Design Document

For

Payment Tracker

Version 1.0

Prepared by

Sindhura.L(01JST17IS050)
Spoorthi.S(01JST17IS062)

15/04/2020

SYSTEM ORGANIZATION:

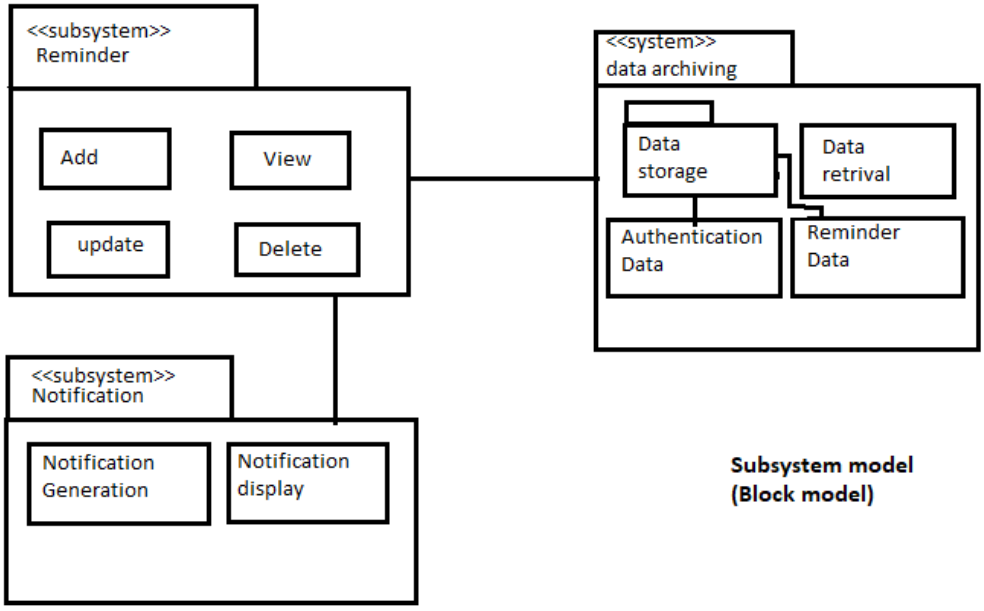
Client server architectural model is used as basic system organizational model for the project. The project uses Firebase as the database works on the basis of client server model. The details entered by the users will be stored in the firebase server. Clients can access their data from any device at any point of time. Good network connection is a necessary requirement for this application. Even if users uninstall the app the user's data will be safe in the server. And also, when app gets connected to Firebase, it does not connect through normal HTTP. It connects through a WebSocket. WebSocket's are much, faster than HTTP and response time is less.

CONTROL STYLE:

Event driven model is used as the control style for the project. The app is driven by externally generated event. When event occur, control is transferred to the subsystem that can handle the event. When the users want to create new reminder, the user should click the new reminder button and create reminder activity does the job.

Here the button click is the acts as event and new reminder is created as a response.

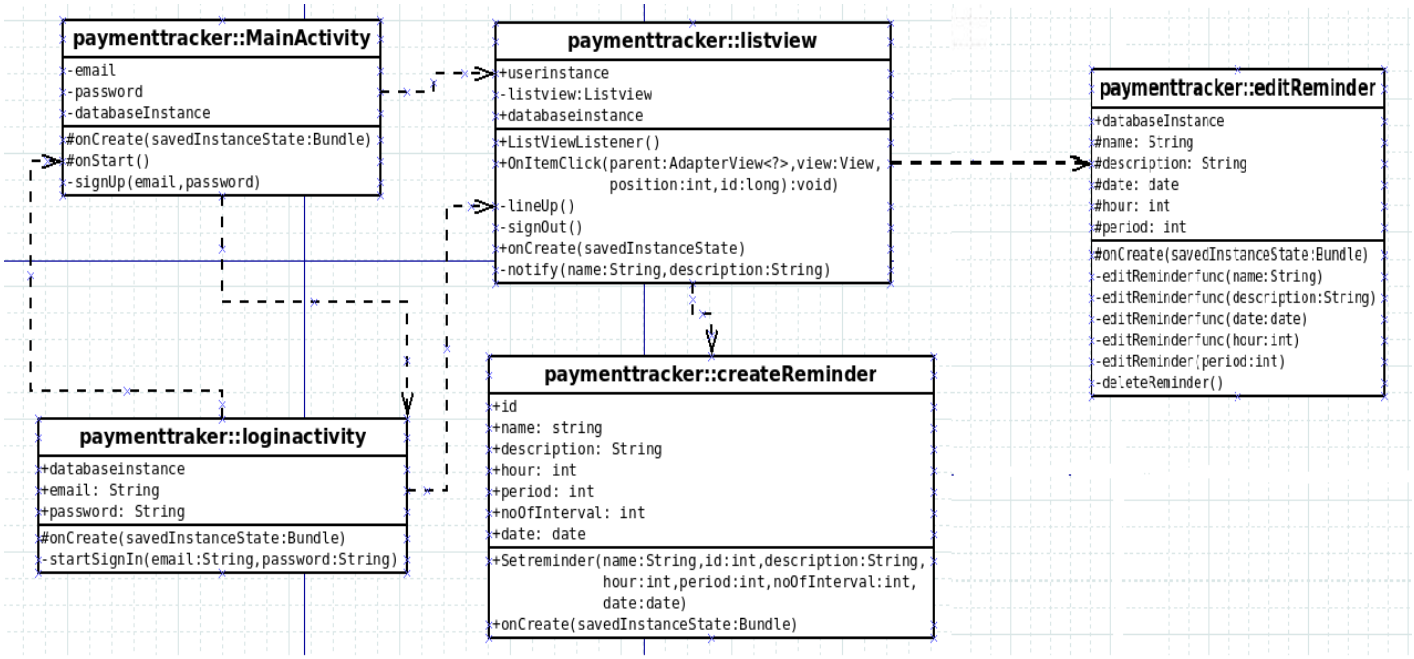
SUBSYSTEM MODEL:



Subsystem model
(Block model)

The payment tracker application mainly consists of three subsystems, reminder subsystem, notification subsystem and data archiving subsystem. The reminder subsystem contains the components/modules that help in performing activities such as adding new reminder, viewing existing reminders, edit reminders and delete reminders. This subsystem interacts with the user to take input regarding adding or updating reminders and displays the created reminders through the graphical user interface. The notification subsystem consists of notification generation and display components. The notification generation component generates notifications on the date specified by user. The display component displays the notification to the user in the notification bar. The data archiving subsystem has data storage subsystem and data retrieval module. The data storage subsystem contains two components that store user authentication data and reminder data. The data retrieval module retrieves data stored in the database and provides it to the reminder subsystem to display the retrieved information.

CLASS DIAGRAM:



Payment tracker:: Main Activity

This activity takes email and password and creates an account for the user. The method onStart() checks if the user is logged in, if yes he is automatically directed to the list view activity.

Payment tracker:: login Activity

This activity takes email and password from the user in and logs the user in.

Payment tracker:: listview Activity

This activity lists all the previously set reminder by the user. notify() method sends notification at the time specified by the user.

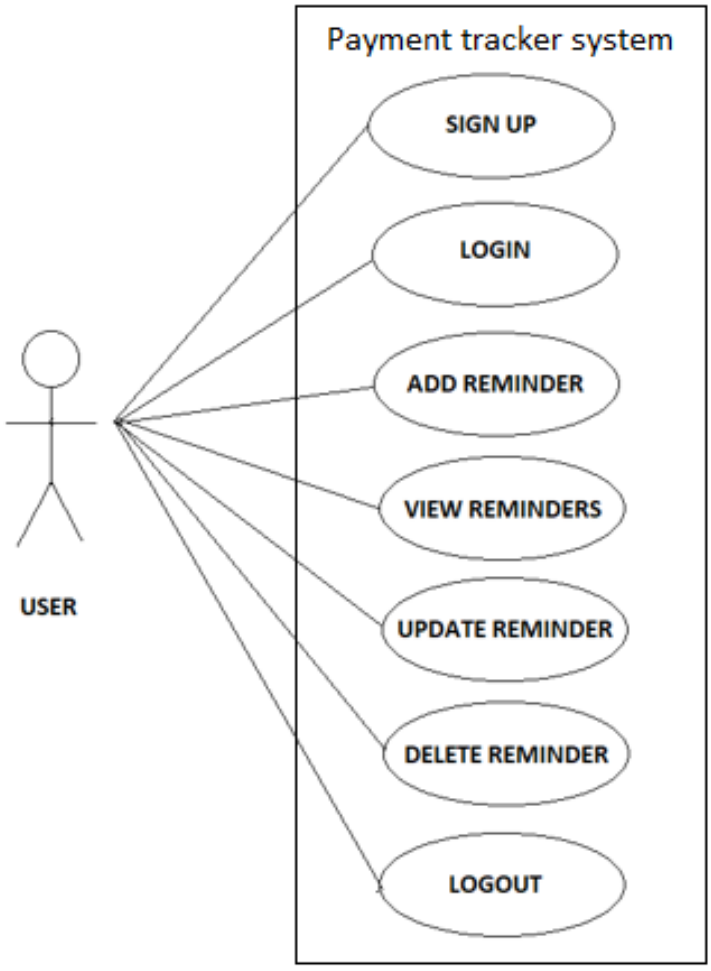
Payment tracker:: createReminder

This activity takes all the necessary information from the users and creates a new reminder.

Payment treacker :: editReminder Activity

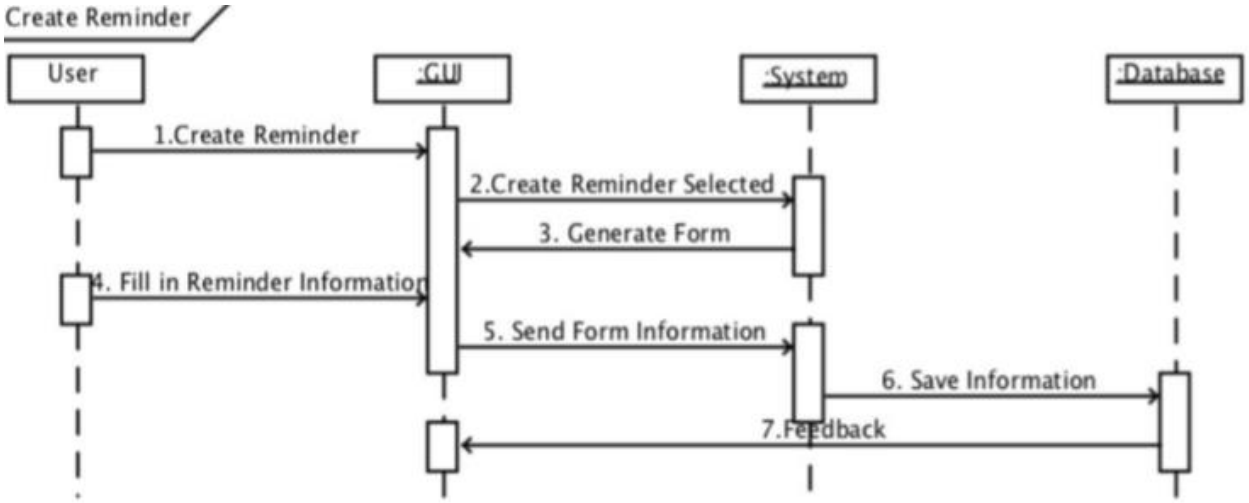
User can edit or delete the reminder information using this activity.

USECASE DIAGRAM:



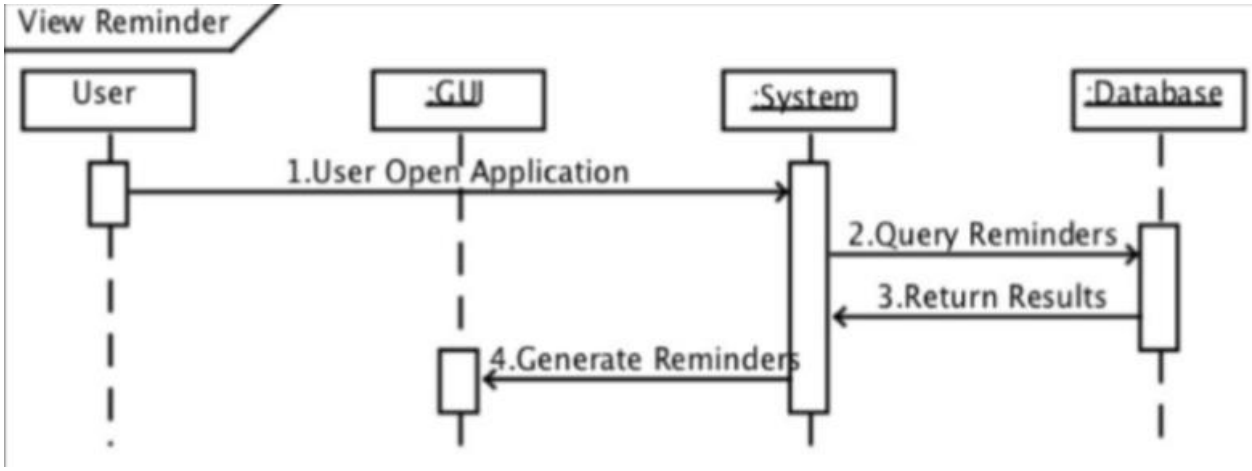
The different use cases are described with the help of sequence diagrams given below.

SEQUENCE DIAGRAM:



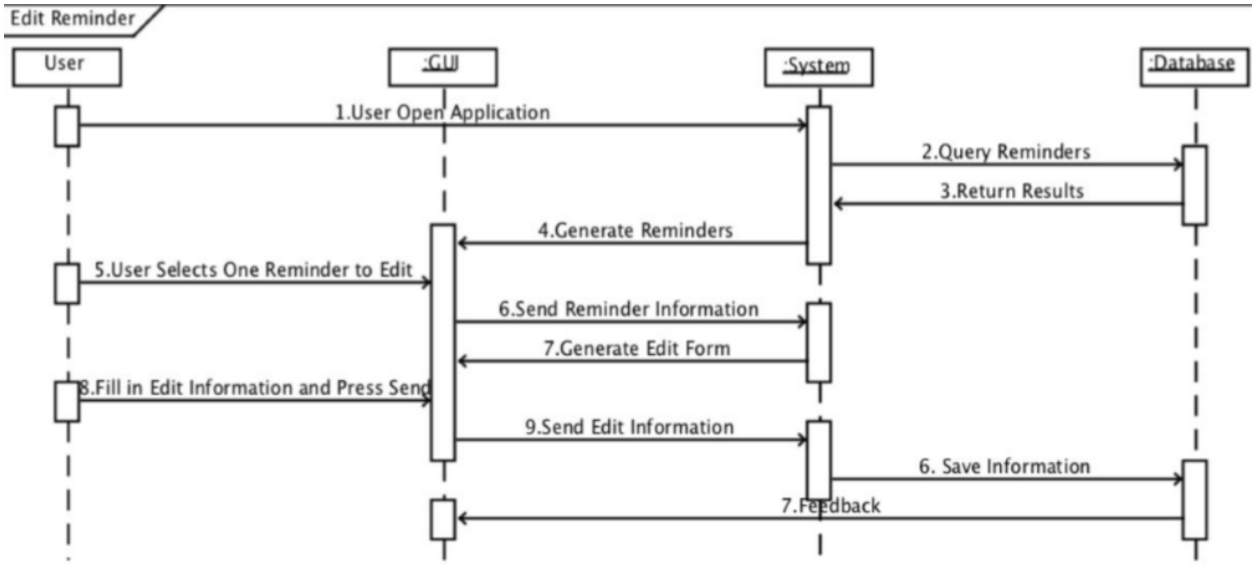
SD1. Sequence Diagram Create Reminder based on Use Case Create Reminder.

The user can create the reminder by selecting the create reminder option. The system generates a form. After entering the details in the form, the system stores the information in the database and sends feedback.



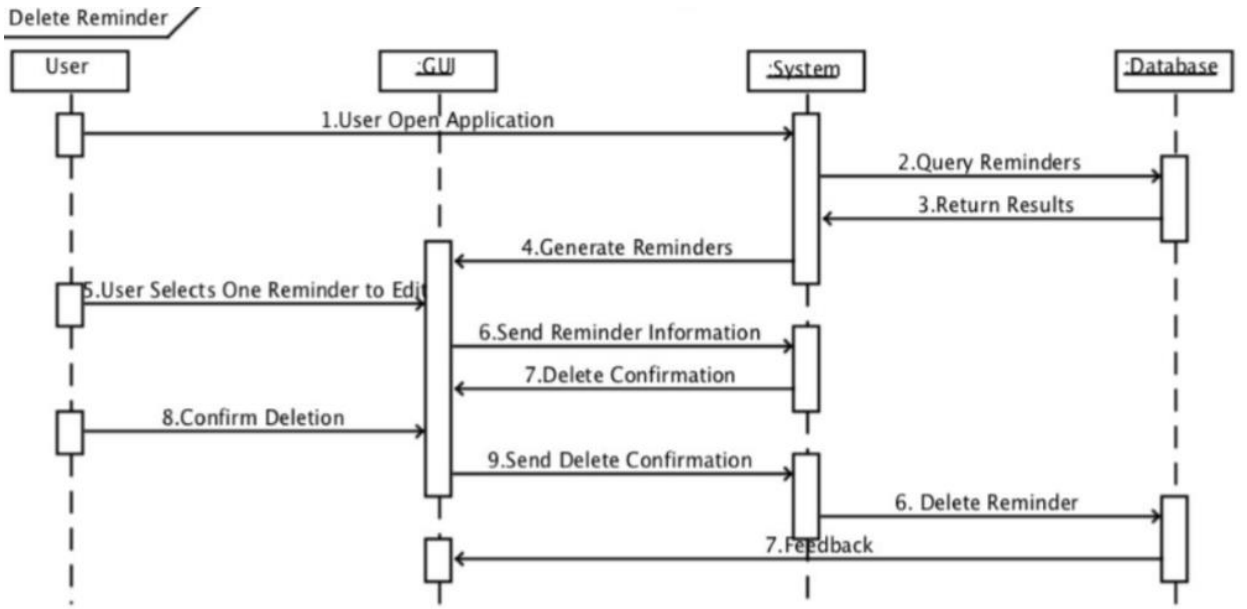
SD2. Sequence Diagram View Reminder based on Use Case View Reminder.

As the user login to the system, the system queries the database for the reminders specific to the user. Database sends the list of already set reminders which is presented to the user



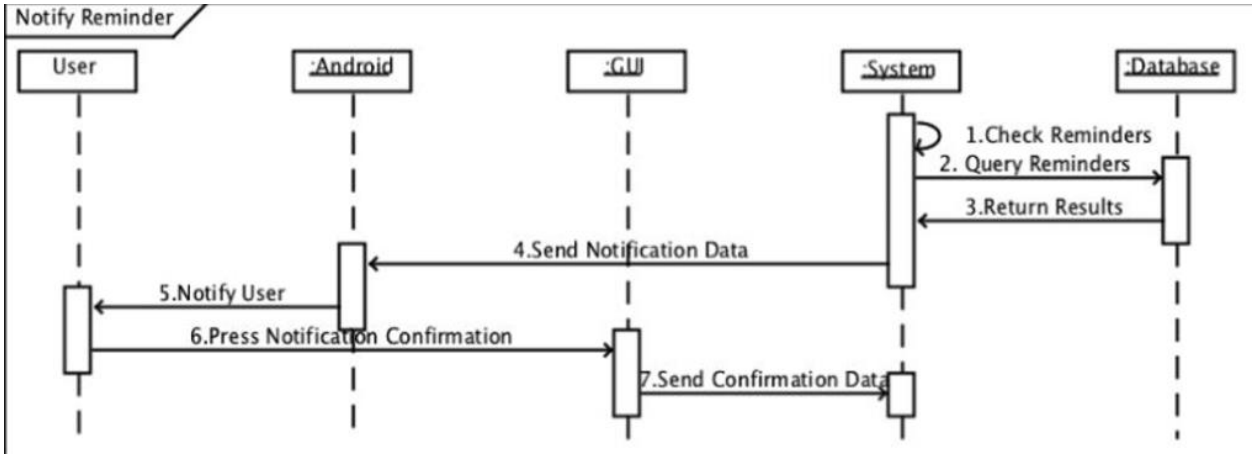
SD3. Sequence Diagram Edit Reminder based on extendable Use Case Edit Remin der.

After viewing the list of reminders, the user can select one among them to edit. User is asked enter new reminder information through a form. Edit information is sent to the database for storage and the database returns feedback to the user.



SD4. Sequence Diagram Delete Reminder based on extendable Use Case Delete Reminder.

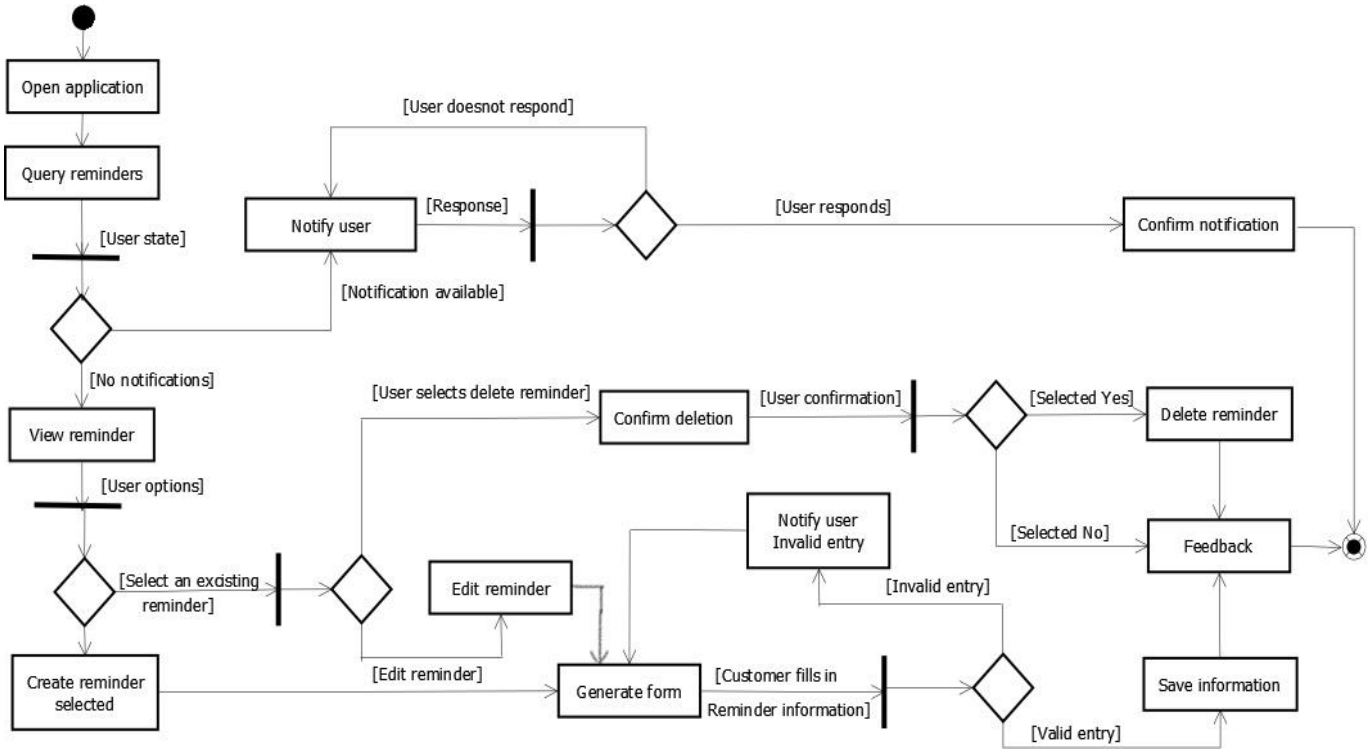
User can select one among the listed reminders and request for deletion of the reminder. The system asks for the confirmation once again. If user confirms deletion the reminder is deleted from the database.



SD5. Sequence Diagram Notify Reminder based on Use Case Notify Reminder.

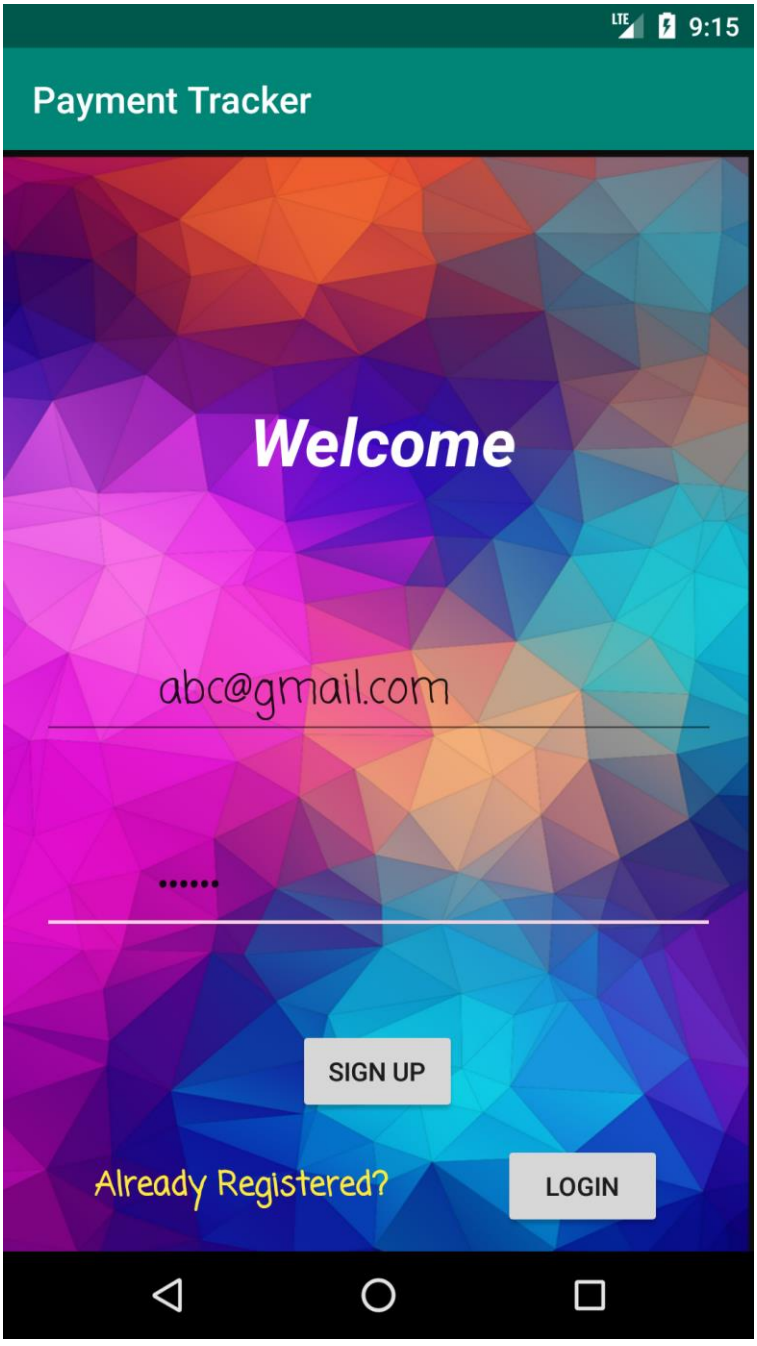
The system checks for the reminders by querying the database. At particular date and time specified by the user the system sends reminder information to the android which is presented as a notification for the user. The users can confirm the notification for acknowledgement.

STATE DIAGRAM:



The state diagram above shows how the application goes to various states upon different responses or interactions from the user. Firstly the user opens the application to view the reminders. If there are any notifications he will be notified until he responds to the notification. If there are no notifications he can view the reminders. Then user can either create a new reminder or select a previous reminder and either edit or delete it. If he selects to create a new reminder a form is generated and user should input the details. If the entries are valid it is saved else user is notified about the invalid entry and is again prompted with the form. If the user desires to edit a reminder then again a form will be generated and data will be saved if entries are valid. If the user wants to delete a selected reminder he will be asked for confirmation and it will be deleted if he responds yes. The user will be provided with a feedback confirming the proper execution of the actions he performed (like adding, deleting or updating).

GRAPHICAL USER INTERFACE:

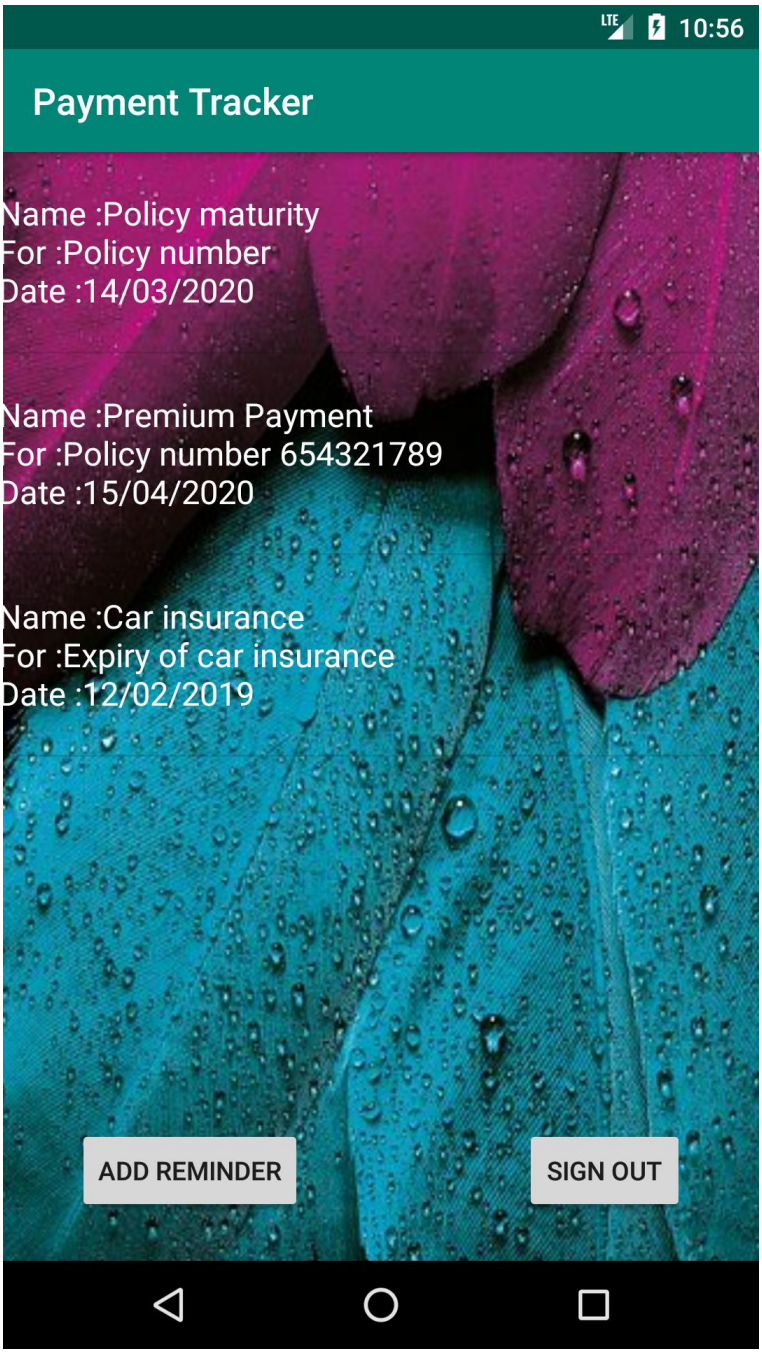


Signup Activity:

It is the opening page of the app. When an unregistered user wants to use the app, it is necessary for him to register by giving his email and password. If the user has an account, he can go to the login Activity.

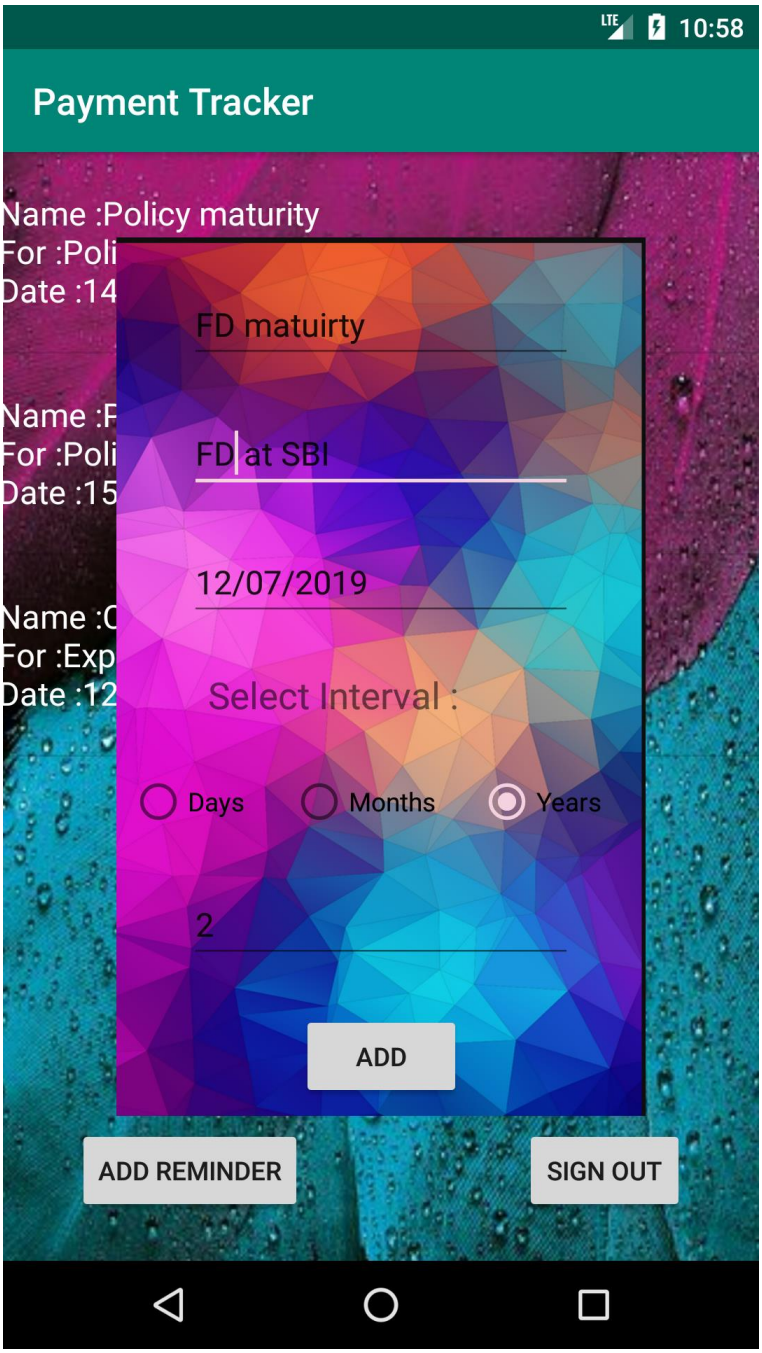
Login Activity:

An already registered user can enter email and password for the login, if the user hasn't logged out then he will be automatically directed to the view reminder Activity.



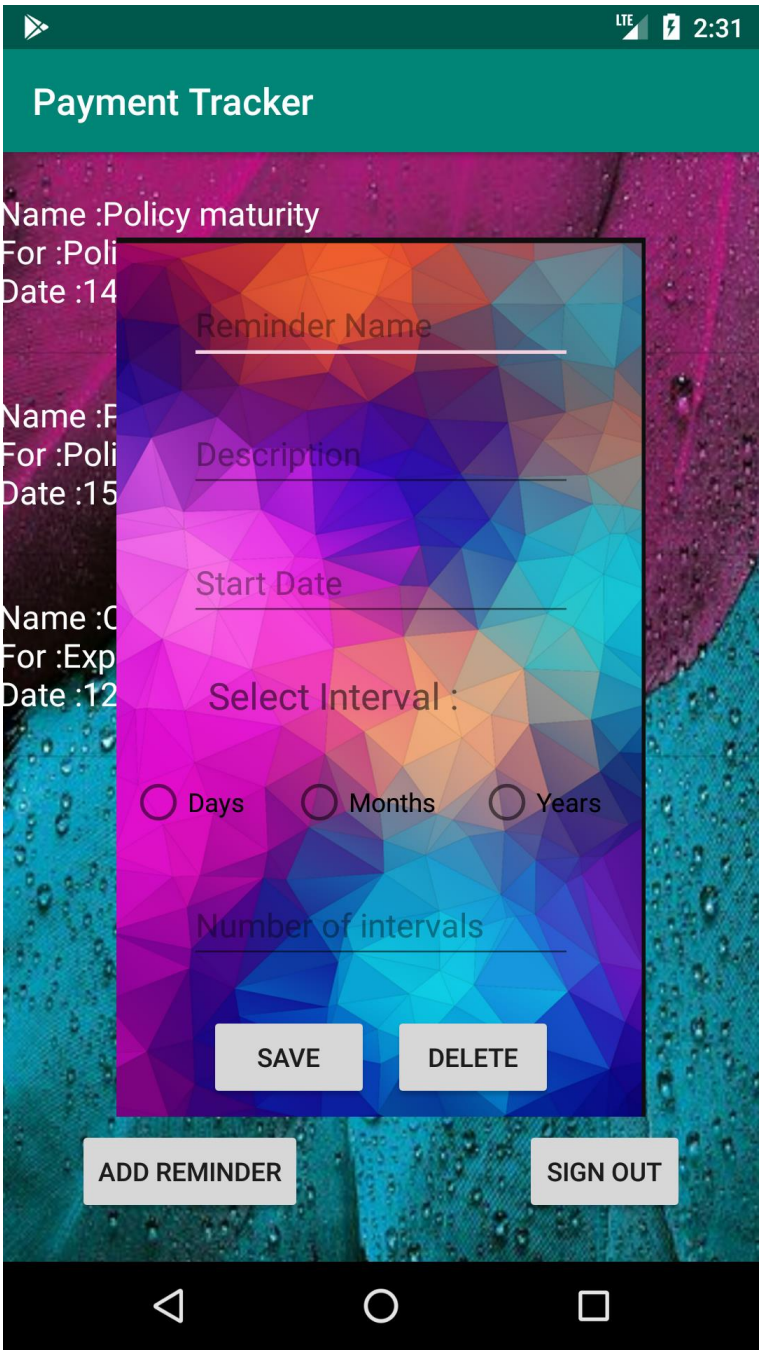
View reminder activity:

This page will have the list of all reminders which are already set by the user. Add new reminder and sign out options is also provided here.



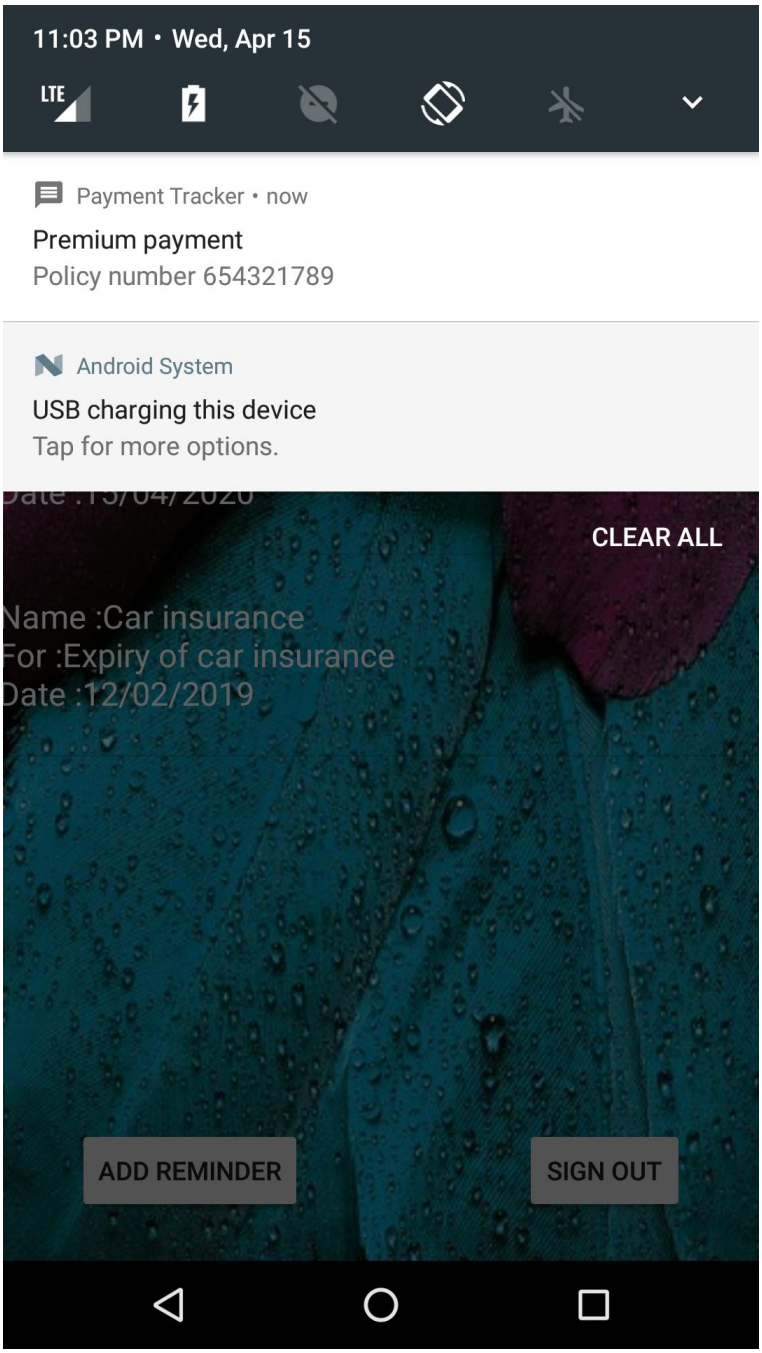
Add reminder activity:

It is a pop-up activity which takes name, description, date, interval at which the reminder has to repeat, and number of repetitions from the user in order to create new reminder.



Update/delete reminder activity:

It is a pop-up activity which displays a form when the user clicks on a reminder. The user can either save changes to the reminder by filling the form or delete the reminder itself.



Notification:

The image shows the notification at time specified by the user. It contains name and description of the reminder given by the user at the time of reminder creation.