

A PROJECT REPORT

ON

Real-Time Chat Application

By

PRAJAPATI NISARG N. (CE-110) (19CEUOS111)

PATEL YASH K. (CE-107) (19CEUON068)

PATEL KRISHIL A. (CE-101) (19CEUOG147)

B.Tech CE Semester-IV

Subject: Software Engineering Practice

Guided by: Prof. Pinkal C. Chauhan

Assistant Professor

Dept. of Comp. Engg.



**Faculty of Technology Department of Computer Engineering
Dharmsinh Desai University**



**Faculty of Technology Department of Computer Engineering
Dharmsinh Desai University**

CERTIFICATE

This is to certify that the practical / term work carried out in the subject of

Software Engineering Practice and recorded in this

journal is the bonafide work of

PRAJAPATI NISARG N. (CE-110) (19CEUOS111)

PATEL YASH K. (CE-107) (19CEUON068)

PATEL KRISHIL A. (CE-101) (19CEUOG147)

of B.Tech semester **IV** in the branch of **Computer Engineering**

during the academic year **2020-2021**.

Prof. Pinkal C. Chauhan
Assistant Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia,
Head,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Table of Content

1 Introduction.....	4
1.1 Project Details: Brief Introduction	4
1.2 Technology and Tools Used	5
2 Software Requirement Specifications	6
2.1 System Functional Requirements	6
3 Design.....	9
3.1 Use Case Diagram	9
3.2 Sequence Diagram.....	10
3.3 Activity Diagram.....	12
3.4 Class Diagram.....	14
3.5 Data Flow	15
3.6 Data Dictionary	16
4 Implementation Details	20
4.1 Functional prototypes	20
5 Screen-shots of the System	23
6 Conclusion	26
7 Limitations and Future Extensions of System	27
8 Bibliography	27

1 Introduction

1.1 Brief Introduction

Real-Time Chat Application” is an online chatting platform. The user Can communicate with many other user in various geographical locations.

This type of system are also used in huge company for their employee to communicate among the teammates and other teams also.

1.2 Tools/Technologies Used

Technologies:

- React.js
- Redux
- Sementic Ui
- Javascript
- JSX

Tools

- Git
- Visual Studio Code

Platform

Local development server
Google Firebase (Database)

2. Software Requirement Specifications

2.1 System Functional Requirements

R1 Authentication

R1.1 Registration

Description: If user is new, it gets register himself with email address and password.

Input: Valid e-mail address and strong password.

Output: Account created and redirected to next screen.

R1.2 Login

Description: User should login through registered e-mail id.

Input: E-mail and password

Output: Successfully login redirected to chat screen.

R2 Messaging

R2.1 Convey a message

Description: The registered person should be able to send and receive instant message to/from any user on his/her user's list given in user panel.

Input: Message to be send.

Output: Message is displayed on the right side on sender's chat box and left side on receiver's end.

R2.2 Deliver Attachments

Description: The person should be able to send or receive videos, documents, audios, images to another user.

Input: Videos, documents, audios, images

Output: Attachments is displayed on the right side on sender's chat box and left side on receiver's end.

R2.3 Group Message

Description: It allow the user to create group of peoples to talk together within the group and share the attachments.

Input: Videos, documents, audios, images

Output: Attachments is displayed on the right side on sender's chat box and left side on receiver's end.

R2.4 Search Message

Description: The user will also get the authority to search a snippet of message in anyone's chat from the whole chat history.

Input: Snippet to be searched.

Output: : Highlight the matched words in yellow color.

R3 Manage Users

R3.1 Block User

Description: The admin can block the targeted user in case of any misbehavior, or abusive language.

R3.2 Settings

Description: It provides the user to customize their settings as per their choice.

R3.3 Profile Picture

Description: User can be able to change his/her profile picture whenever he/she wants to. We can use image (.jpg, .png, .jpeg) for the same.

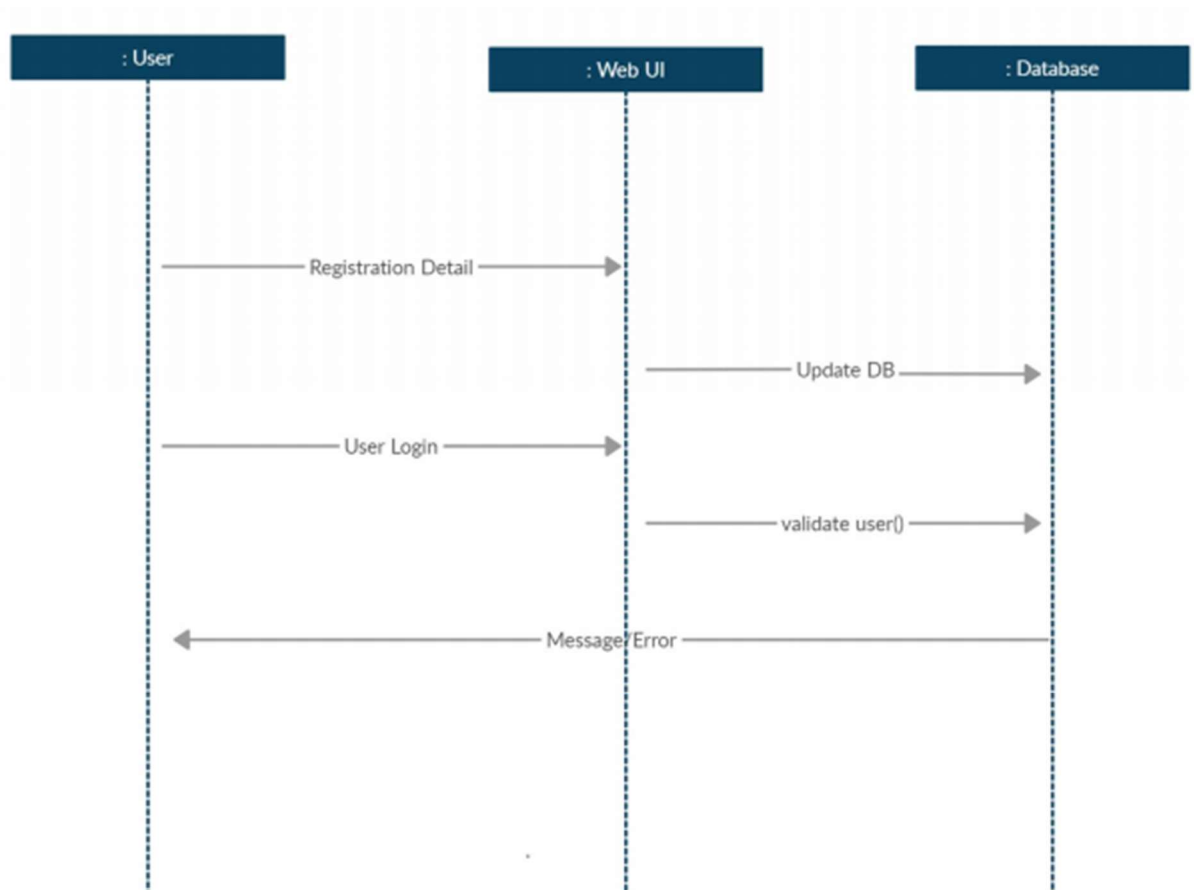
3. Design

3.1 Use Case Diagram



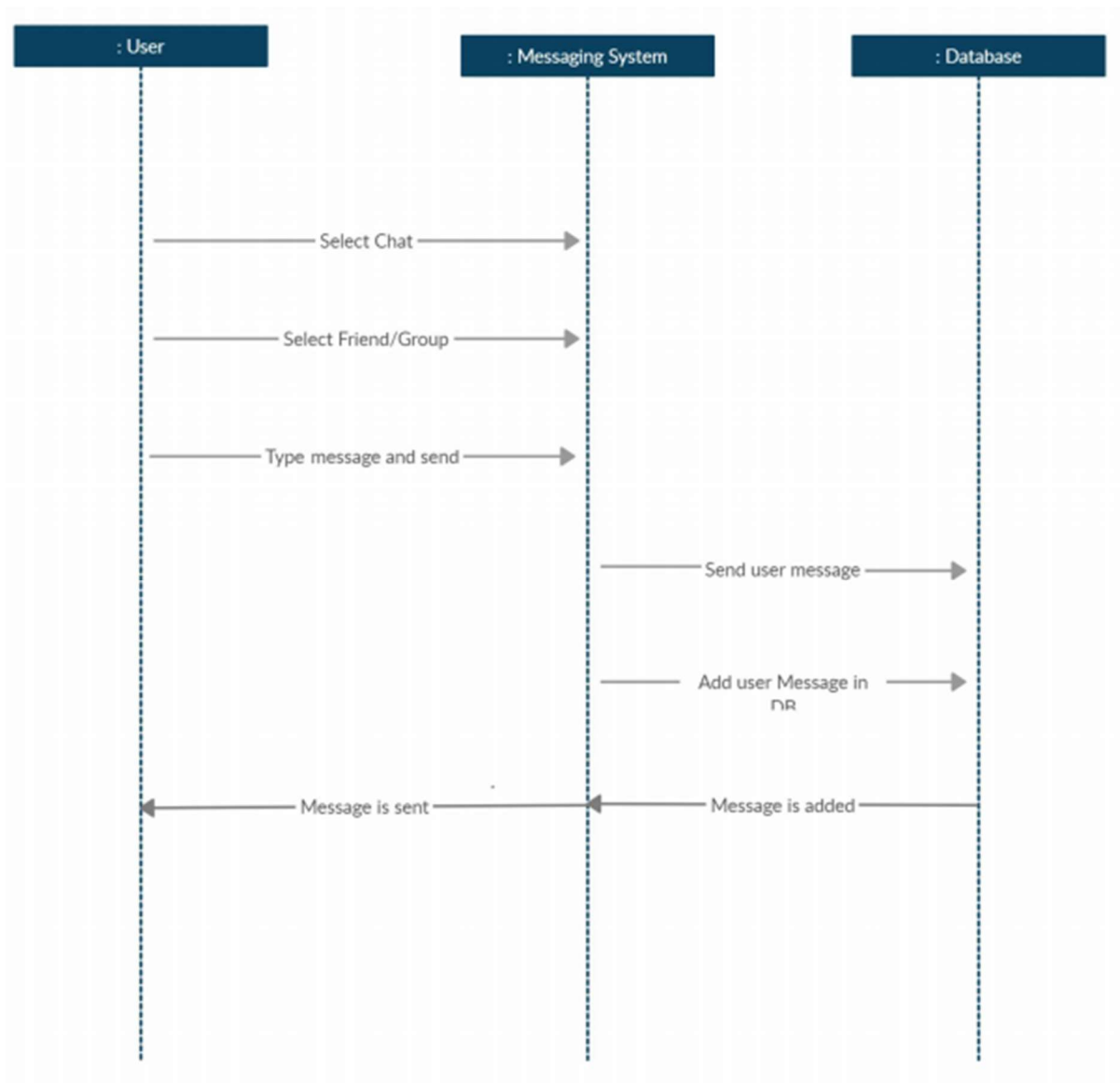
Use case diagram of user

3.2 Sequence Diagram



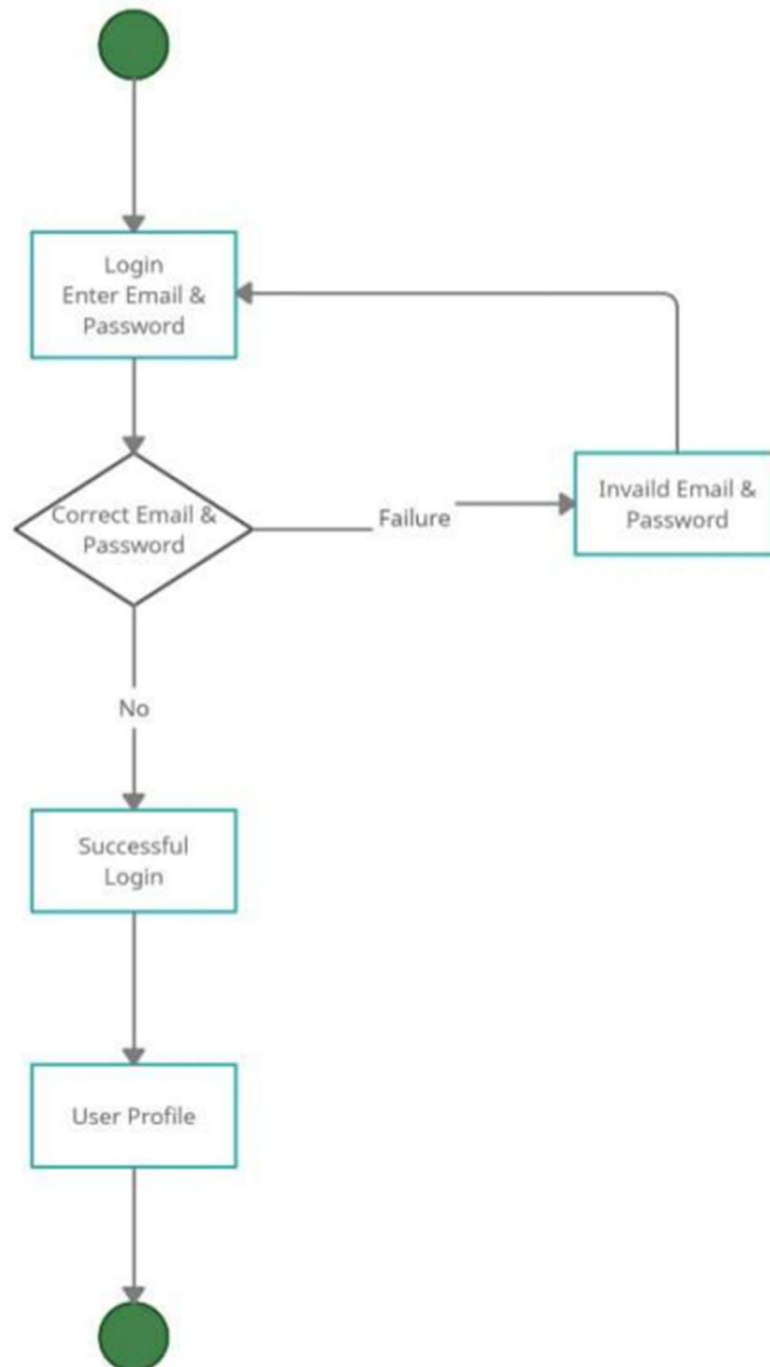
Sequence Diagram of Login/Signup

3.2 Sequence Diagram



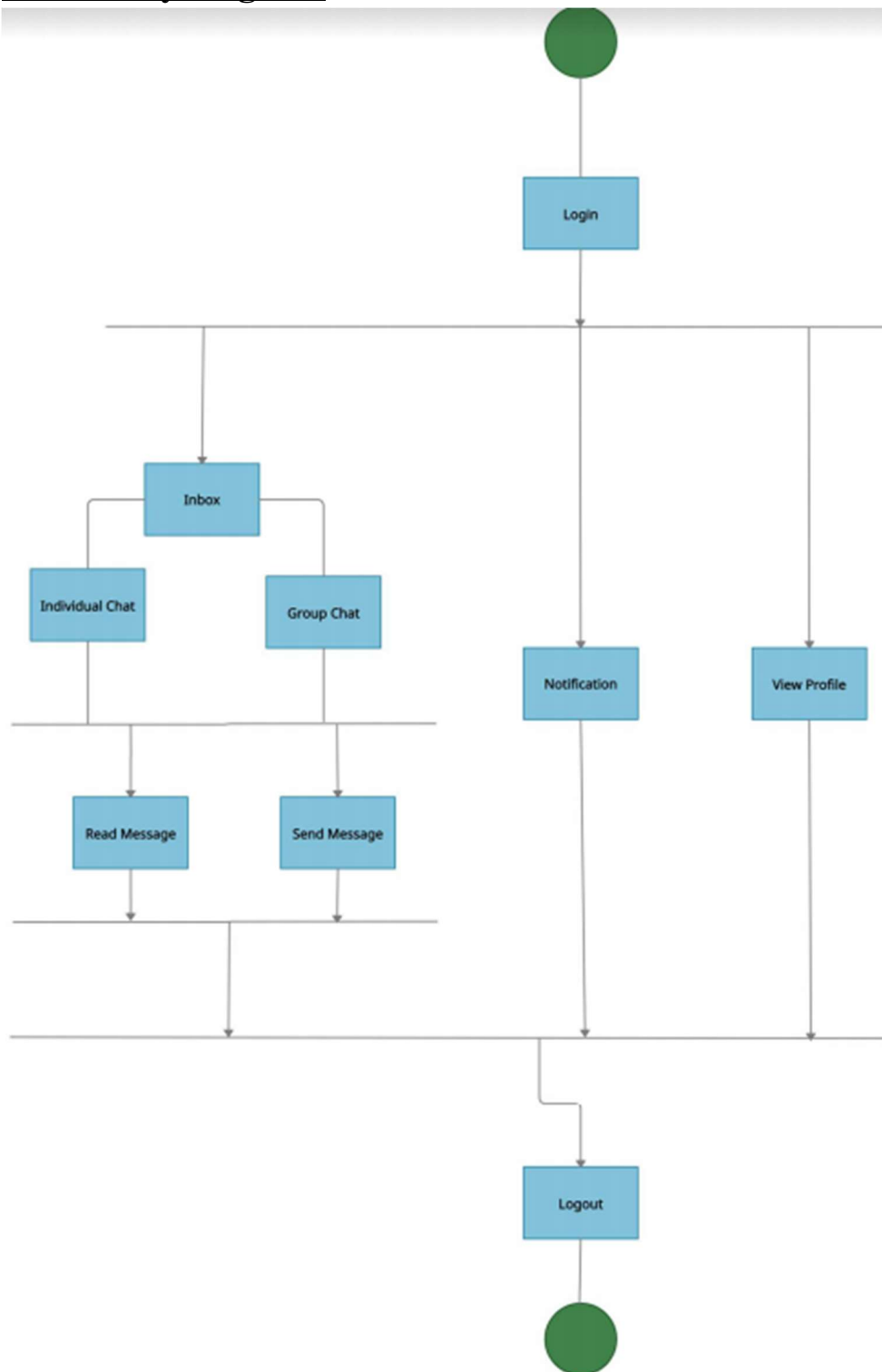
Sequence Diagram of Messaging

3.3 ActivityDiagram



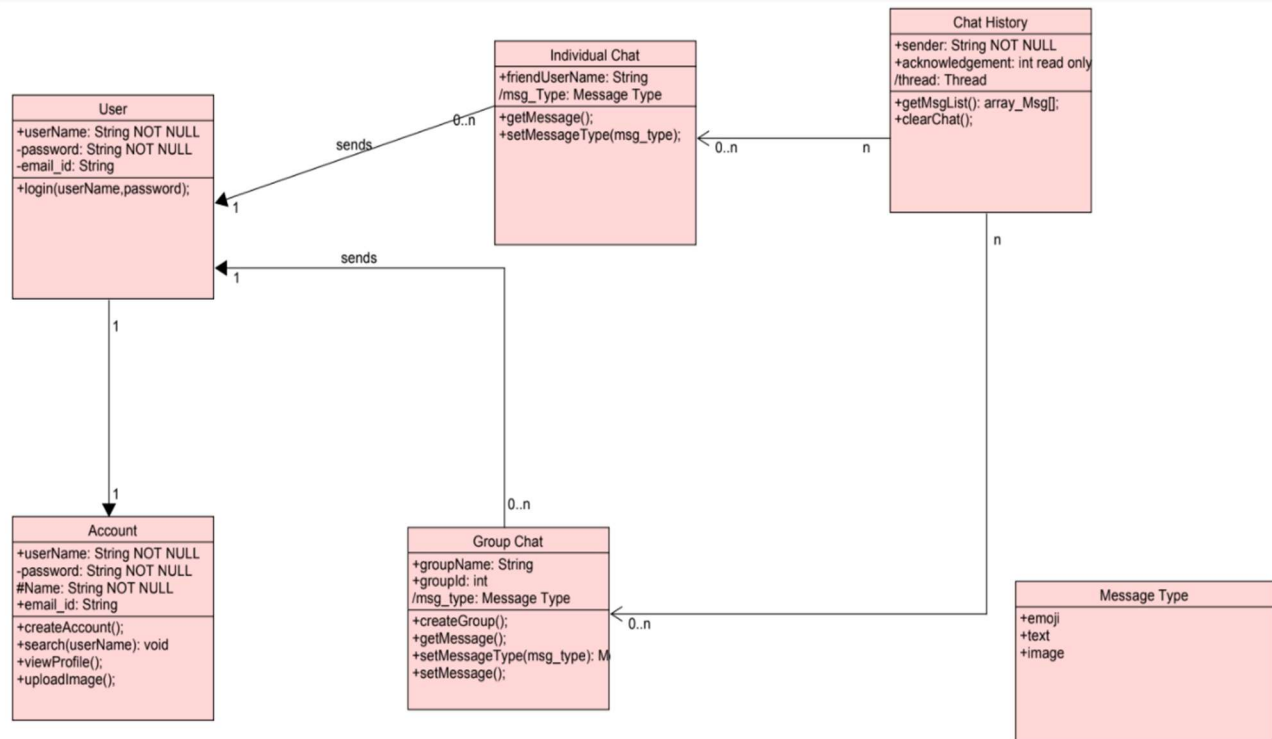
Activity Diagram of Login

3.3 Activity Diagram



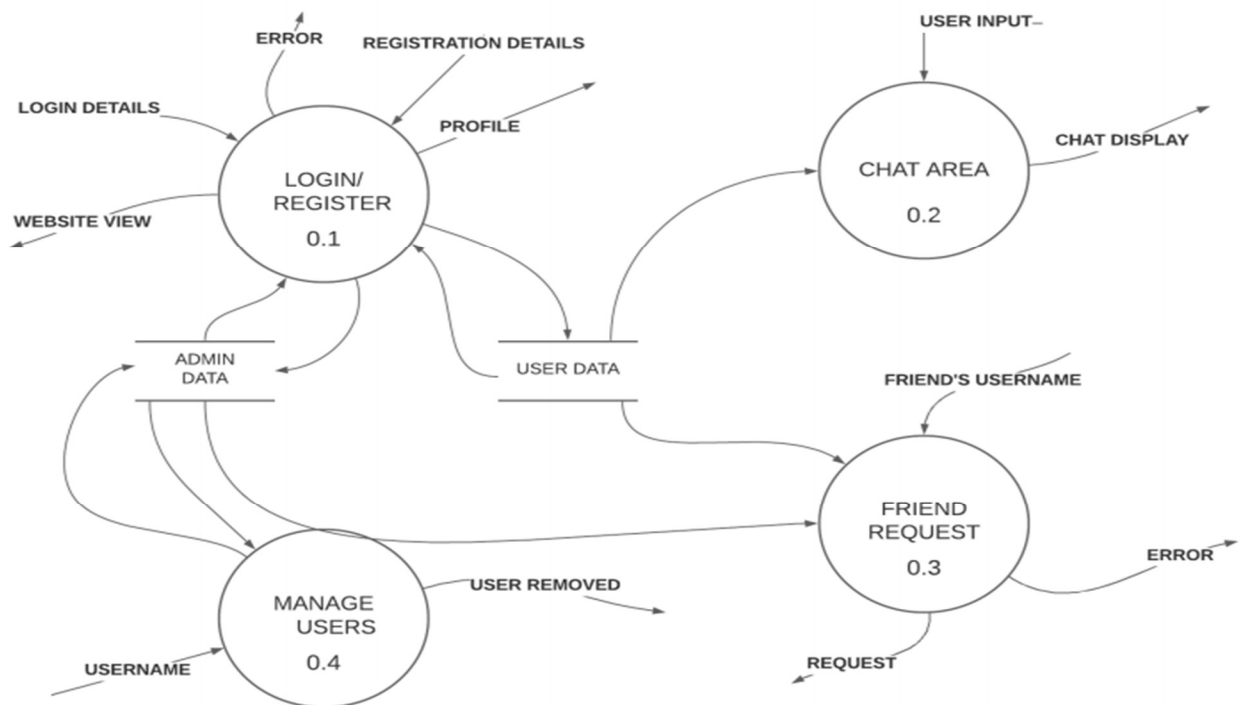
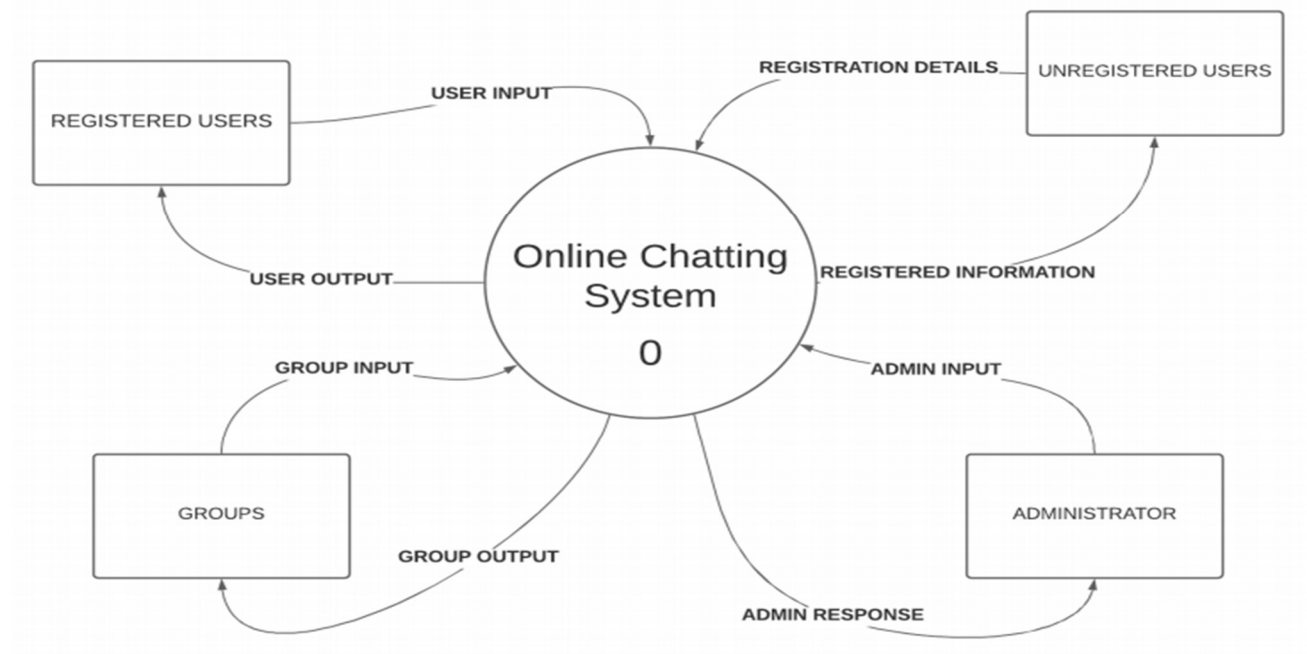
Activity Diagram of Messaging

3.4 Class Diagram



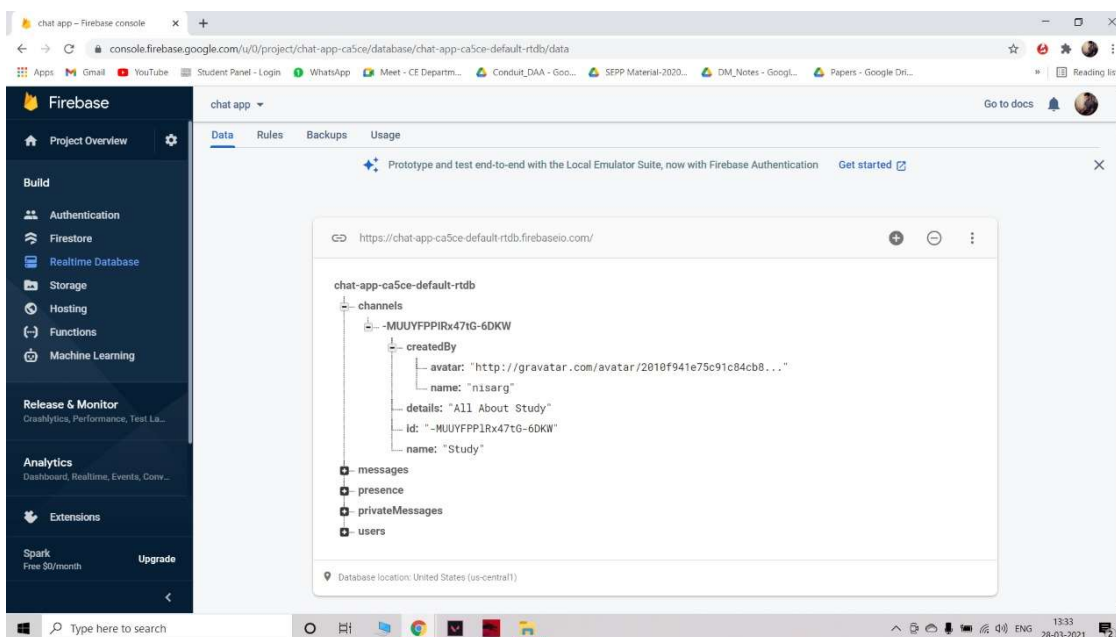
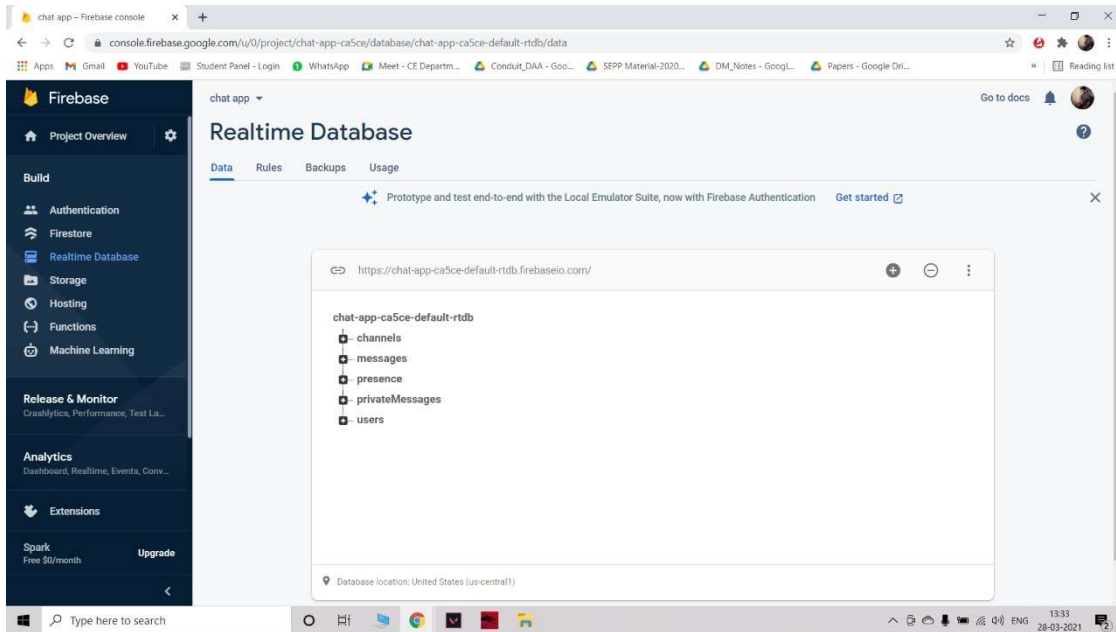
Class diagram

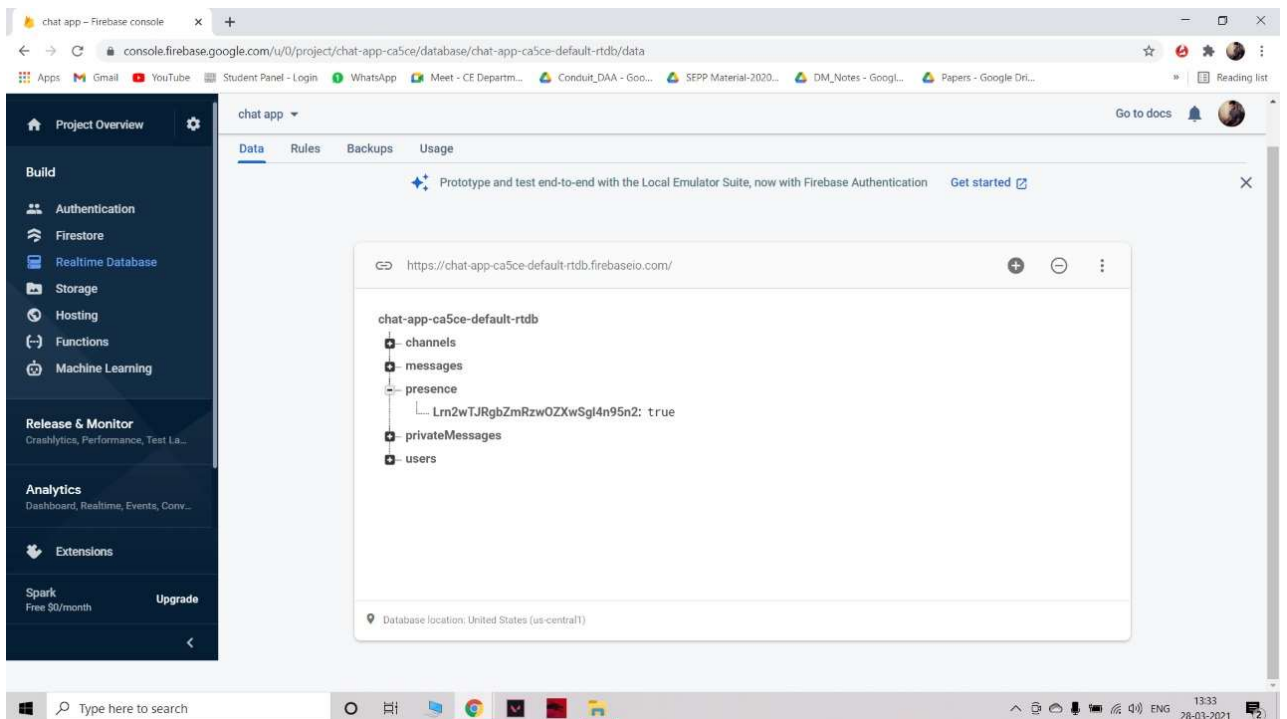
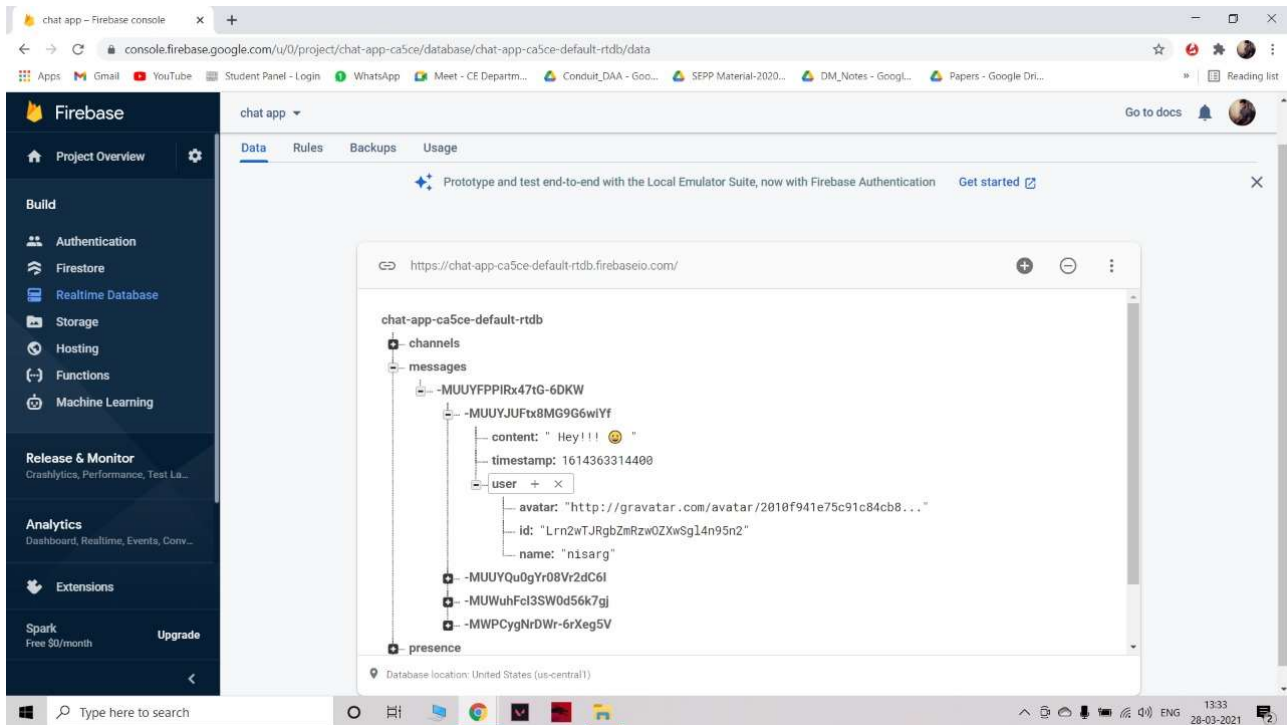
3.5 Data Flow



3.6 Data Dictionary

In our project we have used the NoSQL database so we cannot create a data dictionary as well as entity relationship diagram but we are providing here some pictures of our database. We have used the Google firebase database. It doesn't have a table, row & column but it has a collection document & field as shown in below pictures.





chat app - Firebase console

console.firebase.google.com/u/0/project/chat-app-ca5ce/database/chat-app-ca5ce-default-rtdb/data

Apps Gmail YouTube Student Panel - Login WhatsApp Meet - CE Departm... Conduit_DAA - Goo... SEPP Material-2020... DM_Notes - GoogL... Papers - Google Dri... Reading list

Go to docs

chat app

Data Rules Backups Usage

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication [Get started](#)

<https://chat-app-ca5ce-default-rtdb.firebaseio.com/>

```
presence
  privateMessages
    Lm2wTJRgbZmRzwOZXwSgl4n95n2
      PYHFocGhSRRxLkuAe8Xn2LegzJp2
        MWPB6sLFapWZuLVwzWk
          content: "Hey"
          timestamp: 1616428831811
          user
            avatar: "http://gravatar.com/avatar/20eb35f3bb26acec8aff..."
            id: "PYHFocGhSRRxLkuAe8Xn2LegzJp2"
            name: "Yash Patel"
        XIHyDoNlvaWJt7dVAH61KaZyhuy1
      PYHFocGhSRRxLkuAe8Xn2LegzJp2
      XIHyDoNlvaWJt7dVAH61KaZyhuy1
  users
```

Database location: United States (us-central1)

Type here to search

13:34 28-03-2021

chat app - Firebase console

console.firebase.google.com/u/0/project/chat-app-ca5ce/database/chat-app-ca5ce-default-rtdb/data

Apps Gmail YouTube Student Panel - Login WhatsApp Meet - CE Departm... Conduit_DAA - Goo... SEPP Material-2020... DM_Notes - GoogL... Papers - Google Dri... Reading list

Go to docs

chat app

Data Rules Backups Usage

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication [Get started](#)

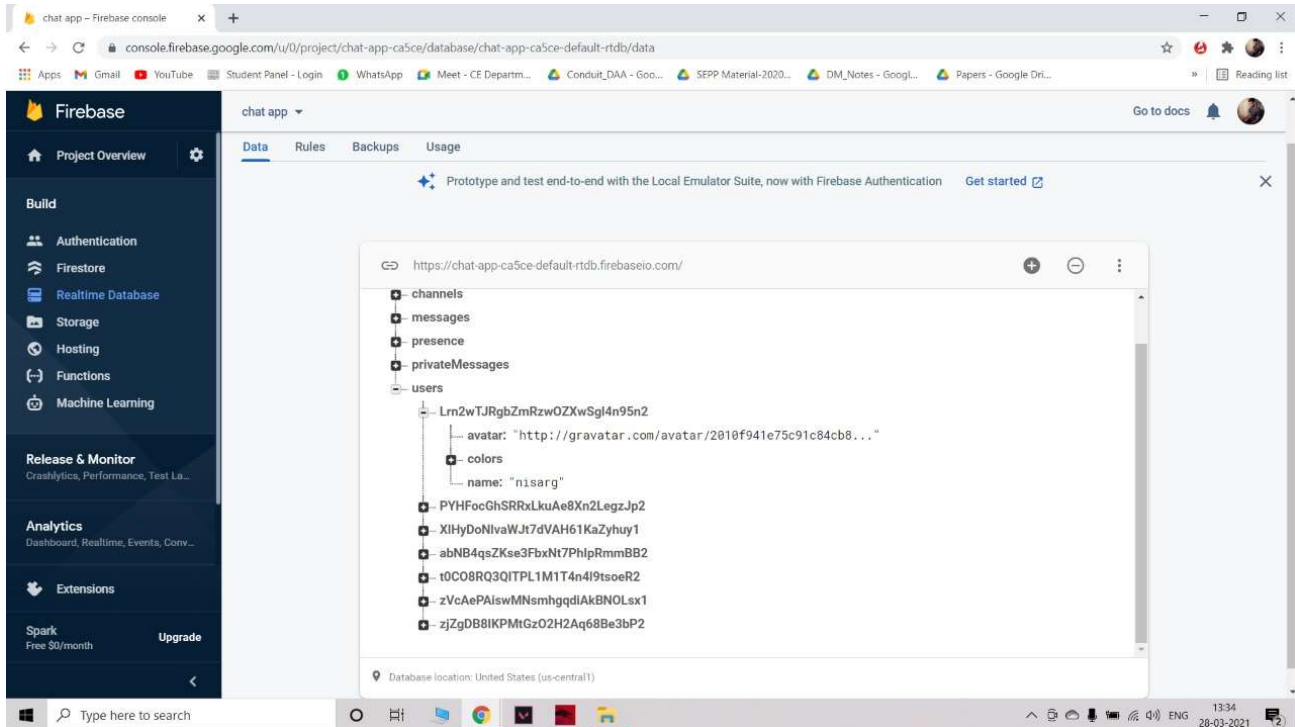
<https://chat-app-ca5ce-default-rtdb.firebaseio.com/>

```
channels
messages
presence
privateMessages
users
  Lm2wTJRgbZmRzwOZXwSgl4n95n2
    avatar: "http://gravatar.com/avatar/2010f941e75c91c84cb8..."
    colors
    name: "nisarg"
  PYHFocGhSRRxLkuAe8Xn2LegzJp2
  XIHyDoNlvaWJt7dVAH61KaZyhuy1
  abNB4qsZKse3FbxNt7PhlpRmmBB2
  t0CO8RQ3QITPL1M1T4n4l9ts0eR2
  zVcAePAiswMNsMhgqdlAkBNOLsx1
  zjZgDB8lKPMtGzO2H2Aq68Be3bP2
```

Database location: United States (us-central1)

Type here to search

13:34 28-03-2021



4. Implementation Details

4.1 Function prototypes

```
handleSubmit = event => {
  event.preventDefault();
  if (this.isFormValid()) {
    this.setState({ errors: [], loading: true });
    firebase
      .auth()
      .createUserWithEmailAndPassword(this.state.email, this.state.password)
      .then(createdUser => {
        console.log(createdUser);
        createdUser.user
          .updateProfile({
            displayName: this.state.username,
            photoURL: `http://gravatar.com/avatar/${md5(
              createdUser.user.email
            )}?d=identicon`
          })
          .then(() => {
            this.saveUser(createdUser).then(() => {
              console.log("user saved");
            });
          })
          .catch(err => {
            console.error(err);
            this.setState({
              errors: this.state.errors.concat(err),
              loading: false
            });
          });
      })
      .catch(err => {
        console.error(err);
        this.setState({
          errors: this.state.errors.concat(err),
          loading: false
        });
      });
  }
};
```

Register User

```

handleSubmit = event => {
  event.preventDefault();
  if (this.isFormValid(this.state)) {
    this.setState({ errors: [], loading: true });
    firebase
      .auth()
      .signInWithEmailAndPassword(this.state.email, this.state.password)
      .then(signedInUser => {
        console.log(signedInUser);
      })
      .catch(err => {
        console.error(err);
        this.setState({
          errors: this.state.errors.concat(err),
          loading: false
        });
      });
  }
};

```

Login

```

sendMessage = () => {
  const { getMessagesRef } = this.props;
  const { message, channel, user, typingRef } = this.state;

  if (message) {
    this.setState({ loading: true });
    getMessagesRef()
      .child(channel.id)
      .push()
      .set(this.createMessage())
      .then(() => {
        this.setState({ loading: false, message: "", errors: [] });
        typingRef
          .child(channel.id)
          .child(user.uid)
          .remove();
      })
      .catch(err => {
        console.error(err);
        this.setState({
          loading: false,
          errors: this.state.errors.concat(err)
        });
      });
  } else {
    this.setState({
      errors: this.state.errors.concat({ message: "Add a message" })
    });
  }
};

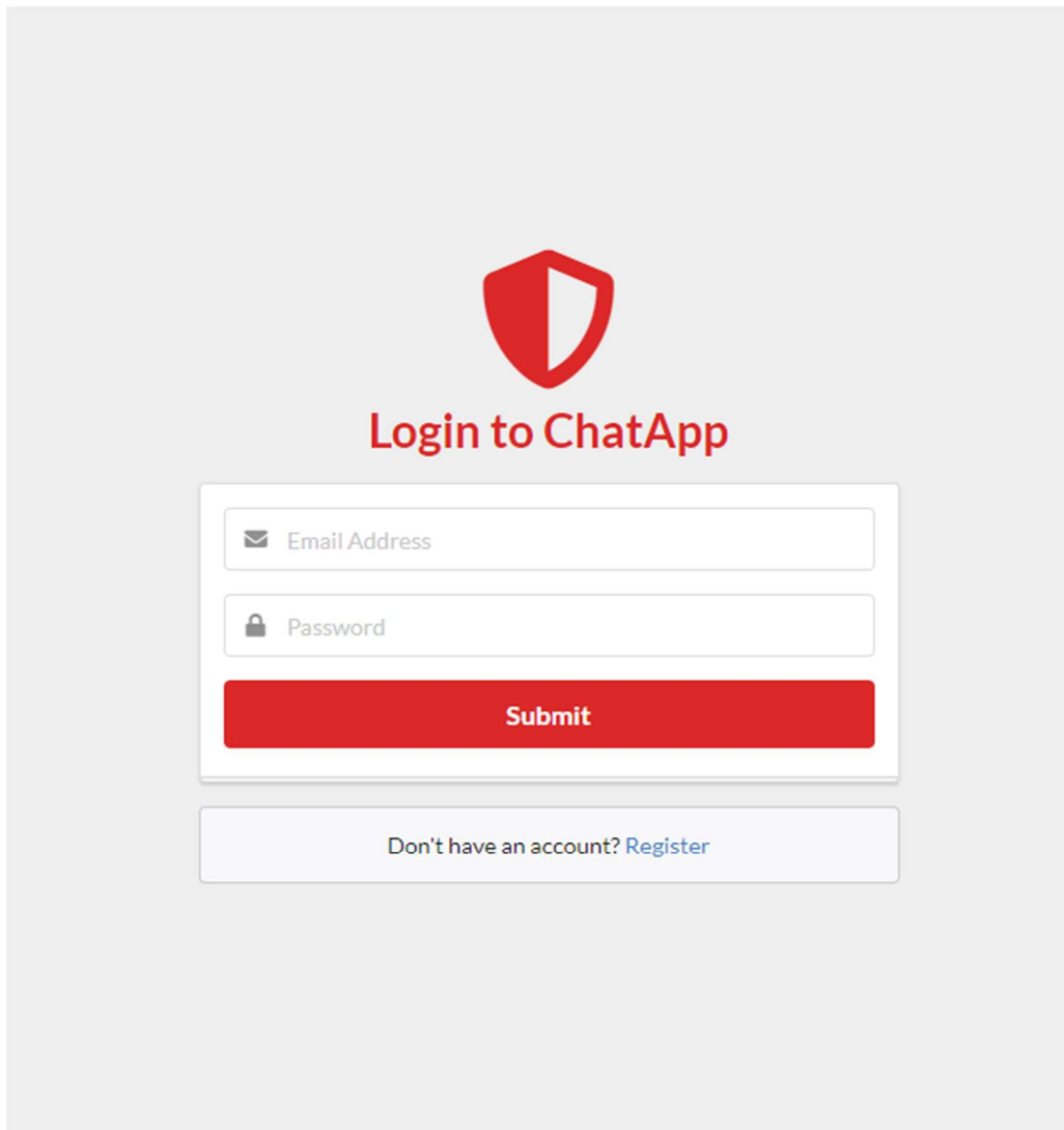
```

Send Message

```
handleSearchMessages = () => {  
  const channelMessages = [...this.state.messages];  
  const regex = new RegExp(this.state.searchTerm, "gi");  
  const searchResults = channelMessages.reduce((acc, message) => {  
    if (  
      (message.content && message.content.match(regex)) ||  
      message.user.name.match(regex)  
    ) {  
      acc.push(message);  
    }  
    return acc;  
  }, []);  
  this.setState({ searchResults });  
  setTimeout(() => this.setState({ searchLoading: false }), 1000);  
};
```

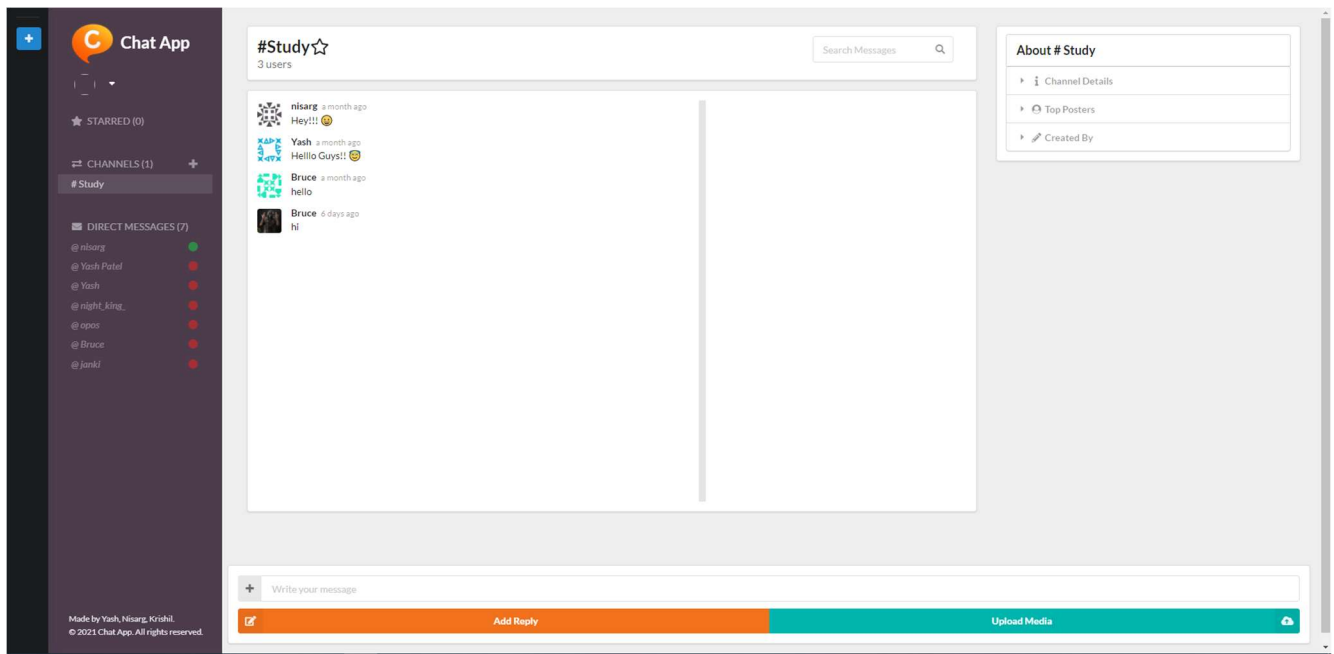
Search Message

5 Screenshots

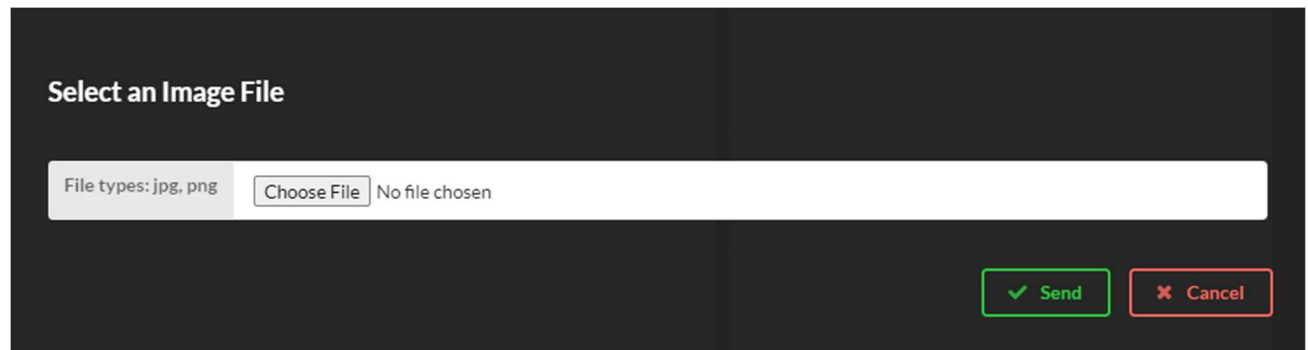


The image shows a login page for 'ChatApp'. At the top center is a red shield icon with a white diagonal line. Below it, the text 'Login to ChatApp' is displayed in red. The login form consists of two input fields: 'Email Address' with an envelope icon and 'Password' with a lock icon. Below these fields is a prominent red 'Submit' button. At the bottom of the form, there is a link that says 'Don't have an account? Register'.

Login-page



Main User Interface



Add Attachments

Add a Channel

Name of Channel

About the Channel

✓ Add

✗ Cancel

Add new channel

The screenshot shows the Visual Studio Code editor with the file `login.js` open. The file contains the following code:

```

1 import React from "react";
2 import firebase from "../../firebase";
3 import {
4   Grid,
5   Form,
6   Segment,
7   Button,
8   Header,
9   Message,
10  Icon
11 } from "semantic-ui-react";
12 import { Link } from "react-router-dom";
13
14 class Login extends React.Component {
15   state = {
16     email: "",
17     password: "",
18     errors: [],
19     loading: false
20   };
21
22   displayErrors = errors =>
23     errors.map((error, i) => <p key={i}>{error.message}</p>);
24
25   handleChange = event => {
26     this.setState({ [event.target.name]: event.target.value });
27   };
28 }

```

The terminal output shows the following messages:

```

Compiled successfully!
You can now view chat-app in the browser.

Local:    http://localhost:3000
On Your Network:  http://192.168.1.3:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

```

TESTING LOCALHOST

6 Conclusion

Functionalities that are successfully implemented in the system are:

- a. User registration containing all the necessary validation on field
- b. Login
- c. User authentication
- d. Logout
- e. User Profile Modifications/Pic etc..
- f. Sending Message
- g. Deliver Attachment
- h. Search a message
- i. Sending E-mojis

7 Limitations and Future Enhancements

We are able to implement the functionality model of the “Real-Time Chat Application”. We aim to make this product ready to be used.

Here if not logout and close browser then other can see our status online, it's a bug.

In future we can make user interface little bit better and fast ,we can add Google login or Facebook login or Github login ,and remove some bugs mention above.

8 Reference / Bibliography

Following links and websites were referred during the development of this project:

<https://stackoverflow.com/>

<https://reactjs.org/>

<https://react.semantic-ui.com/>

<https://firebase.google.com/docs>

<https://github.com/>

<https://youtu.be/-OKrloDzGpU> (Firebase Auth)