1. INTRODUCTION

Every institution nowadays has a lot of events going on all the time so it is vital for every institution to deliver all these updates to each and every student on time. Currently institutes are bounded by conveying these information through notice boards and messages to the students which makes it difficult and tedious for the institute as well the students as they have to regularly visit notice boards for any updates like attendance list, exam dates, upcoming events, etc. There may be a possibility when the conveyed information by fellow students is rather dubious or not reliable. Although Institutes are very much providing the information to the students through websites and text messages but there is no common medium to get all the information.

This is where Medicaps-Messenger (Me-Me) application comes into play. This application focuses on all the affairs in the University. It aims to aid the students with a common platform which will help them to grab all the updated information effortlessly. This will help students to focus more on their fruitful work instead of giving their valuable time for gathering such information. It also provides a means to allow a student to communicate to other users for any guidance they require.

Use of this application will make the life much easier for all the students and faculties as it would save the time and the efforts and make all relevant information accessible in an effective and sophisticated manner.

1.1 Objective

The main objective of providing Me-Me application to the institute's students and faculties is to give them a common platform for the purpose of getting all the relevant information all at one place with convenience and focus more on their studies rather than collecting information from scattered sources which becomes a tedious task.

The major driving force behind developing this application is to make the information available to the students at all times whether they are physically present at the institute or not. This is very crucial when students get their preparation leave or not able to attend the college for some reason.

With the help of this application all the students will get the opportunity to know their college better by interacting with their seniors and participating in each and every college event. The application brings a common platform for all the students to interact so if any of the students find any difficulty on any college matter he/she can get help instantaneously. It also features a real time chat system which will help students to communicate better. Application will give regular notifications to the student so that they do not have to constantly visit all the notice boards or even the application itself.

It aims to solve all this tedious task in an uncluttered manner and making the information accessibility much easier than currently is.

1.2 Problem Domain

In current scenario institutes find it difficult to provide students with the updated information and somewhat lacks in proper communication of various institute related things to all the students with the help of a common platform. Websites are traditionally used for serving this purpose however the current trend are mobile platforms and thus this problem needs to be solved by providing a common platform that is accessible to each and every one.

1.3 Solution Domain

Institutes have their own websites which can only provide news to the students and faculties only and they need to regularly visit those in order to get updated. Me-Me is an Android application that shows news feed of college as well as real time

chatting, it has a user friendly interface and is loaded with extra features that includes easy puzzle games which will help students to kill boredom and stress. It is a reliable and faster way of spreading important information in the college premises. Me-Me is developed on android, as android is the most used smartphone OS and it will allow most of the students to get benefited from the application.

1.4 System Domain

Tools, technology, environment, platform and hardware and software specifications needed for implementation and deployment of the proposed idea are as follows: -

Table 2.1

IDE (Integrated Development Environment)	Android Studio v2.3.3
APIs (Application Programming Interface)	Firebase
Languages used	Java, XML
Hardware used	Windows 10 system
Software version Required	Android v4.0 and above
Cloud	Firebase

2. SYSTEM REQUIREMENT ANALYSIS

System Requirements Analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project

2.1 Information Gathering

Information Gathering is the act of collecting information from various sources through various means.

2.1.1 Functional Requirements

Functional Requirement defines a function of a system or its component. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define *what* a system is supposed to accomplish.

ID – Defines the Tag of requirement

Title – Defines Title/Topic of requirement

Desc – Describes the requirement

Dependency – Describes the dependency of one requirement on another

ID: FR1

Title: Sign up / Registration

Desc: All the students and faculties will be able to register on the application to view the noticeboard and access all the features of application.

Dependency: None

ID: FR2

Title: Login Activity

Desc: Only the registered users will be able to login using their credentials, which will be verified with the credentials present on cloud.

Dependency: FR1

ID: FR3

Title: Edit notice board

Desc: Admin can add, update or delete notices on the notice board in the application to make it available for the users of the application.

Dependency: None

ID: FR4

Title: View notice

Desc: Users will be able to view the notices on the application and will remain till the event is completed or is passed.

Dependency: FR3

ID: FR5

Title: Send Message

Desc: Users will be able to send each other messages through the Real-time messaging service.

Dependency: FR2

2.1.2 Non – Functional Requirements

Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. They are contrasted with functional requirements that define specific behaviour or functions.

ID: NFR1

Title: Maintenance and Manageability

Desc: Data in the form of news feeds will be updated and displayed regularly by the admins of the application.

Dependency: None

ID: NFR2

Title: Authentication

Desc: Login credentials for students and faculties will be used to provide easy and secure access.

Dependency: NFR4

ID: NFR3

Title: Security

Desc: Real-time messaging feature will be encrypted hence only sender and receiver will have the chat log.

Dependency: NFR2

ID: NFR4

Title: Availability

Desc: It will be available to every students and faculties in the college who has a smartphone running Android and application will be available to use 24×7 as cloud database is used.

Dependency: None

ID: NFR5

Title: Usability

Desc: Real-time messaging and notice board will be usable to all the users.

Dependency: NFR3

ID: NFR6

Title: Performance

Desc: Application will be very fast as it will include only student and faculty related features and also it consumes very less memory and storage space. Firebase cloud database is used which do not fails under any circumstances.

Dependency: None

2.2 System Feasibility

Feasibility study is an assessment of the practicality of a proposed project or system.

2.2.1 Technical Feasibility

The application requires Android OS 4.0 and above, which is most commonly available in almost all devices hence there is no need for the users to update their existing Operating System. The application is developed using Android Studio IDE, which is also an open source software so working with the software is hassle-free. It will be very compact in size so it will be easily installed in any smartphone, as well as cloud database will be used to store all the information hence it will store most of the data in the cloud, making the application consume less storage space in smartphone.

2.2.2 Economic Feasibility

The application will be free to download and will have no in-application purchases, hence the user of the application does not need to spend any money, making the application economically feasible.

2.2.3 Operational Feasibility

Application will be operational only if there is internet connectivity as most of the data is in the firebase cloud database which will be fetched and accessible by the user. The application will not crash in normal conditions and data security is provided by firebase database. It will be easily operable as it will be fast, due to less memory it will require to run on any smartphone. The user interface will be user-friendly hence no tutorial or such will be required to operate the application.

3. SYSTEM DESIGN

3.1 Use Case Diagram

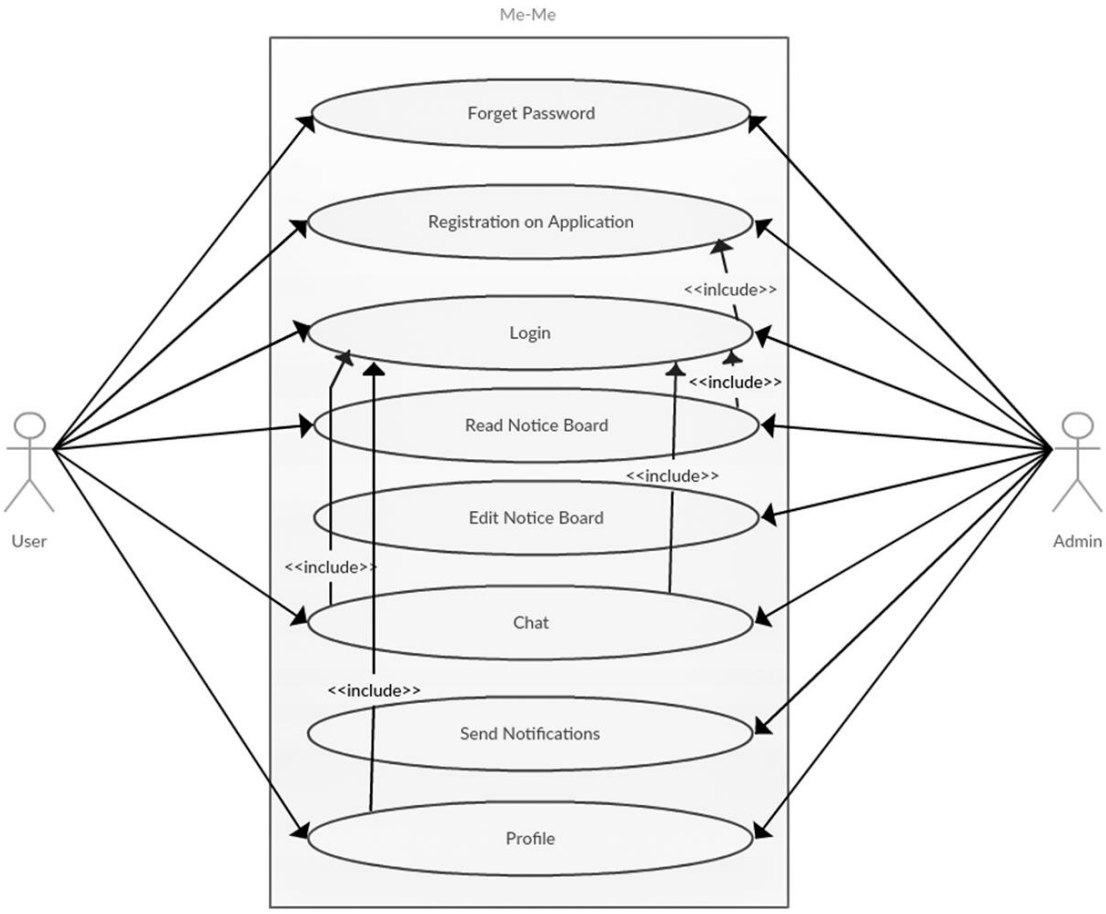


Figure 3.1 Use Case Diagram

Use case diagram shows the functionality that the system provides to its user and what other functionalities are performed in collaboration with the user. The user can register and login on application to access and read notice board, chat, read notifications. Admin can also perform these operation as well as edit the notice board by using a admin password exclusively known to admins only. Once a user is logged in, whenever he/she opens the application in future they will be automatically redirected to the notice board by default. Admin can upload notices and delete notices through the app itself. Notifications can be sent through web console whenever new notices are posted.

3.2 Class Diagram

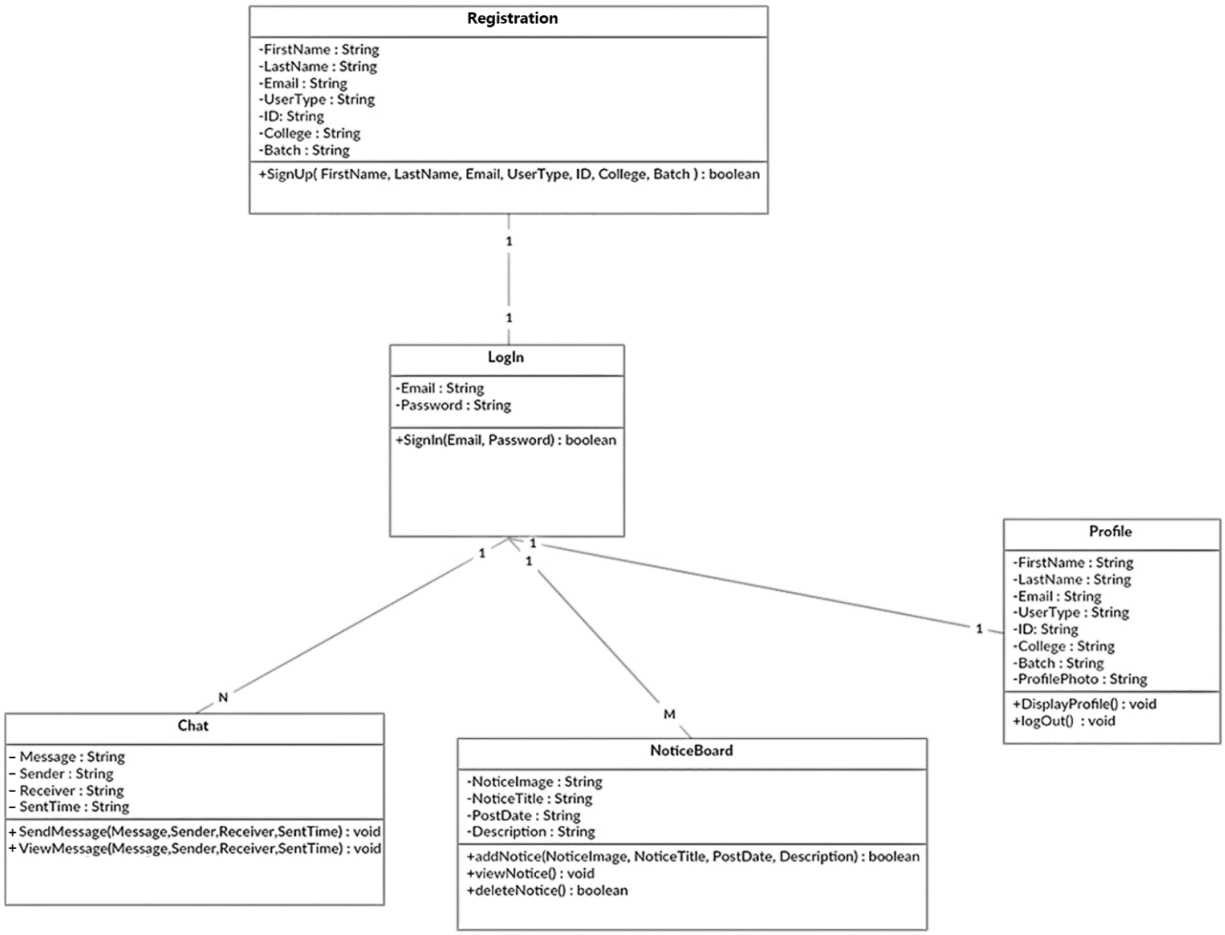


Figure 3.2 Class Diagram

Class diagram describes the structure of a system by showing the system's classes, their attributes, methods and the relationship among objects. In this class diagram, we have 5 classes, the Registration class has data members to store the data about the user, using these data, the member function Signup is called through which the user is registered. Login class has data members for storing credentials and using these the user is logged in upon calling of the function Signin.The Chat class stores data about the message received or sent, has 2-member functions, one for sending message and the other for displaying the message to both the parties. Noticeboard class stores data about the notice that is posted, having 3 functions, one for adding notices, others for viewing and deleting it. Profile class is used for displaying the details about the user, its data members hold the user information and provide 2 functionalities, namely, displaying of profile and logging out of the application.

3.3 E–R Diagram

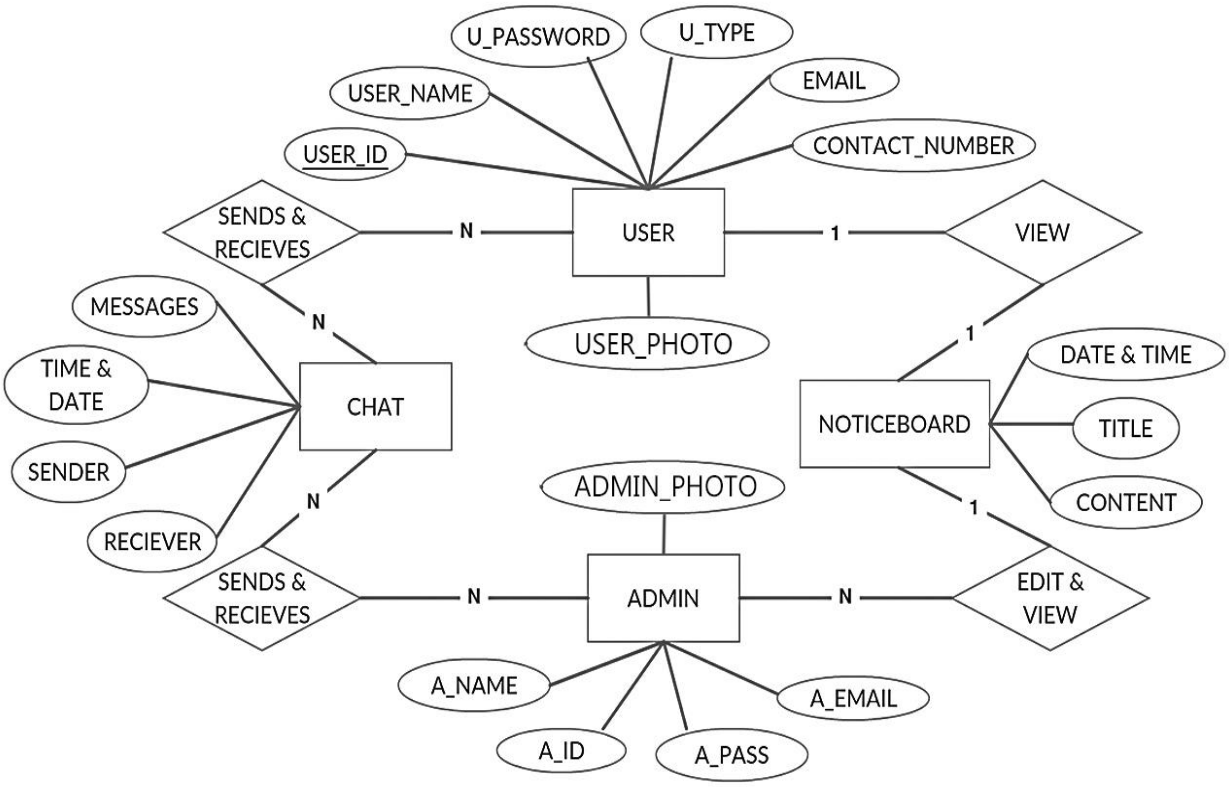


Figure 3.3 E – R Diagram

E - R diagram is defined as the relationship between entities.

ER Diagram shows the attributes associated with all the entities involved in the project. Numbers and letters between the lines represents cardinality.

Each USER will have a unique user id, email through which he/she may login. All the user data such as USER_ID, USER_NAME, U_TYPE, etc. will be saved in the database.

Each ADMIN will have the same attributes as a USER but one more attribute is there which is A_PASS which he/she will use to edit noticeboard.

Each Notice in the Noticeboard will have date & time, title, content.

All the NOTICEBOARD and CHAT data are visible to respective Users.

3.4 Data Flow Diagram

Level 0:

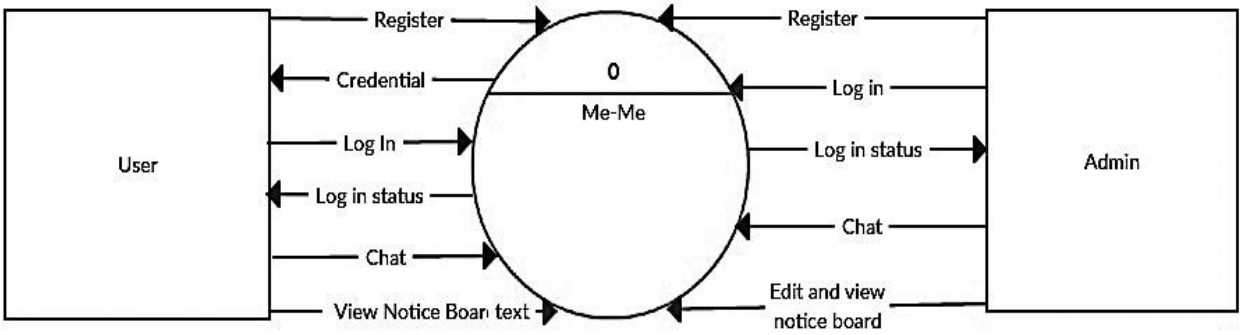


Figure 3.4 DFD Level 0

DFD level 0 diagram is showing the data flow when the user first registers on the application, then the user data is verified when the user login into the application. After the successful login, user can view noticeboard or chat with any other user.

Level 1:

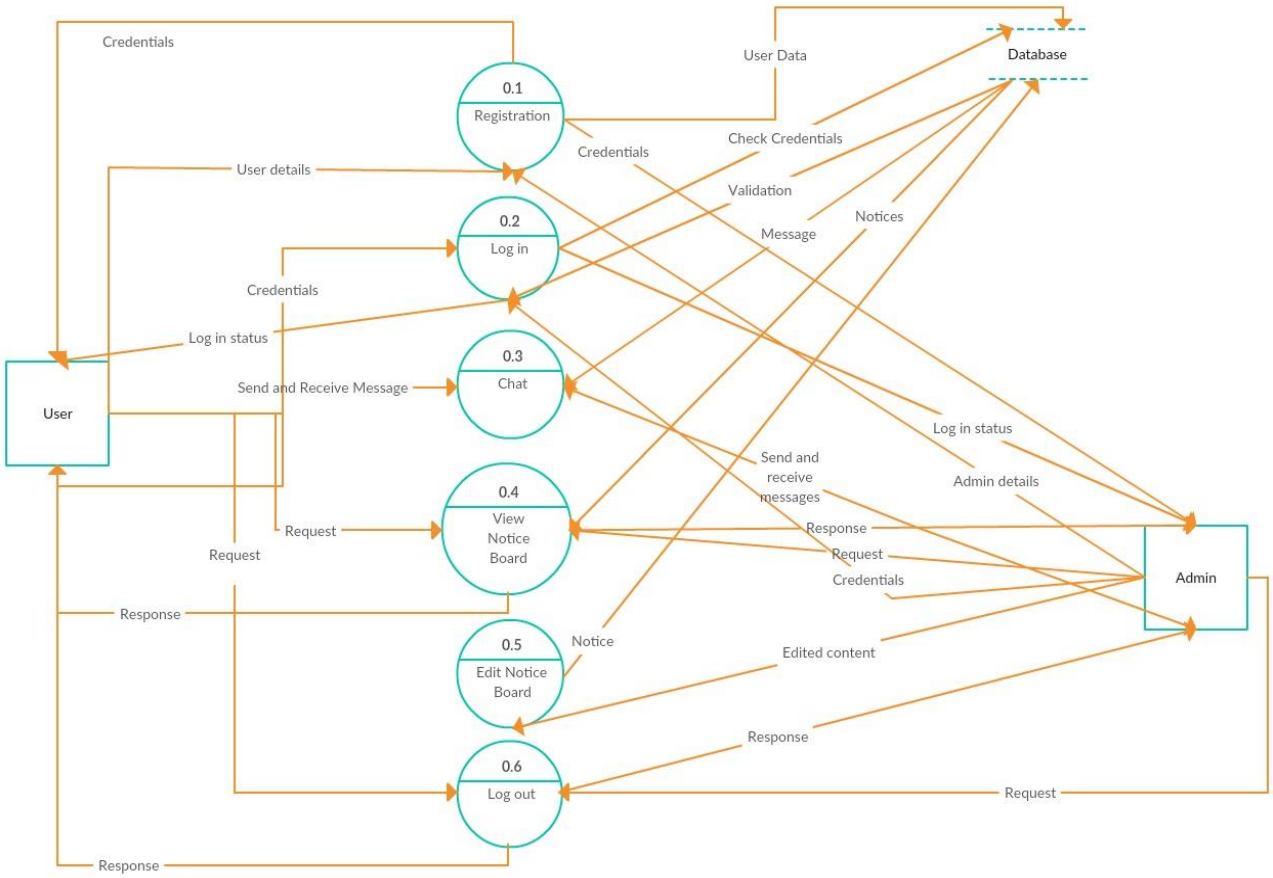


Figure 3.5 DFD Level 1

DFD level 1 shows the flow of data. The rectangles are used here to represent the two types of users, i.e. Admin and users of the application.

The circle represents the application where the admin can register, login, chat, view or edit the noticeboard and the user can Register, login, chat and view the noticeboard.

Initially the user will be taken to login screen where he/she can register, if not registered already. After successful registration of the user, the user needs to verify their email address through a verification mail sent to their respective mail id. After that user is able to login using their registered credentials and can access notices directly. To send messages to other users they can perform it through the chat section.

3.5 Sequence Diagram

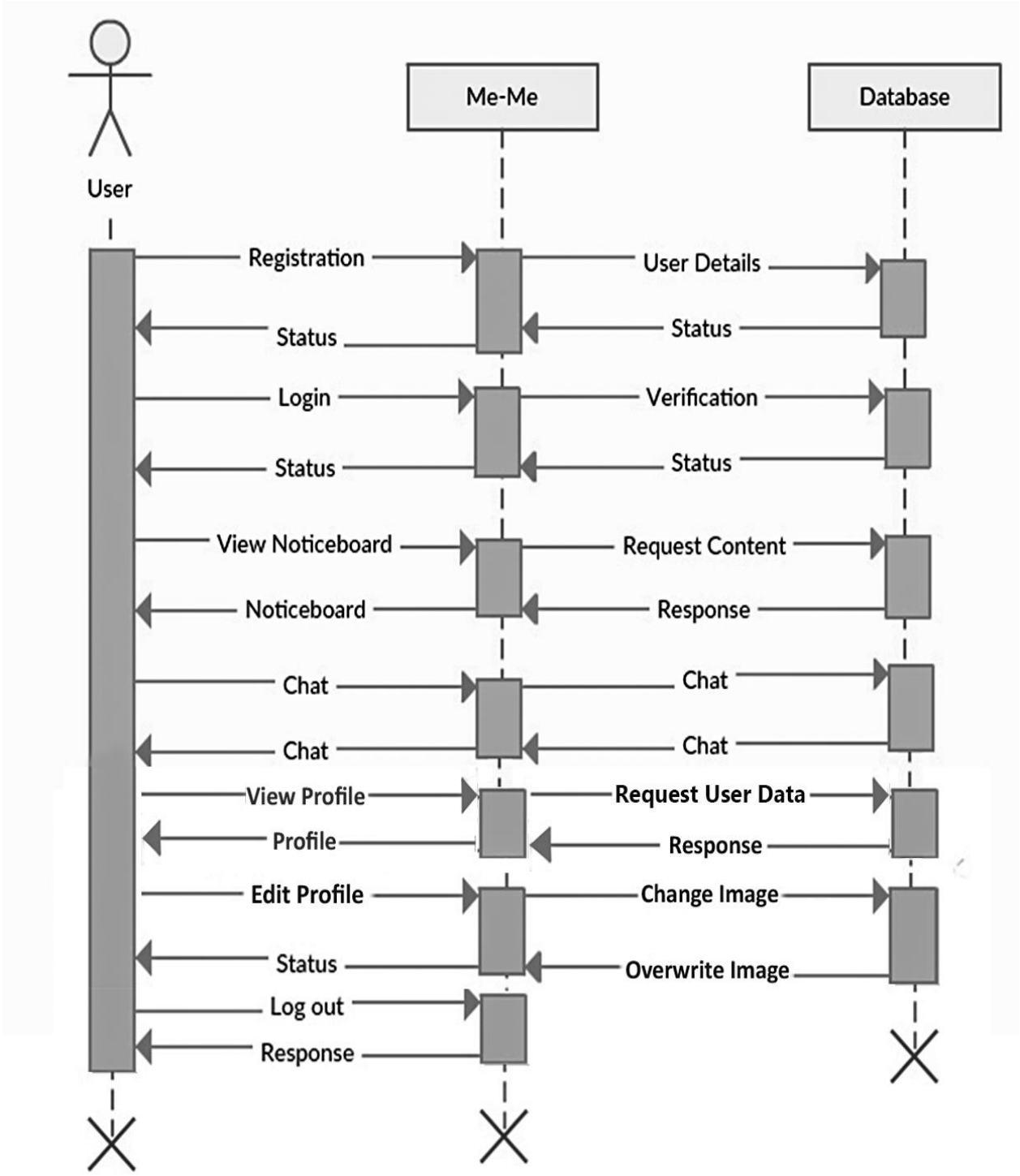


Figure 3.6 User Sequence Diagram

Sequence diagram shows the sequence in which objects of the classes interact with each other. Sequence diagrams are a popular dynamic modelling solution. Dynamic modelling focuses on the interactions occurring within the system. Sequence diagrams specifically focus on the "lifelines" of an object and how they communicate with other objects to perform a function before the lifeline ends. User first registers through the registration page then they will be redirected to the login page to login in the application. After successful login they will be presented with the Noticeboard Section of the application through which they can access all notices. To utilize chat functionality, they can do so by swiping to the chat section of the application. User can upload their profile photo if they want to through the overflow menu.

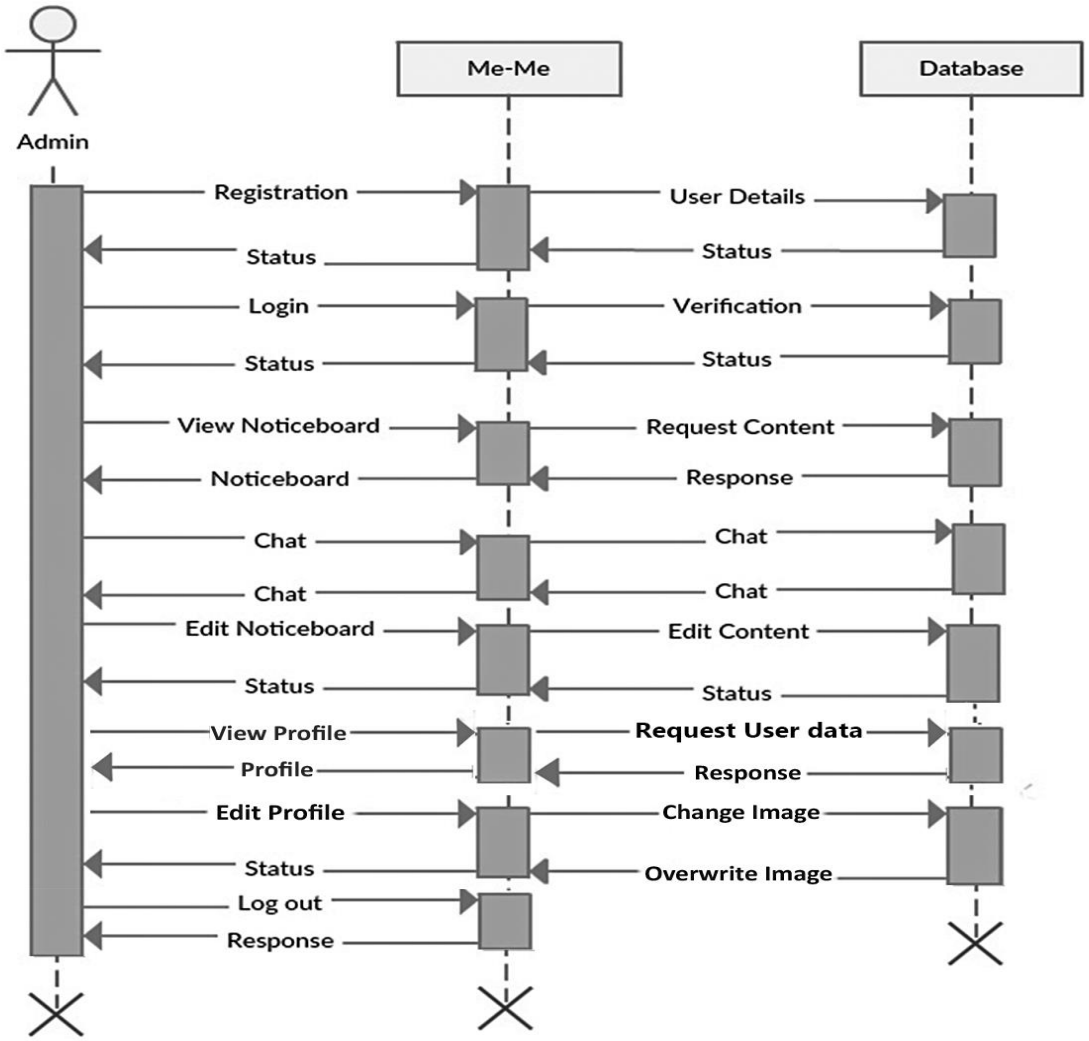


Figure 3.7 Admin Sequence Diagram

Admin follows the same sequence as the user to register and login in the application and access features of the application but admin will be having special privileges to edit noticeboard through the application itself and will be able to delete the post provided that the respective notice was uploaded by he/her. Admin will be able to send notifications about the updates in the noticeboard through a web console.

3.6 State Chart Diagram

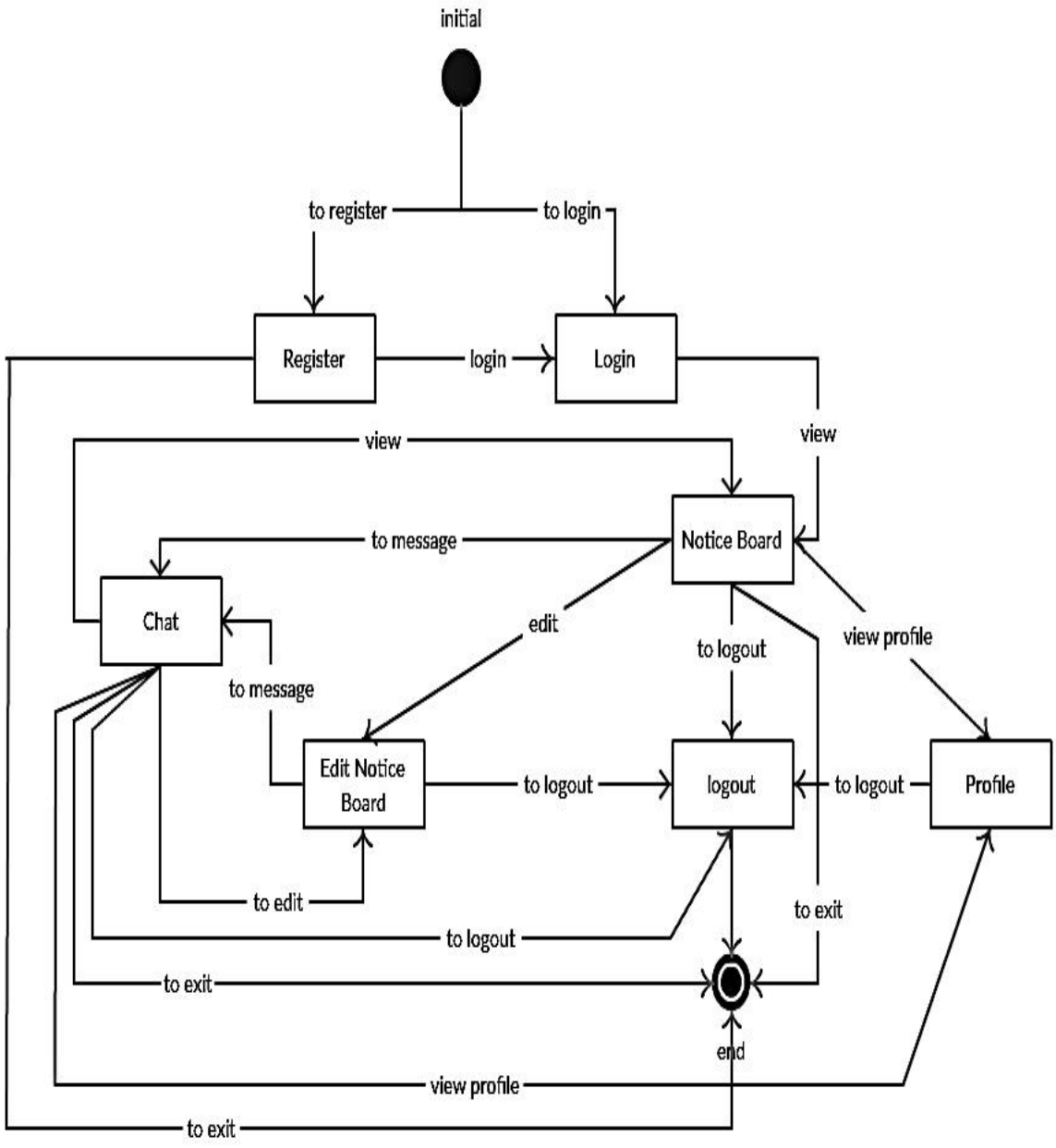


Figure 3.8 Admin State Chart Diagram

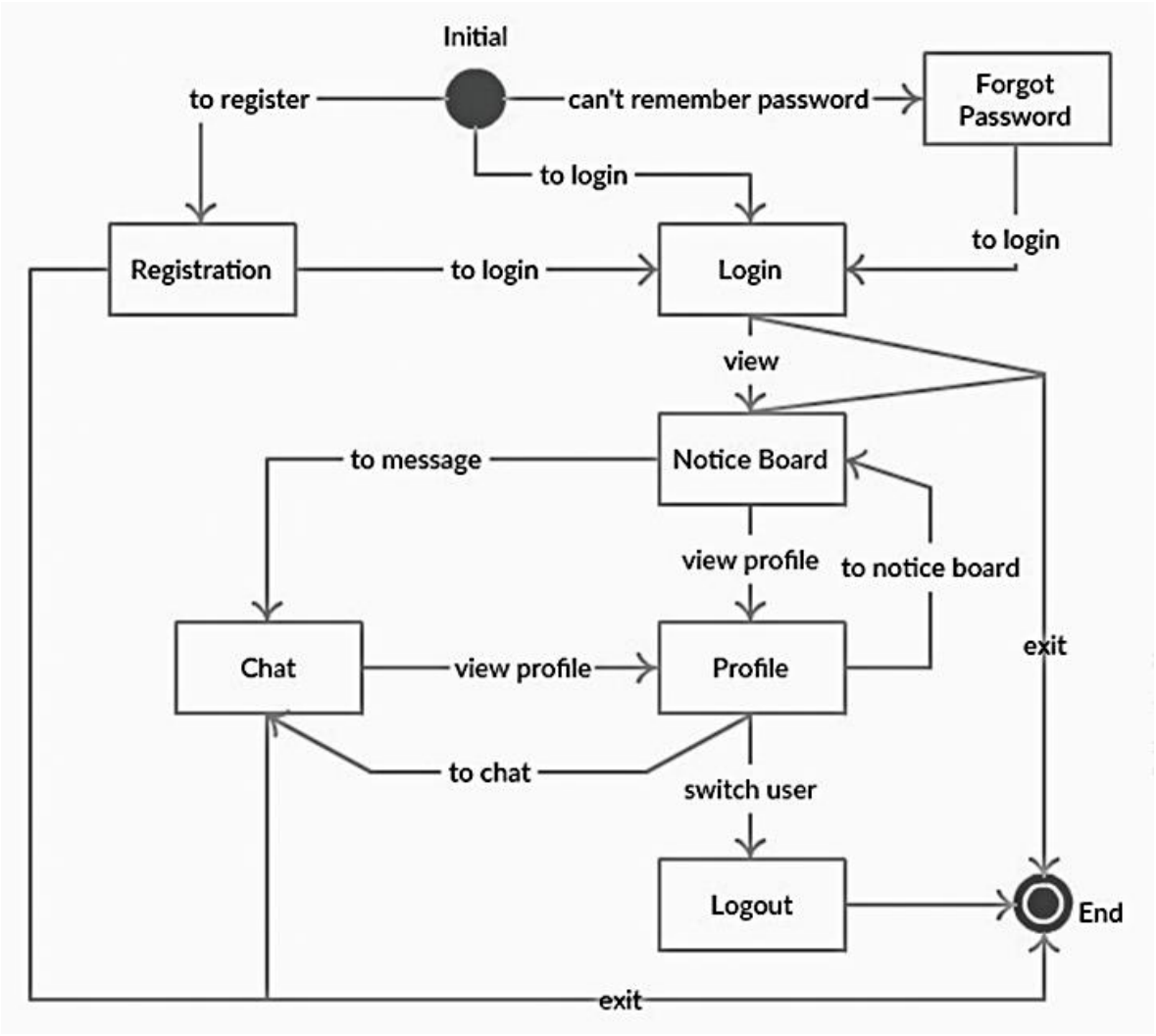
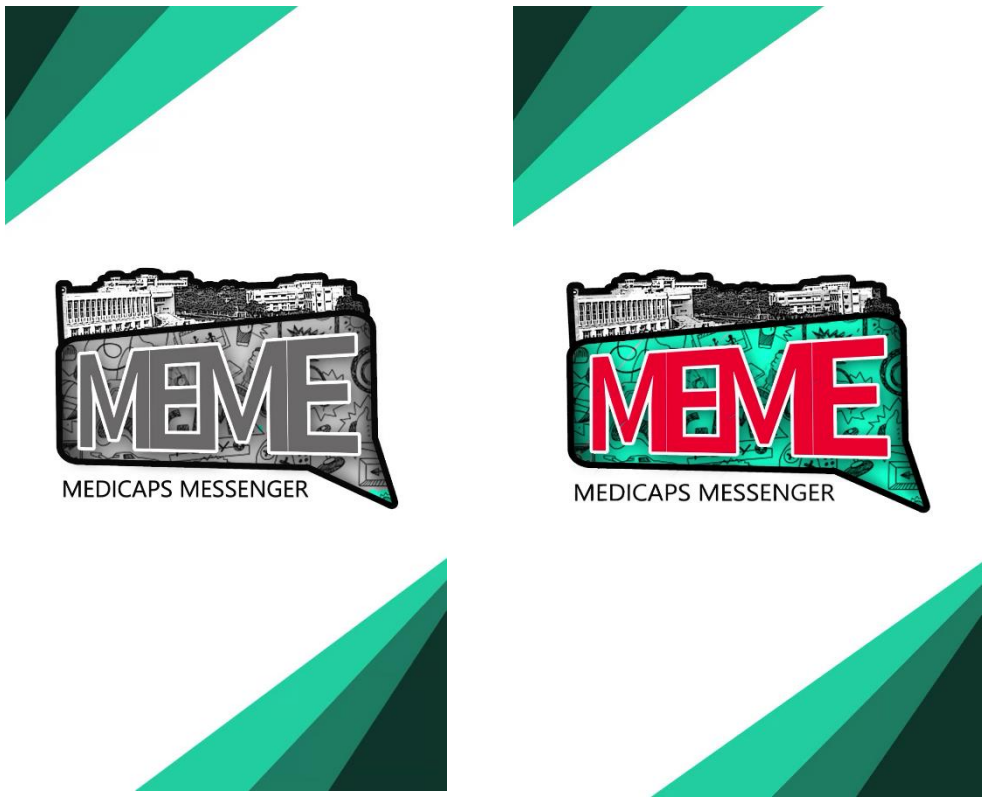


Figure 3.9 User State Chart Diagram

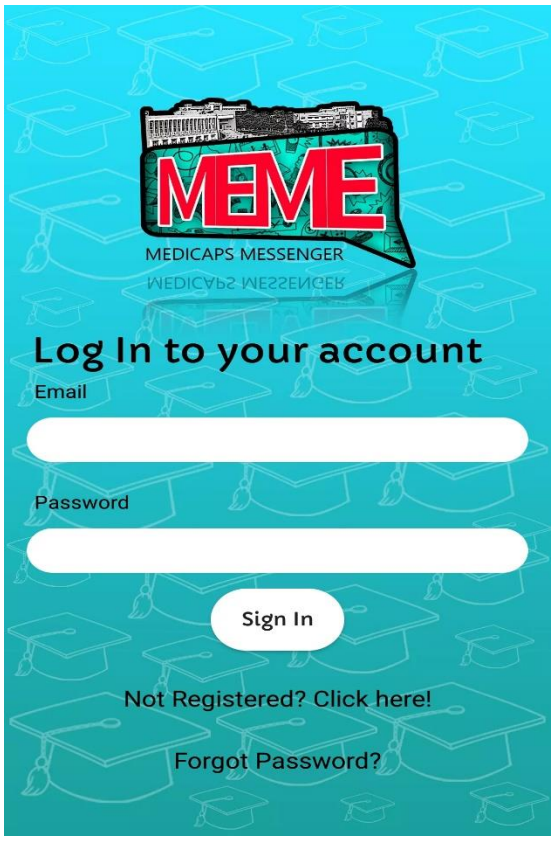
State chart diagram shows the state of the system at any instant. The State Chart Diagram represents the state of the application, it consists of initial state through which a user may either register or login, Noticeboard is either preceded by messaging state or Logout. For chatting with other users of the application users can do so at anytime using the chats section and can update their profile at anytime they want to (optional). Admin can upload or delete notices by using the application's Edit option in the overflow menu. Notifications for new notices will be sent to all the application's users through a web console and users get notifications every time they receive new messages.

4. IMPLEMENTATION

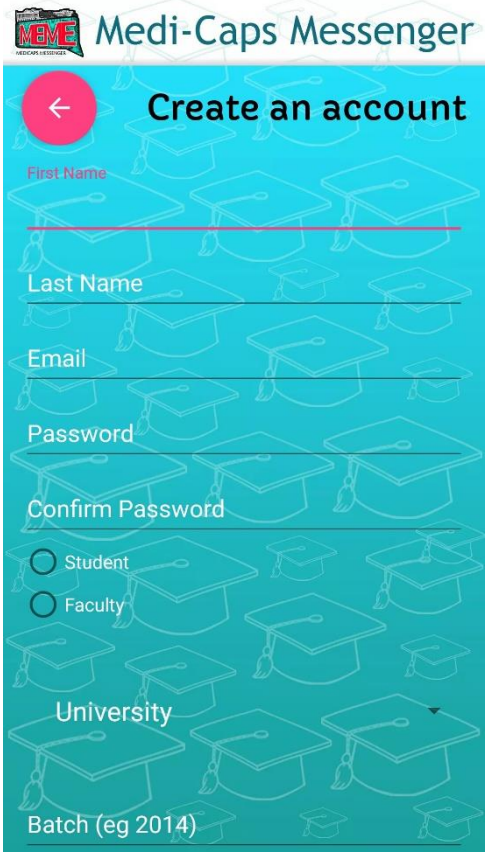
4.1 Implementation of Modules

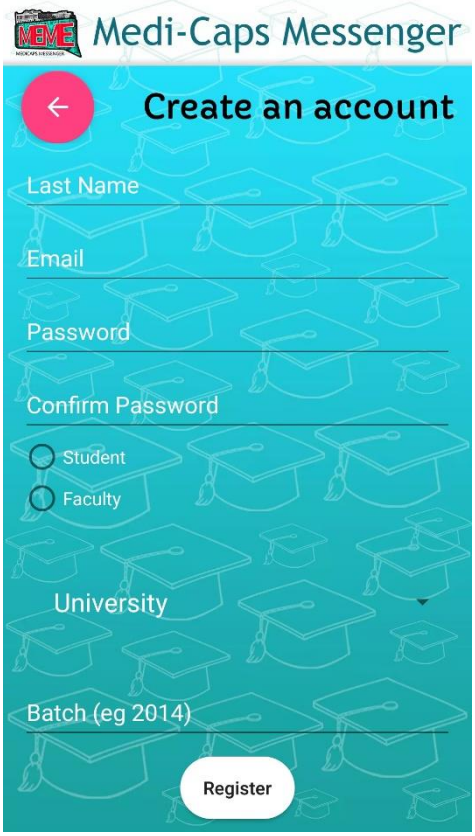


As soon as application is launched user is treated with a short 2 sec video of the app

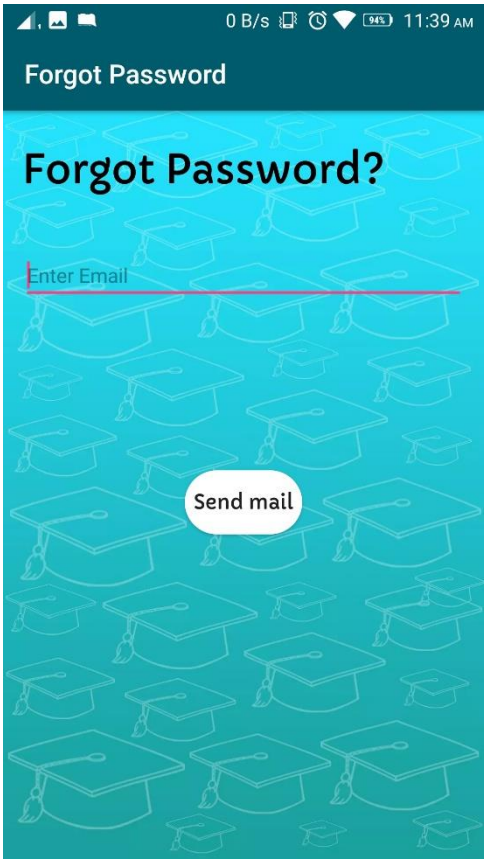


After the end of video user will be presented with the login activity where they can login using their email and password.

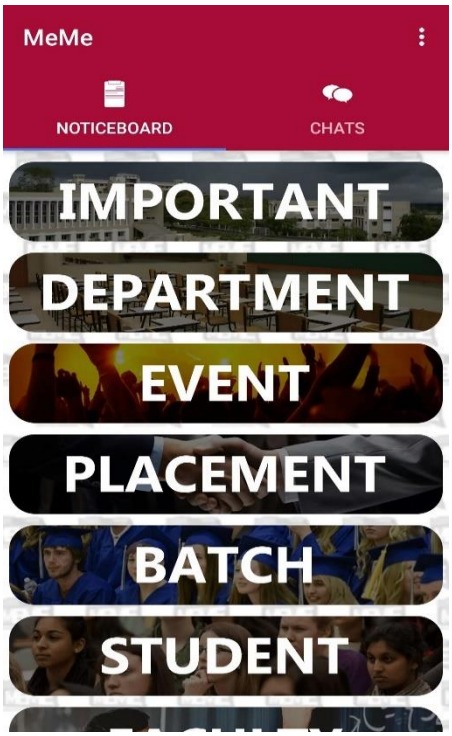


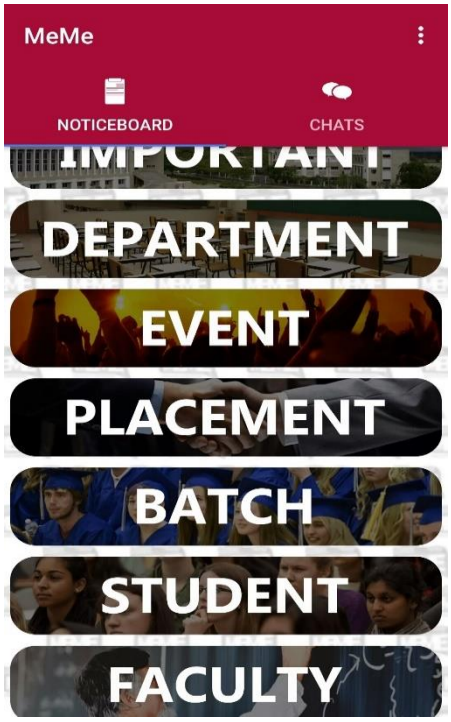


If user is not already registered they can register using the registration activity. After Successfully registered, user will be redirected to login activity.



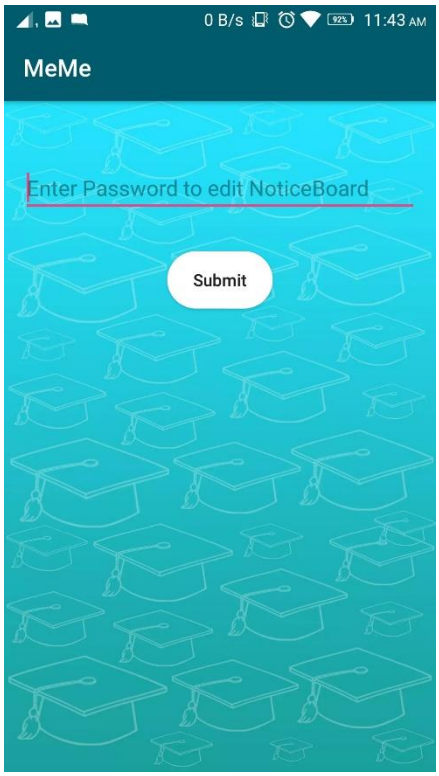
In case if the user forgets their password, they can send a forgot password mail to their email id from the login screen



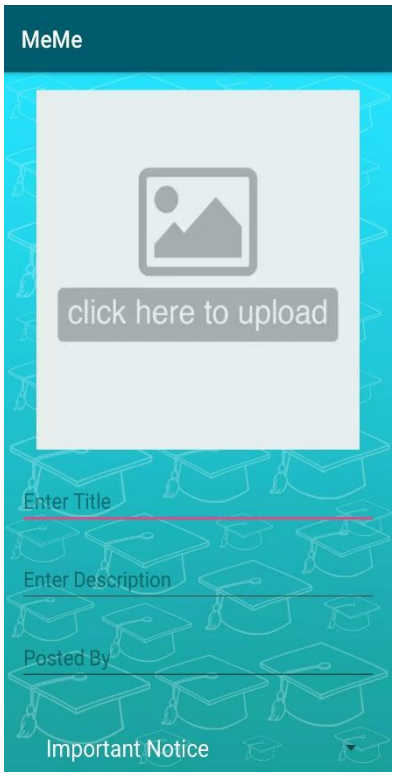


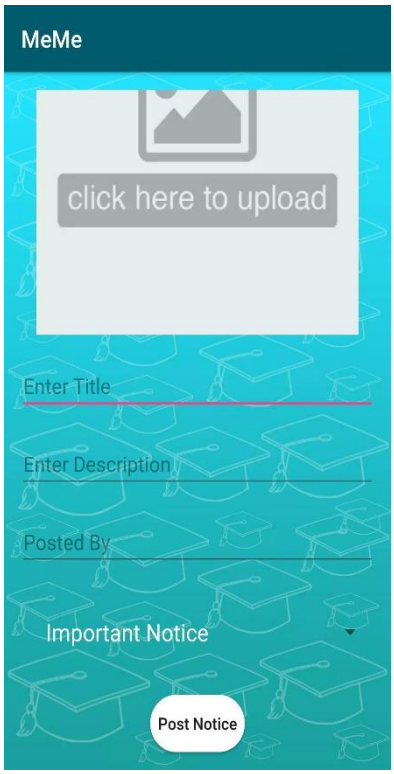
After successful login, users will be redirected to the noticeboard tab by default where they can view all the notices. Until the user logout from the app, they will be directly redirected to noticeboard every time they start the application

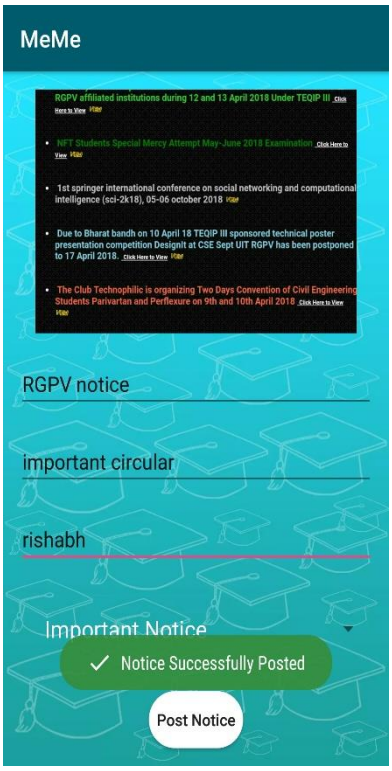
The logged in user can see their profile in the above-mentioned manner and can upload a profile photo. They can also send reset password mail and logout from their profile activity.



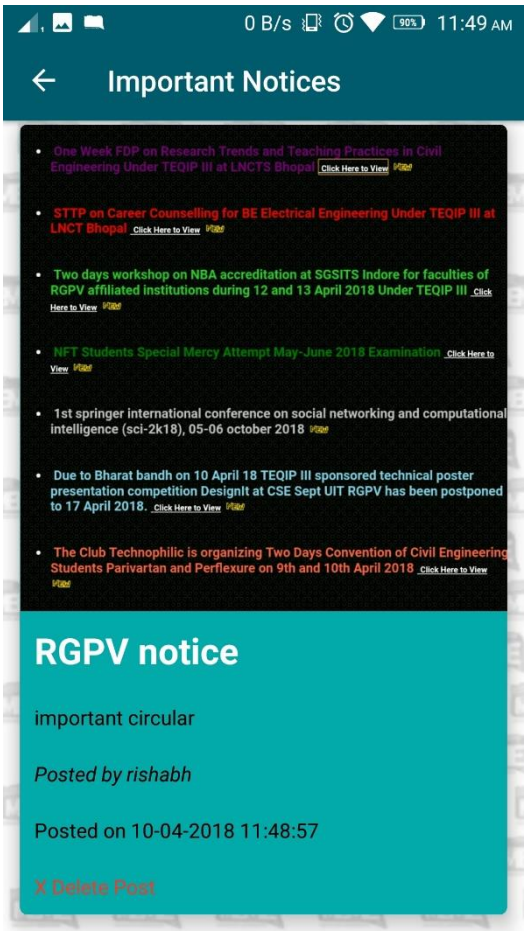
By tapping the Edit menu item, they will be presented with an activity which asks the admin password to upload new notices on the noticeboard

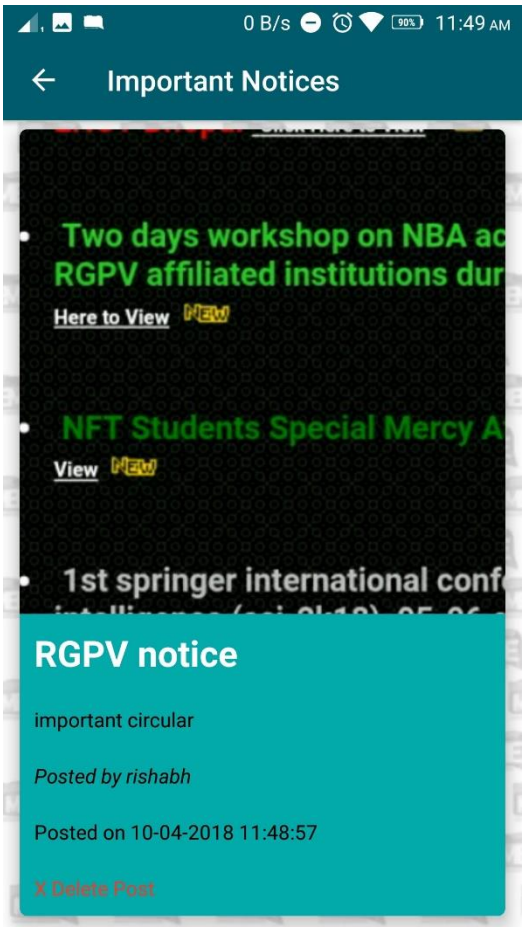




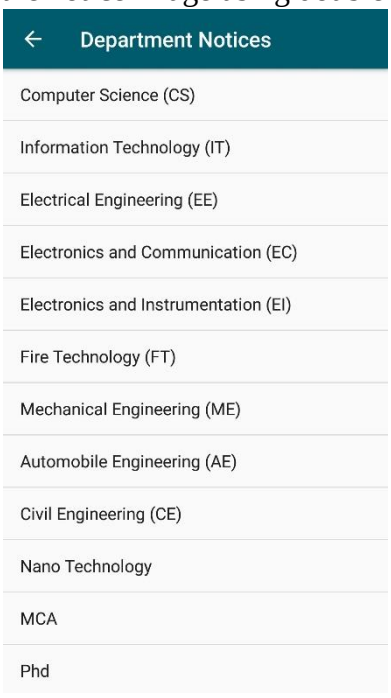


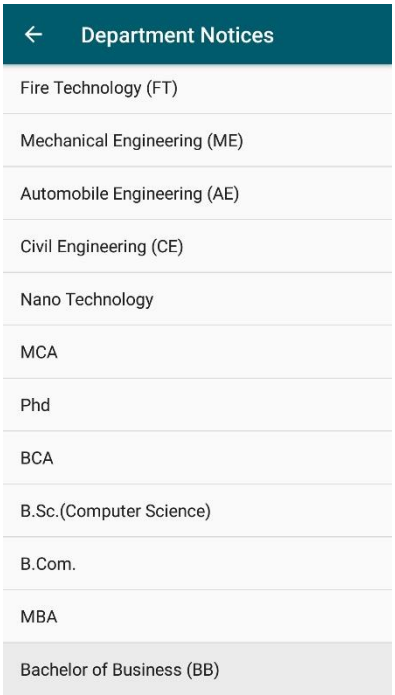
From this activity admin can upload notice image with notice details directly to the noticeboard in real-time. An example is shown here

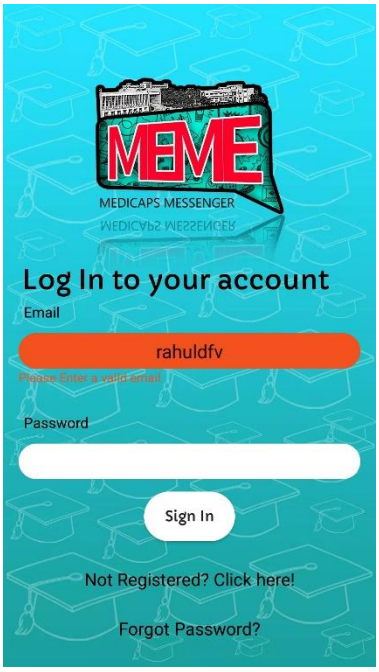


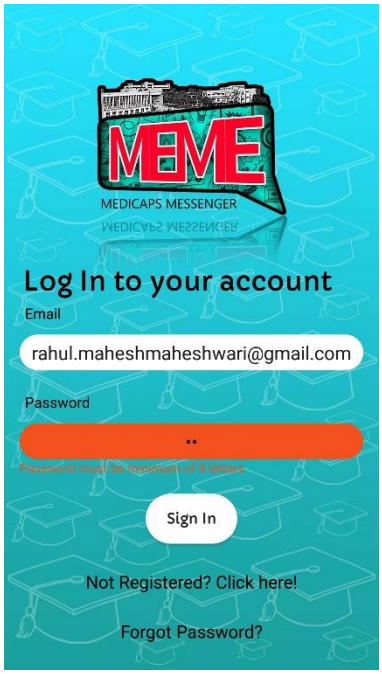


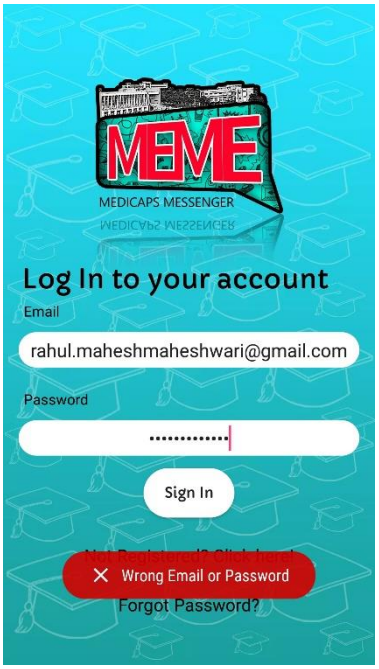
This is how notice will be shown on the respective notice board. Users can also zoom the notice image using double tap gesture.





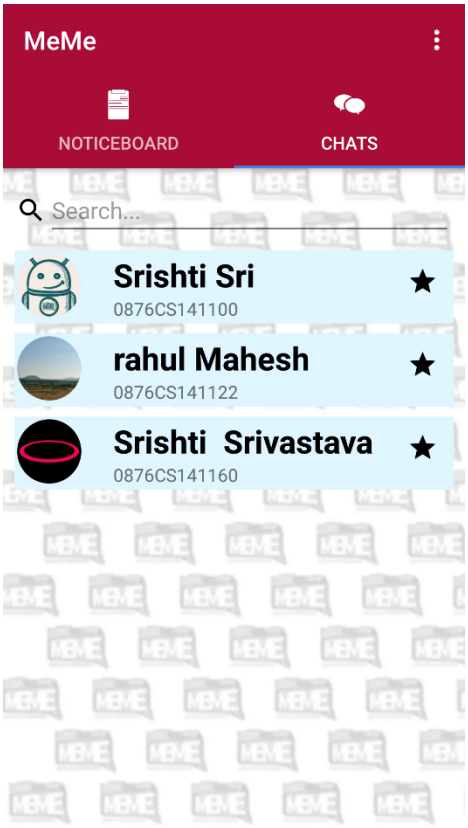




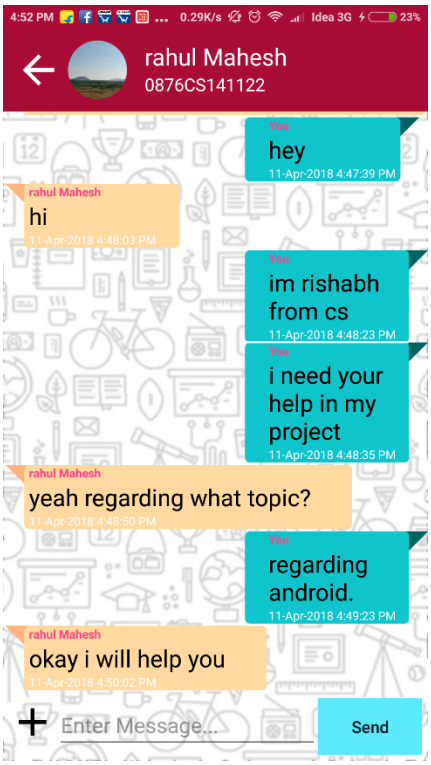


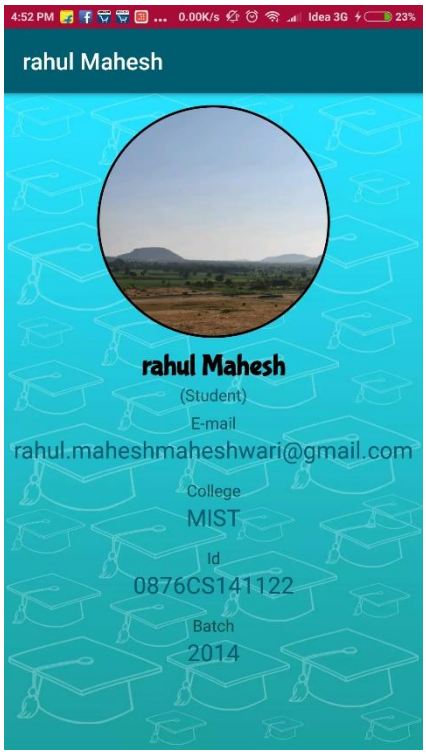
These are all the departments in the department tab shown on noticeboard tab.

Validations regarding Login activity

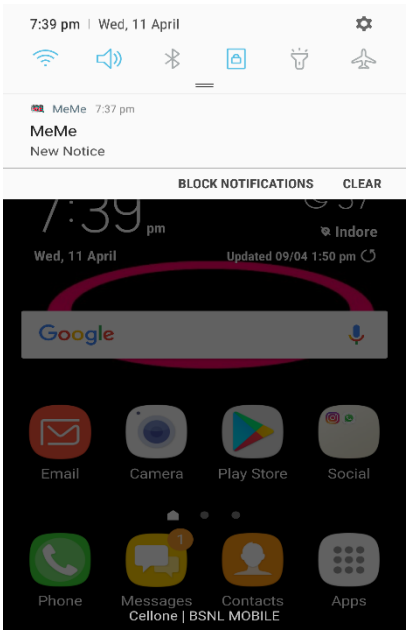


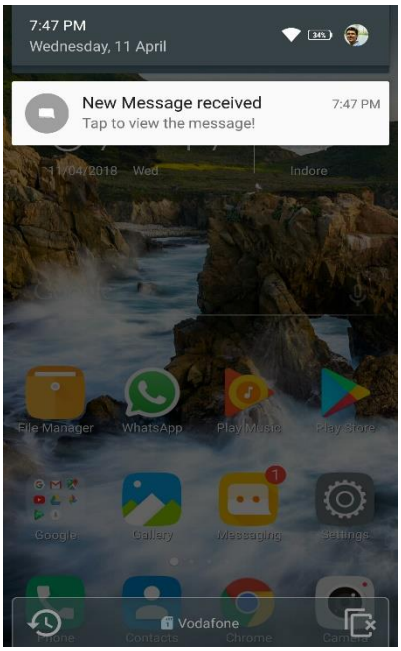
By swiping left users can see the chat activity where all the registered users will appear. Search bar is provided to search for a specific user to which user wants to chat.

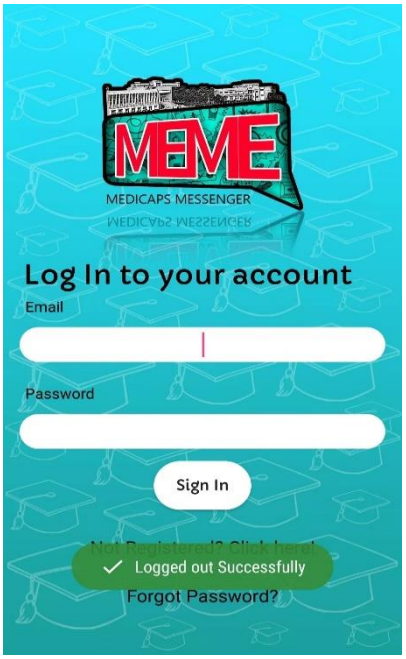


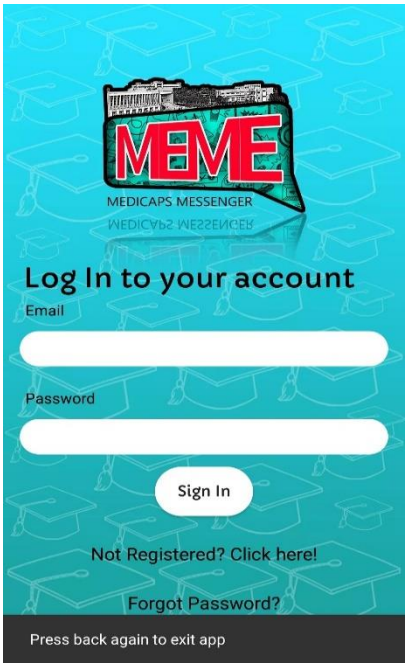


After tapping on the name of users from the list appeared in chat activity, the user will be redirected to a one to one chat window where they can communicate. To view the profile of the user to whom a user is chatting they can tap on the top bar showing the name of the user receiving the message which will open the profile of receiver of messages. Notifications will be displayed in the following manner.



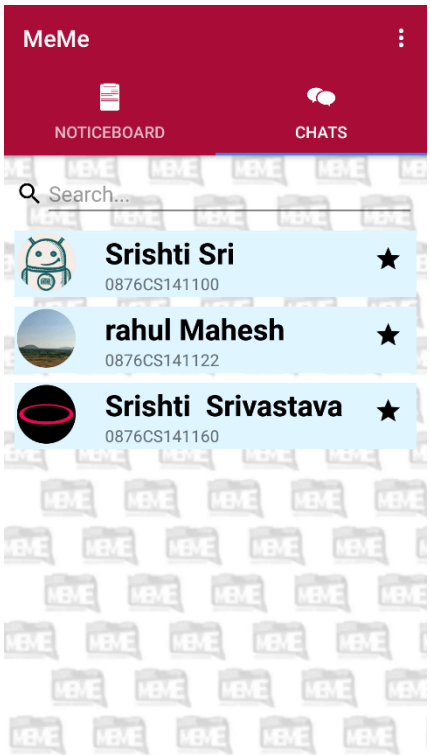






User can logout from the application from the profile or from the overflow menu (three dot menu). To exit the application user needs to press the back button twice. (to prevent accidental exits)

4.2 Results



As expected the functionalities are implemented and working perfectly.

5.Testing

5.1 Test Cases

S.No.	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass / Fail
1	Don't enter anything in the registration	Leave all the fields of registration empty and press register	Fields=""	It should show error message saying "First name required and must contain only alphabets"	As expected	Pass
2	Enter password less than 8 characters	Enter passcode having length lesser than 8 and press register	Password ="1234"	It should show error message saying "Password must be minimum of 8 letters"	As expected	Pass
3	Enter valid user details in registration	Enter all the valid user details and click register button	User details	Display message "Registration done" followed by message "Check your mail for verification"	As expected	Pass
4	Don't enter anything in email in login	Leave the email field empty, enter the password and press Sign in button	Email=""	Display message "Enter valid email"	As expected	Pass
5	Don't enter anything in password in login	Leave the password field empty, enter the password and press Sign in button	Password= ""	Display message "Password must be minimum * letters"	As expected	Pass
6	Enter valid credentials	Enter the registered email and password and press Sign In button	Email=abc@example.com Password= "12345678"	Redirect to notice board	As expected	Pass
7	Click on delete post when you did not post the	Click on delete post on the notice that has not been		Displays message "you are not authorized to	As expected	Pass

	notice	posted by you		delete the post”		
8	Post notice without any content	Click on post notice without entering any data	Fields=””	Displays message “Please select an image first”	As expected	Pass