MBSTU Chat Box

A software development Project

Submitted By

Md. Mehedi Hasan Khan

ID: CE18024

Jotirmoy Roy

ID: CE18028

Md Hasan Uz Zaman

ID: CE18029

Supervised By

Md Hadifur Rahman

Associate Professor

Department of Computer Science and Engineering (CSE)

Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902

Bangladesh

Table of Contents

MBSTU Chat Box

Description

‘MBSTU Chat Box’ is a web-based application, which used for communicating among people easily. Here, we create an attractive interface for users to chat other users.

Firstly, user need to create a profile for using this application. Then, users make connection or friend, which he/she choose. Users can easily chatting with each other’s if they are friend and also we added here for posting, picture sharing, commenting, reacting on post(Like, Love). Users can make group to users for making group chatting. They can easily connected using group and get extra advantages to send one text to all other users in the group.

The purpose of the project is to build an application program to reduce the manual work for managing the chat profile, chat user, multi chat, and chat history.

1. Analysis and domain modeling

Conceptual model: Conceptual model is represented the total system. Here, users can send message the other users. And he/she can see the notification of incoming messages. Also, a lot of features is here for the users.

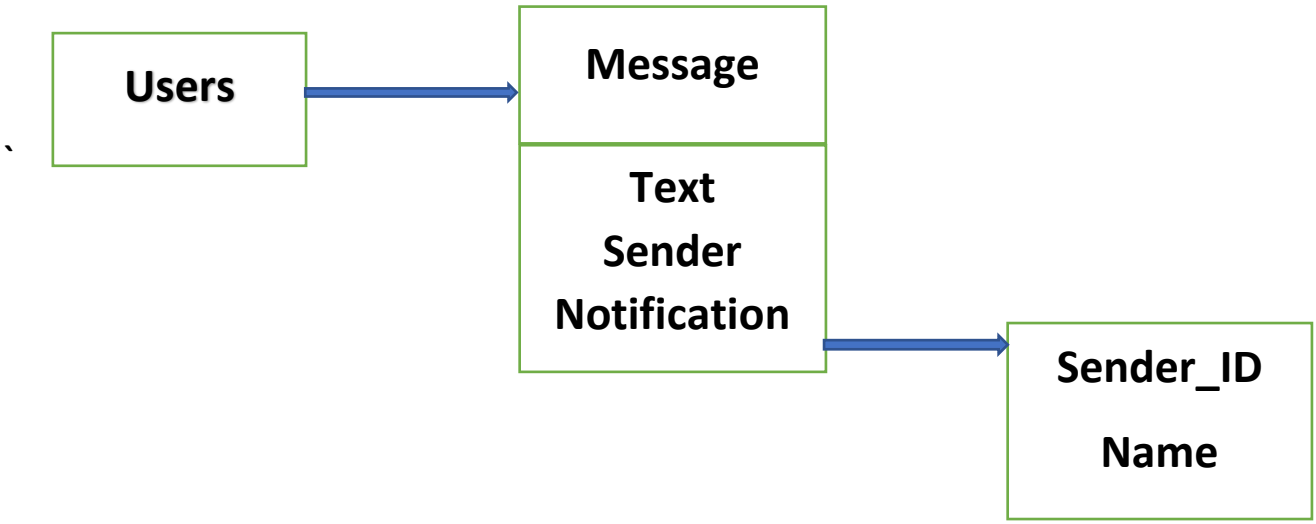


Figure: Conceptual model of chat box

Data model:

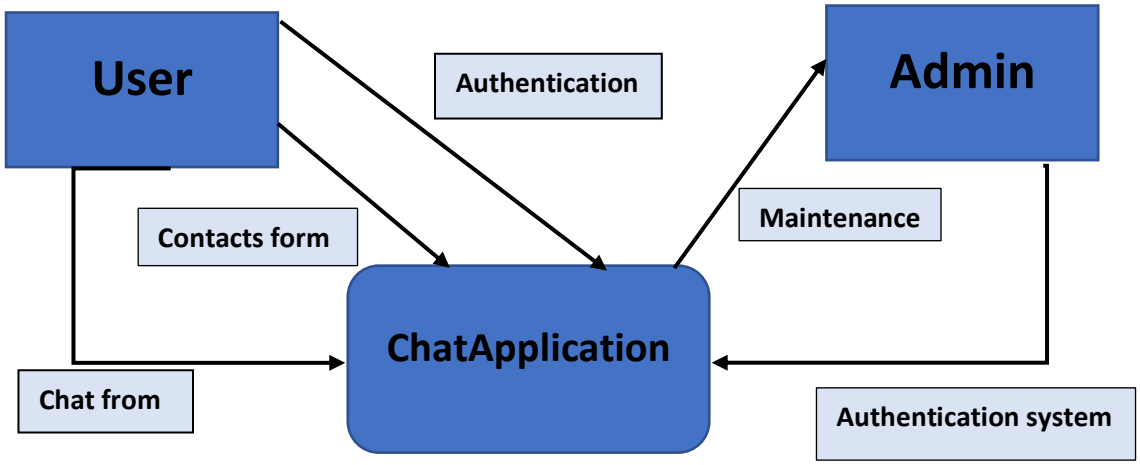


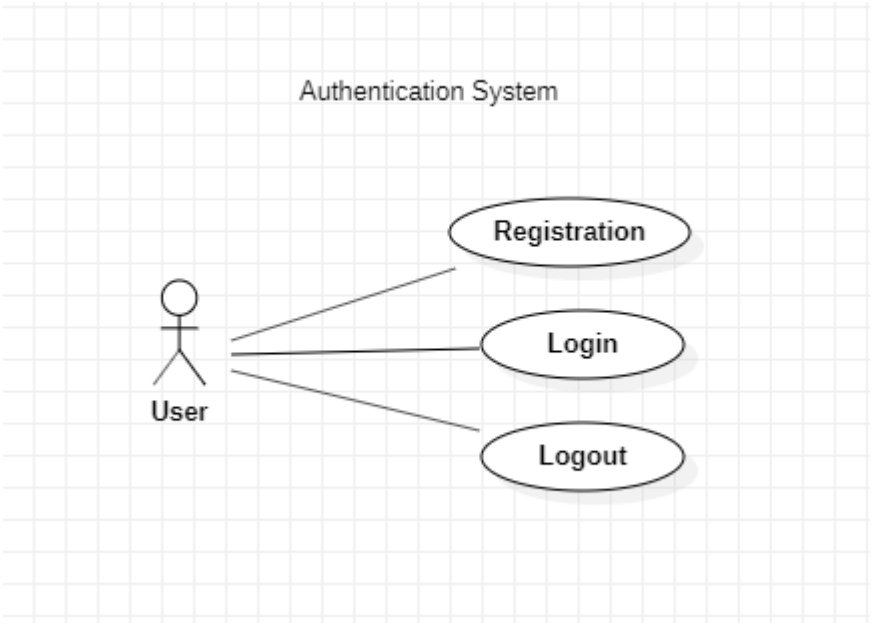
Figure: Data Flow diagram

2. Interaction diagram

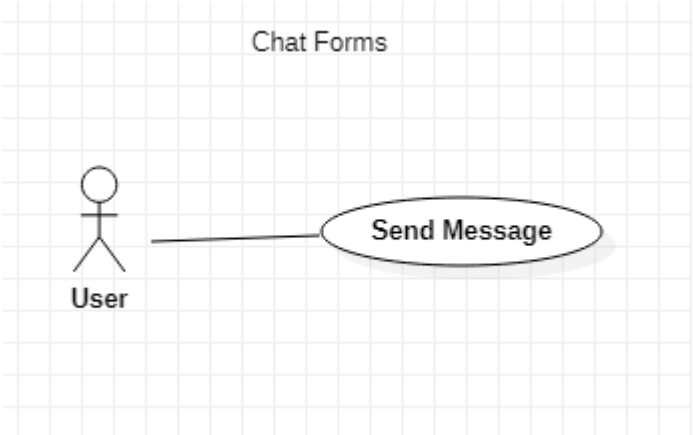
Use case diagram:

Here, we discuss the application with the use case diagram. We divided into some parts of the project for understanding better.

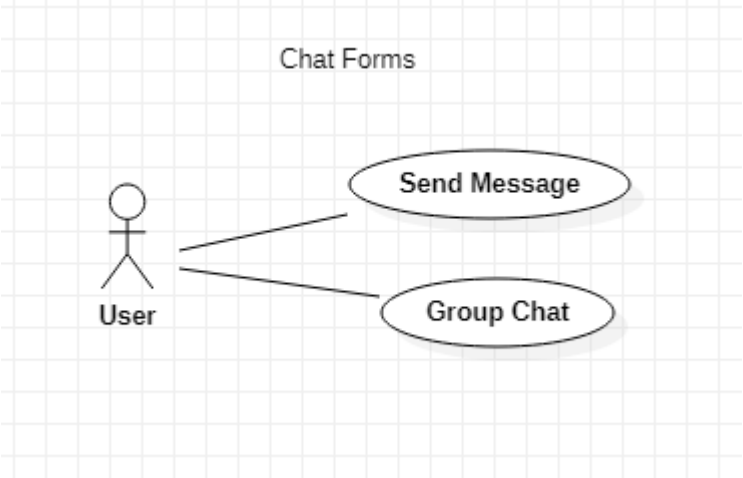
Authentication system use case diagram:



Chat use case diagram:

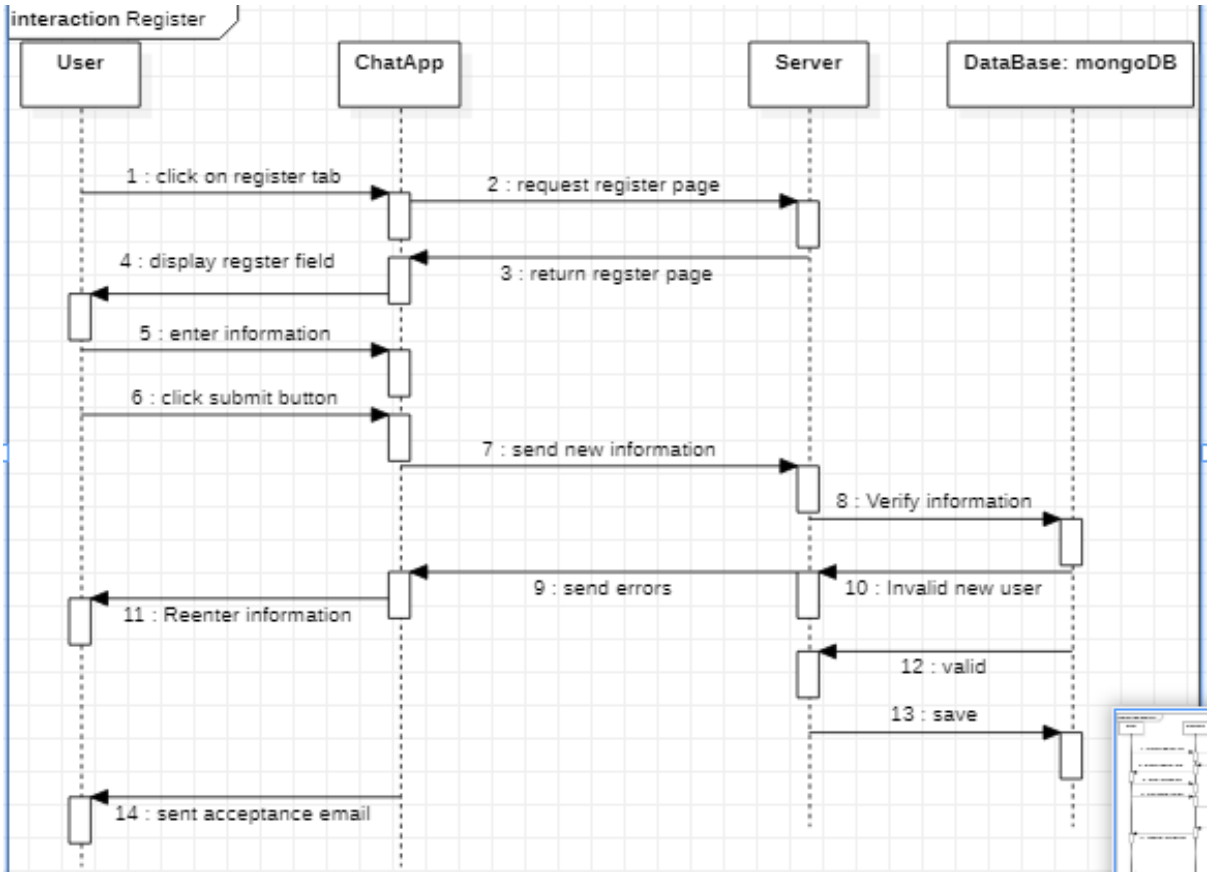


Chat form use case diagram

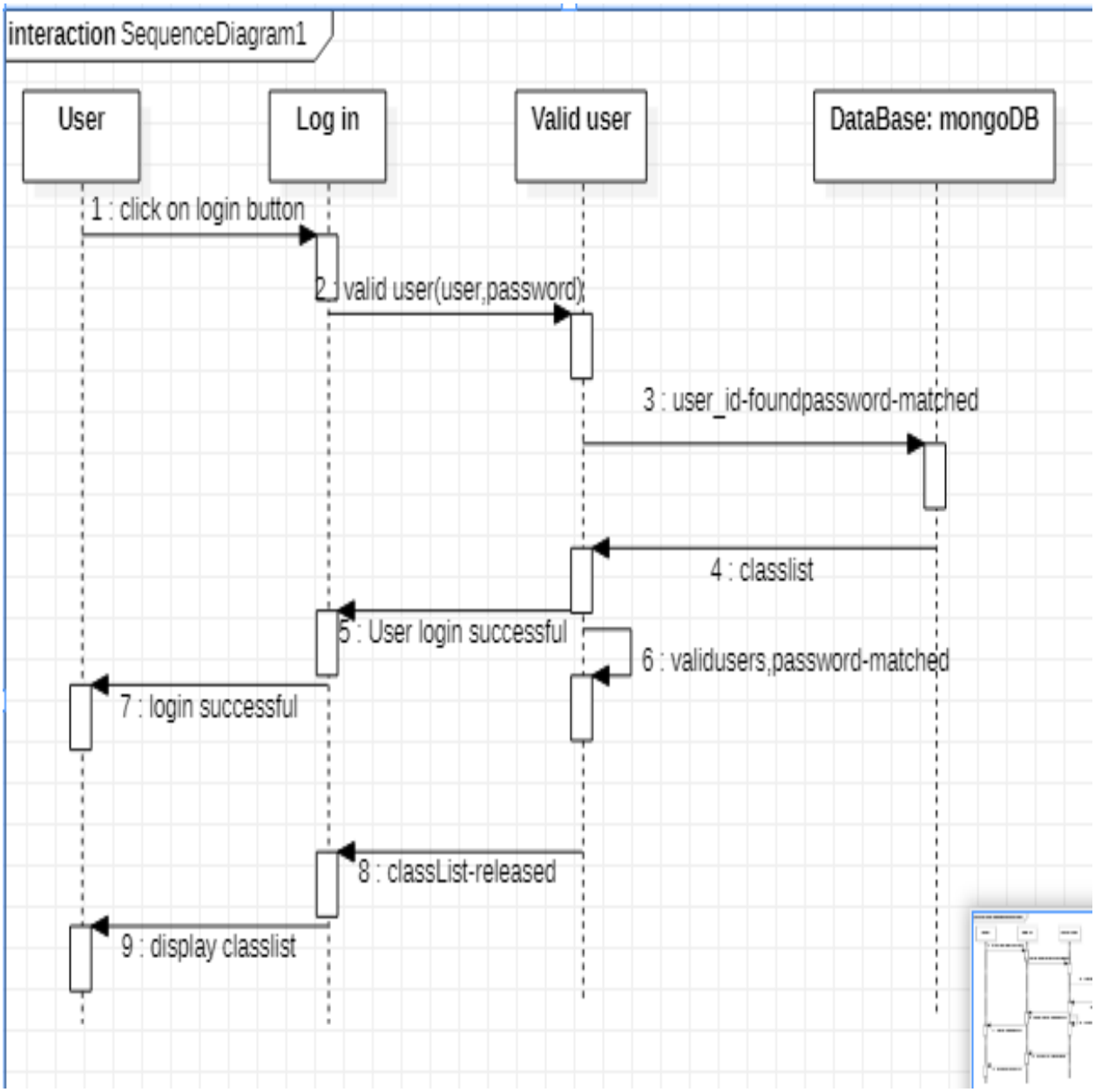


<u>Sequence diagram:</u> Here we are showing the sequence of messages between objects in an interaction.

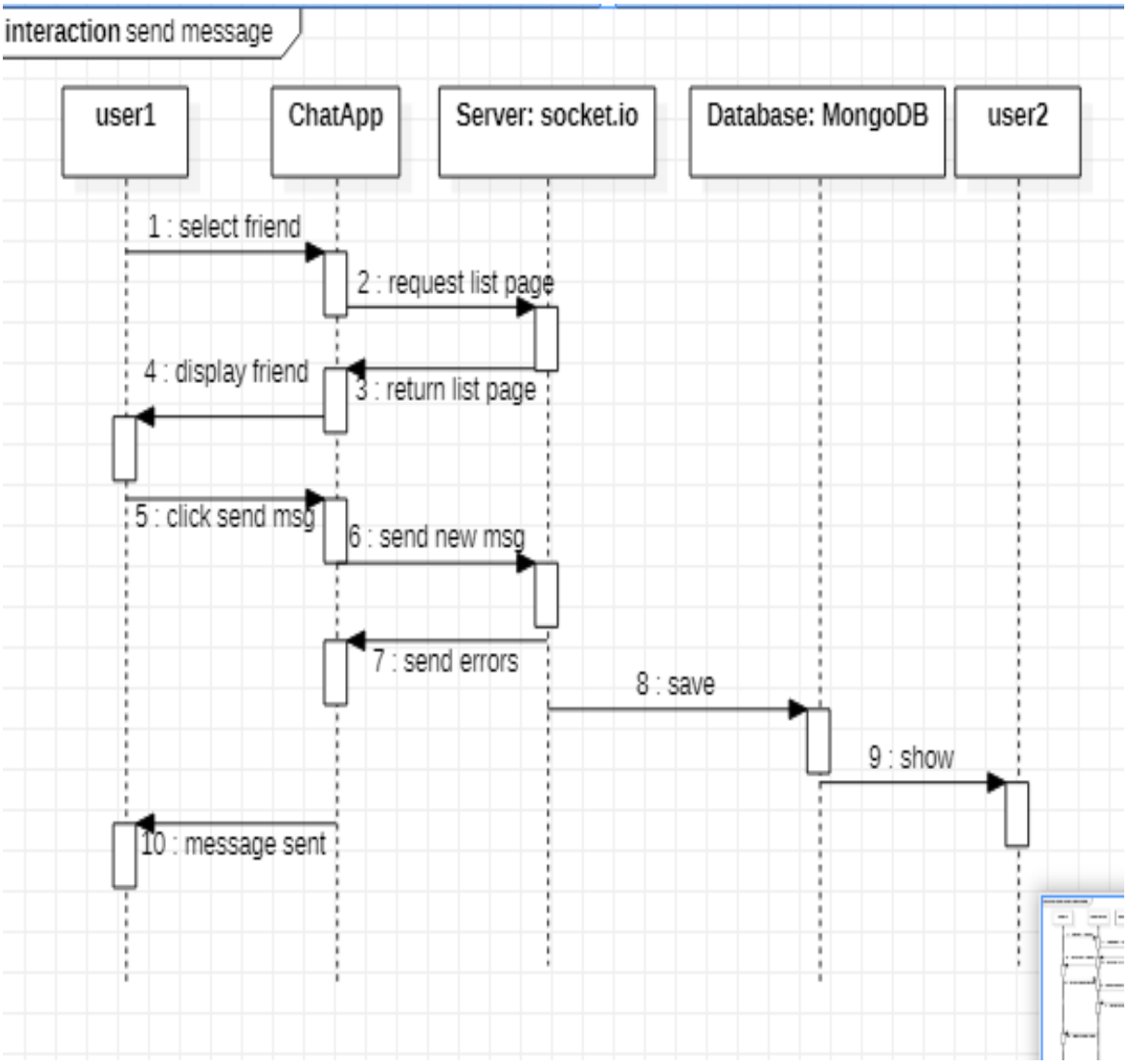
Register of a user's sequence diagram:



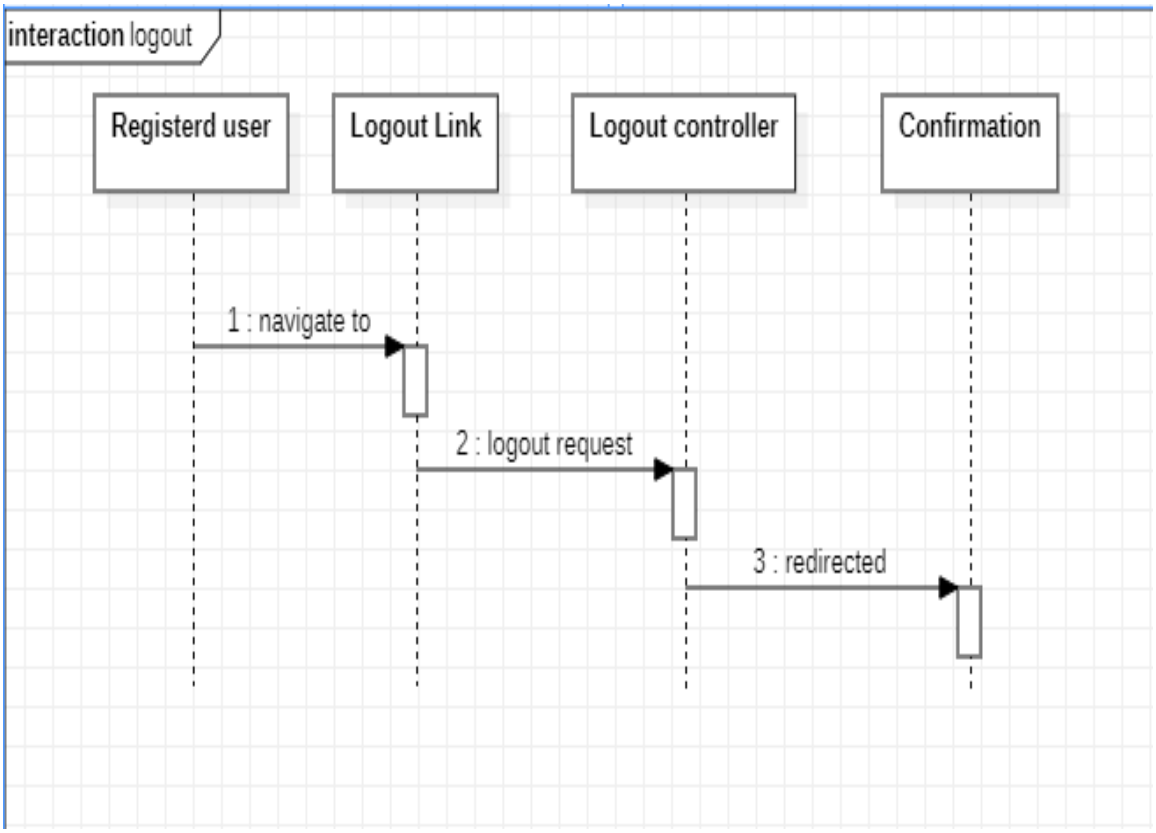
Login of a user's sequence diagram:



Send message of a user's sequence diagram:

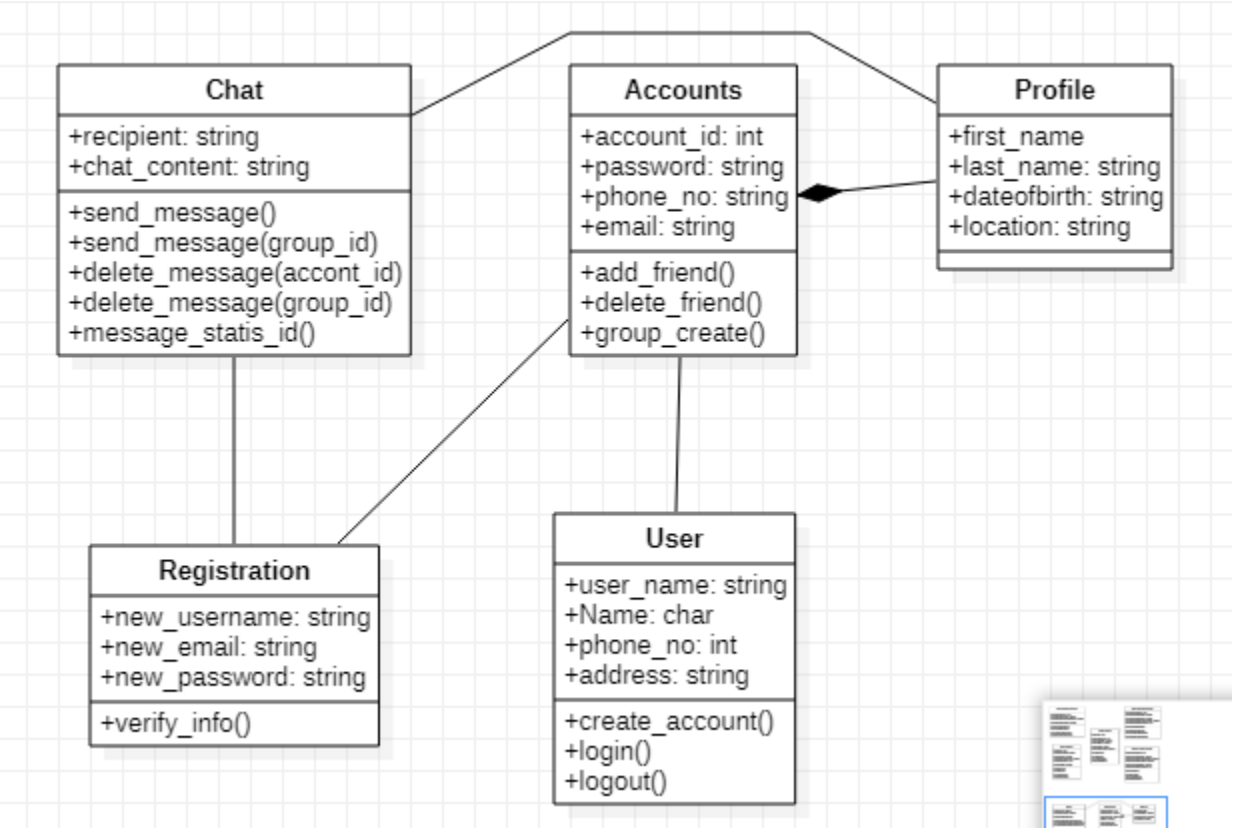


Logout sequence diagram:



3. Class diagram and interface specification

Class Diagram: Class diagram of the chat application is under where we draw a class diagram model.



Data Types and Operation Signatures:

Classes of chat application and their operations:

Chat Class: Manage all the operation of Chat

Chat Class operations: send_message(), messages_state_id()

User Class: Manage all the operation of users.

User Class operations: create_account(), login(), logout()

Accounts Class: Manage all the operation of accounts of the users.

Accounts Class operations: add_friend(), delete_friend()

Profile Class: Manage all the operation of profile of the users.

Registration Class: Manage all the operation of registration.

Registration Class operations: verify_info()

4. Algorithm and Data structures

Algorithm:

Complex algorithm is not use here. Just normal easy algorithm used here like searching, sorting, binary searching etc etc.

Data structure:

Here, we used some data structures. Like arrays, queues. Arrays are used for string data, which come from database (mongoDB). In addition, queues are used for message part. Here, storing text message, which come first and show first. Also it used in notification sorting.

5. User interface design and implementation

User interface (UI) is an important part for making software. User interface present the software or application to the users. Here, we try to make an interesting interface for the uses.

Firstly, we take ideas from Facebook and Twitter. Moreover, UI is designed combining Facebook and Twitter. Here, first users need to register a good-looking interface by giving data of users. Then need to login there. Then users can use the features of the application. User can make friend by following, make follower, sending messages, giving post with image etc.

User interface is implemented using JavaScript, HTML, CSS, Reactjs, bootstrap, ANPD design etc. all are used to implemented the UI.

6. Design of tests

Unit Testing:

Here, we test every section of our project. Is everything working properly or not.

Register: Here, we check the mail address is valid or not. Then check the mail is used before or not. Then checking the password is strong or not.
If anything is going wrong or showing incorrect then need to change.

Login: Here, we check the mail and password. If anything is going wrong or showing incorrect then need to change.

Messaging: Here, we check selected users is not selected or not. Then Check the text message is not going to its destination or not. If anything is going wrong or showing incorrect then need to change.

Making friends: Here, we check making friends is working properly or not.
If anything is going wrong or showing incorrect then need to change.

Unfollow friends: Here, we check making unfollow is working properly or not. If anything is going wrong or showing incorrect then need to change.

Posting: Here, we check post making in timeline is working properly or not.
If anything is going wrong or showing incorrect then need to change.

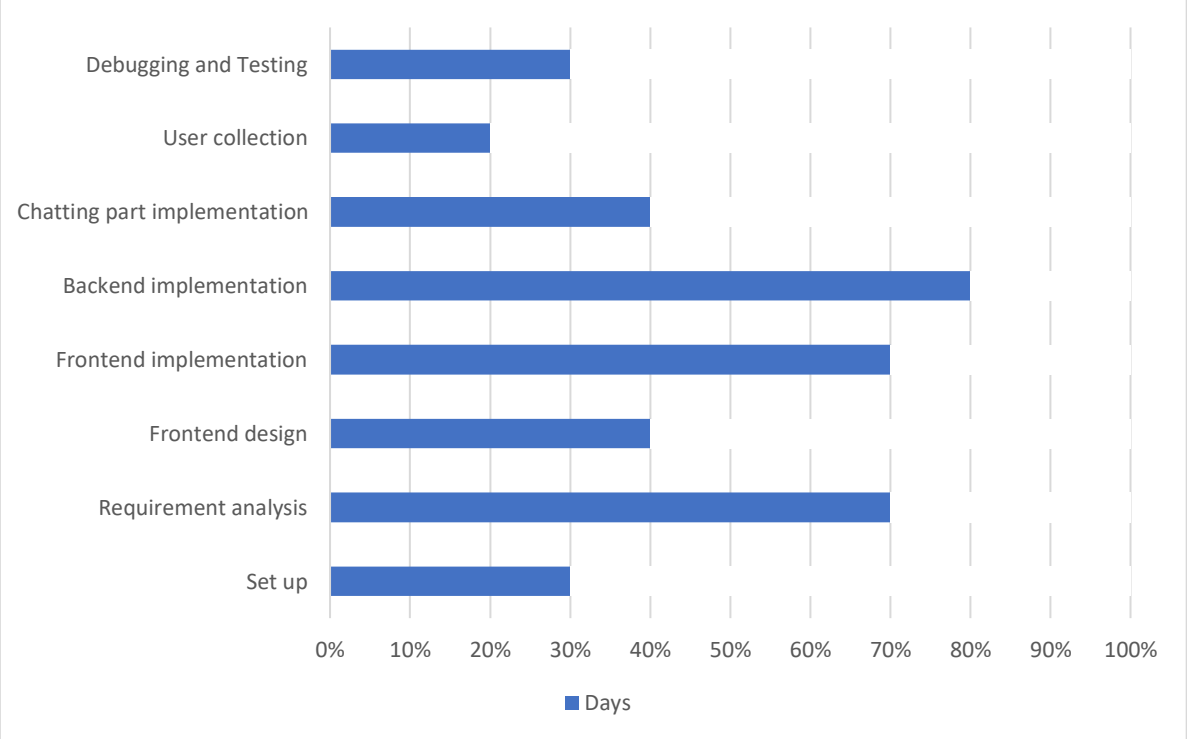
Logout: Here, we check logout is working properly or not.
If anything is going wrong or showing incorrect then need to change.

7. Project management and plan of work

Merging the Contributions from Individual Team Members:

List	Md Mehedi Hasan Khan	Md Hasan Uz Zaman	Jotirmoy Roy
Requirement Specification	34%	33%	33%
Software Design	33%	34%	33%
Codding	40%	30%	30%
Debugging and Testing	33%	33%	34%
Report & presentation Preparation	33%	33%	34%

Plan of work:



Breakdown of Responsibilities:

1. **Md. Hasan Uz Zaman:**

   Users and posts API design and implementation.

   API implementation for sending messages.

   Password hashing.

2. **Md. Mehedi Hasan Khan:**

   Frontend design and implementation.

   API testing and integrating from team members.

   User authentication by using Firebase.

3. **Jotirmoy Roy:**

   API Conversation and message design and implementation.

   API implementation for sending messages.

   Password hashing.

Project link:

Live link: https://mbstu-chat-box.web.app/home

Client code: https://github.com/Mehedi-Developer/mbstu-chat-box-client

Server code: https://github.com/Mehedi-Developer/mbstu-chat-box-server/blob/main/README.md

8. References

https://reactjs.org/docs/getting-started.html

https://expressjs.com/en/api.html

https://firebase.google.com/docs

https://docs.mongodb.com/

https://devcenter.heroku.com/categories/reference

https://nodejs.org/en/docs/

https://sequencediagram.org/instructions.html

https://www.tutorialspoint.com/uml/uml_class_diagram.htm

https://docs.dhtmlx.com/gantt/