):
The importance of an intuitive UI can't be understated, especially in today's competitive app market. Initial feedback from peers suggests the design is on track, but a broader sample size for user testing might offer more comprehensive insights.
User Authentication and Profile Management:
Security remains a top priority, especially with increasing concerns over data breaches and privacy. Regular updates and perhaps third-party audits might be beneficial to ensure the app's authentication system remains secure.
UI Customization:
While the customization features are promising, it's essential to ensure they don't compromise the app's performance. Rigorous testing across different devices will be crucial.
Chatting Capability:
The chat function will be the heart of the app. As it's developed, considerations for features like encryption, multimedia sharing, and other modern chat functionalities will be vital.
Market Launch and User Feedback:
A phased launch, starting with a beta version for select users, could be a strategic approach. This would allow for real-world testing and feedback, ensuring any glitches or shortcomings are addressed before a full-scale launch.

VIII. CONCLUSION

Our Android chat app development adventure has been both tough and enjoyable. As we progressed through the design, development, and refinement stages, it became clear that the app's emphasis on user customization and powerful backend connection would set it apart in the modern messaging world. While the app has made tremendous progress in terms of UI/UX design and security measures, the lack of fundamental chatting functionality highlights the project's developmental nature. Nonetheless, the preliminary work accomplished thus far provides a solid framework for future developments.

IX. Future Scope

- **Chat Functionality:** The impending inclusion of the chat feature will be revolutionary. Future versions will look into group chats, and possibly multimedia sharing.
- **Advanced security** measures will grow in tandem with cyber threats. End-to-end encryption and two-factor authentication could help to increase user trust and data security.
- **Voice and Video Call Integration**: Future versions of the app may include voice and video call functionality to make it a more versatile communication tool.
- **Integration with Other Platforms**: Our software has the ability to integrate with other popular platforms or services, hence increasing its utility and user base.
- **Machine Learning and AI:** Using chatbots for instant customer service or AI-driven algorithms to advise user customisation can be investigated to

improve the user experience.

- **Feedback and Iterative Development:** Collecting user input after launch will be critical. This will aid with the app's refinement, issue fixes, and the introduction of features that are popular with the user population.
- **Expansion to Other Operating Systems:** While our first focus was on Android, the app's popularity may pave the road for its development on other platforms, such as iOS.

REFERENCES

Mittal, J., Garg, A., & Sharma, S. (2020). Online Chat Request. International Journal of Research in Engineering, IT and Social Sciences, ISSN 2250-0588.

Gayathri, R., & Kalieswari, C. (2020). [Title of the paper if available]. International Journal of Engineering and Advanced Technology (IJEAT).

Henriyan, D., Subiyanti, D. P., & Fauzian, R. (2016). Designing and deploying a real-time web-based chatserver. International Conference on Engineering and Technology (ICSET).

Dandane, T. C., & Jogalekar, U. A. (2015). A study of web chat applications with its performance measurement parameters. International Journal of Engineering Sciences & Research Technology, 4(1), 300.

React Native Firebase Group, "Realtime Database Usage with React Native Firebase," Accessed November 2023, https://rnfirebase.io/database/usage.

React Native Firebase Group, "Phone Authentication with React Native Firebase," Accesses.

React Native Firebase Group, "Cloud Firestore Usage with React Native Firebase," Accessed October 2023, https://rnfirebase.io/firestore/usage.

React Native Community, "Getting Started with ReactNative," Accessed October 2023, https://reactnative.dev/docs/getting-started.

React Navigation Contributors, "Stack Navigator in React Navigation," Accessed October 2023, https://reactnavigation.org/docs/stack-navigator