AALAP – A CHATTING APPLICATION

By

Proloy Karmakar
Roll: 1907051

&

Md. Sakib Hasan
Roll: 1907052

Supervisor:

Kazi Saeed Alam
Lecturer
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology

Signature

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
November 2023

Acknowledgements

We developed our project, "Aalap - a chatting application," as part of the CSE 3200: System Development Project course. We would like to thank our supervisor, Kazi Saeed Alam, Lecturer in the Department of Computer Science and Engineering at Khulna University of Engineering & Technology (KUET), for his invaluable advice and support. We would also like to thank Sunanda Das, Assistant Professor in the department of Computer Science and Engineering at Khulna University of Engineering & Technology (KUET). Without their help, this project would not have been achievable.

Authors

Proloy Karmakar

Md Sakib Hasan

Abstract

"Aalap" is a project that presents an easy-to-use chat application that aims to make communication between users effortless. Our system strives to create a personalized and interesting chatting experience in response to the increasing need for effective and feature-rich messaging services. Real-time communication, connection, and information sharing among users promotes teamwork and a sense of connectedness. The application's user-centric design ensures accessibility and usability. Real-time messaging and a dynamic user interface are important characteristics. The development of the project is based on best practices in software engineering and system design and development principles. "Aalap" provides a platform where users may easily communicate and engage, marking a step forward in improving digital communication experiences.

Contents

List of Figures

1 Introduction

1.1 Background

The demand for effective and user-friendly communication tools is constantly rising in the modern digital environment. The explosion of virtual connectivity requires creative ways to improve instant messaging user experiences. In answer to this need, "Aalap" appears, aiming to offer a smooth and feature-rich chat program for users to communicate and engage.

1.2 Objectives

Provide a User-Friendly Interface:
To improve the user experience while interacting, create a visually appealing and friendly user interface for "Aalap".

Real-Time Messaging:
To enable fast communication between users, put in place a reliable real-time messaging system.

Scalability:
Create a scalable application so that it can accommodate future increasing in the number of users without experiencing performance issues.

Responsive Design:
To make sure that application works well on a range of screens and devices to encourage accessibility for a wide range of users.

Future Improvements:
Establish the framework for future feature additions and integrations while taking updates into account to satisfy changing user requirements.

1.3 Scope

The project's scope includes every stage of "Aalap's" development, from ideation to execution. Post sharing and real-time text messaging will be supported by the application. The architecture of the system will be scalable to handle future user base expansion. The project establishes the framework for upcoming improvements and integrations, even though the first release will concentrate on key features. Maintaining established security protocols to protect user data and privacy is also included in the scope.

1.4 Unfamiliarity of the problem

The motivation for the project was the realization that there was a growing need for a cutting-edge, user-focused chat application. Findings showed that there was a need for a platform that prioritizes user experience and seamlessly combines post sharing and real-time messaging. The problem is unfamiliar because of the subtle complexities involved in developing a unified and effective communication tool that accommodates a wide range of user preferences. addressing the changing needs of a changing user base presented difficulties not anticipated when the project began. It required extensive study, original problem-solving, and a dedication to keeping up with the rapidly changing landscape of communication technology. The foundation of the project is its reaction to the new environment, with the goal of providing a solution that not only satisfies users' expectations in digital communication.

1.5 Project planning

The project planning for the feature-rich and user-friendly chat application Alaap involves taking a methodical approach to creating a platform. The schedule includes the following stages: requirements collecting, system design, user interface development, online user tracking, chat functionality integration, post sharing, editing and updating profiles, newsfeed implementation, message notifications, testing and debugging, deployment, and continuous maintenance. This organized strategy guarantees the timely and effective completion of the project.

A Comprehensive Timeline for Developing Alaap: A Chatting Application

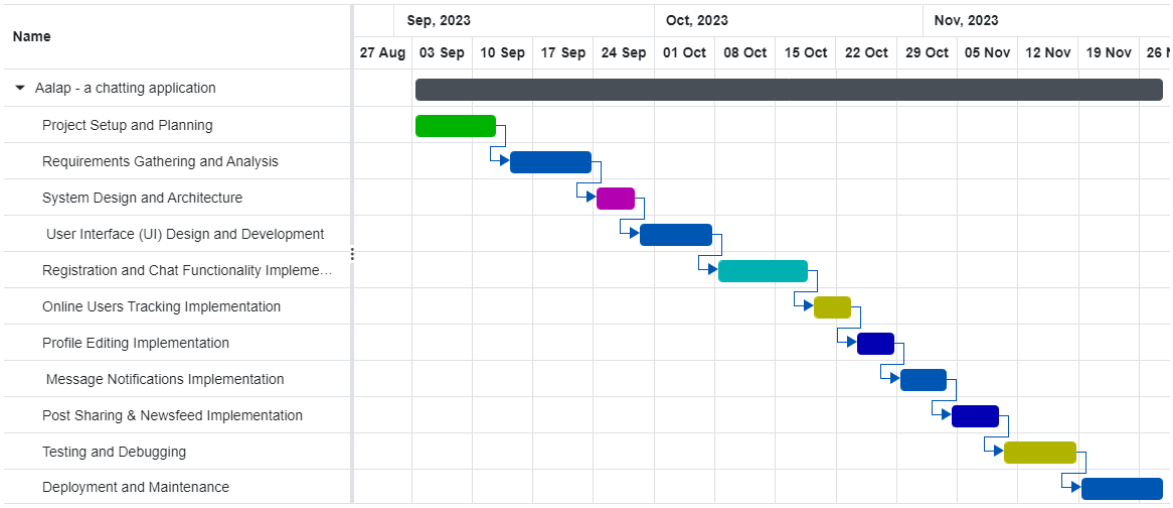

Figure 1.1: Gantt chart of project planning

This Gantt chart shows the proper time division of our application development. It is seen in the chart that; we have done total 11 steps in our application development. Time taken to develop each step is showing in the Gantt chart according to the date.

2 System Design

2.1 Analysis of the system

Workflow diagram

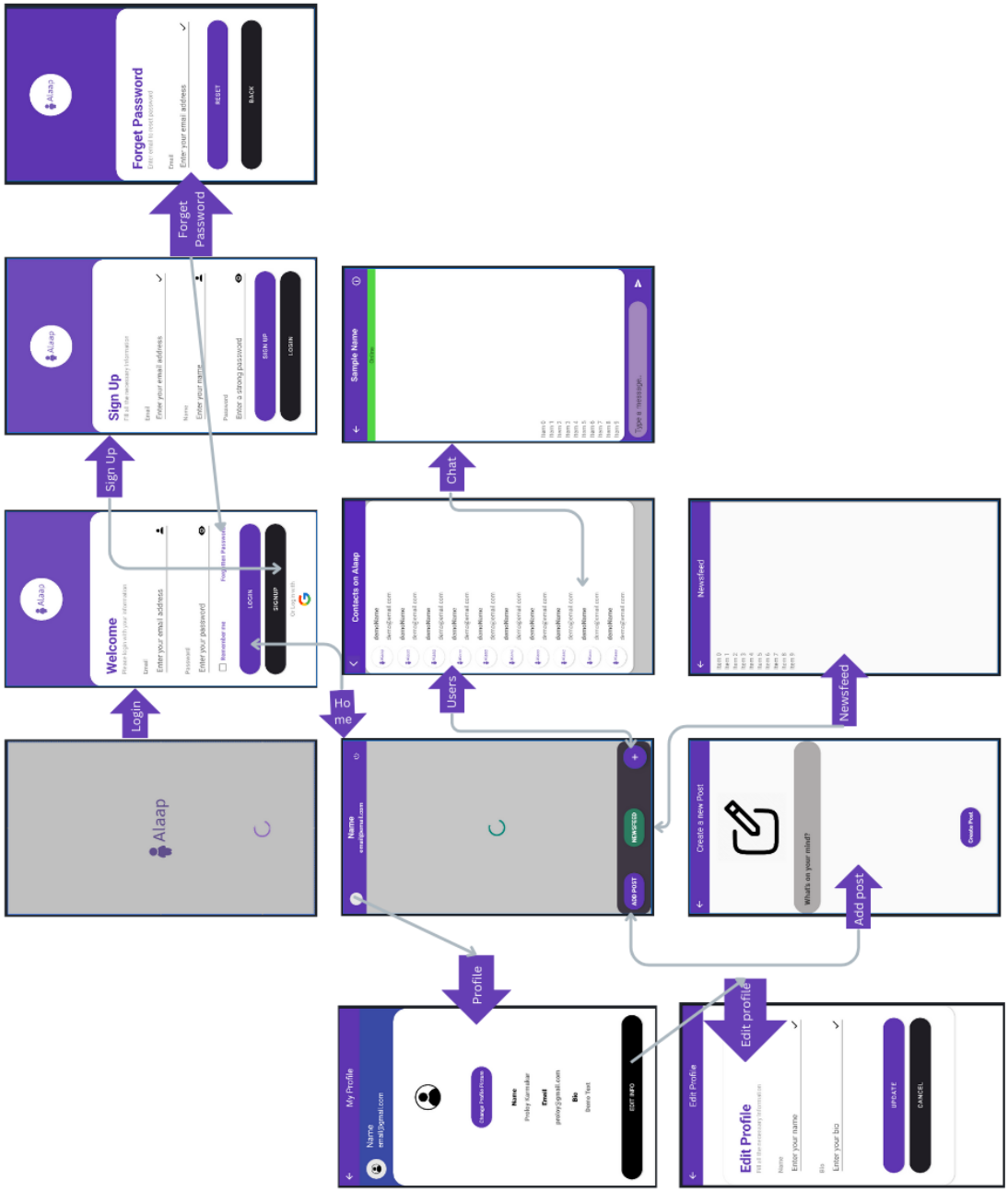

Figure 2.1: Workflow diagram of Aalap

This diagram shows the complete workflow of our application. From which page user will pass to the next page is shown in this diagram by arrow sign. It can be seen that, at first Splash screen will appear, then login page, from there by clicking sign up button Sign up page will appear and so on. More details explanation is included in the next section.

Use Case Diagram

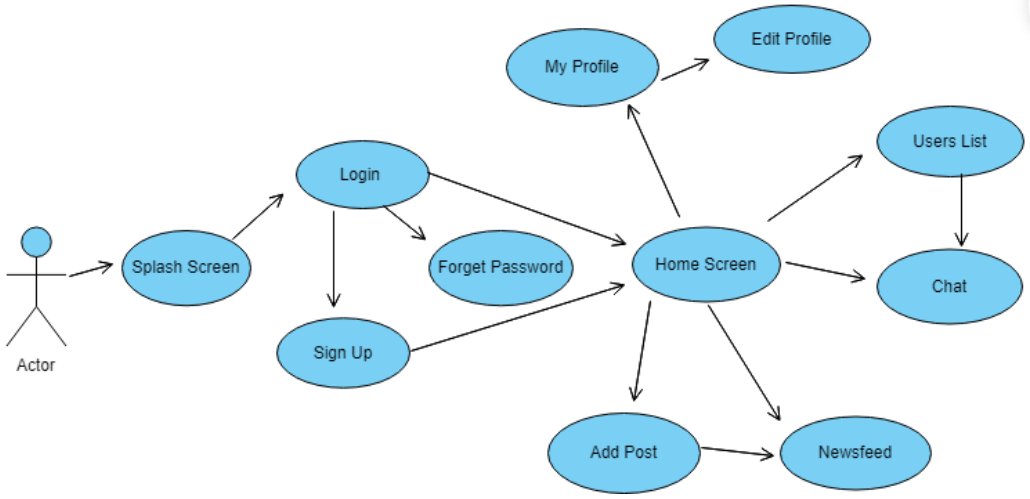

Figure 2.2: Use Case Diagram

Basically, a use case diagram shows, the interaction between user and the application. It also determines, how much a user get satisfaction while interacting with an application. Its main focus is to find that, how much user friendly an application is.

In our case, it is shown that, a user first interacts with splash screen. From There Login page and sign-up page. Then the user gets into the Home Screen of the application and so on.

2.2 System architecture

UML Class Diagram

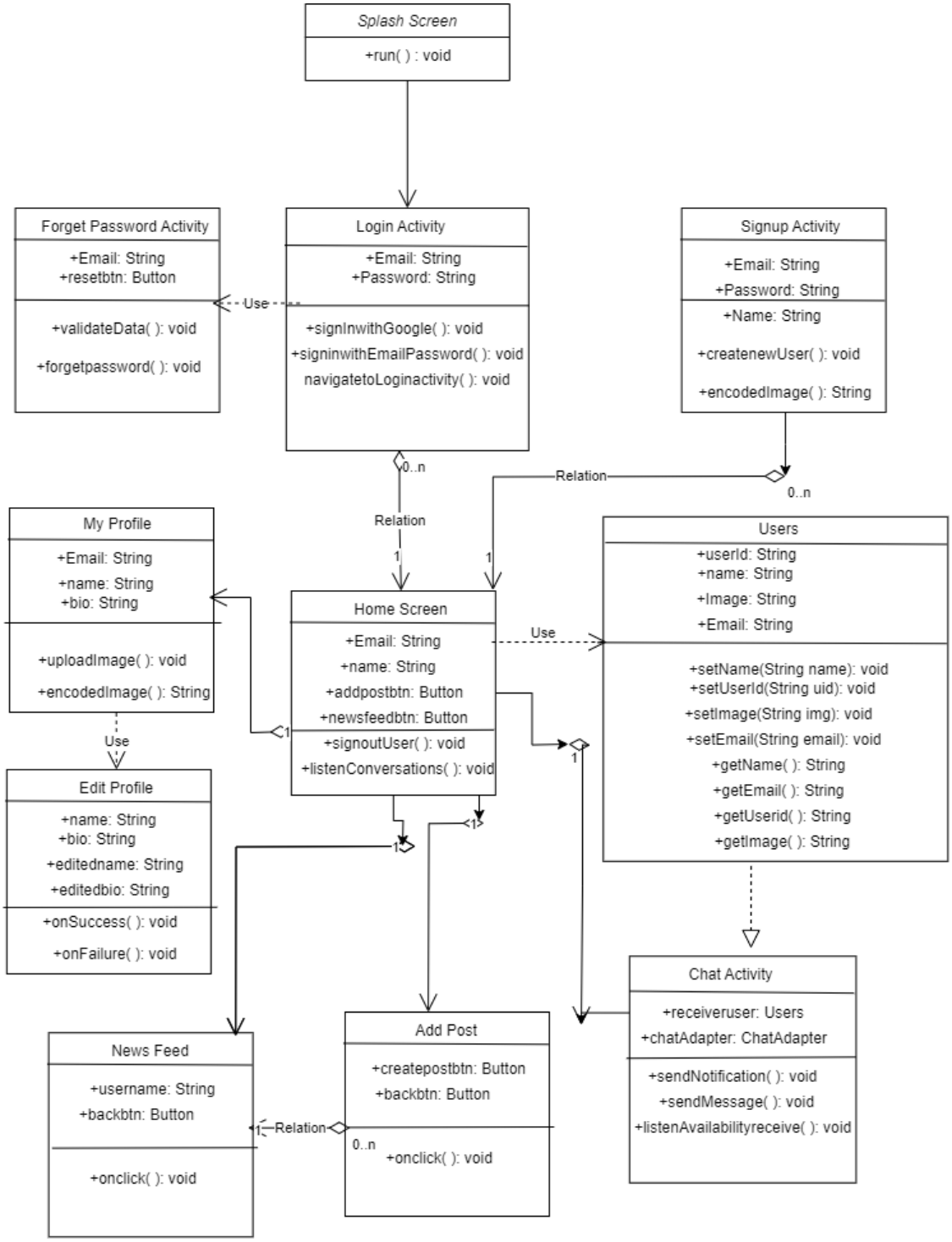

Figure 2.3: UML Class Diagram

UML class diagram is the visual representation of the architecture, design and implementation of a complex software system.

This class diagram shows that, how many classes does this application has, which class contains which types of properties and methods and relationships between every class by generalization, specification, assertation etc.

2.3 Tools / Platform used

Since "Aalap" chatting application is an android application, we had to use android studio as our main tool. We had to use some other tools like firebase, retrofit etc. Detailed description of each tool is given below:

2.3.1 Android Studio

Description: Android Studio is the reliable environment (IDE) for Android app development. It consists of a complete set of equipment for designing, coding, checking out, and debugging Android programs.
Usage: Android Studio served because the primary IDE for developing the “Aalap “chatting utility, providing a well development surroundings with capabilities tailored for Android app development.

2.3.2 Firebase Firestore

Description: Firebase Firestore is a NoSQL cloud database provided by way of Google Firebase. It lets in for real-time facts synchronization and seamless integration with Android packages.
Usage: Firestore turned into employed to store and retrieve chat messages and user Information in actual-time, allowing efficient and synchronized communication inside the utility.

2.3.3 Firebase Authentication

Description: Firebase Authentication offers a stable and easy approach for user authentication. It supports numerous authentication strategies, together with e-mail/password, Google Sign-In, and greater.
Usage: Firebase Authentication turned into utilized to authenticate and control consumer get entry to to the chatting utility, making sure a steady and personalized enjoy.

2.3.4 Firebase Realtime Database

Description: Firebase Realtime Database is a cloud-hosted database that supports real-time data synchronization. It allows for the storage and retrieval of data in a JSON format.
Usage: Realtime Database complemented Firestore, serving specific needs such as storing presence information and other real-time data crucial for the functioning of the chatting application.

2.3.5 Firebase Cloud Messaging (FCM)

Description: Firebase Cloud Messaging is a cloud solution for messages on iOS, Android, and web applications. It facilitates the reliable and efficient transmission of messages to devices.
Usage: FCM was integrated into the application to enable push notifications, ensuring users receive timely updates and messages even when the app is not actively in use.

2.3.6 Retrofit

Description: Retrofit is a popular HTTP client library for Android, simplifying the process of sending network requests. It is commonly used for API communication.
Usage: Retrofit was specifically employed for handling notification requests, providing a streamlined approach to interact with the server for sending and receiving notifications within the "Aalap" application.

3 Project Implementation

3.1 System implementation

3.1.1 Splash Screen

A splash screen is basically a welcome page of an android application. It starts at the time of starting of the application. It shows app logo or some others welcoming massages. And, it lasts for a short amount of time and after the duration is over, the next screen according to the application login will appear.

From figure no 2.1.1, in our application, there is our app logo and a progress bar in the splash screen page. And, after A short duration, login page will appear and progress bar will also be disappeared.

3.1.2 Login Page

From figure no 2.1.2, this login page contains two Edit text used for entering email address and password from the user, two buttons. One is login button which is used to ensure login after providing valid email address and password. And the later is, signup button, which is used for a new user who doesn't have an account already. By clicking this signup button, the signup page will appear where the user can create his/her account by providing all the information's required according to the requirements.

We have also used another functionality for logging into the application which is "Login with Google" option. By clicking the "Google Button" google intent will be appear and from there user can login by providing a valid email address.

There is a checkbox named "Remember me". If an user checked the checkbox while logging into the application, he/she doesn't need to login again to enter into the application later.

There is also an option to reset password if an user forget his/her password. While clicking "Forgotten Password" options there will appear a page where user can

provide his/her email address and a password reset confirmation email will be sent to the email address given by the user immediately.

We have used “Firebase Authentication” to authenticate that whether an user is authorized or not. We get the email and password from the edit text and match using firebase authentication methods. It is must that, user should provide a valid email address with maintaining proper email format and a valid password to get logged into the application. Otherwise, the application will show an error message through the Toast option.

3.1.3 Signup Page

From figure no 2.1.3, this sign-up page is for the users those who are new to the application. So, they need to create an account for each user as their own to get into the application.

In this page, there has 3 edit texts. First one for entering email address, second one for entering name and the last one for entering a password. The password must be at least of 6 characters.

There have two buttons. One is “Sign Up” button which is used to create account for an user. But before creating account for an user, it will check such conditions for example where the email address is proper formatted or not, whether password is at least of 6 characters or not, all the three information is provided or not.

If all the required information is provided properly, we use firebase authentication methods “createuserwithemailandpassword” to create an account for the user. If successfully account is created the completion message will be shown by a Toast message and the user will be passed directly to the “Home Screen” of the application.

Otherwise, a particular error message will be shown according to the requirements failure Via Toast message.

The second button is “Login” button. By clicking this button, user will be passed to the Login page of the application. The uses of this button is, if an user has an account already, but he/she unwantedly comes to this Sign Up page, when he realizes that, he can go back to the login page and can logging into the application.

3.1.4 Home Screen

From figure no 2.1.6, after successfully Login or Sign Up the user will be in this Page of the application. In this page, there is a top bar which contains the profile icon image of the user, user’s name and email address in which the user is logged in recently. And there is a Logout button at the top right corner of the top bar. By clicking this button, the user will be logged out and passed to the Login page. Thus, he/she needs to login again to interact with the application again.

There is a Bottom bar. In which there are 3 options. First one is a button named “Add Post” button. By clicking this button, a page will appear in which user can write his/her feelings or thoughts and can create the post.

The second button is “News-Feed” button. By clicking this button, news-feed page will appear and there the user can see any other user’s posted thoughts or feelings including his/her ones.
At the middle of this Home Screen page, recent chatting person’s list will be shown as a list and user can chat with them by clicking from the list of persons whom he has messaged already.

At the bottom right corner of the bottom bar the is a Button named “Contacts” Button. By clicking this the user can be able to see all the contacts of his/her in a page as list and from there also he/she can exchange messages with them.

We use “Firebase Fire-store” to store the image, name and email address of the user. Each user has different user-id. We first get the current user id by creating an instance of the Firebase Fire-store . Then using this user id, we retrieve the data of the current user’s from the Fire-Store like image, name and email and set them into

the appropriate view components into the Home Screen page we defined in our XML file.

3.1.5 Forget Password Page

If a user forgot his/her password, he/she can reset password by providing a valid email address.

From figure no 2.1.4**,** there is an edit text to able to user to enter the email address. There is button named “Reset” button. By clicking this button, it will check that whether the email address is valid or not. If it found to be valid it will immediately send an confirmation email for resetting user’s password.

There is also another Button named “Back” button. By clicking this button, user will go back to the previous page which is nothing but Login Page.

3.1.6 Users Page

From figure no 2.1.7**,** this page will show all the contacts or all the users of a particular as a list and the user can exchange their thoughts and emotions by messaging each other.

Here each user’s name and email address will be shown. We use Firebase Fire-store to retrieve their image icon, username, and email address from the firebase fire-store by getting an instance of the Firebase Fire-store and user of the users. Users’ information is stored in firestore under “users” collection. Each user is uniquely identified by using UID. Under the UID sub section the details of users is saved.

A recycler View is used to display the users list. We retrieved the users list from firestore and using a custom adapter showed the list using the recycler view. Anyone who registers in the application is shown in the users list.

3.1.7 Chat Page

From figure no 2.1.8**,** In this page, users can exchange messages with each other. This is done by FCM (Firebase Cloud Messaging). We stored the chats in firestore under “chat” section. Here, sender id and receiver id are stored to identify the sender and receiver of the message. Also, the timestamp is stored to show the time when the message was sent. We also have another collection called “conversations” in firestore to store all the details of sender and receiver. The last message is saved here to show on the home page. Each conversation is also given a unique id to identify.

Here, user can be able to find out that whether another user is in online or offline right now. If a user is in online, the other users can see a sign written “online” with a background color as green. The messaging time will be shown at Realtime. This functionality was done by adding a field called “availability” under users. It stored when the user was active in the application.

The messaging functionality is done by “Cloud Messaging FCM” of Firebase. We first tested FCM by sending tokens from firebase. The application was receiving the tokens successfully. The messages are stored in the Fire-store and retrieved from there when messaging. We designed two XML layout files for sent message and received message. Then matched the sender id when a user sent a message. If a match was found then we used the sent message XML layout to show the messages of the sender. Similarly, the receiver id was matched to whom the message was being sent. Opening a conversation with a user provided the receiver id. The recycler view was stacked from end to show the last message.

The user’s name will be shown at the top of the screen. We retrieve the user's name from the Firebase Fire-store according to the user’s current user id.

There is a “Back Button Icon” at the top left corner of the page. By clicking this button, user will be passed to the previous which is the “Users Page”.

3.1.8 Add Post Page

From figure no 2.1.10, in this page, user can be able to share his/her thoughts as a post. He/she can write the post and there is a button named “Create Post”. By clicking this button, the text or the feelings or the thoughts of the user will be posted and it will be added to the “Newsfeed” page. Then other users can see this post.

We get the text from the Edit Text and store it into “Realtime Database” by creating a child “Post”. And retrieve all the posts into the “News-Feed” page.

There is a “Back Button Icon” at the top left corner of the page. By clicking this button, the user will be passed to the previous page which is nothing but “Home Screen” page.

3.1.9 Profile Page

From figure no 2.1.5**,** on this page, user can be able to see his/her own profile. Here, at the top bar, there is a image icon, a name text field and an email text field. There is a Profile picture image view at the middle of the page. Below that, there is Name text filed, Email text filed and Bio text field.

We retrieve this information from the Firebase Fire-store and set them into the appropriate XML components defined in the XML file.

There is a Button at the bottom of the Profile Picture image view. By clicking this button, image intent will be opened and from the phone’s gallery user can update their image as new Profile Picture.

By using a firebase authentication variable, we get the current user id by using “.getinstance( ).getuid( )” method.

We first get an instance of the Fire-store and by this we update the image into the collection name “users” into the Fire-store.

But before that, most importantly, we convert the image into String and then save the string into the fire-store.

As soon as, the user update his/her profile picture, his/her round image icon at the top right corner will also be updated.

There is a button at the bottom of the page named “Edit Info”. By clicking this button, the user will be passed to another activity where a user can edit his profile information like name, bio etc.

3.1.10 Edit Profile Status

From figure no 2.1.9, on this page, user can update his/her name and bio. There is one edit text for entering name and another edit text for entering bio from the user.

There are two buttons at the bottom part of the page. The first one is “Update” button. Which is used for the confirmation of updating his/her name and bio. This updated name and bio will be stored in the Fire-Store under the collection “users”. After that this name and bio will also be updated to the previous page’s all name and bio XML components.

The later button is “Cancel” button. By clicking this button, the user will be passed to the previous page which is nothing but “Profile page”.

3.1.11 News-Feed Page

In this page, every user can see the posts posted by different users on their contact list. Here user name is fetched from the Realtime Database. This is done by getting the current user id using the Firebase authentication instance.

The user posts are also retrieved by the same process. To store the name and post we used a Java Class named User Class.
There is “Back Button” at the top left corner of the page. By clicking this button, the user will be passed to the previous page which is nothing but “Home Screen”.

We have used a card view in XML design of the page. So, every post is separated from the other.

3.2 Morality and Ethical Considerations in the Project

3.2.1 Proper Citations and Acknowledgment

Description: The project strongly emphasizes correct citation of sources and acknowledgment of outside contributions in accordance with ethical standards. When using third-party frameworks, libraries, or code snippets, proper credit to the original creators or organizations must always be given.

Significance: Accurate citation guarantees intellectual integrity, recognizes the contributions of others, and fosters an environment of openness and deference among the development community.

3.2.2 Plagiarism Prevention

Description: We are dedicated to stop plagiarism by closely following moral coding guidelines. Every code segment, algorithm, and design principle is created from the ground up, with existing solutions properly referenced when necessary.

Significance: Preventing plagiarism promotes creativity, protects academic and professional integrity, and makes sure that credit is given appropriately. This method also lessens the possibility of facing legal repercussions for infringement of intellectual property.

3.2.3 User Data Privacy and Security

Description: The "Aalap" chatting application prioritizes user data privacy and security. Stringent measures, including encryption for sensitive data, secure authentication processes, and compliance with data protection regulations, are implemented to safeguard user information.

Significance: Respecting user privacy is not only an ethical obligation but also crucial for building trust. Adhering to data protection standards helps protect users from potential security breaches and identity theft.

3.2.4 Transparency in User Communication

Description: The project emphasizes transparent communication with users regarding data usage policies, system updates, and any potential risks. Clear and accessible terms of service and privacy policies are provided within the application.

Significance: Transparency fosters trust between the development team and users, ensuring informed consent regarding how their data is handled and enhancing the overall ethical standing of the project.

3.2.5 Accessibility and Inclusivity

Description: The project aims to be inclusive by prioritizing accessibility features, ensuring that the application is usable by individuals with diverse abilities. This includes considerations for visual, auditory, and motor impairments.

Significance: Emphasizing accessibility promotes ethical considerations for users with different needs, aligning with principles of fairness and equal opportunity in digital spaces.

3.2.6 Community Engagement and Open-Source Contributions

Description: The project team thinks about contributing to open-source projects and actively interacts with the developer community. This entails giving back to the community, working together, and exchanging knowledge.

Significance: Ethical accountability goes beyond the project at hand to make a constructive impact on the larger development community. Contributions to open source aid in the advancement of industry ethics and collective knowledge.

3.3 Socio-economic impact and sustainability

3.3.1 Societal Impact

"Aalap" chatting application contributes positively to society by providing enhanced communication and connectivity. It enables users to stay connected with family, friends and colleagues irrespective of geographical locations.

3.3.2 Health and Safety

The project positively impacts mental health by providing a platform for social interaction place. A common place to connect to family, friends and colleagues potentially mitigating the feeling of loneliness and stress.

3.3.3 Legal Compliance

The project compliances with legal frameworks, particularly in terms of user privacy and data protection. User authentication is provided in the app to ensure safety of account.

3.3.4 Environmental Impact and Sustainability

While the impact of the project on the environment is relatively low, the project was committed to sustainable practices in its development process.

3.3.5 Economic Impact

The project may have economic impacts by creating opportunities for entrepreneurship and innovation. If the application gains widespread popularity, it could create economic opportunities for developers, content creators, and businesses.

3.4 Financial analyses and budget

Since our application is a software based mobile application, we didn't need any cost to pay. Because if this would be a Hardware based project then we would have needed to buy some hardware. In that case, we could need some cost to buy this hardware.

But we have plan to extend our application and to add some more functionalities in future. We may be needed to integrate some API or payment gateway, then we will be needed to pay cost to integrate these things into our application**.**

4 Conclusion

4.1 Conclusion and challenges faced

In conclusion, the development of our system project "Aalap" has been a fulfilling journey marked by innovation, collaboration and a commitment to deliver a user-centric communication platform. The project aimed to connect the dots in the digital communication realm, emphasizing real time messaging, sharing posts, seeing others posts and a seamless user experience. As the project concludes, it leaves a great impact on both the development team and users' community.

The successful development of “Aalap” shows the dedication of our team to develop high quality software, ethical considerations and a sustainable approach to a problem. The app not only meets user requirements but also exceeds them with the principle of privacy, cultural sustainability.

The journey has been marked with valuable lessons, the never giving up mindset of our team to overcome different challenges. The collaborative effort of our team, guidance from our supervisors have been the major factor for the successful deployment of the project.

Challenges faced:

4.1.1 Technical Challenges

Overcoming technical difficulties related to real-time messaging, sharing posts, profile editing and ensuring the scalability of the application was the biggest challenge for us.

4.1.2 Resource Limitations

The project faced limitations in terms of time and budget. Balancing the project’s goals with available resources required strategic prioritization.

4.1.3 User Feedback Integration

Incorporating user feedback in real time was a challenge, as it required quick response mechanism to address suggestions and concerns.

4.1.4 Security and Privacy concerns

Ensuring security and privacy concerns required great attention to details. The team faced challenges in implementing secure authentication and data protection standards.

4.1.5 Remote Collaborations

With team members working remotely, effective and continuous communication and collaboration required virtual collaboration and connection establishment for efficient remote work practices.

4.2 Future Study

"Aalap" application can remain innovative, competitive and dynamic with the landscape of digital communication with the following future study areas on focus:

4.2.1 Feature Expansion

Explore opportunities to further expand and add features like voice and video calling to provide comprehensive communication experience.

4.2.2 Cross Platform Compatibility

Developing versions of the application for other platform like iOS and web browsers to broaden the use base.

4.2.3 Machine Learning Integration

Integrate machine learning algorithms to personalize content recommendations, analysis and smart chat assistance to further improve user experience.

4.2.4 Advanced Security Measures

Consider integration of advanced security features such as end to end encryption for messaging and multimedia. Use latest advanced security measures and two factor authentication.

4.2.5 Accessibility improvements

Prioritize users with feature rich application. Dynamic app with customization to enhance user experience.

References

[1] D. Griffiths and D. Griffiths, "Head First Android Development", 2nd ed. O'Reilly Media, 2017.

[2] S. Hasani, Firebase for Android Developers: Develop a Real-time, Serverless, and Secure App. Apress, 2017.

[3] F. Zukerberg and J. S. Chirapurath, Firebase Essentials - Android Edition: Get to Know Firebase Realtime Database and Authentication on Android. Independently published, 2018.

[4] Google, "Add Data to Firestore," Firebase Documentation. [Online]. Available: https://firebase.google.com/docs/firestore/manage-data/add-data.[Accessed: November 23, 2023].

[5] Square, "Retrofit: Type-safe HTTP client for Android and Java," Retrofit Documentation. [Online]. Available: https://square.github.io/retrofit/. [Accessed: November 18, 2023].

[6] Google, "Firebase Cloud Messaging," Firebase Documentation. [Online]. Available: https://firebase.google.com/docs/cloud-messaging. [Accessed: October 19, 2023]