Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B. Tech. CE Semester – VI

Subject: System Design Programming

Project Title:

Location Based Task Reminder Application

By:

Darshan Gohel-CE037 (17CEUBG002)

Neel Dani-CE021 (17CEUON039)

Rushabh Dholakia -CE029 (17CEUOS088)

Guided By:

Prof. Sheetal S.Shah

Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

CERTIFICATE

This is to certify that System Design Programming Project entitled "**Location Based Task Reminder Application**" is the bonafied report of work carried out by

1) Darshan Gohel (17CEUBG002)

2) Neel Dani (17CEUON039)

3)Rushabh Dholakia(17CEUOS088)

Of Department of Computer Engineering, Semester VI, academic year 2019-2020, under our supervision and guidance.

Guide	**HOD**
Prof. Sheetal S. Shah Assistant Professor of Department of Computer Engineering, Dharmsinh Desai University, Nadiad.	**Dr. C. K. Bhensdadia** Head of Department of Computer Engineering, Dharmsinh Desai University, Nadiad.

Table of Contents

Abstract

Our project is on location based task remaindering application. As we all know that in our day to day life we have many task to do. So instead of remembering each and every task why not to develop an app which does it for us. Our system will locate each and every task user enters and integrate the location with Google maps which makes the place easy to find even if we don’t know the route. Also if the user forgets the responsibility for remaindering the task rest on our system.

Introduction

2.1. Brief Introduction

Location Based Task Reminder is a react-native (android) application where a user can enter their daily tasks as a to-do list and can perform or complete them within travelling the minimum distance. Our app can calculate the distance between the user's current position and the place entered by user and accordingly finds the best shortest path based on the minimum distance. Now even if the person is new to a place, with help of Google maps integrated in our app, user finds it easy to locate a particular place.

2.2 Tools/Technologies Used

Technologies: React-Native
JavaScript
CSS 3
Bootstrap 4

Platform: Expo App
Firebase

Tool: Visual Studio Code

3. Software Requirement Specifications

3.1 Types of Users:-

1. End user (Customer)

3.2 System Functional Requirements

R.1:End-User:

R.1.1 : Register user
Description: user needs to register by providing details.
Input : user details.
Output : Success message.

R.1.2 : Manage Task
Description: user can add, view, start, delete his/her task.

R.1.2.1 : Add Task
Description: user can add the task.
Input : user select add-task option
Output : appropriate user interface.

R.1.2.1.2 : Task Details
Description: user can add the task.
Input : Task name and place.
Output : Google map showing the place entered by user.

R.1.2.2 : View Task
Description : user can view the task.
Input : user select view-task option
Output : list of all the tasks showing in Google map.

R.1.2.3 : Start Task

Description: user can start the task.
Input : user select start-task option
Output : appropriate user interface.

R.1.2.3.1 : Task Order

Description: user can know which task to execute the first.
Input : user select the start button.
Output : Notification of the task.

R.1.2.4 : Finish/Delete Task

Description: user can delete the task after it is finished.
Input : user select finish-task option
Output : appropriate user interface.

R.1.2.4.1 : Task list

Description: user can finish/remove the task.
Input : user select one of the task shown in the list.
Output : appropriate user interface.

3. 3 Other Nonfunctional Requirements

1 Performance

Performance of the system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays.

2 Safety

User details should be securely stored to the server. The main security concern is for user account hence proper login mechanism should be used to avoid hacking.

3 Reliability

The capability to maintain the specified level of performance is what meant by reliability. This application will run on any android device.

4 Database

System requires to access users data fastly to maintain the performance.

5 Portability

The capability adapted for different specified environments

Without applying action or means other than those provide for this

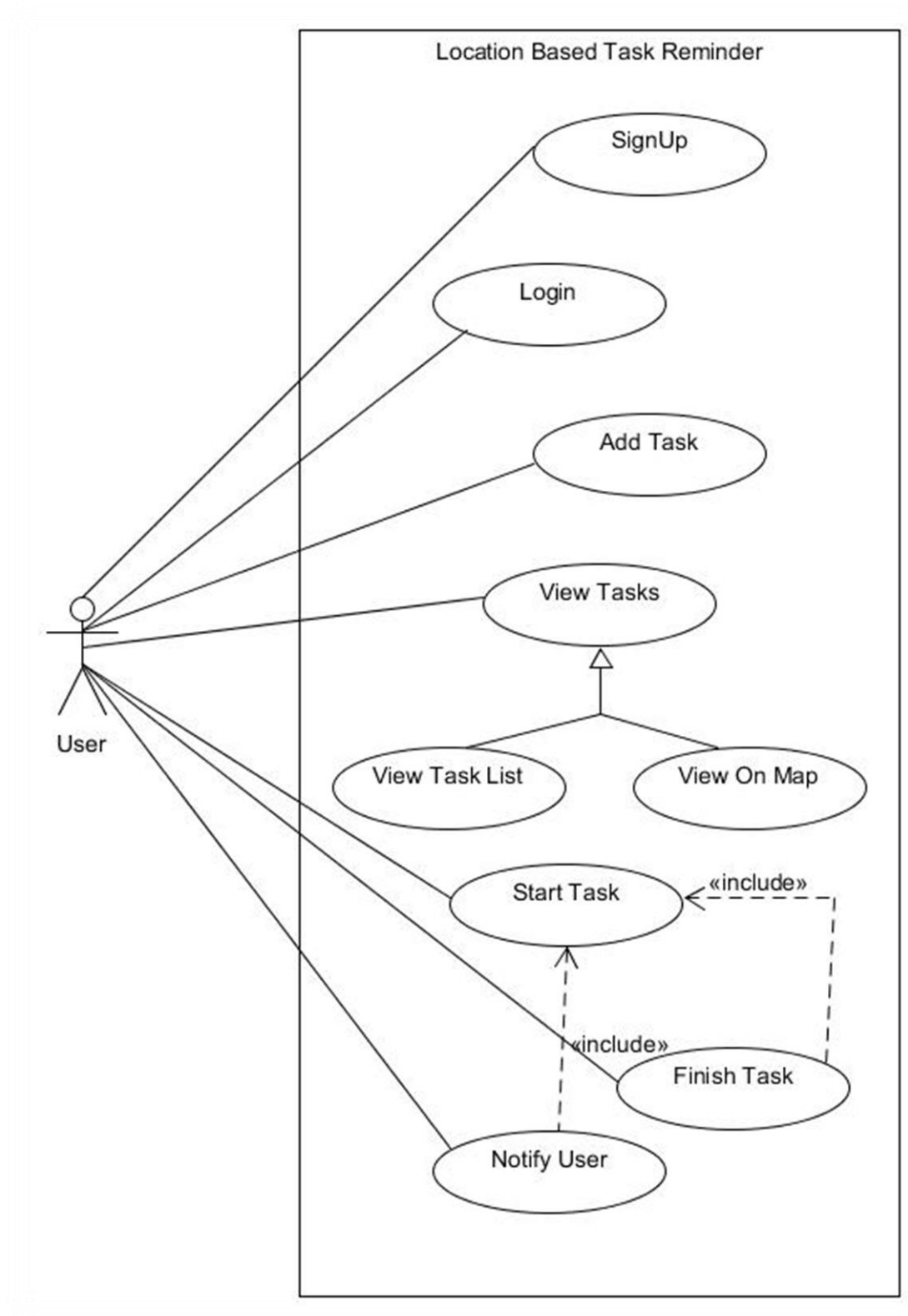
Purpose in the product. Since, phones are portable, so do the application.

6 Availability

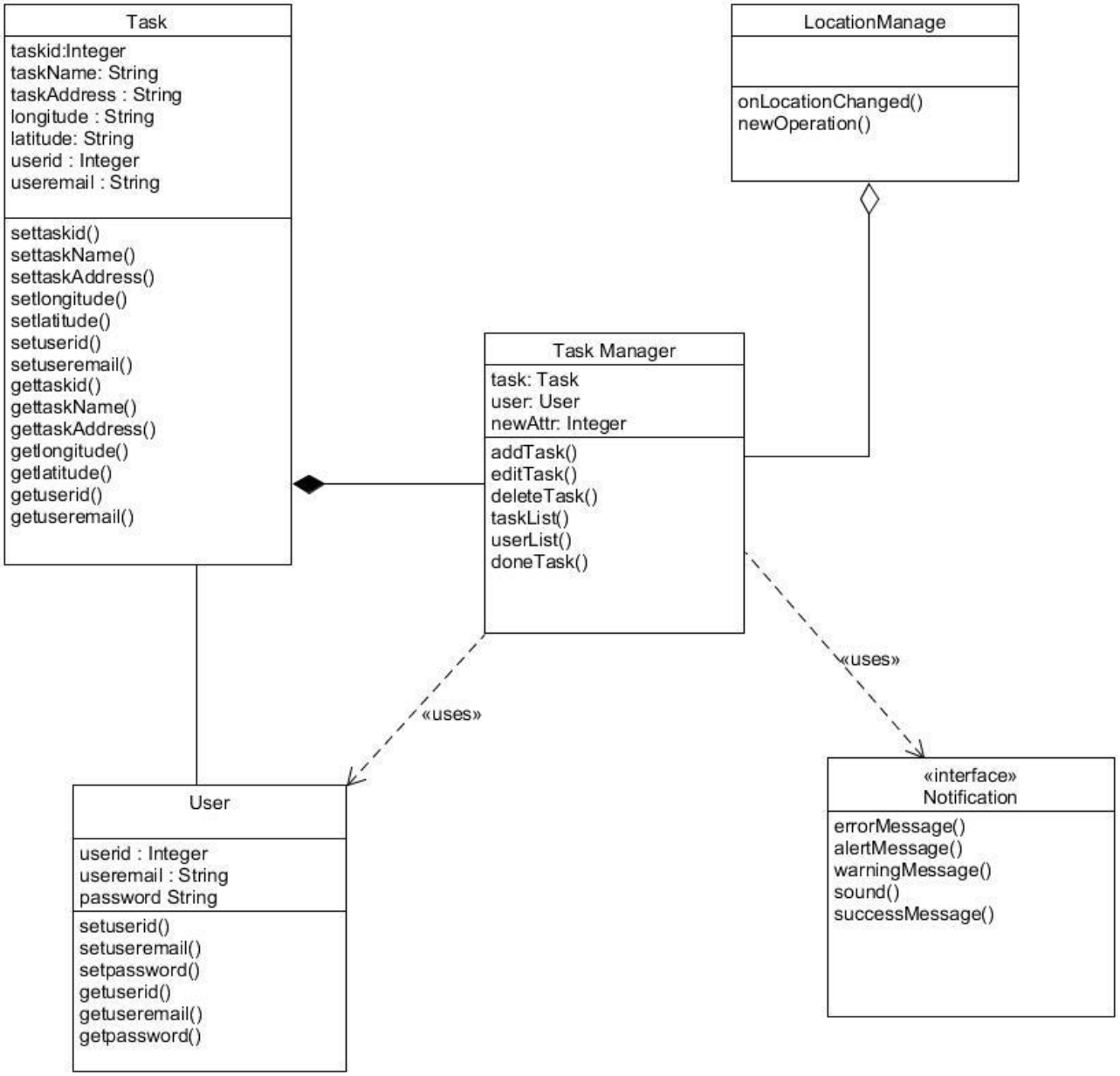
The application is run 24X7 if internet is available.

4 Design

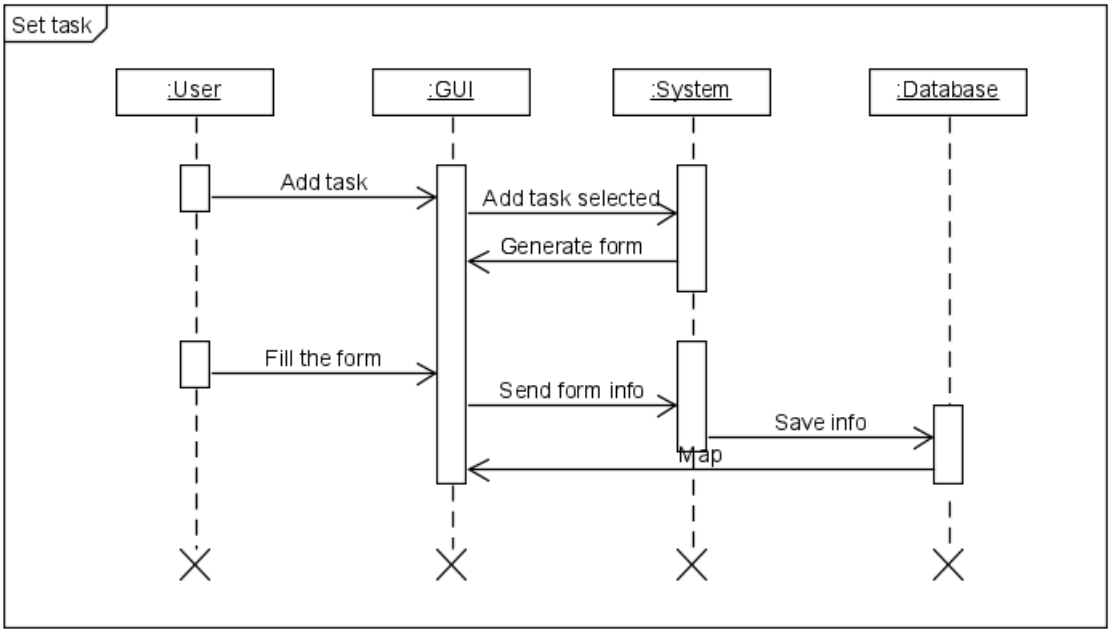
4.1 Use Case Diagram

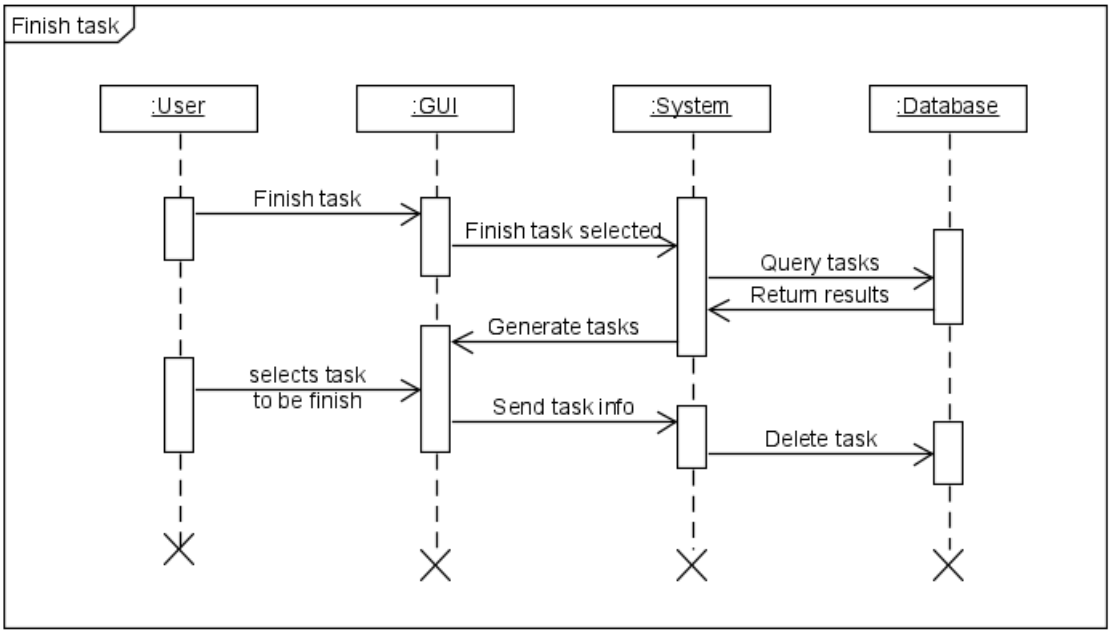


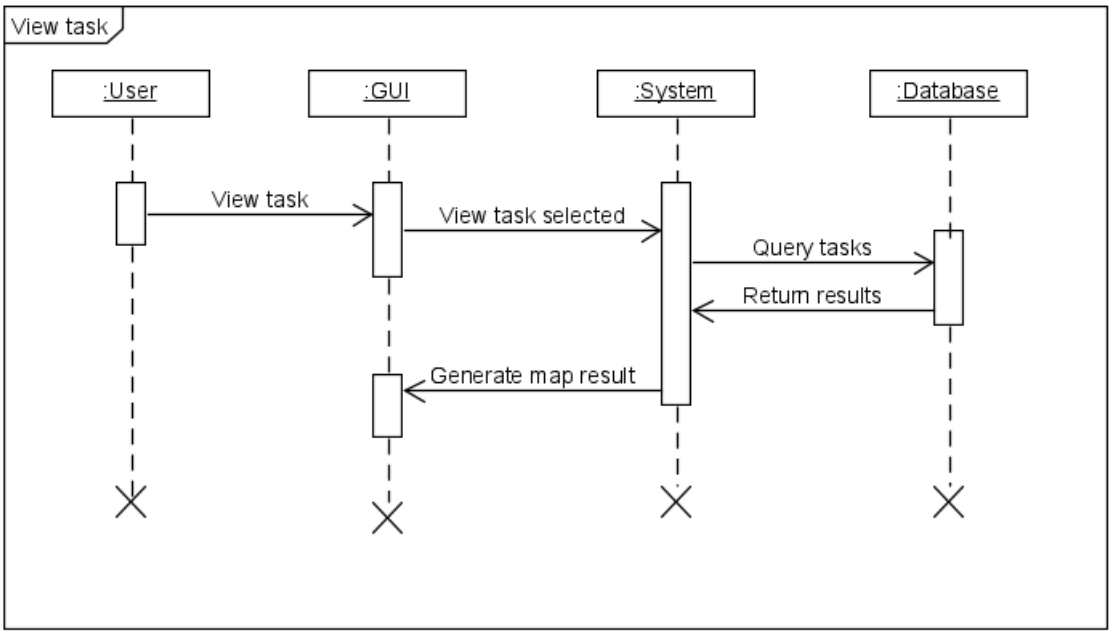
4.2 Class Diagram

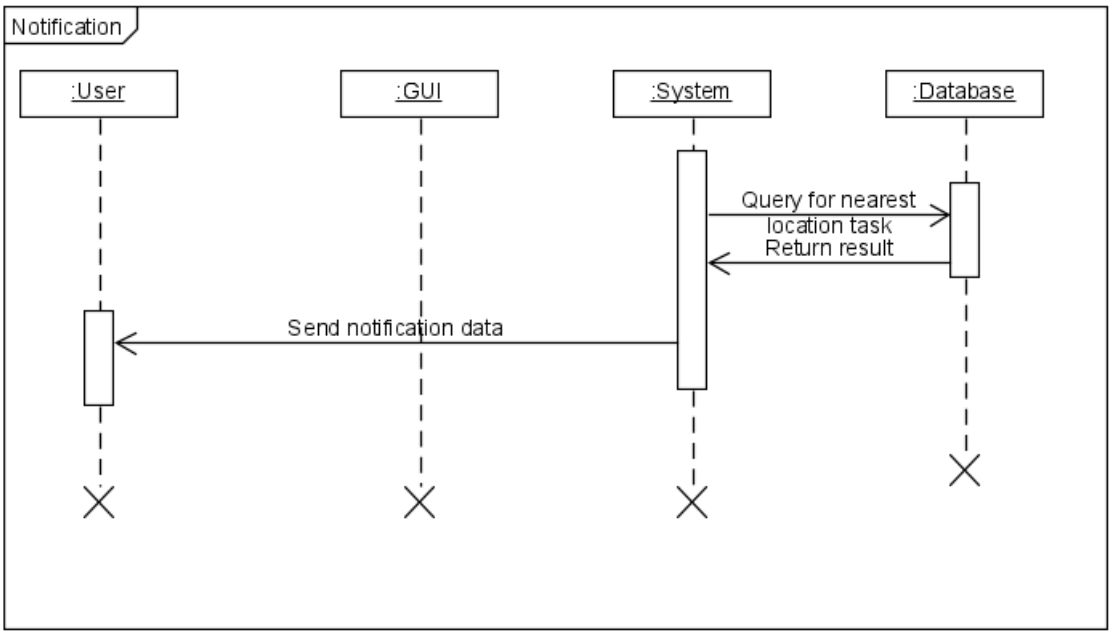


4.3 Sequence Diagrams

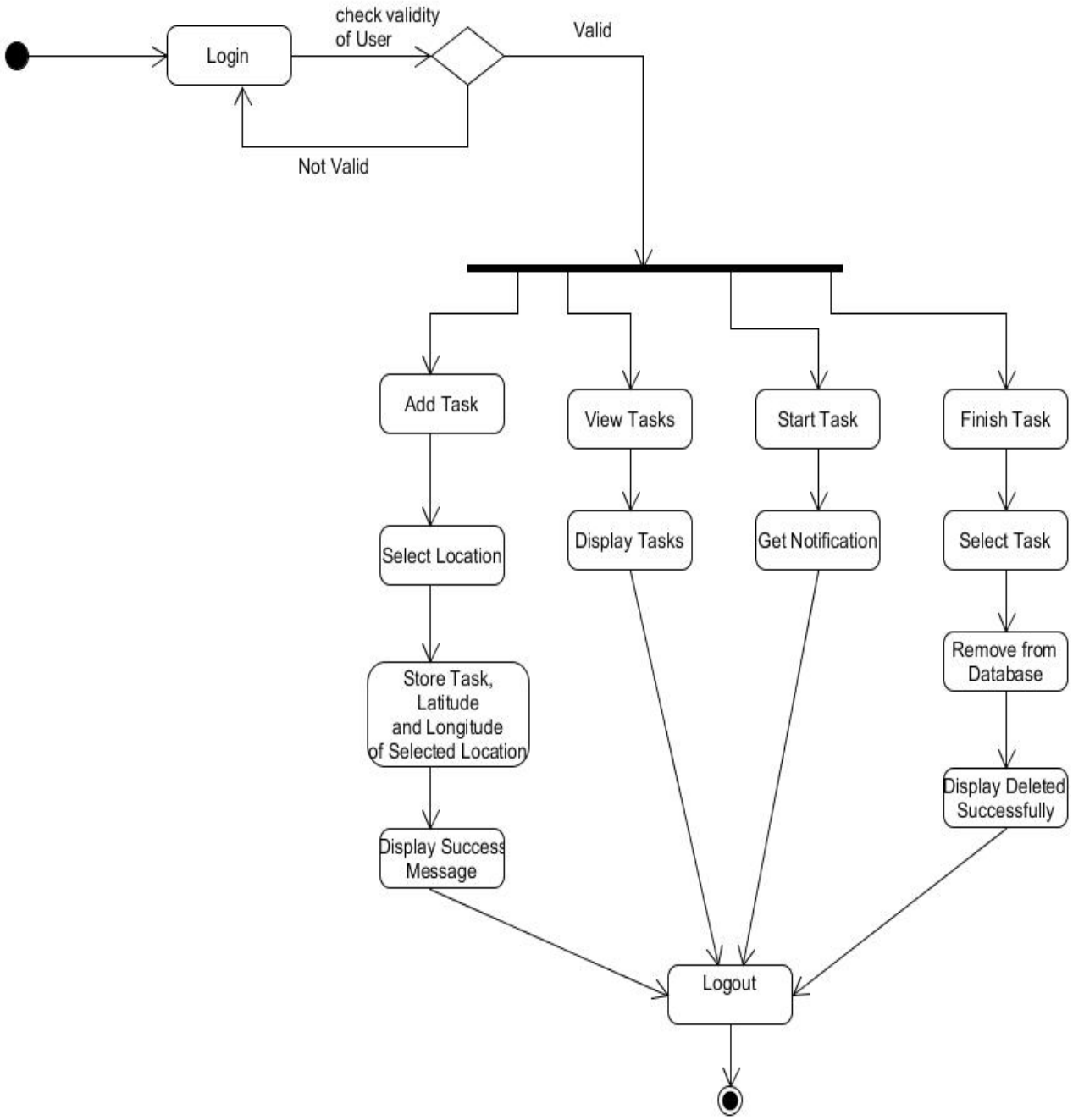




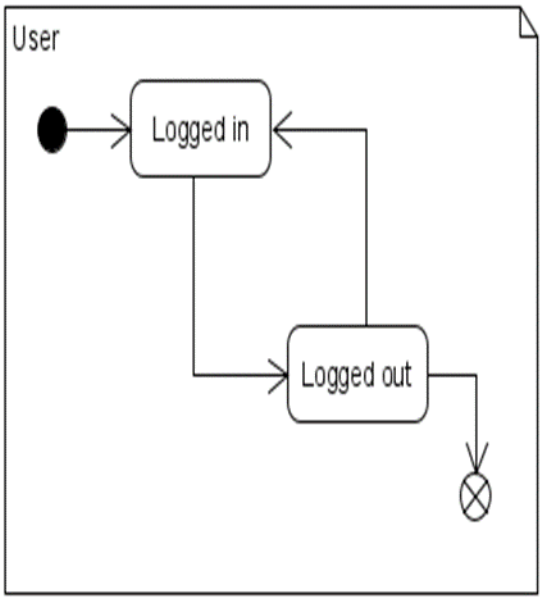

View task
:User
:GUI
:System
:Database
View task
View task selected
Query tasks
Return results
Generate map result

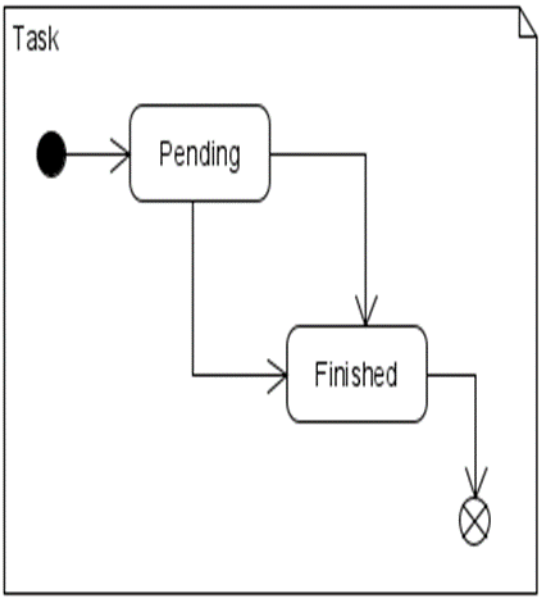

Notification
:User
:GUI
:System
:Database
Query for nearest
location task
Return result
Send notification data

4.4 Activity Diagram

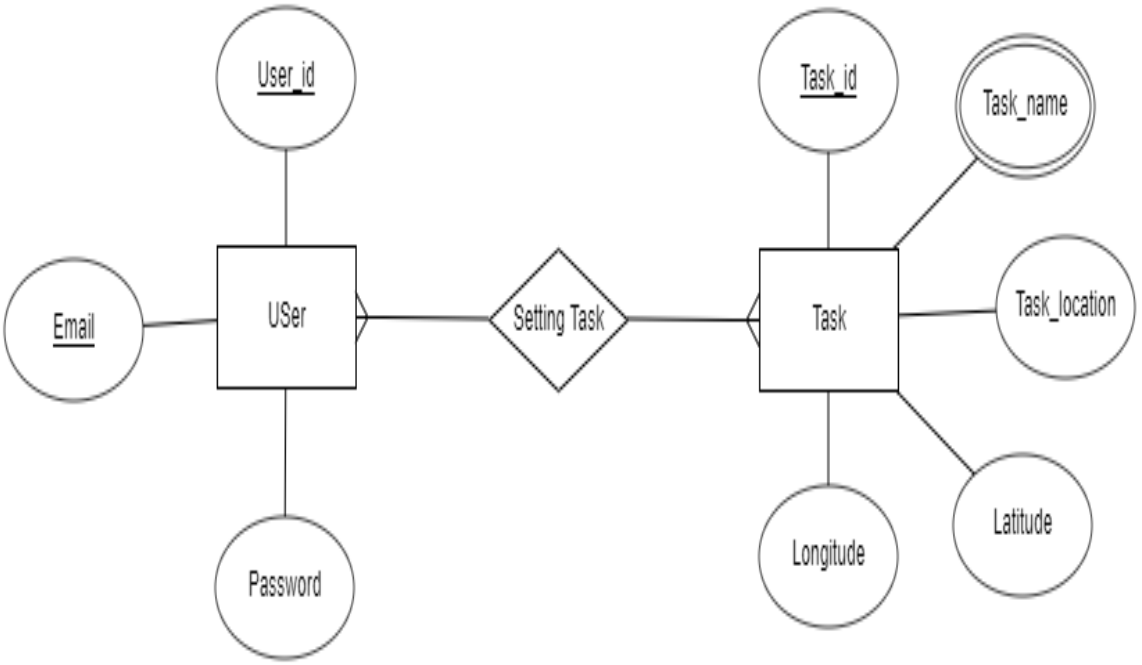


4.5 State Diagram





4.6 E-R Diagram



4.7 Data Dictionary

User								
Sr. No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	User_id	Varchar2	10	Yes	Yes	PK		
2	Email	Varchar2	20	No	Yes			
3	Password	Password	12	No	No			

Task								
Sr. No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Task_id	Varchar2	10	Yes	Yes	PK		
2	Task_Name	Varchar2	40	No	Yes			
3	Task_Location	Varchar2	50	No	No			
4	Latitude	Number	10	No	No			
5	Longitude	Number	10	No	No			

5 Implementation Details

5.1 Modules Description

Main User Module:

This module gives the main home screen of the application. It contain different unfinished task added by user and one floating action button which gives you different options to add task, view task. This module does needs authentication.

Input: User Selection

Output: Corresponding Response

Registration Module:

This module is used to store user's data to the Database and enables the user to login the application. All the fields in this in this module contain required validations. User can also navigate to login page if he/she has already registered.

Input: User's Information

Output: User registered and redirected to login screen.

Processing: Validating user's data and then storing them to database.

Login Module:

This module takes users credentials and then verifies it with registered users, if user is not registered the invalid credentials is shown else if they match with database then login user.

Input: User Credentials
Output: Logging user.
Processing: Verifying user credentials with the Database.

Task Detail Module:

This module is display task in detail when any one unfinished tasks from task list (user home) is click. This module display task name, place, and view on map option and also provides finish option to finish the task.

Input: User Selection
Output: Task Detail Screen
Processing: Fetching data of clicked task from database.

6 Testing

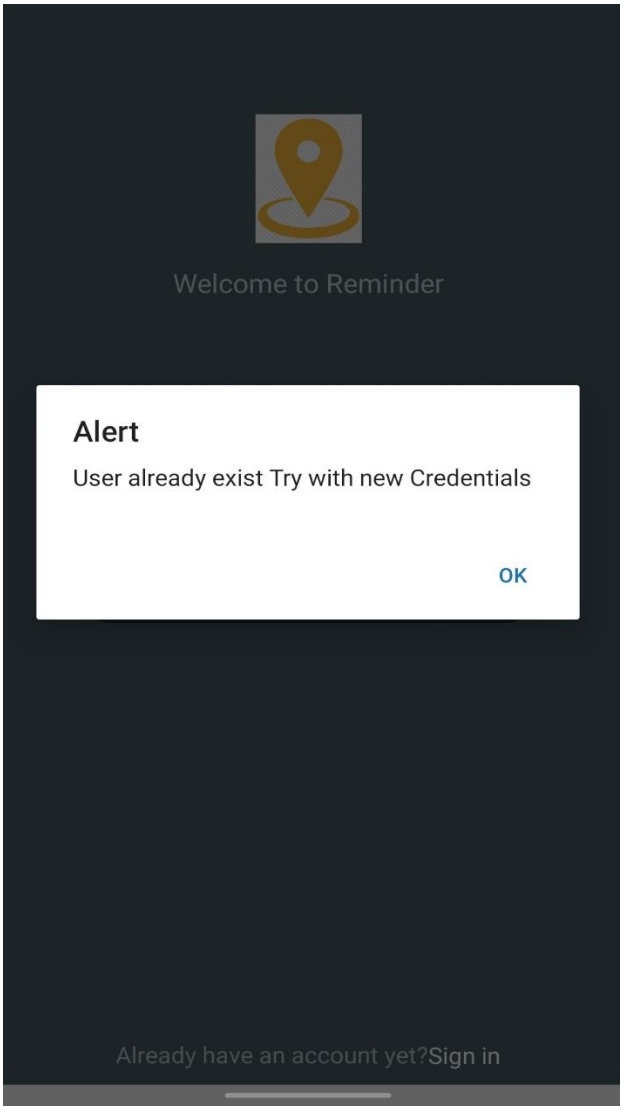
6.1 Testing Method:

We have performed Black-box testing for the testing purpose.

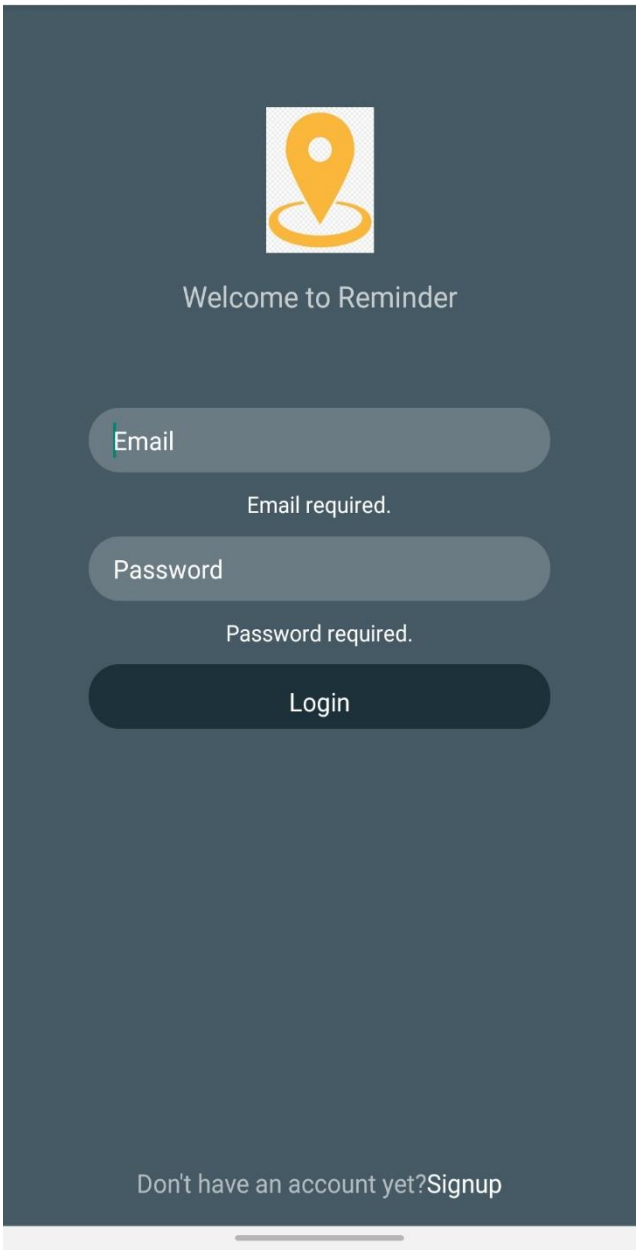
6.2 Test Cases:

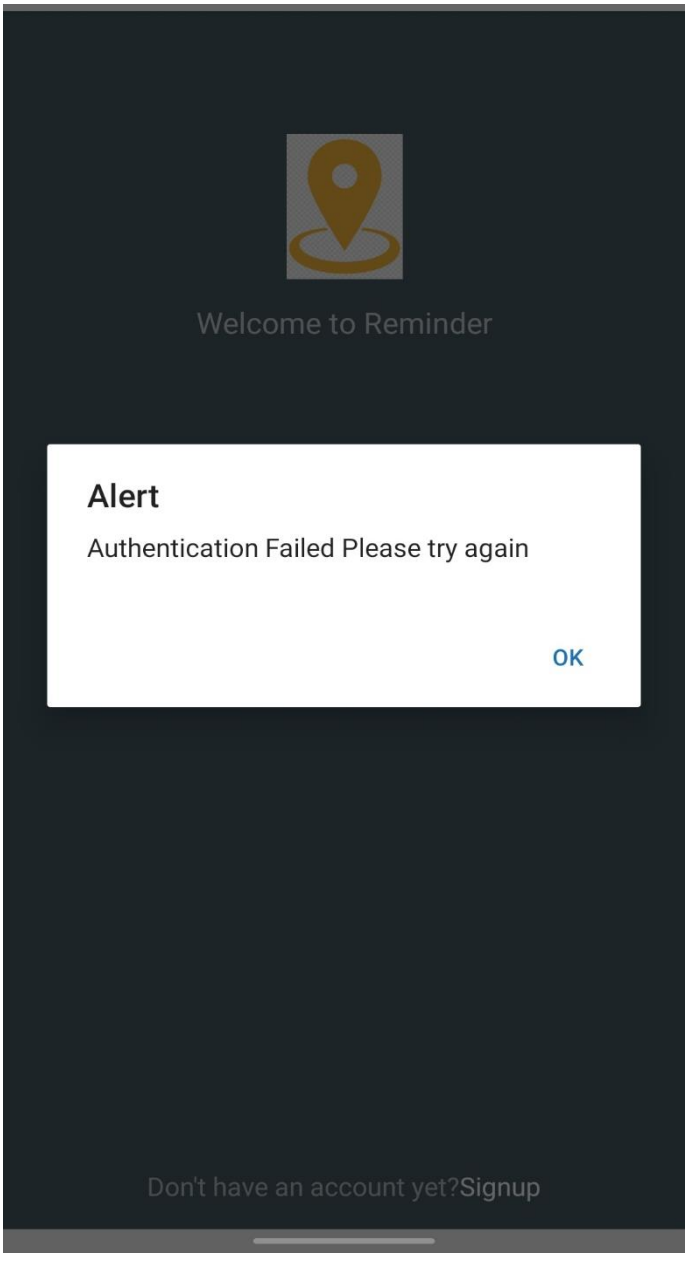
For Registration:

Email and Password must be unique otherwise it gives error as below.

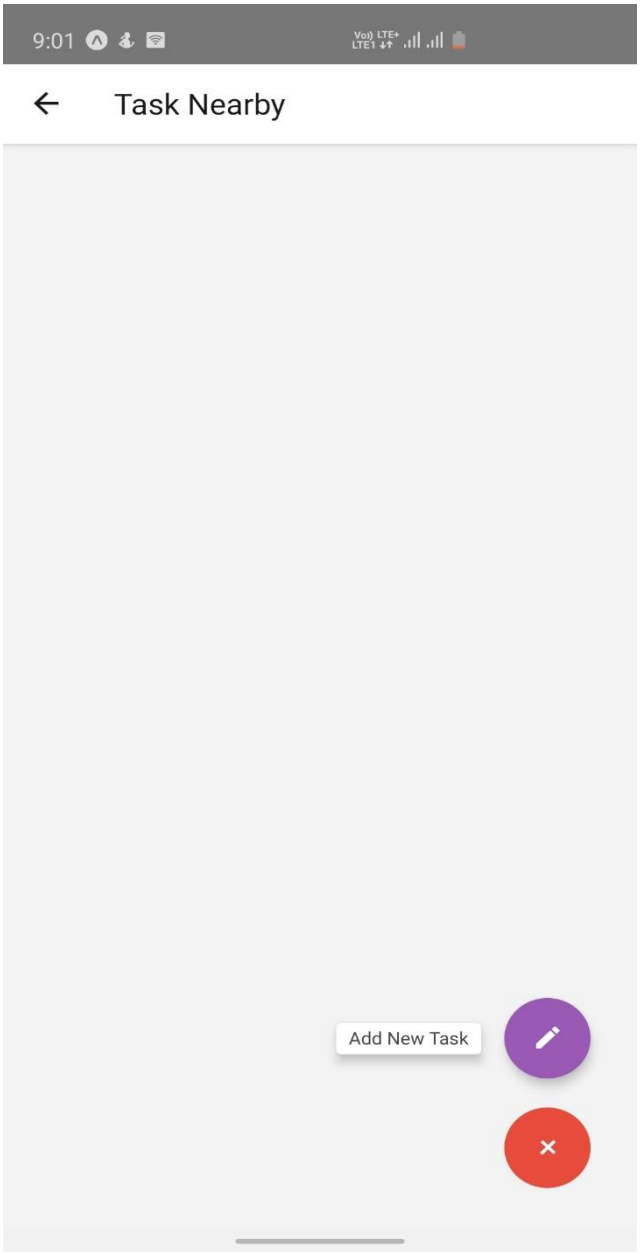


For Login:
If all field data are not entered by user than it shows error as below





7 Screenshots



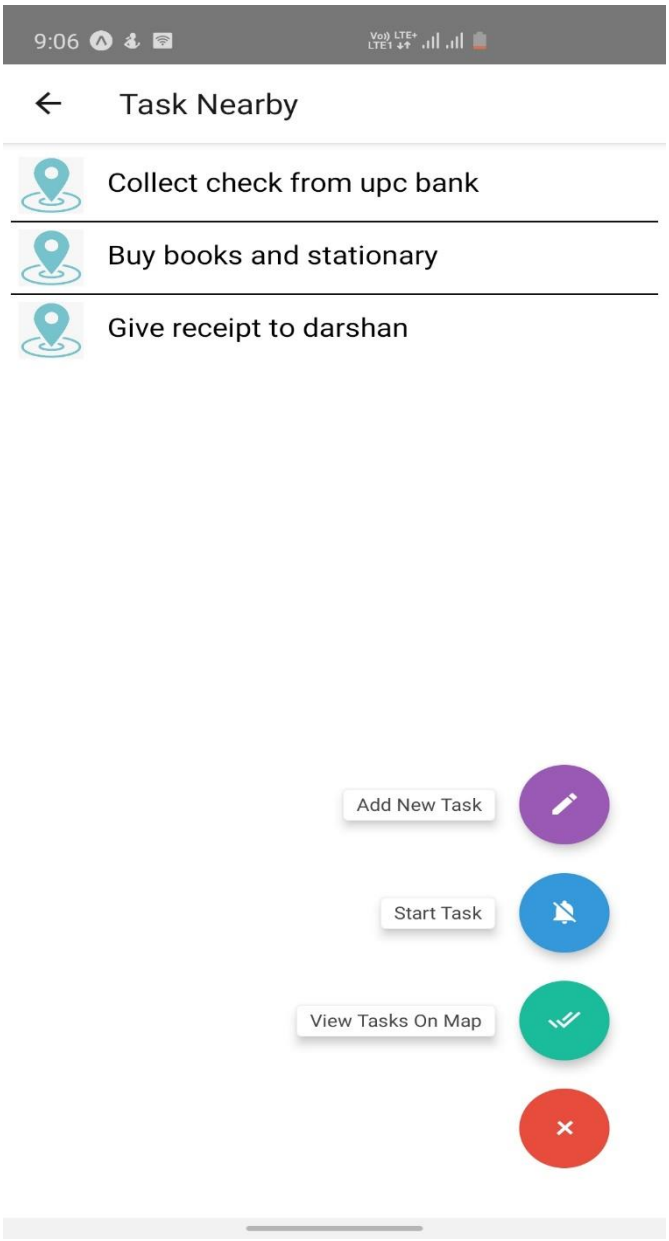


Fig 7.1: After Login Successfully User's Home Page before any task added.

Fig 7.2: User's home Screen after adding some Ta

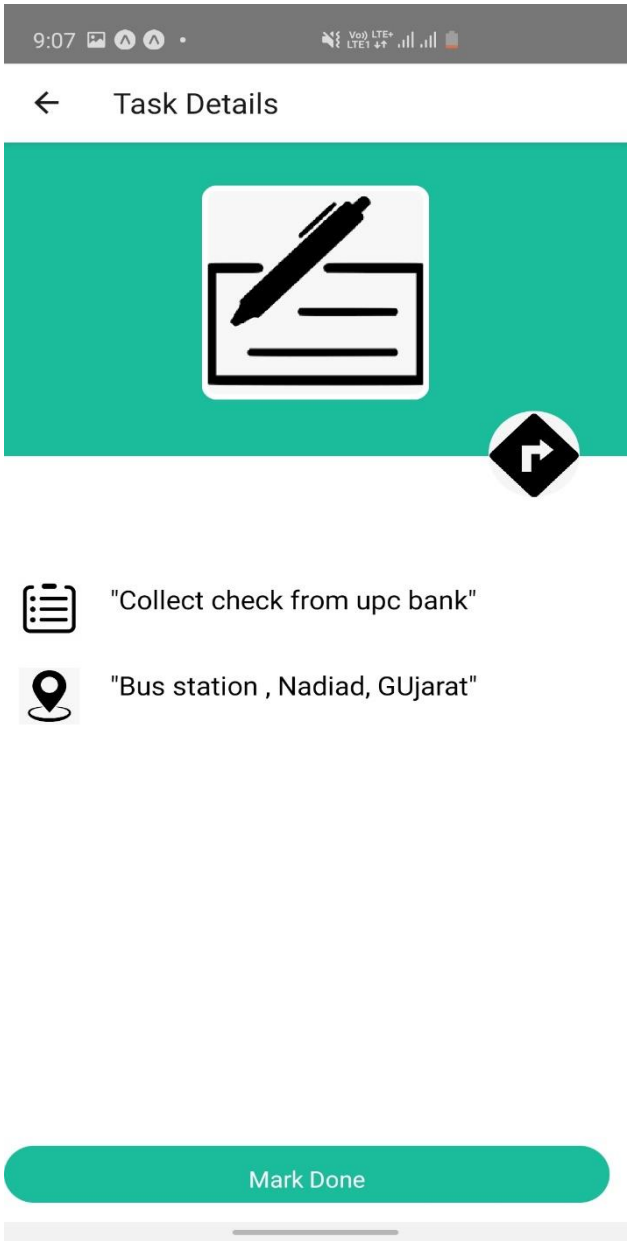


Fig 7.3: Task Detail Screen after clicking any task from Home screen.

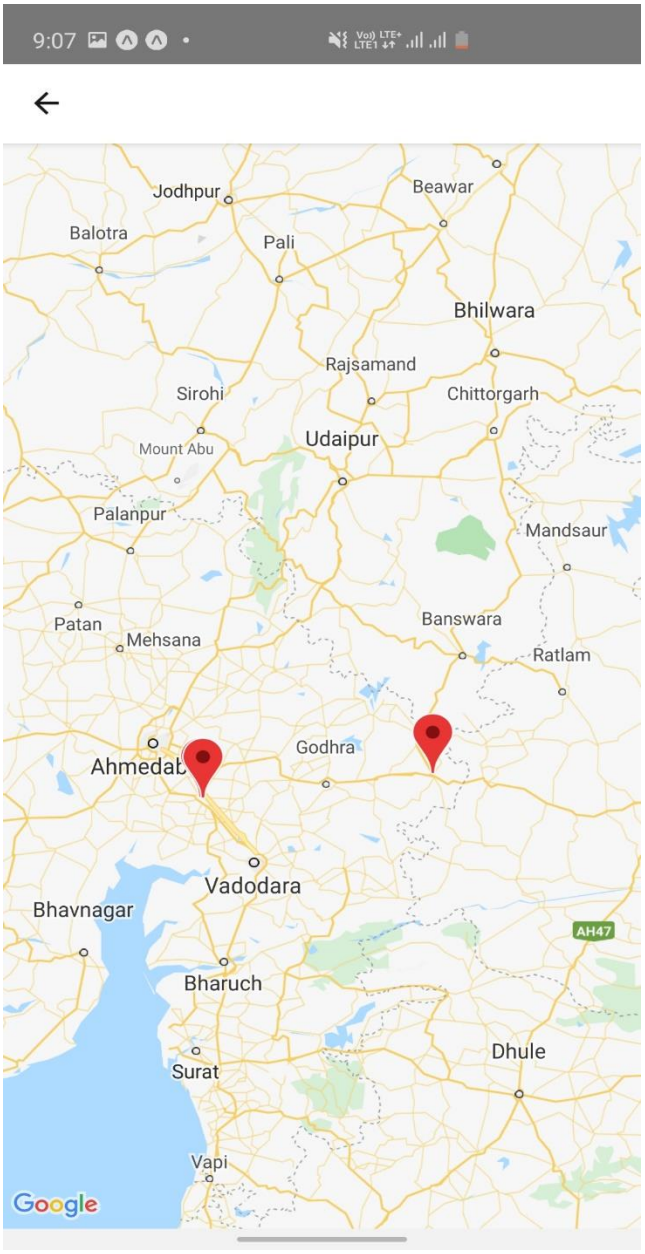


Fig 7.4: After Clicking View Tasks on map Button.

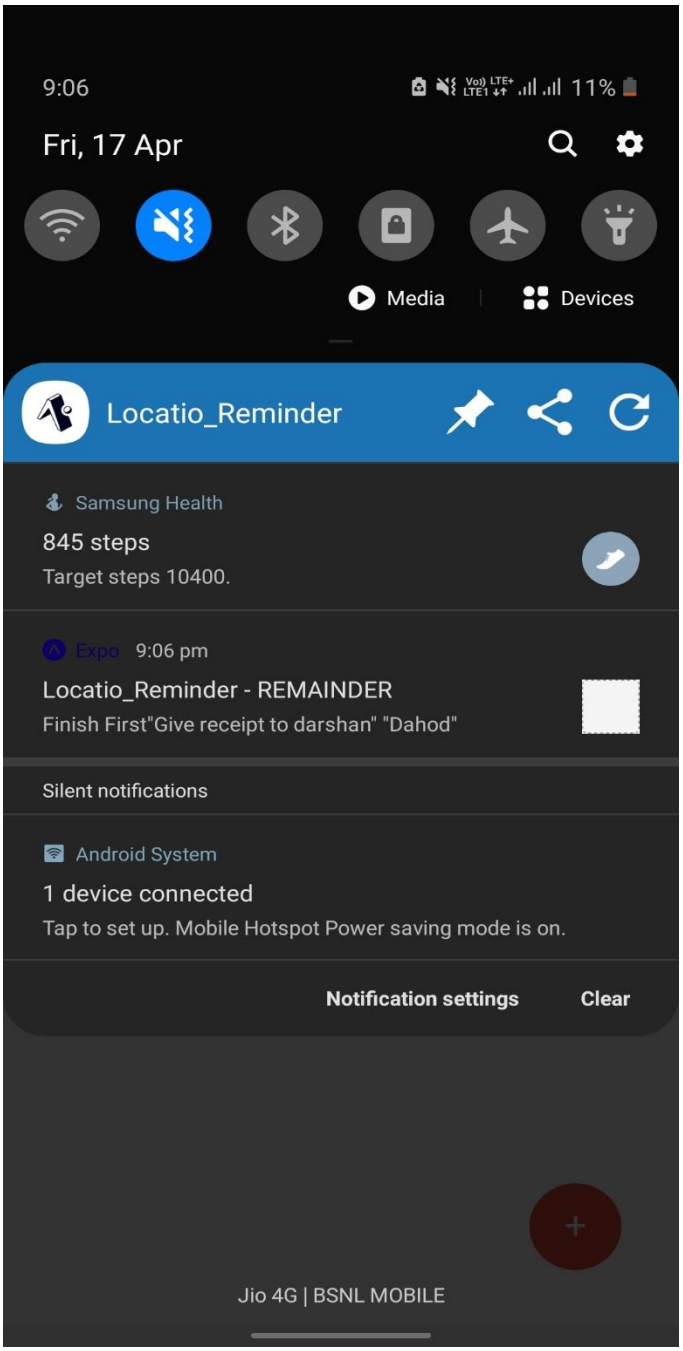


Figure7.5: User get notification of nearest task after clicking start task.

Conclusion

The functionality implemented in the system was done after understanding all The system modules according to the requirements.

Functionalities that are successfully implemented:

- User Registration containing all the necessary validations on fields.
- Login
- Forgot Password
- User Authentication
- Session Management
- Add Task
- View Tasks On a Map
- Start Task
- Notify User
- Finish Task

After the implementation and coding of system, comprehensive testing was performed on the system to determine the loopholes and possible flaws in the application.

Limitations and Future Enhancements

Limitations

- Application has lacks in terms of security aspects.

Functionalities not implemented

- In this application scheduling of task is not implemented. In which, User can schedule task on particular date and user get notification on that day.

Future Extension

- Using Voice User Interface (VUI) to perform different operations like add task, view task, start task and finish task.
- Add other functionalities like task scheduling, maintain notes, images and add them using VUI

Bibliography

Websites:

1. https://www.w3schools.com –For Html, CSS, Bootstrap, JavaScript.
2. https://reactnative.dev/docs/- For Learning React-Native
3. https://stackoverflow.com –For solving problems.
4. https://medium.com/- For solving doubts.

Useful Links:

1. https://docs.expo.io/ - For installing Expo CLI.
2. https://www.youtube.com/watch?v=_K41vd_W2qE– For understanding.
3. https://play.google.com/store/apps/details?id=app.tasknearby.yashcreations.com.tasknearby&hl=en_IN- Reference Application