Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2023), B.Sc. in CSE (Day)

Messenger Chat Application

Course title: Integrated Design Project II

Course Code: CSE 406 **Section:** 202 D3

Students Details

Name	ID
A.K.M. Atiqur Rahman	202002035

Course Teacher's Name: Shimul Dey Katha

[For teachers use only: Don't write anything inside this box]

Lab Project Status

Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Introduction

Messenger Chat Application is the title of the project. It is also known as a messaging app. A chat application is a platform that enables a user to connect and message another person through the use of their computers or mobile device by login or signed-in their account. Some few examples of chat applications are Facebook Messenger, WeChat, WhatsApp, Viber and many more. In our project, we are making a communication application which is a replicated idea of those example's chatting apps. Here, In this chat app, first user have to sign up if a user is new on the chat application then after successfully signed up, the user will be redirected to the home interface where user who have signed in to this chat app will appear there full name, profile picture, active status and other details. User can select any other user to start chatting and share their feelings with them. Then, they can share file, link and emoji. The user will be able to create a group off specific friends or members and collaborate with each other. The user also can see the sending and receiving time. This project is based on the real life application of socket programming and also for communication systems.

1.2 Motivation

There can be several motivations behind creating a messenger chat application project, including:

- Meeting a Market Need: The primary motivation behind creating a messenger chat application could be to meet the market need for a communication platform that is fast, secure, and user-friendly. By providing a messaging app that addresses these needs, the app could become a popular choice among users.

- Innovation: Another motivation could be to introduce new features and functionalities to the messaging app space. By creating a chat application that incorporates innovative features, such as chatting, file sharing, link sharing, sharing feelings using different types of emoji, group chatting. The app could stand out from competitors and attract more users.

- Pursuing Personal Interests: The motivation for creating a messenger chat application could also be driven by personal interest or passion for technology, design, or communication. Developers may enjoy working on a project that allows them to explore their creativity and technical skills.
- Improving Collaboration and Communication: The motivation for creating a messenger chat application could also be driven by the desire to improve collaboration and communication among teams or groups of people. By creating a chat application that is tailored to a specific industry or group, it could enhance communication and streamline workflow processes.

1.3 Problem Definition

1.3.1 Problem Statement

A problem statement is a brief description of a specific issue or challenge that needs to be addressed. In the context of a messenger chat application project, a problem statement could be:

- Existing messaging apps have limitations in terms of efficiency, security, and user-friendliness.
- Users are dissatisfied with the features offered by current messaging apps.
- There is a need for a new messenger chat application that provides a comprehensive solution that caters to all the needs of users.
- Current messaging apps are not able to provide a more efficient, secure, and user-friendly experience for users.
- There is an opportunity to create a new messaging app that addresses these limitations and offers a better solution for users.

1.3.2 Complex Engineering Problem

Here is a complex engineering problem statement for a messenger chat application project: Designing a messenger chat application that can handle a massive number of concurrent users while maintaining fast and reliable message delivery, secure end-to-end encryption, and an intuitive user interface is a complex engineering challenge. The problem involves developing an architecture that can handle high traffic loads while minimizing latency, ensuring data privacy and security through advanced encryption methods, and implementing user-friendly features that enable seamless communication. Additionally, the application must be optimized for various devices and operating systems, including desktops, smartphones, and tablets. Solving this engineering challenge will require expertise in various technologies, including network architecture, encryption methods, user interface design, and software development.

1.4 Design Goals/Objectives

In this project, our objectives is about -

- To develop a messenger chatting application.
- To know about how to make a communication technology based project using socket programming.
- To understand the mechanism of a chatting system application and how to implementing a project using Agile method.

1.5 Application

The application of a messenger chat application project written in given below-

- A software application designed to facilitate instant messaging and communication between users.
- Enables users to exchange messages in real-time, either individually or within groups.
- May include additional features such as file sharing, link sharing, emoji sharing and group chatting.
- Available on various platforms, including desktops, smartphones, and tablets.
- Allows individuals and businesses to communicate quickly and easily across the globe.
- Popular examples include WhatsApp, Facebook Messenger, and Telegram.

Chapter 2

Requirement Analysis and Design

2.1 Requirement Analysis

2.1.1 Functional Requirement

System Requirement

(a) User registration and login: System will be able to store all registration and login information of users.

(b) Messaging: System will be able to store sent and received text messages, images, and videos.

(c) Group messaging: System should be able to store created groups and members details.

(d) Push notifications: System will be able to show push notifications when user will receive new messages or calls.

(e) Emoji and sticker support: System will provide some predefined emoji package.

(f) File sharing: System will provide reliable file transfer using the latest security mechanism.

User Requirement

(a) User registration and login: Users should be able to create an account and log in securely.

(b) Messaging: Users should be able to send and receive text messages, images, and videos.

(c) Group messaging: Users should be able to create groups and add members for group messaging.

(d) Push notifications: Users should receive push notifications when they receive new messages or calls.

(e) File sharing: Users should be able to share files, such as documents and images, with each other.

(f) Emoji and sticker support: Users should be able to use emojis and stickers to express themselves in messages.

2.1.2 Non-functional Requirement

(a) Performance:
The application should be fast and responsive, with low latency and no significant delays in message delivery

(b) Scalability:
The application should be able to handle a large number of users and concurrent connections.

(c) Security:
The application should be secure and protect user data with end-to-end encryption, and two-factor authentication.

(d) Availability:
The application should be available 24/7 with minimum downtime for maintenance or upgrades.

(e) User interface:
The application should be easy to use, with an intuitive user interface that allows users to navigate and use the app with ease.

(f) Compatibility:
The application should be compatible with various devices and operating systems, such as iOS, Android, Windows, and Mac.

(g) Efficiency:
The application will also provide a error free and faster connection. System will be able to perform in real time.

(h) Usability:
The user friendly interface and all functionality will help the user to get use to with the system in no time.

(i) Maintainability:
System can be updated and upgraded as needed. When any update or upgrade will be required, it will be easy to update and upgrade as requirement without any server or system dawn.

2.2 Tools and Techniques

Below given uses tools and technologies for the whole project -

1. Desktop / Laptop: These are the devices on which the Messenger chat application can be used. Users can access the application using their desktop or laptop computers.

2. Apache NetBeans IDE 15: Apache NetBeans IDE is an integrated development environment used for Java development. It provides features like code editing, debugging, and project management. NetBeans IDE 15 is specifically mentioned as the development environment for the project.

3. XAMPP: XAMPP is a cross-platform software package that includes the Apache web server, MySQL database, and PHP. It is used to set up a local development environment for testing and running the Messenger chat application.

4. Microsoft Edge, Google Chrome, Firefox (Browser)

5. Java: Java is the primary programming language used for developing the Messenger chat application. Java provides a robust and secure platform for building server-side applications.

6. MySQL: MySQL is used as the database management system to store and manage user data, chat history, and other relevant information. It is used to execute SQL queries for database operations such as creating tables, inserting data, updating records, and retrieving information.

7. Socket Programming: Socket programming is a concept used in network communication to establish a connection between the server and clients. It enables the exchange of messages between different devices over a network. In the Messenger chat application, socket programming is employed to facilitate real-time messaging between users.

8. Mechanism of Chat System: This refers to the overall architecture and design principles used in developing the chat system. It includes concepts such as message routing, user authentication, message encryption, chat room management, and handling various scenarios like message delivery, offline messaging, etc.

Overall, the mentioned tools and techniques provide a foundation for developing the Messenger chat application, covering programming languages, database management, network communication, and the necessary development environment.

2.3 Gantt Chart of Project Development Timing

A planned project titled “Messenger Chat Application” will be launched within 4 months (16 weeks). This software development timing is described below by a Gantt Chart:

Activities	Weeks															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Planning and Analysis	✓								✓							
Design		✓	✓							✓	✓					
Development				✓	✓	✓						✓	✓			
Testing							✓	✓						✓	✓	
Launch																✓

Table 2.1: Development Timing Gantt Chart

2.4 Use Case Diagram

A Use Case Diagram is a graphical representation of a system that shows the different interactions between users and the system. In the context of a Messenger chat application, a Use Case Diagram can be used to represent the various actions that a user can perform within the application. This can include actions such as sending and receiving messages, creating and joining chat groups, and managing their account settings. The Use Case Diagram can help developers understand the different features and functions of the Messenger chat application, and can be used as a tool to identify any potential gaps or areas for improvement in the system's design. Overall, the Use Case Diagram is an essential tool for ensuring that the Messenger chat application meets the requirements and expectations of its users.

2.5 Data Flow Diagram

A data flow diagram is a graphical representation of the data flow in an information system. It can represent incoming data stream, outgoing data stream, and stored data. The DFD doesn't mention anything about how data flows through the system. It is a traditional visual representation of the information flows in a system.

A clear and sharp DFD can graphically represent the Systemropriate amount of system re- quirements. It tells how data enters and leaves the system, what changes the information

and where the data is stored.Levels or layers are used in DFDs to represent progressive de- grees of detail about the system or process. DFD0, DFD1, and DFD2 are different levels of

abstraction in Data Flow Diagrams. These levels include:

- Level 0: Also known as a ”context diagram,” this is the highest level and represents a very simple, top-level view of the system being represented.
- Level 1: Still a relatively broad view of the system, but incorporates sub processes and more detail.
- Level 2: Provides even more detail and continues to break down sub processes as needed.

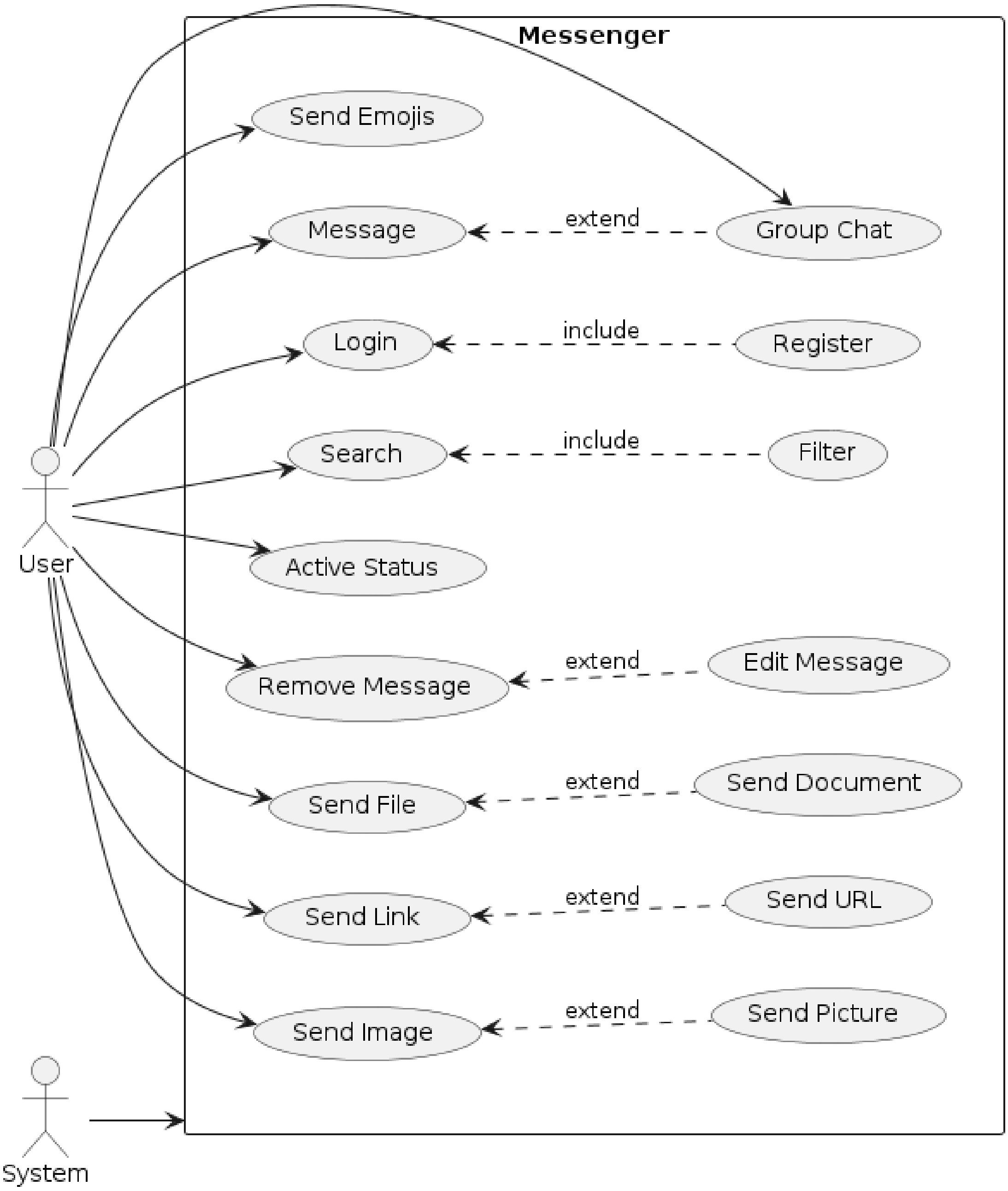


Figure 2.1: Use Case Diagram.

In theory, more levels are possible, but they are rarely used and would likely represent more detail than a data flow diagram would normally convey.

2.5.1 Data flow diagram(Level 0)

DFD level 0, also known as Context Diagram, is the highest-level view of the Data Flow Diagram used to represent the system's interaction with the external entities. In the case of a Messenger chat application, the Context Diagram represents the application's interaction with the external entities such as the user, the server, and other applications. It provides a high-level view of the system and helps in identifying the data inputs and outputs, processes, and entities involved in the system. By analyzing the DFD 0, we can gain a better understanding of the system's overall structure and its interactions with external entities. This information is essential for designing and developing the Messenger chat application.

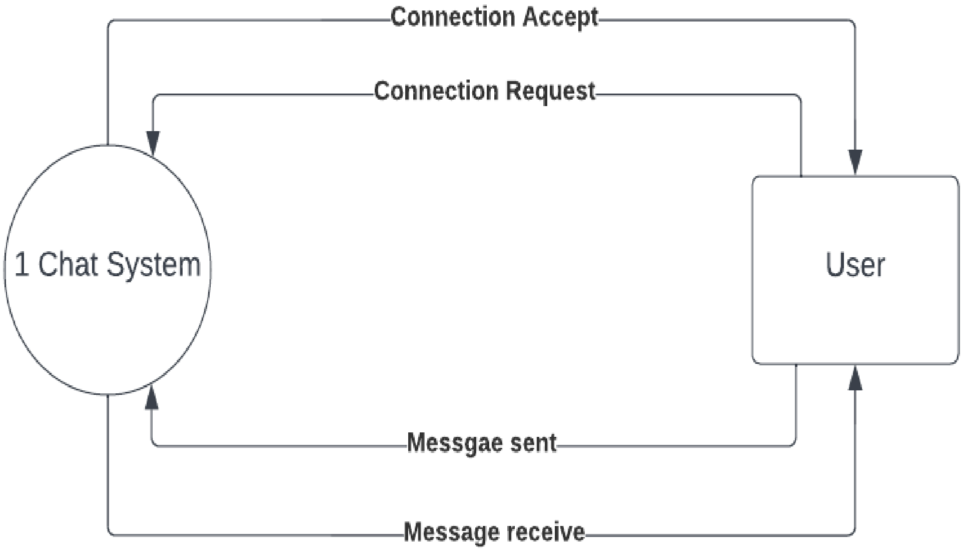


Figure 2.2: Data Flow Diagram level 0.

2.5.2 Data flow diagram(Level 1)

DFD Level 1, a context diagram is decomposed into multiple bubbles/processes. DFD 1 for a Messenger chat application is a high-level diagram that provides an overview of the system's functionality and data flow. It depicts the external entities that interact with the system, such as users and external services, and the different components of the system, including the chat application, database, and server. This DFD level provides a clear understanding of the system's boundaries, inputs, and outputs, which is important for designing the system's architecture and identifying potential areas for improvement. Overall, DFD 1 is a useful tool for analyzing the flow of data and interactions between different components of a messenger chat application.

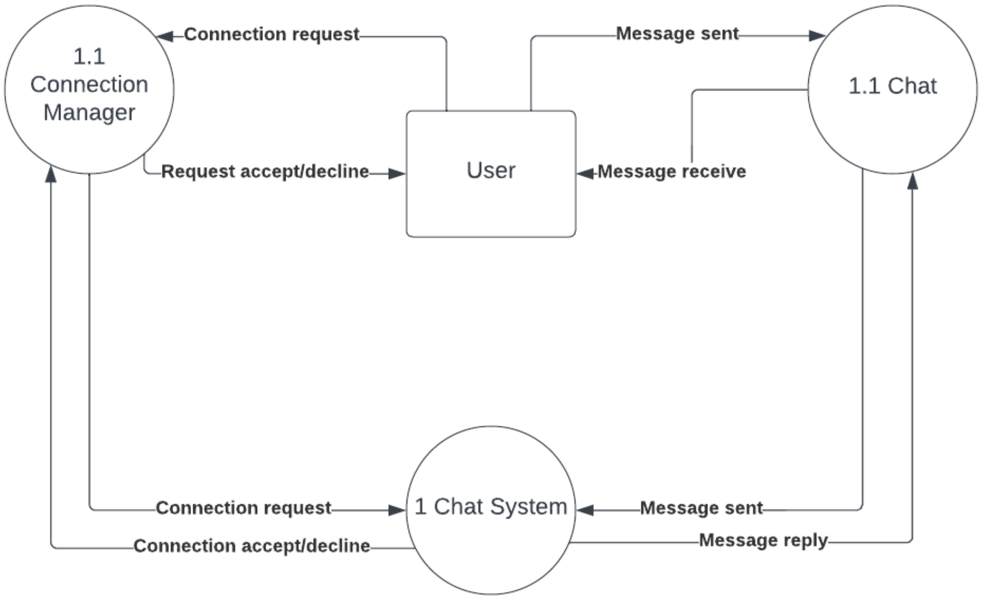


Figure 2.3: Data Flow Diagram level 1.

2.5.3 Data flow diagram(Level 2)

DFD Level 2, DFD shows how data flows inside the modules mentioned in Level 1. DFD 2 for a messenger chat application represents a more detailed view of the system components and their interactions than DFD 1. It includes data stores, external entities, and processes that were not shown in DFD 1. DFD 2 helps in understanding the system's functionality and identifying potential areas for improvement or optimization. It also allows developers to identify the data sources and sinks, which can be helpful in debugging the system. Overall, DFD 2 is a valuable tool for analyzing the messenger chat application's functionality and ensuring that it meets the user's requirements.

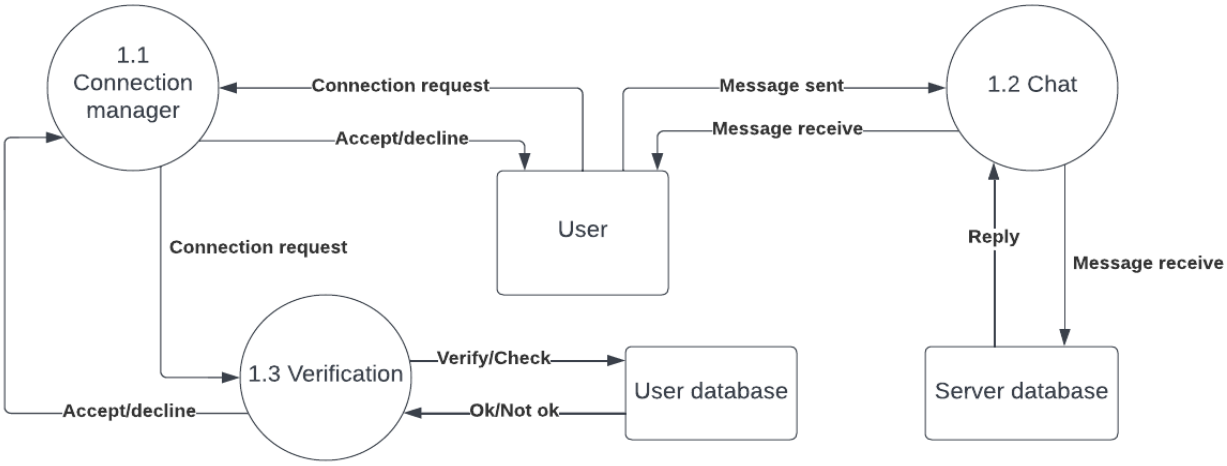


Figure 2.4: Data Flow Diagram level 2.

2.6 E-R diagram and Database Schema

2.6.1 Logical Schema

File (FileID, FileExtension, BlurHash, Status)

User (UserID, UserName, Password)

User_Account (UserID, UserName, Gender, Image, Status)

2.6.2 E-R Diagram

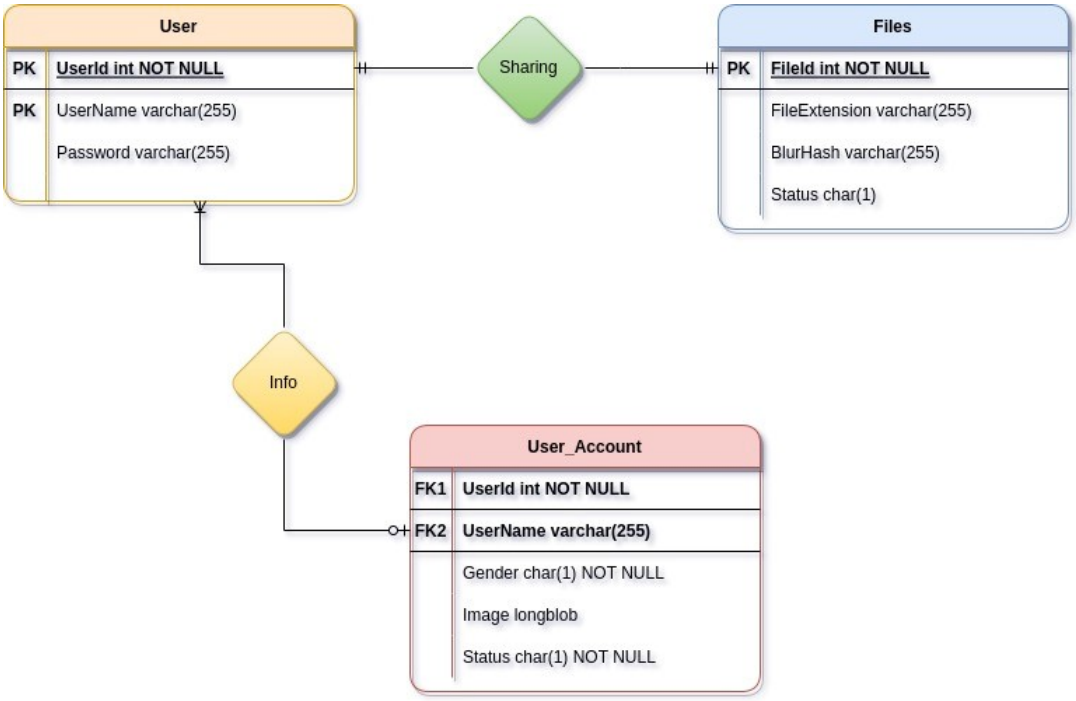


Figure 2.5: ER Diagram of Messenger Chat Application.

2.6.3 Physical Database

FileID	FileExtension	BlurHash	Status
0	JPG	NULL	1
1	JPG	NULL	1
2	PNG	NULL	1
3	GIF	NULL	1

Table 2.2: File Table

UserID	UserName	Password
0	Prangon	Prangon@35
1	Washim	Washim@34
3	Atiq	A@202002035
2	Akram	W@202002034

Table 2.3: User Table

UserID	UserName	Gender	Image	Status
0	Prangon	M	NULL	1
1	Washim	M	NULL	1
2	Atiq	M	NULL	1
3	Akram	M	NULL	1

Table 2.4: User_Account Table

2.7 System Architecture

The success of a software project depends on its system architecture, which provides a foundation for the entire development process. This lab report presents the results and analysis of an experiment to develop a System Architecture for a Messenger chat application system project. In software engineering, system architecture refers to the overall structure and organization of a software system, which includes its components, their interactions, and their relationships. It provides a roadmap for the entire development process, guiding the design and implementation of the system. System architecture can be broken down into two main types of design:

1. High-level Architecture
2. Low-level Architecture

2.7.1 High-level Architecture

The high-level design for the Messenger chat application consists of several modules, including the User Display, User Keypad, Login/Signup Management, Account Information Database, Media Sending, Control System, and Chatting Management modules. These modules work together to facilitate real-time communication between users in a secure and user-friendly manner. The User Display module provides a clear interface for users to manage their contacts, chat histories, and notifications, while the User Keypad module enables users to input messages using various methods. The Login/Signup Management module handles user authentication and account management, and the Account Information Database module stores user profiles and contact lists. The Media Sending module allows for secure transmission of media files, while the Control System module manages overall system operation. Finally, the Chatting Management module ensures private communication and manages chat messages. Together, these modules provide a solid foundation for a reliable and secure communication system that offers users a seamless communication experience. Below is the high level design for a Messenger chat system:

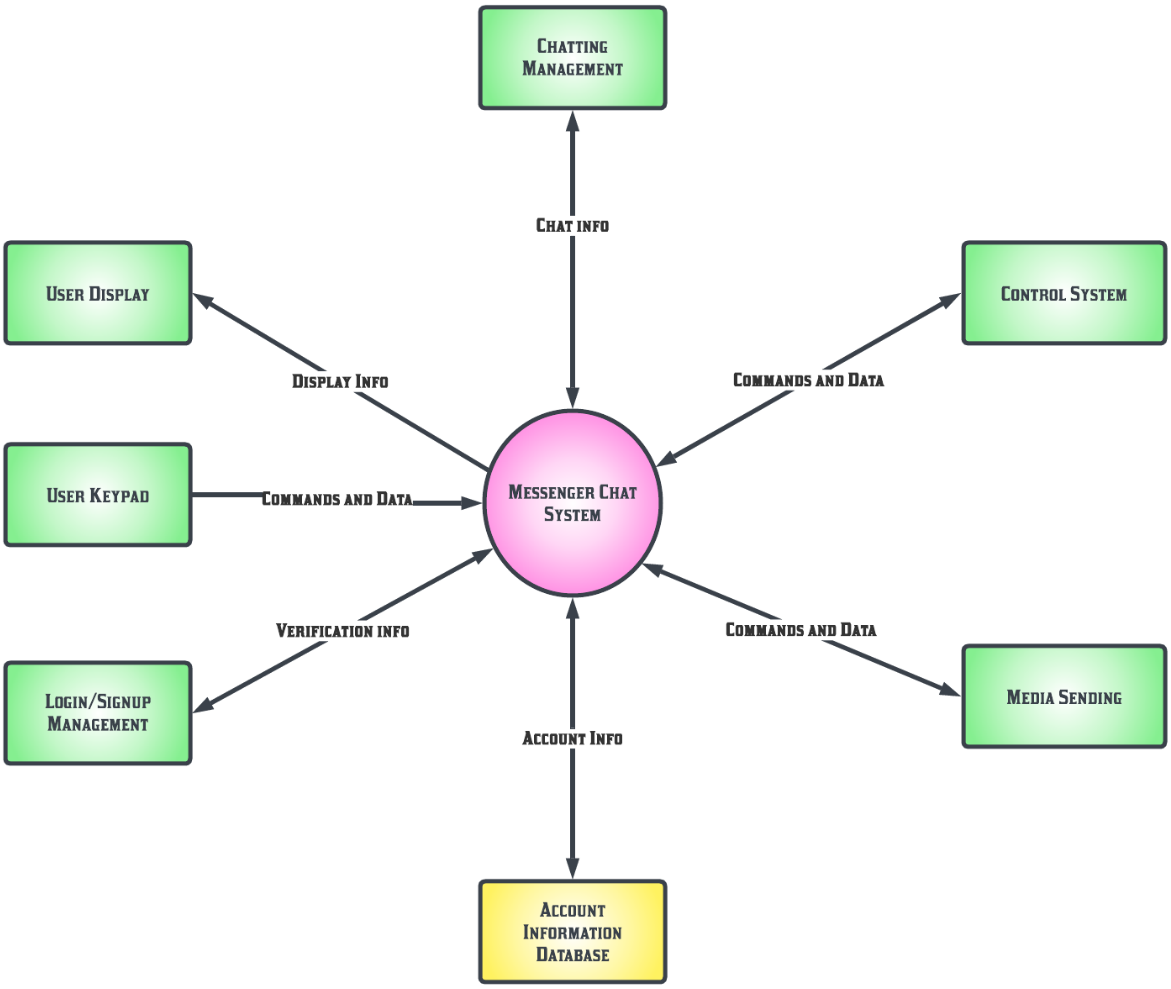


Figure 2.6: High level architecture (abstract view) of the system of a Messenger Chat Application based on context based architecture.

2.7.2 Low Level Design

Low-level design for a Messenger chat application project would involve designing the internal workings of each component of the system, including the algorithms, data structures, and interfaces. For example, the low-level design of the chat server would specify how messages are sent and received, how the server handles user authentication and authorization, and how the server stores and retrieves message data from the database. Similarly, the low-level design of the user interface would specify the layout and behavior of each interface component, including buttons, text fields, and menus. In short, low-level design in this project would focus on the implementation details of each component, ensuring that they work together seamlessly to provide a reliable and secure messaging platform for users. Below is the low-level design for a Messenger chat system:

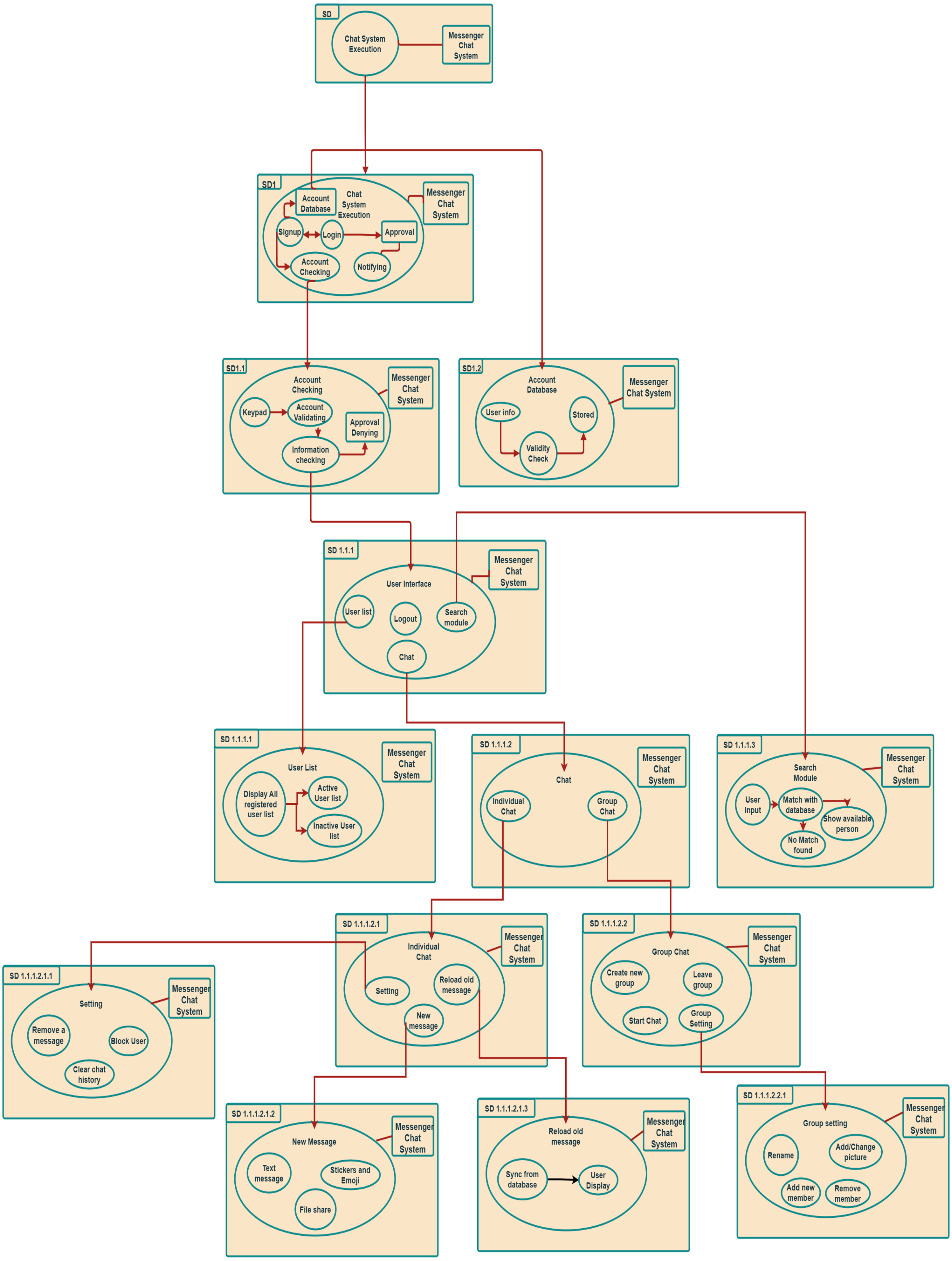


Figure 2.7: Trends to high level view to low level view using context based architecture of A Messenger Chat Application.

2.8 Conclusion

In conclusion, the requirement analysis and design phase of the Messenger chat application project have provided a comprehensive understanding of the functional and non-functional requirements of the system. The functional requirements cover essential features such as user registration and login, messaging, group messaging, push notifications, emoji and sticker support, and file sharing. The non-functional requirements address performance, scalability, security, availability, user interface, compatibility, efficiency, usability, and maintainability aspects of the application. The use case diagram has depicted the various interactions between users and the system, while the data flow diagram has illustrated the flow of data within the application at different levels. The Gantt chart has provided a visual representation of the project development timing, and the tools and techniques have outlined the necessary technologies and development environment. The E-R diagram and the physical database schema have presented the logical and physical structure of the database. Finally, high-level and low-level architectures provide a solid foundation for the development and implementation of the Messenger chat application. Overall, the analyses and designs form the foundation for the successful development and implementation of the Messenger chat application.

Chapter 3

Interface Design and Implementation

3.1 Design UI/UX

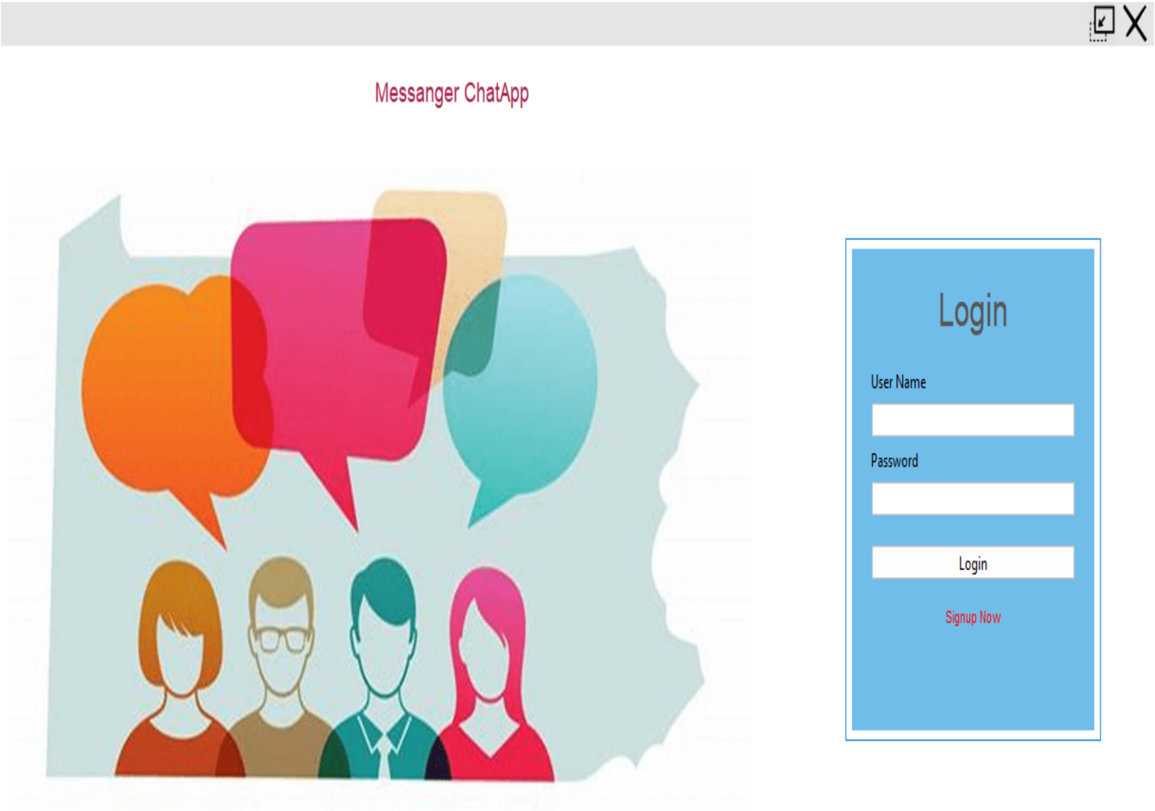
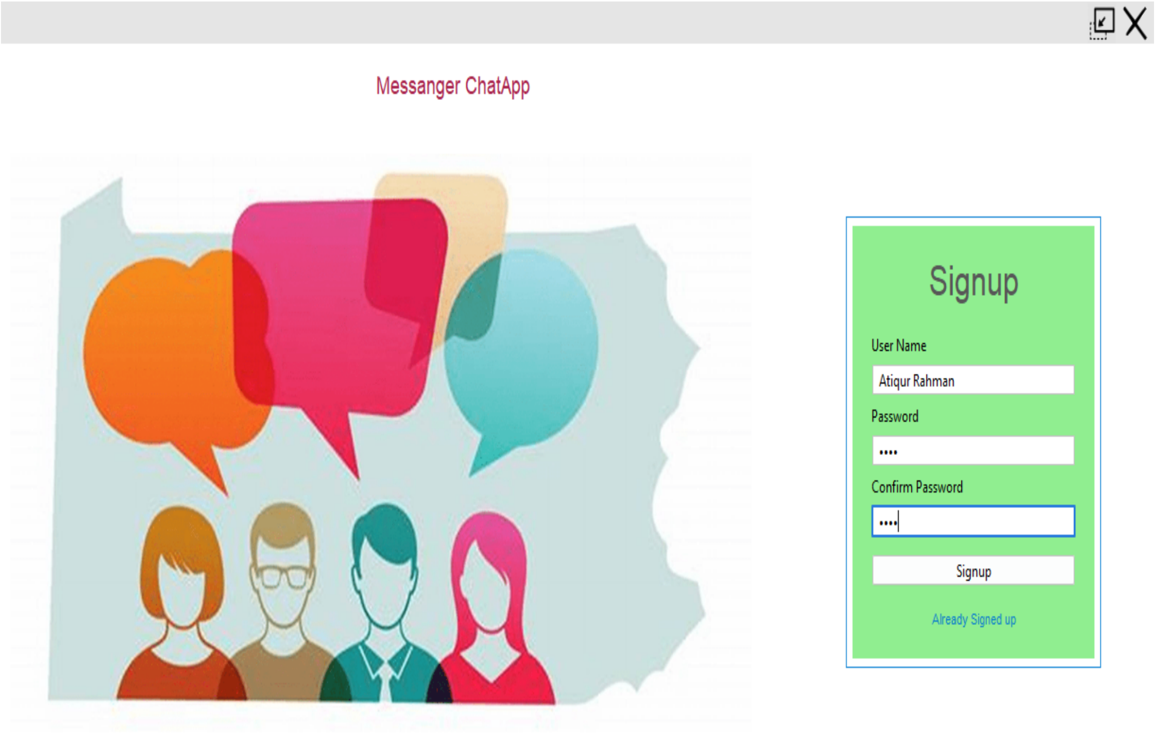


Figure 3.1: Login Interface



User Name

Atiqur Rahman

Password

••••

Confirm Password

••••

Figure 3.2: Signup box for signup

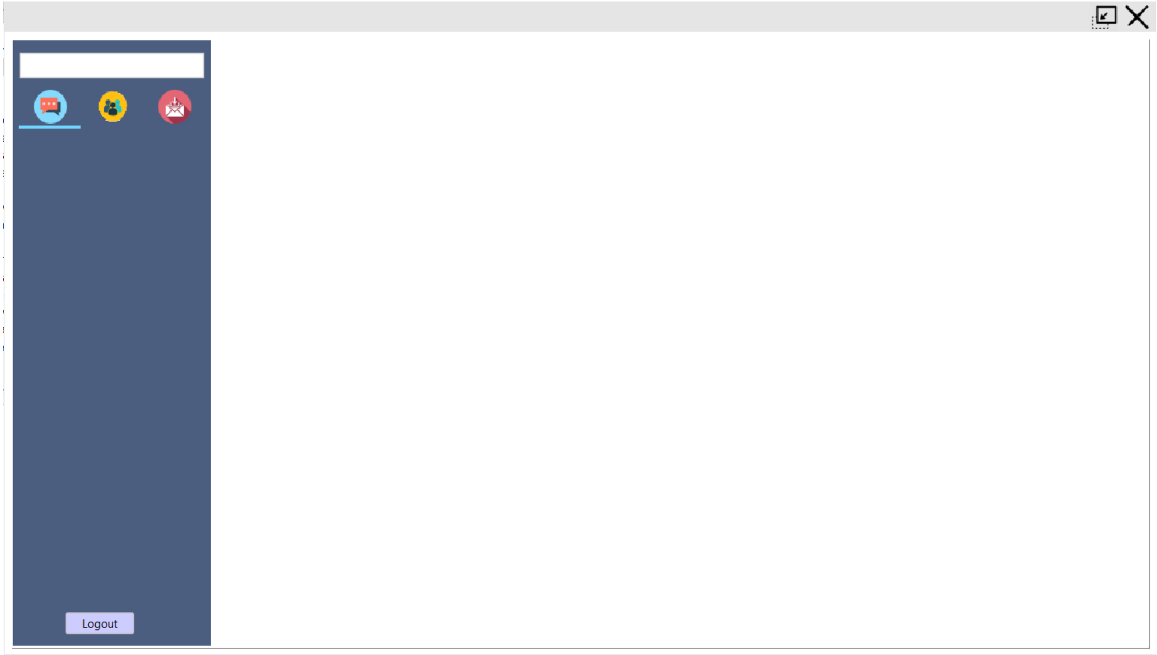


Figure 3.3: After Login the home page for chat

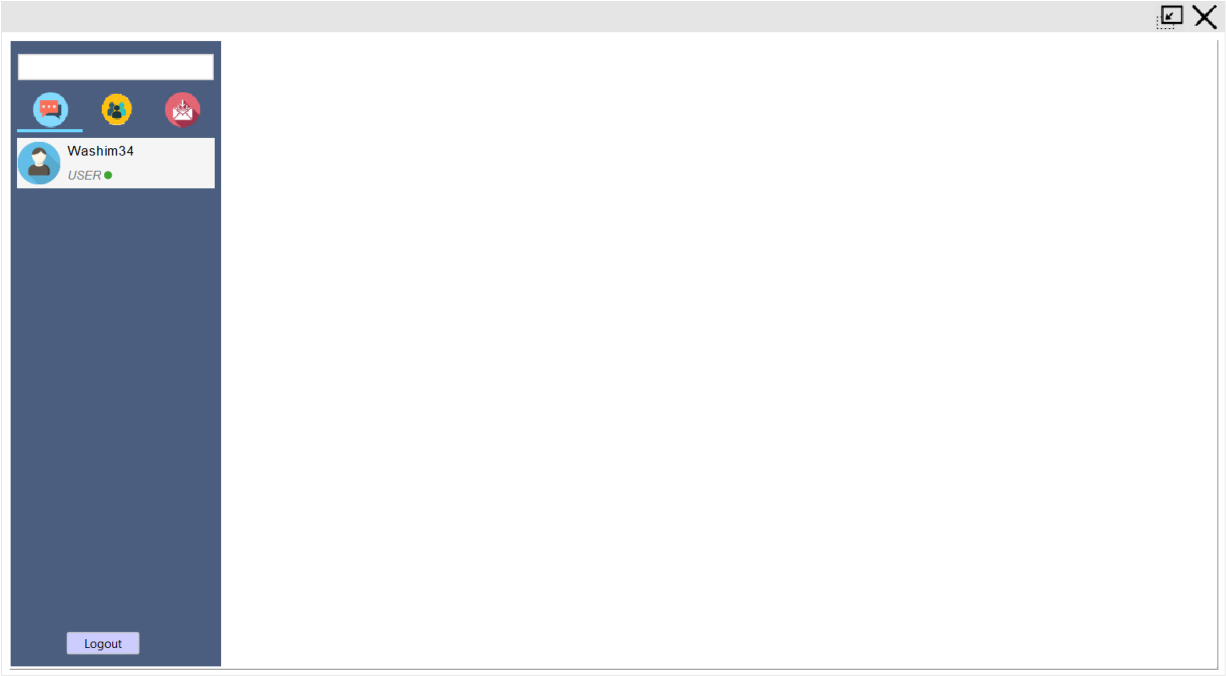


Figure 3.4: After Login Active User

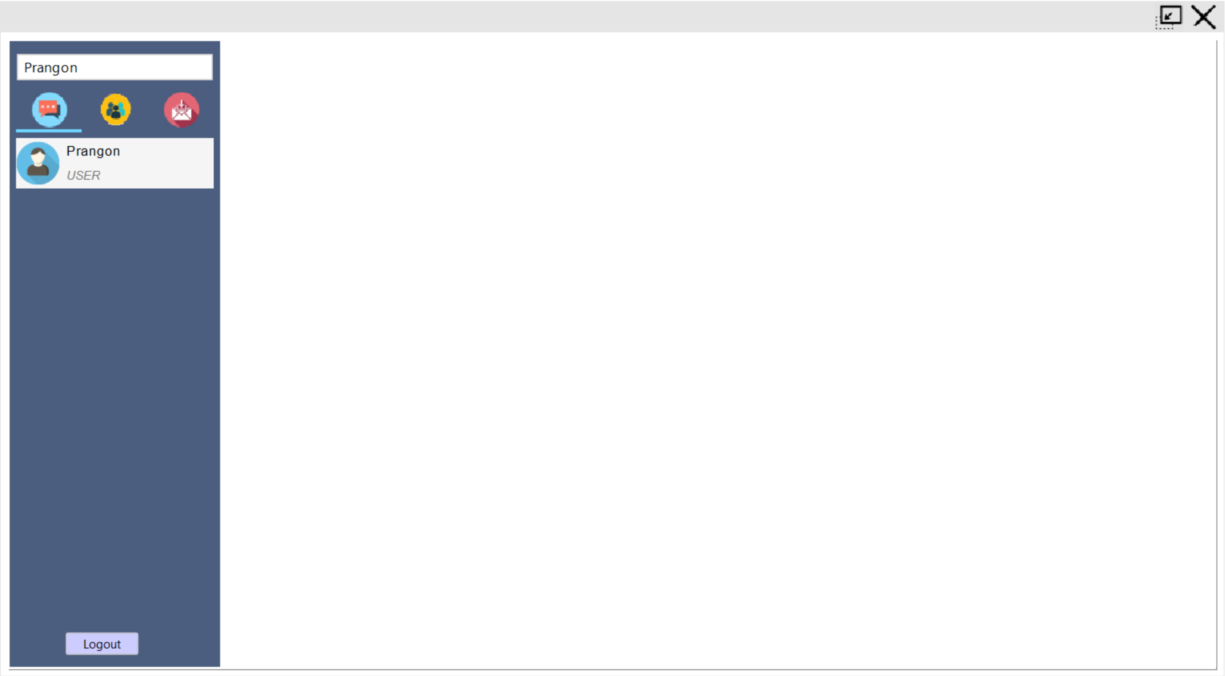
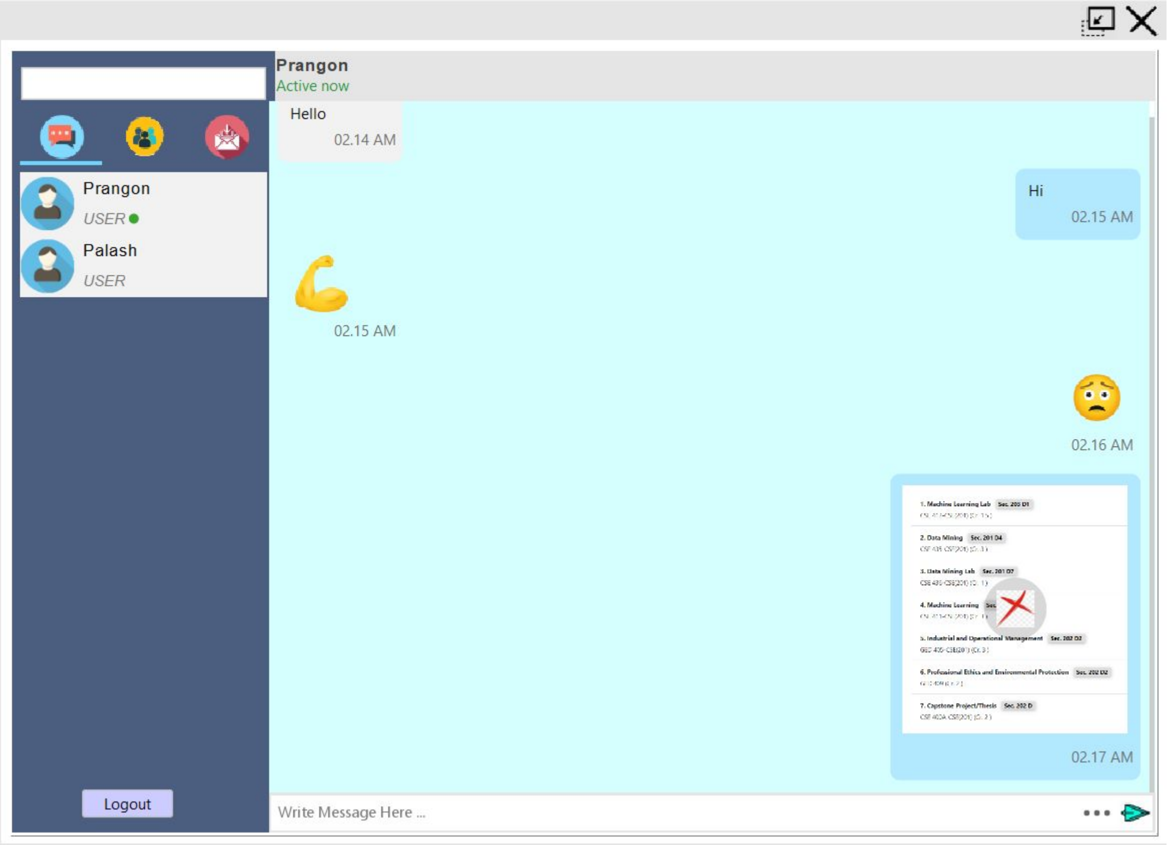


Figure 3.5: Seaching Person by Name


Prangon
Active now
Prangon
USER
Palash
USER
Logout
1. Machine Learning Lab Sec. 205 D1
2. Data Mining Sec. 201 D4
3. Data Mining Lab Sec. 201 D7
4. Machine Learning
5. Industrial and Operational Management Sec. 202 D2
6. Professional Ethics and Environmental Protection Sec. 202 D2
7. Capstone Project/Thesis Sec. 202 D
Write Message Here ...

Hello
02.14 AM

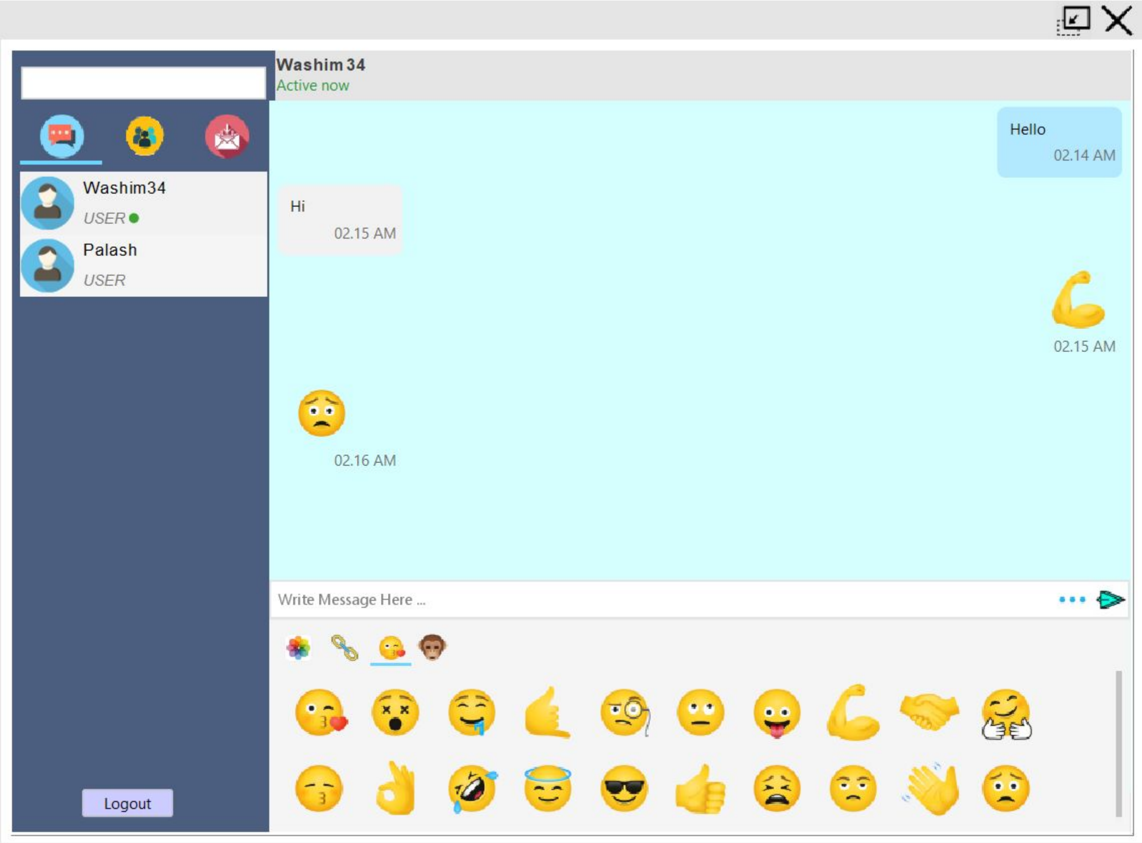
Hi
02.15 AM

💪
02.15 AM

😟
02.16 AM

02.17 AM

Figure 3.6: Chatting Clients with each other



Washim 34
Active now

Hello
02.14 AM

Hi
02.15 AM

02.15 AM

02.16 AM

Figure 3.7: Emoji Sending

3.2 Implementation Procedure Overview of the project

The project is about Messenger Chat application. This will create a chat Application project inside IDE. First create form for right-click on the project – > New – > Select JFrame Form. You have to make two Jframe forms, one for Client and one for Server. Jframe is used to make a design of your Application. it has a simple configuration, Also, it has a decent toolbox with drag and drop functionality. Now, create two Jframe and name them Client.java and Server.java. Then drag and drop these controls – TextArea, TextField, Button and Label. TextArea is used to view the incoming messages and TextField is used to write the message. Here, after creating all forms by drag and drop then add icons. Then creating socket programming for communication client and server. Then adding file sharing and emoji sharing feature. After all that We are connecting Database. That’s all about Implementation procedure overview of this project.

3.2.1 Project Implementation Methodology Diagram

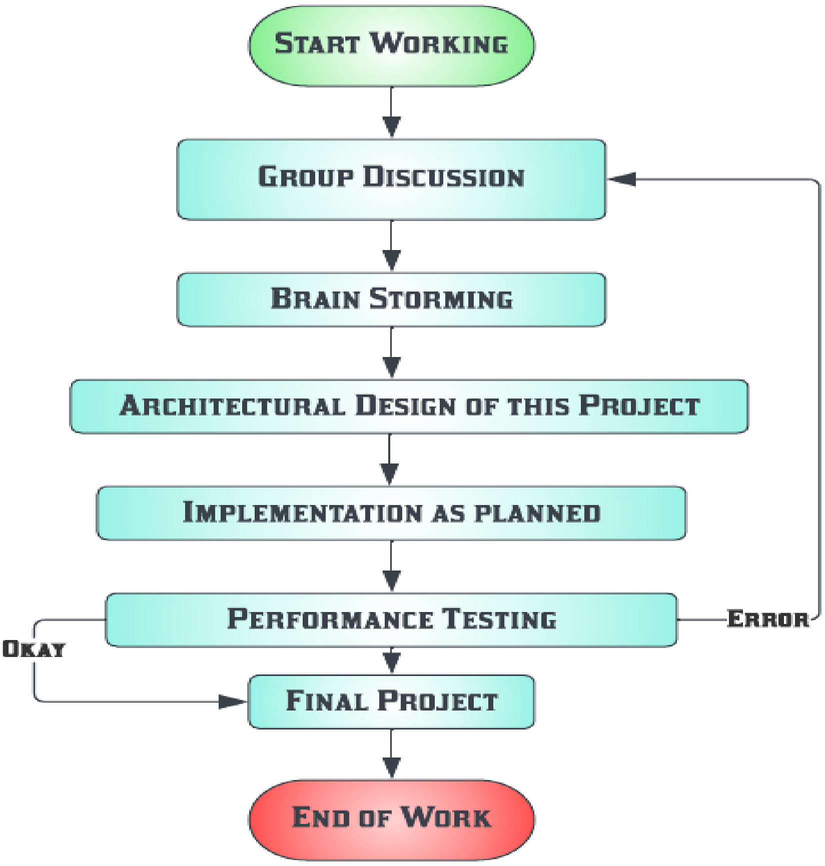


Figure 3.8: Diagram of Methodology for this project

3.3 Achievement

By completing this project, I achieved all the objectives with the main concept of this project which is developing a chatting application and also get the desired outputs.

3.3.1 Challenge Face during implementation

In this project, we faced some challenges when doing the tasks which are given below -

- When connecting database it gives error with message no driven found which is recovered after trying many times with some guidelines.
- When working with GUI and import into others class then faced little bit problem.
- When Working with File sharing system and adding multiple option, there is also facing some challenges.

Chapter 4

Testing & Result Analysis

4.1 Test Cases

Sign-up

Username	Password	System Status	Remark
Washim	34	Not ok	Required at least 8 digits and a special, Lower-capital character mixed password
Prangon	Prangon@35	ok	Successfully Sign-up
34	Washim@34	Not ok	Digit not allowed as username
@@		Not ok	Special character is not allowed as username and must require all the fields

Login

Username	Password	System Status	Remark
Prangon	Prangon@35	ok	Login successfully
Prangon	35	Not ok	Login failed as password do not match with database
Washim	Washim@34	ok	Login successfully
IDPtwo	()ww	Not ok	Login failed as Password and username do not match with database

Text Message

Message send	Message Receive	System Status	Remark
Hi Ma’am	Hi Ma’am	ok	Message sent and received perfectly
What’s up bro	What’s up bro	ok	Message sent and received perfectly

File sharing (Emoji, Image, Link):

Send	Message Receive	System Status	Remark
emoji.png	emoji.png	ok	Emoji send successfully
CLPfullmarks.jpg	CLPfullmarks.jpg	ok	Received

Logout

Click	System Status	Remark
Logout	Ok	Perfectly logged out

Delete

Delete	Action	System Status	Remark
Hi Ma’am	Blank	Not ok	Partially working

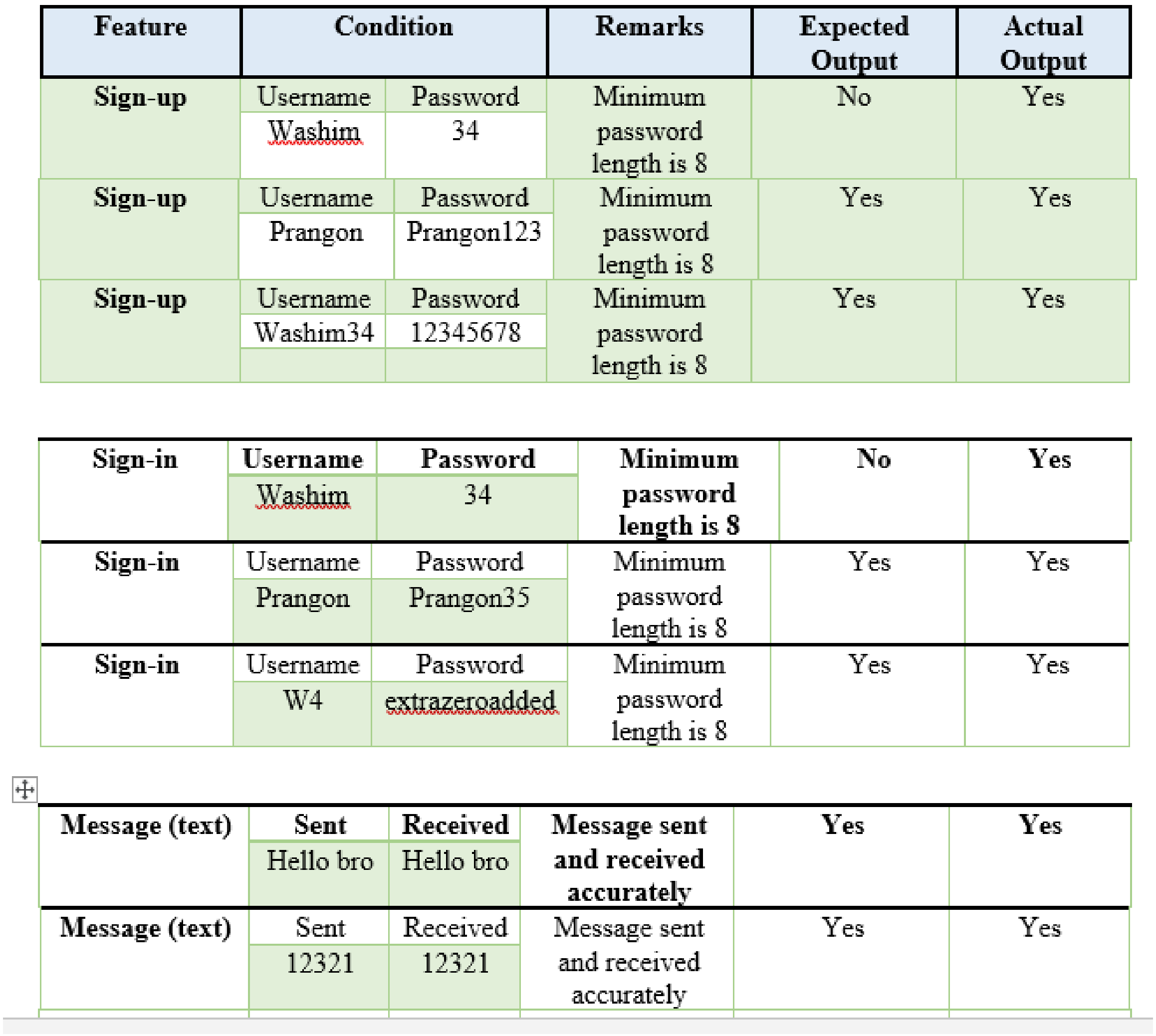
Group Creation

Group Name	Group member	System Status	Remark
CLP	2	Not ok	Less than requirement group member
IDP2	5	ok	Group created successfully

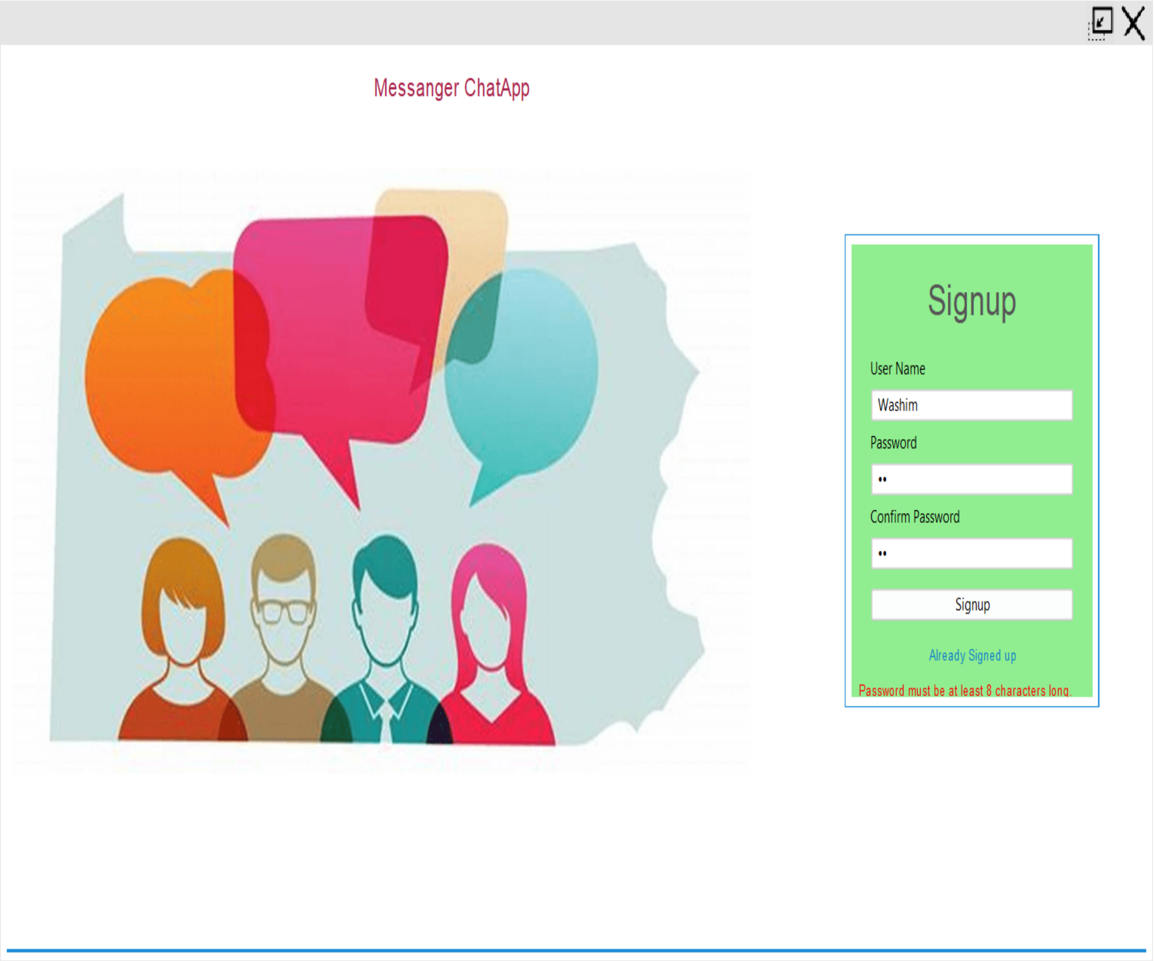
Group Message

Send Message	Received Message	System Status	Remark
Member-1 (Hi Friends)	Member-3(Hi Friends)	Not ok	Not Received all members.

4.2 Manual Testing



Feature	Condition		Remarks	Expected Output	Actual Output
Sign-up	Username Washim	Password 34	Minimum password length is 8	No	Yes
Sign-up	Username Prangon	Password Prangon123	Minimum password length is 8	Yes	Yes
Sign-up	Username Washim34	Password 12345678	Minimum password length is 8	Yes	Yes
Sign-in	**Username** Washim	**Password** 34	**Minimum password length is 8**	**No**	**Yes**
Sign-in	Username Prangon	Password Prangon35	Minimum password length is 8	Yes	Yes
Sign-in	Username W4	Password extrazeroadded	Minimum password length is 8	Yes	Yes
Message (text)	**Sent** Hello bro	**Received** Hello bro	**Message sent and received accurately**	**Yes**	**Yes**
Message (text)	Sent 12321	Received 12321	Message sent and received accurately	Yes	Yes



Messanger ChatApp

Signup

User Name

Washim

Password

••

Confirm Password

••

Signup

Already Signed up

Password must be at least 8 characters long.

Figure 4.1: Login Interface

4.3 Limitations

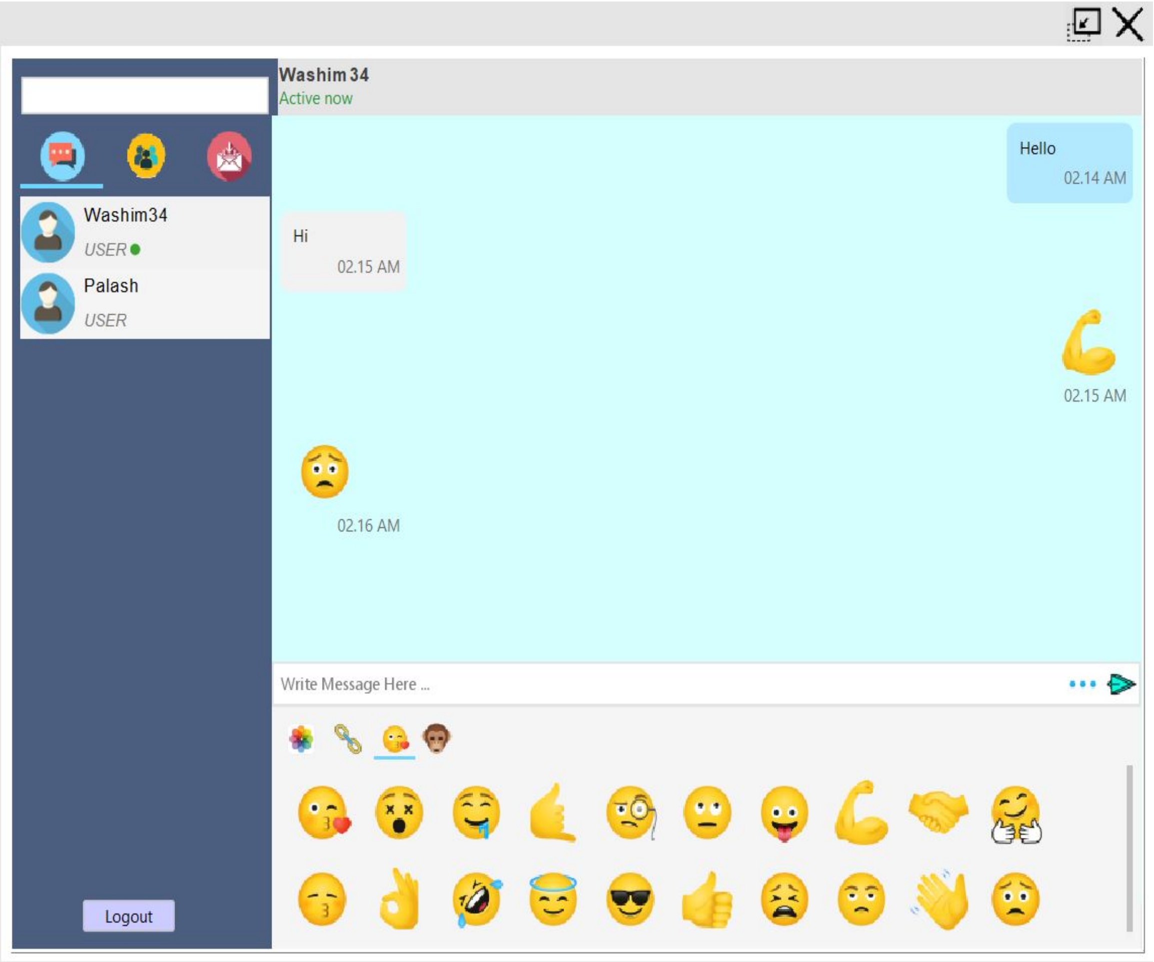
The Messenger Chat Application project has the following limitations:

1. Lack of Clear Module Implementation: The project lacks a clear modular structure, making it difficult to manage and maintain the codebase. This can lead to challenges in scalability, code reusability, and overall project organization.

2. Absence of Group Chat Functionality: The Messenger Chat Application does not include a group chat feature. Users are limited to one-on-one conversations, which can be a significant limitation for users who require group collaboration or communication.

3. Incomplete Inbox Functionality: The application lacks a fully developed and comprehensive inbox feature. Users may face difficulties in managing and organizing their messages efficiently, including archiving, filtering, and searching for specific messages.

4. Limited Frame Handling: The application struggles with the handling of separate frames or windows for different chat conversations. This limitation can lead to



Hello
02.14 AM

Hi
02.15 AM

02.15 AM

02.16 AM

Figure 4.2: Login Interface

challenges in managing multiple chat sessions simultaneously and may result in a less user-friendly experience.

5. Non-Convertibility to APK File: The current state of the project does not allow for the direct conversion of the application into an APK (Android Application Package) file. This limitation prevents easy distribution and installation of the application on Android devices.

It is important to address these limitations in future iterations of the project to enhance the overall functionality, user experience, and security of the Messenger Chat Application.

4.4 Overall Result Discussion

In this result analysis, we have successfully implemented the test cases for key function and demonstrated key functionalities of the Messenger Chat Application, including login, signup, searching, logout, active status, database connection, and file sharing. These functionalities have been thoroughly tested and validated, providing users with a secure and efficient chat experience. However, certain limitations and bugs have been

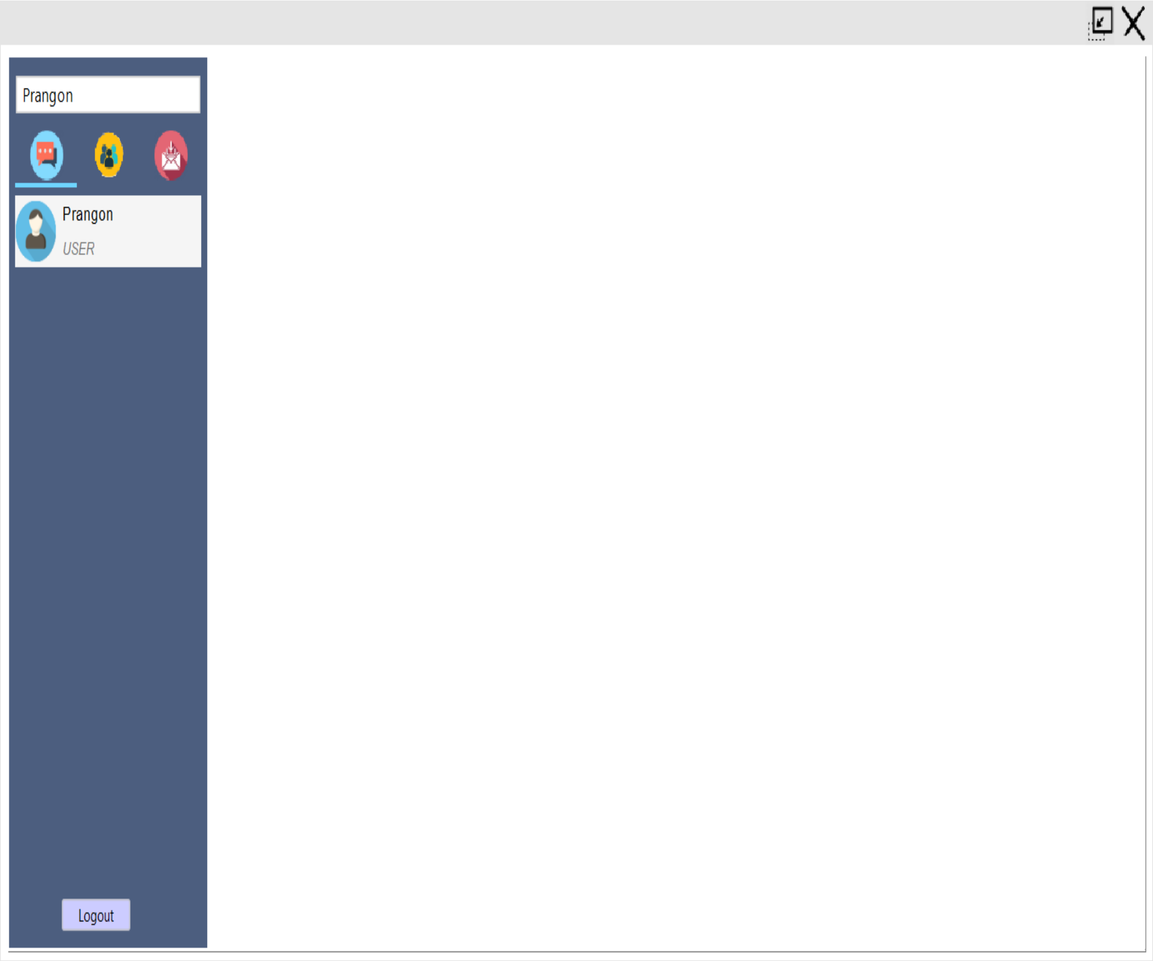
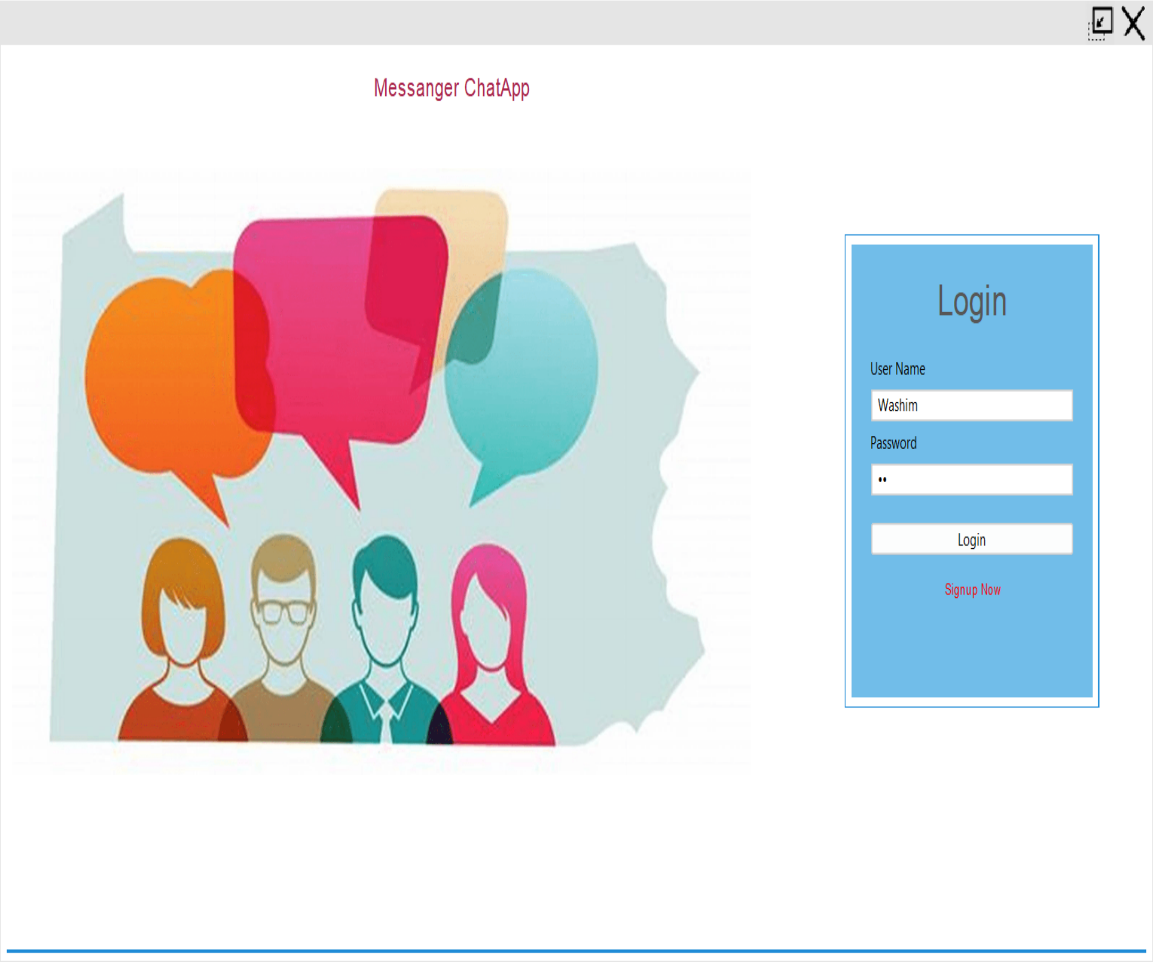


Figure 4.3: Login Interface

encountered, such as the lack of clear module implementation, incomplete inbox functionality, and limited frame handling. To address these limitations, future work should focus on implementing a modular structure, enhancing the inbox functionality, and improving frame handling. Additionally, integrating group chat functionality, shareable links, and addressing security and privacy concerns would enhance the overall feature set and user satisfaction. Despite these limitations, the project serves as a successful demonstration of core functionalities and provides a solid foundation for further development and improvements in the Messenger Chat Application.



Messanger ChatApp

Login

User Name

Washim

Password

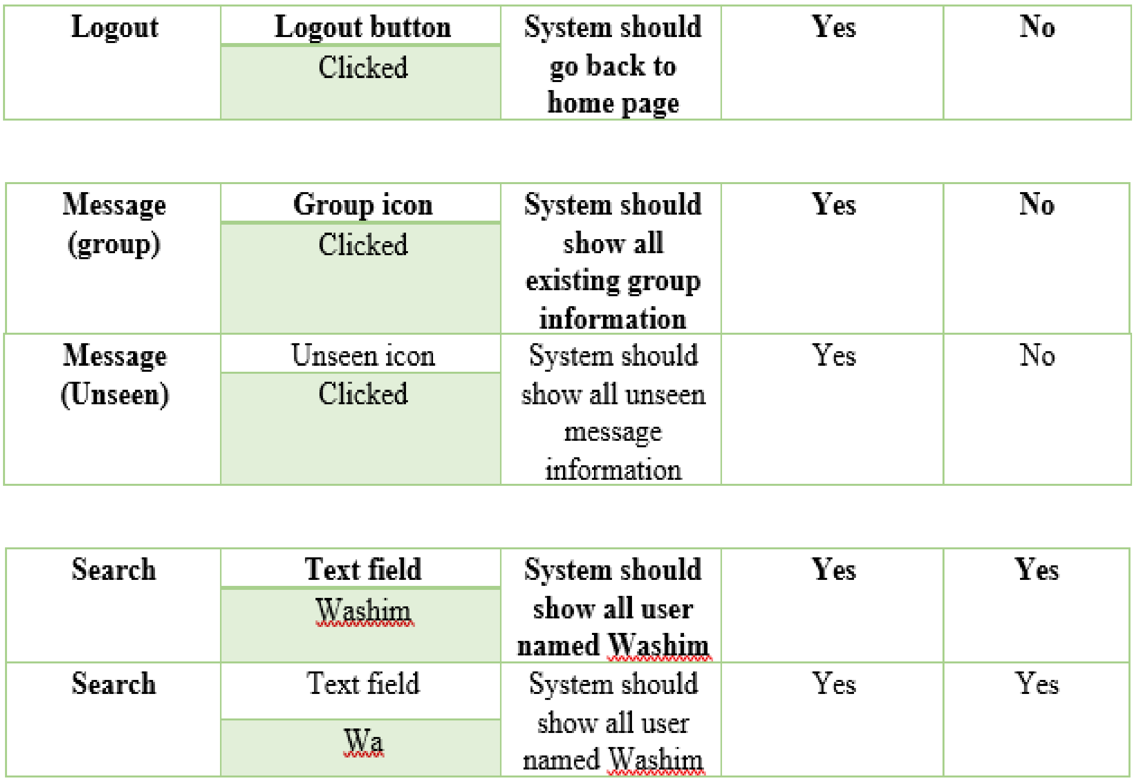
••

Login

Signup Now

Figure 4.4: Login Interface

Message (Emoji)	Sent 😁	Received 😁	Message sent and received accurately	Yes	Yes
Message (Image)	Sent Extra.jpg	Received Null	Message (image) sent and received accurately	Yes	No



Logout	**Logout button** Clicked	**System should go back to home page**	**Yes**	**No**

Message (group)	**Group icon** Clicked	**System should show all existing group information**	**Yes**	**No**
Message (Unseen)	Unseen icon Clicked	System should show all unseen message information	Yes	No

Search	**Text field** Washim	**System should show all user named Washim**	**Yes**	**Yes**
Search	Text field Wa	System should show all user named Washim	Yes	Yes

Figure 4.5: Various Manual Testing Condition generate for the system test.

Chapter 5

Conclusion

5.1 Practical Implication

The messenger chat application project have practical implication which are given in below-

- To keep up in this fast-paced world, we need to be able to adapt to constant changes. Communication is one of the most critical factors in any sector. So, this project can easily be used for communication.
- Furthermore, chat applications allow a person to engage with one another and communicate together to share their private talk and important topics by texting because it is more efficient for communication.

5.2 Scope of the future work

The messenger chat application project should have some scope of future work which are given in below-

- This project also motivated us to work in the future and we can add an individual information searching option in future.
- We can be work on group chatting system and individual incoming message box.
- This project can be adding more new looks with some dynamic options like delete, record option in future work.

5.3 Discussion

The project is about a messenger chat application which helps to learn about to use of communication application based projects. Then it can help to know about how to develop a communication application with socket programming. It was developed by java and for database connection using Mysql. Here, the whole project concept is about mechanism or chat system and socket programming. Inside this chat application, it will have some different features like login and signed-up page chatting page, active status shown and file and emoji transferring and many more features. In future, we can be added some new features group chatting system and incoming message box and also add individual search bar and GIF sending option and try for full bug free chat system. It can motivate us to work in the future with daily life problems and motivates us to work with communication technologies using socket programming. It does not solve any new problems as chatting based softwares already exists. What this project will do is that, it will reveal the underlying working principle and replicate all the existing chatting based softwares. This project is not fully bug free but enough to be an exhibition. So, that we can said that our project is completed.

5.4 References

1. Gaurav Tank. (2021). Chat program using Socket Programming. Results Retrieved from https://gauravtank2203.medium.com/chat-program-using-socket-programming-b6fb69834586 (2022, 19 December).

2. Mahmoud hamwi.(2020, October). Chatting App Design. https://www.youtube.com/watch?v=pV2331oV3yUt=1262s (2022, December)

3. W3School.(1999). Java Tutorial. Results Retrieved from https://www.w3schools.com/java/ (2022,December).