BSCS-F19-017

03-134162-001 AAMNA MOHSIN

03-134162-046 NAWAL NASIR

Android Application for IoT based Telemetry

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: Muhammad Zunnurain Hussain

Department of Computer Sciences

Bahria University, Lahore Campus

July 2020

C e r t i f i c a t e

We accept the work contained in the report titled

"Android Application for IoT based Telemetry"

written by

AAMNA MOHSIN

NAWAL NASIR

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Muhammad Zunnurain Hussain

_____ (Signature)

July 20, 2020

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134162-001	AAMNA MOHSIN	
03-134162-046	NAWAL NASIR	

Date : July 20, 2020

Specially dedicated to
my family and friends
(AAMNA MOHSIN)
my family and friends
(NAWAL NASIR)

ACKNOWLEDGMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our research supervisor, Mr. Zunnurain Hussain for his invaluable advice, guidance, and his enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parents and friends who had helped and encouraged me.

AAMNA MOHSIN
NAWAL NASIR

Android Application for IoT based Telemetry

ABSTRACT

Within the past decades, technology significantly changes the daily routine task of an individual. This increases the work efficiency and comfort of mankind, the Internet of Things (IoT) is an emerging technology that is making our world smarter. In the modern age of automation, better living standers are introduced. Home Telemetry System (HTS) has been designed for mobile phones having the Android platform to automate micro-controller which controls many home appliances like lights, fans, bulbs, etc. This project presents the automated approach of controlling the devices in a household that could ease the tasks of using the traditional method of the switch.

Its focus on providing a mobile application in which a user keeps an eye on his billing estimation, reduce his energy consumption. HTS will help you to monitor and control your house even from far away using networking and IoT.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

HTS	Home Telemetry System
API	Application Programming Interface
UI	User Interface
IoT	Internet of Things

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Through IoT, almost every object of our daily life in a home can be connected to the Internet. IoT allows monitoring and controlling all of these connected objects regardless of time and location. The purpose of a smart home is to improve living standards, security and safety as well as save energy and resources. The smart home plays an important role in the development of society [1]. Wi-Fi technology capable solution has proved to be controlled remotely, provide home security and it is low cost as compared to the previous systems

The IoT based smart home has many benefits. Firstly, it is compatible with current household appliances. It doesn't mean discarding current technologies, but to collaborate with them to provide a better life. Secondly, it is scalable. Any new appliance complying with this architecture and protocols can be added to the system. The current appliance can be added to the system through interface [13].

IoT can be described as connecting everyday objects such as smartphones, Internet television, sensor, and actuators to the internet where the devices are smartly linked together to allow a new form of communication between people and themselves.

Home Telemetry one of the emerging technologies which are getting popular day by day with different variations. We introduce you to a system with a mobile application that helps people to monitor home and have a check on their budget consideration.

1.2 Problem Statements

Despite energy shortages in Pakistan, it is estimated that households waste 25% of their power because of efficient appliances in the country and a lack of mindfulness when using electricity. The long-standing issue of load shedding arises in Pakistan as well. Pakistan is currently facing up to 18 hours of electricity outage a day, especially in summers, is expected to face more if not dealt with in time. To overcome this at least in our home, we are developing an app that helps people to control their energy consumptions.

1.3 Aims and Objectives

The objectives of the project are shown as following:

i. To control the home appliances
ii. To tackle energy crises
iii. To monitor electric consumption

1.4 Scope of Project

Home Telemetry System (HTS) world a step forward to innovate living. It will help you to monitor and control your home from anywhere in the world. This project will focus on proving a mobile application to HTS. The Android platform to monitor and control an automated home (or workplace) using a web server, Database, and APIs to provide better security and cost management. The application will give UI and control to the user. The user will be able to monitor and control the house anywhere from the world. This application has the following features

- Bill estimation
- Peak hour notification
- Set and Notify Bill limit
- Units per hour
- Notify if voltage increase or decrease (or on/off automatically)

- Main supply switch control
- Register rooms and switches
- Push notification about shutdown and load shedding

CHAPTER 2

LITERATURE REVIEW

2.1 Overall Description

The description and product perspective of the Home telemetry system described in detail.

2.2 Other Non-Functional Requirements

The non-functional requirements of the home telemetry system are given below.

2.2.1 Safety Requirements

In the development stage, there are no security concerns. The reason is that there no storage of data right now.

2.2.2 Security Requirements

Users should be login to the system.

2.2.3 Software Quality Attributes

- **Reliability**

  The system can be used by more than one user at a time. Any user can access the system by using even a low-performance mobile phone.

- **Availability**
  The system is available all the time to a user.

- **Portability**
  Users can log on to the system anywhere at any time.

2.3 Software Requirements Chart

The following table has the software requirements.

Table 2-1: Software Requirements Chart

ID	Priority	Type	Source	Contained in use case	Description
HTS-R1	High	Functional	Place owner/admin	HTS-01-001	Authentication access
HTS-R2	Medium	Functional	Place owner/admin	HTS-01-002	To register a new user
HTS-R3	High	Functional	Place owner	HTS-01-003	User forgets password
HTS-R4	Medium	Functional	Place owner/admin	HTS-01-004	Remember password for future
HTS-R5	High	Functional	Place owner	HTS-02-001	Display all register switches with their status (on/off) of the room)
HTS-R6	Medium	Functional	Place owner/admin	HTS-02-002	Add a new room into an application.
HTS-R7	Medium	Functional	Place owner/admin	HTS-02-003	Add new switch into register room

HTS - R8	Medium	Functional	Place owner/admin	HTS-02-004	Change the current state of a switch
HTS-R9	High	Functional	Place owner/admin	HTS-03-001	Display number of units consumed
HTS-R10	High	Functional	Place owner/admin	HTS-03-002	Display expected bill for used units
HTS-R11	High	Functional	Place owner/admin	HTS-03-003	User can set the limit and able to know how he is exceeding from the set limit
HTS-R12	High	Functional	Place owner/admin	HTS-03-004	Display percentage of used electricity of set limit
HTS-R13	High	Functional	Place owner/admin	HTS-04-001	Display notification when electricity fluctuates
HTS-R14	High	Functional	Place owner/admin	HTS-04-002	Display notification when a shutdown or load shedding is scheduled

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Design

This chapter overviews the design of Drive It. The system architecture design gives the total perspective of the system. This will enable developers and clients to see and check the design plan in detail. Following artifacts incorporated in this Chapter.

1. Use case diagrams
2. Use case descriptions
3. Sequence diagrams
4. Collaborative Diagram
5. Domain Model
6. Entity Relationship Diagrams
7. Design Class Diagram
8. Operation Contract

3.2 Use Case Diagram

Use case diagram of the whole system given below.

3.2.1 System Use Case Diagram

The following Figure 1 shows the system Use-case diagram.

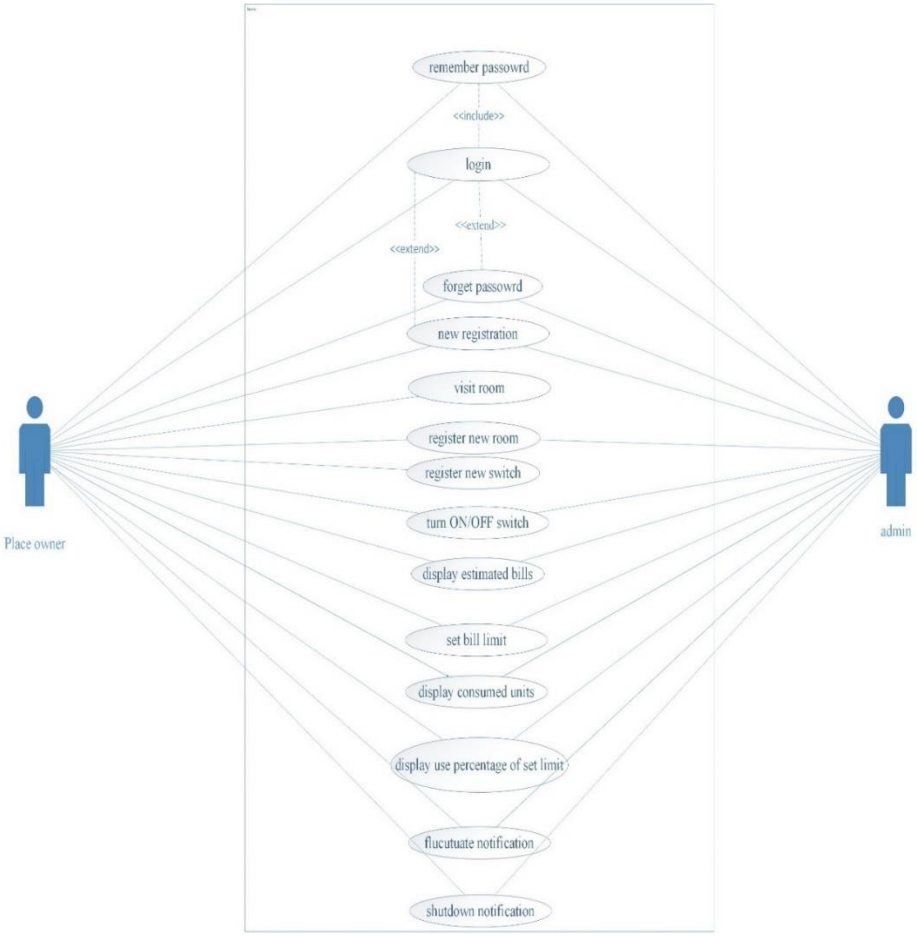


Figure 3-1: System Use Case Diagram

3.2.2 Login Use Case Diagram

The following Figure 2 shows the Login Use-case diagram.

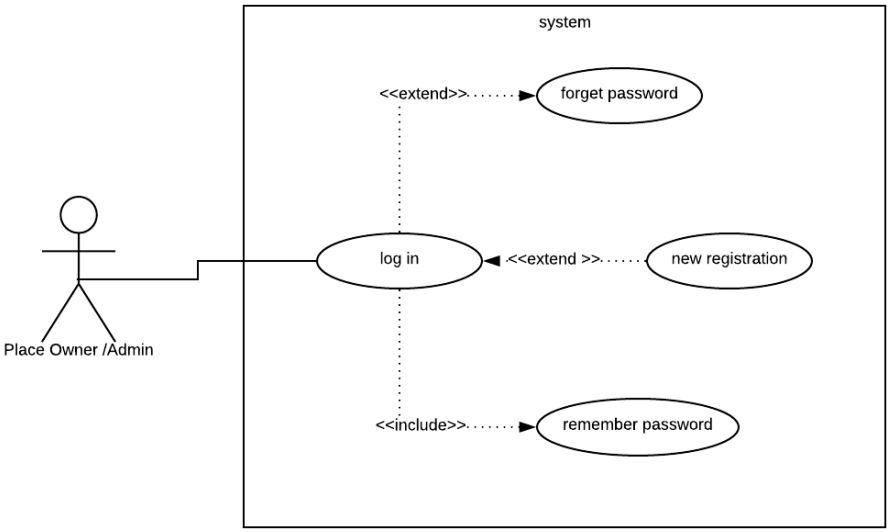


Figure 3-2: Login Use Case Diagram

3.2.3 Display and Set Bill Use Case Diagram

The following Figure 3 shows the Display and Set Bill Use-case diagram.

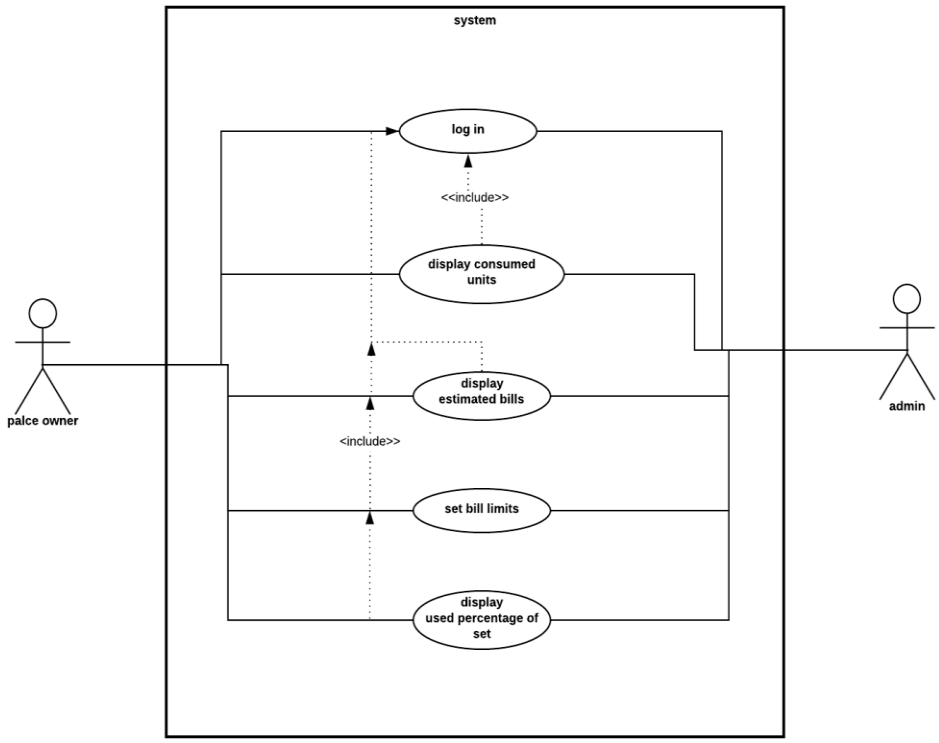


Figure 3-3: Display and Set Bill Use Case Diagram

3.2.4 New Entries Use Case Diagram

The following Figure 4 shows the New Entries Use-case diagram. Users can add a new switch and register a new room.

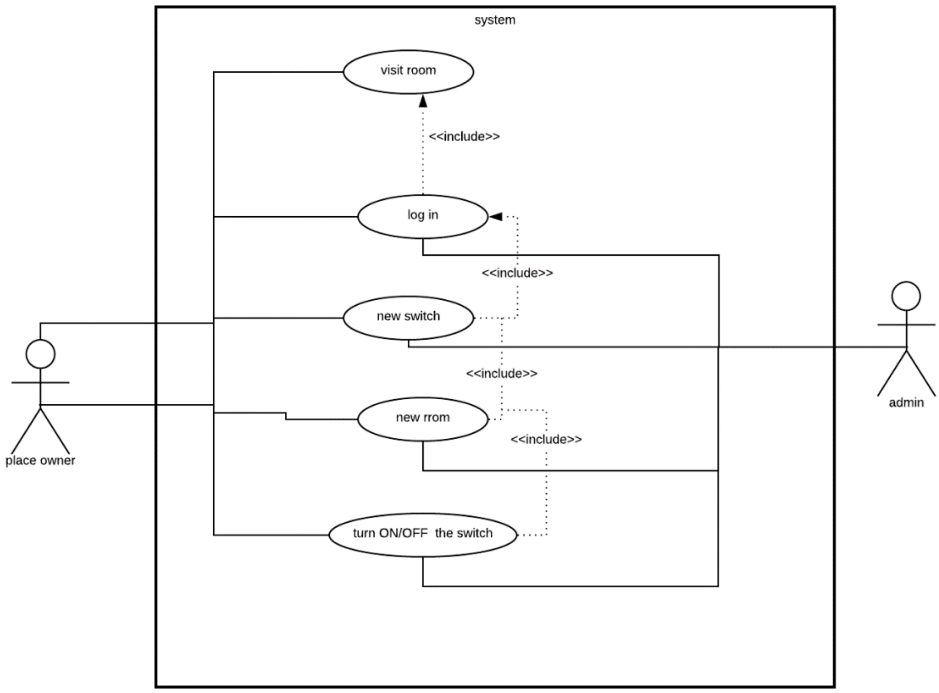


Figure 3-4: New Entries Use Case Diagram

3.3 Use Case Descriptions

A use-case description is a text that captures the detailed functionalities of a use-case. The description of all use-cases is written down in this section.

3.3.1 Log-In (HTS-01-001)

Place owner or admin sign in the application. After authentication user can log in to the application.

Table 3-1: Log-in Use Case

Name	Log-In
Unique Identifier	HTS-01-001
Objective	authenticate the access
Priority	High
Actors	Place owner / Admin
Basic Flow	1. Open the application. 2. Fill credentials. 3. Click the sign-in button.
Alternative Flow	Already login Register new user
Preconditions	User must register before login
Post Conditions	User successfully login
Notes/Issues	Internet may not be working

3.3.2 New Registration (HTS-01-002)

New registration is done by place owner or admin after login.

Table 3-2: New registration Use Case

Name	New Registration
Unique Identifier	HTS-01-002
Objective	To register a new user
Priority	Medium
Actors	Place owner / Admin
Basic Flow	1. Open the application. 2. Click Register new user button 3. Fill credentials.

	4. Click the register button.
Alternative Flow	User already registered
Preconditions	The app must be installed
Post Conditions	User successfully register
Notes/Issues	Already account exits on email

3.3.3 Forget Password (HTS-01-002)

In case the user forgets the password, he can reset.

Table 3-3: Forget Password Use Case

Name	Forget password
Unique Identifier	HTS -01-003
Objective	To reset password
Priority	High
Actors	Place owner
Basic Flow	1. Open the application. 2. Click to forget the password.
Alternative Flow	None
Preconditions	Enter valid email
Post Conditions	Enter code for verification
Notes/Issues	Internet connection

3.3.4 Remember Password (HTS-01-004)

If the user wants to remember password in application and does not want to enter password all time.

Table 3-4: Remember Password Use Case

Name	Remember password

Unique Identifier	HTS -01-004
Objective	Remember the password for the future.
Priority	Medium
Actors	Place owner
Basic Flow	1. Open the application. 2. Fill credentials. 3. Check the “remember password” checkbox. 4. And click the login button.
Alternative Flow	Log in without checking to remember me
Preconditions	None
Post Conditions	Successfully login
Notes/Issues	Credentials may not be correct

3.3.5 Visit Room (HTS-02-001)

Display all the register switches with their status (on/off) of the room.

Table 3-5: Visit Room Use Case

Name	Visit Room
Unique Identifier	HTS -02-001
Objective	Display all the register switches with their status
Priority	High
Actors	Place owner
Basic Flow	1. Open application 2. Click on a relevant room
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in.
Post Conditions	Display all switches and their status
Notes/Issues	The room may not load due to a connection issue.

3.3.6 Register New Room (HTS-02-002)

Users can add a new room.

Table 3-6: Register New Room Use Case

Name	Register New Room
Unique Identifier	HTS -02-02
Objective	Add a new room into an application.
Priority	Medium
Actors	Place owner/ admin
Basic Flow	1. Open application. 2. Select add room 3. Enter name.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in.
Post Conditions	A new room will be registered.
Notes/Issues	No, an issue for this use case.

3.3.7 Register New Switch (HTS-02-003)

User can add a new switch in a registered room

Table 3-7: Register New Switch Use Case

Name	Register new switch
Unique Identifier	HTS -02-003
Objective	Add new switch in a registered room
Priority	Medium
Actors	Place owner/ admin

Basic Flow	1. Open application. 2. Open the room where the new switch to be added. 3. Select add switch. 4. Enter the name and IP address of the switch.
Alternative Flow	No, alternative flow for this use case.
Preconditions	The room should be registered.
Post Conditions	The switch will successfully register and will display.
Notes/Issues	Connectivity error

3.3.8 Turn on/off a Switch (HTS-02-004)

The user can change the status of the appliance that is he can turn on/off switch.

Table 3-8: Turn on/off a Switch Use Case

Name	Turn on/off a Switch
Unique Identifier	HTS-02-004
Objective	Change the current state of a switch.
Priority	Medium
Actors	Place owner/admin
Basic Flow	1. Open the app. 2. Click the room 3. Click the toggle button of the switch.
Alternative Flow	No, alternative flow for this use case.
Preconditions	The switch should be registered before with a valid IP address.
Post Conditions	The state of the switch will change.
Notes/Issues	Connectivity issue

3.3.9 Display consumed Units (HTS-03-001)

Users can have all information about consumed units. This use case displays information about the consumed units.

Table 3-9: Display Consumed Units Use Case

Name	Display consumed Units
Unique Identifier	HTS-03-001
Objective	Display the number of units consumed.
Priority	High
Actors	Place owner/admin
Basic Flow	1. Open application. 2. Select statics from options.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in with the valid electric meter
Post Conditions	Display consumed units.
Notes/Issues	Connectivity issue.

3.3.10 Display Estimated Bill (HTS-03-002)

Display the expected bill to the user according to the bill set limit.

Table 3-10: Display Estimated Bill Use Case

Name	Display estimated bill
Unique Identifier	HTS-03-002
Objective	Display the expected bill for used units.
Priority	High
Actors	Place owner/admin
Basic Flow	1. Open application. 2. Select statics from options.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in with the valid electric meter

Post Conditions	Display the expected bill.
Notes/Issues	Connectivity issue.

3.3.11 Set Bill Limit (HTS-03-003)

The user can set the bill limit.

Table 3-11: Set Bill Limit Use Case

Name	Set Bill Limit
Unique Identifier	HTS-03-003
Objective	The user can set Limits and able to how he is exceeding from the set limit.
Priority	High
Actors	Place owner/admin
Basic Flow	1. Open application. 2. Select statics from options. 3. Click Set limit.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in with the valid electric meter
Post Conditions	Display the expected bill.
Notes/Issues	Connectivity issue.

3.3.12 Display use Percentage of Set limit (HTS-03-004)

This use case displays the use percentage of the set bill limit.

Table 3-12: Display Use Percentage of Set limit Use Case

Name	Display use percentage of set limit
Unique Identifier	HTS-03-004
Objective	Display the percentage of used electricity of set limits.
Priority	High
Actors	Place owner/admin
Basic Flow	1. Open application. 2. Select statics from options.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Set a limit and logged in with the valid electric meter
Post Conditions	Display the expected bill with the percentage of used electricity for the budget.
Notes/Issues	Connectivity issue.

3.3.13 Electricity Fluctuate Notification (HTS-04-001)

This use case notifies the user about the electricity fluctuations. Users can measure precautions accordingly.

Table 3-13: Electricity Fluctuate Notification Use Case

Name	Electricity Fluctuation Notification.
Unique Identifier	HTS-04-001
Objective	Display notification when electricity fluctuates.
Priority	High
Actors	Place owner/admin
Basic Flow	1. When electricity fluctuates a notification will push into notification bar.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in with the valid electric meter
Post Conditions	Display a notification on the notification bar.
Notes/Issues	Connectivity issue.

3.3.14 Load shedding and Shutdown Notification (HTS-04-002)

This use case notifies the user about the load shedding and also gives notification whenever shutdown is going to happen.

Table 3-14: Load shedding and Shutdown Notification Use Case

Name	Load shedding and Shutdown Notification.
Unique Identifier	HTS-04-002
Objective	Display notification when a shutdown or load shading is a schedule.
Priority	High
Actors	Place owner/admin
Basic Flow	1. When electricity fluctuates a notification will push into notification bar.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in with the valid electric meter
Post Conditions	Display a notification on the notification bar.
Notes/Issues	Connectivity issue.

3.3.15 Peak hours Notification (HTS-04-003)

When peak hours started this use case gives the notification to the user.

Table 3-15: Peak Hour Notification Use Case

Name	Peak hours Notification.
Unique Identifier	HTS-04-003
Objective	Display notification when peak starts and ends.
Priority	High
Actors	Place owner
Basic Flow	1. When peak hour starts or ends a notification will push into the notification bar.
Alternative Flow	No, alternative flow for this use case.
Preconditions	Logged in with the valid electric meter
Post Conditions	Display a notification on the notification bar.
Notes/Issues	Connectivity issue.

3.4 Sequence Diagrams

Following are the Sequence diagrams

3.4.1 Sequence Diagrams of Login

Sequence diagram of login. If the user enters valid details he can log in to the application.

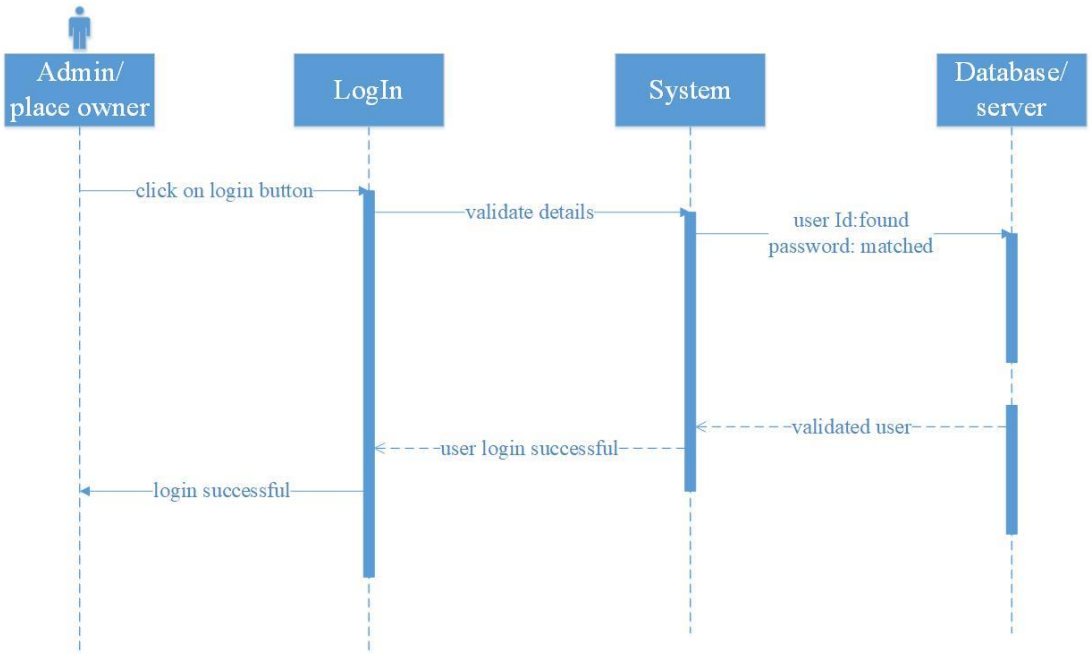


Figure 3-5: Log-in Sequence Diagram

3.4.2 Sequence Diagrams of New Registration

The following diagram shows the sequence of registration of a new user.

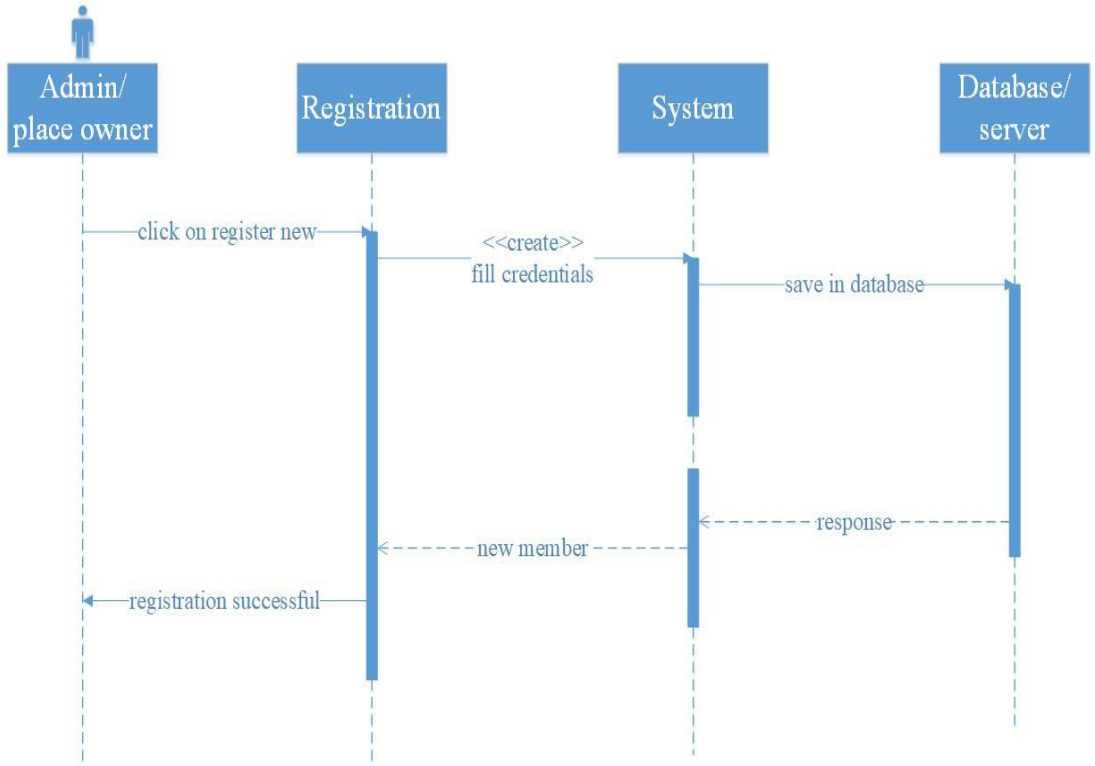


Figure 3-6: New Registration Sequence Diagram

3.4.3 Sequence Diagrams of Forget Password

The following sequence diagram shows the sequence of forgetting the password. If the user forgets his password, he can reset it.

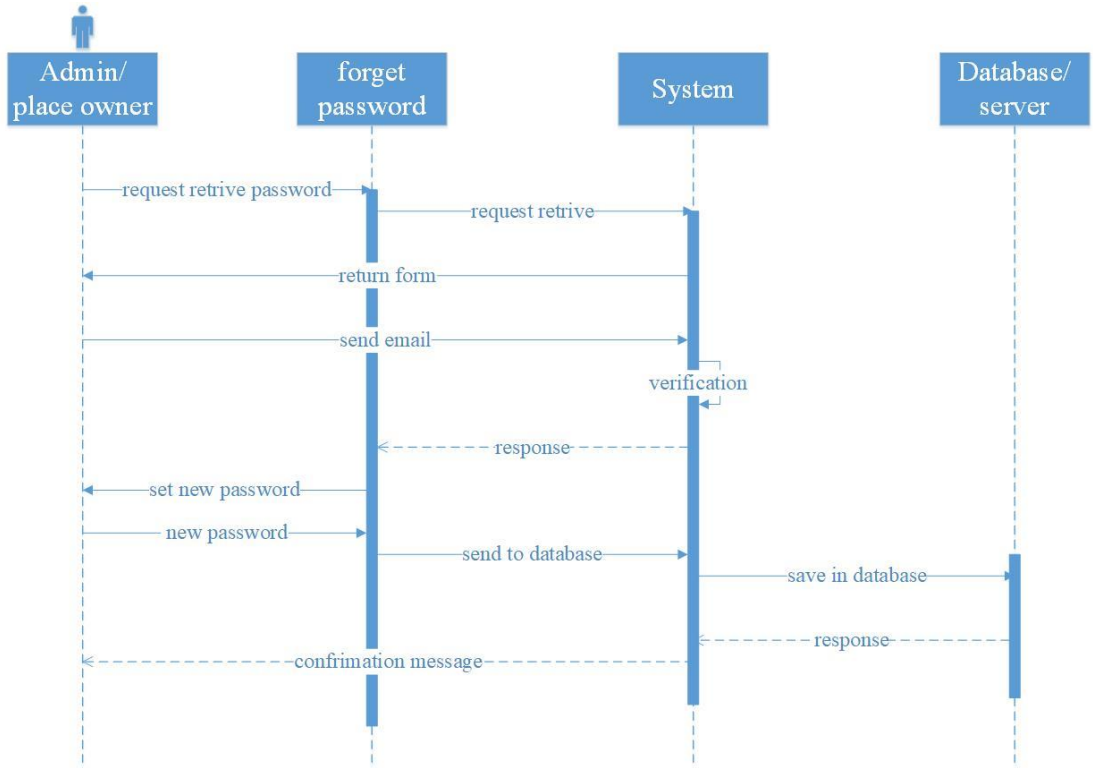


Figure 3-7: Forget Password Sequence Diagram

3.4.4 Sequence Diagrams of Registration New Room

To add a new room in the application, the diagram shows the sequence to register the new room.

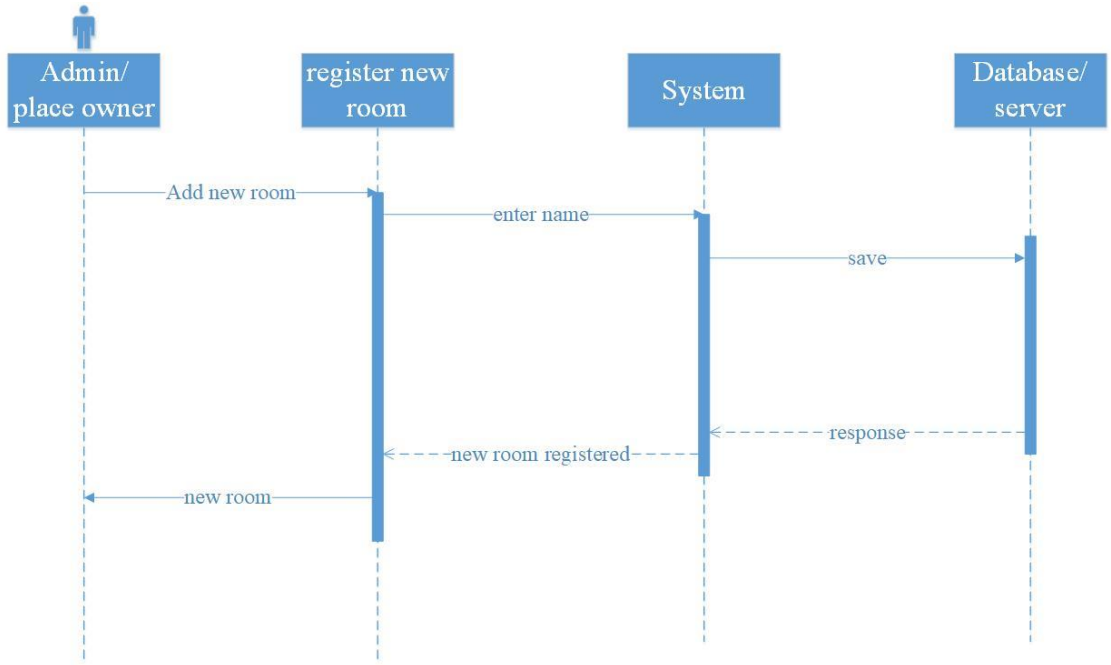


Figure 3-8: Registration New Room Use Case

3.4.5 Sequence Diagrams of Register New Switch

To add a new switch in the room, the diagram shows the sequence to register the new switch in the room.

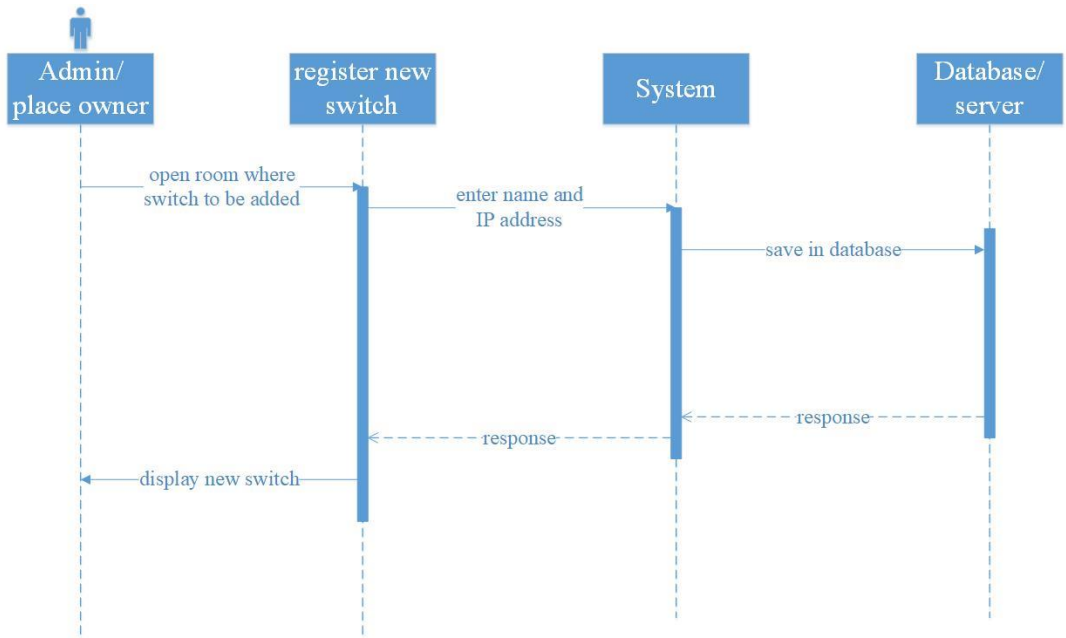


Figure 3-9: Register New Switch Use Case

3.4.6 Sequence Diagrams of System

The following diagram shows the sequence of the home telemetry system.

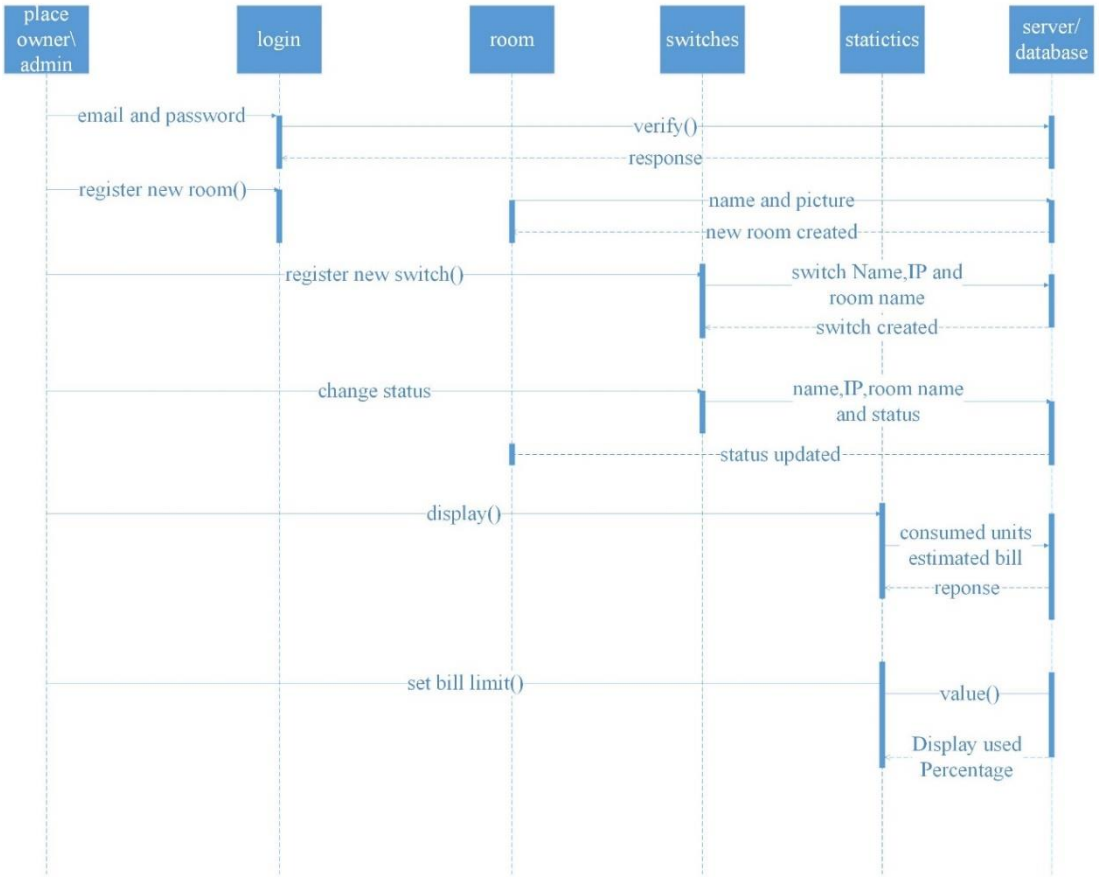


Figure 3-10: System Sequence Diagram

3.5 Domain Model

Following is the Domain Model of the project. A conceptual model of the domain that incorporates both behaviour and data.

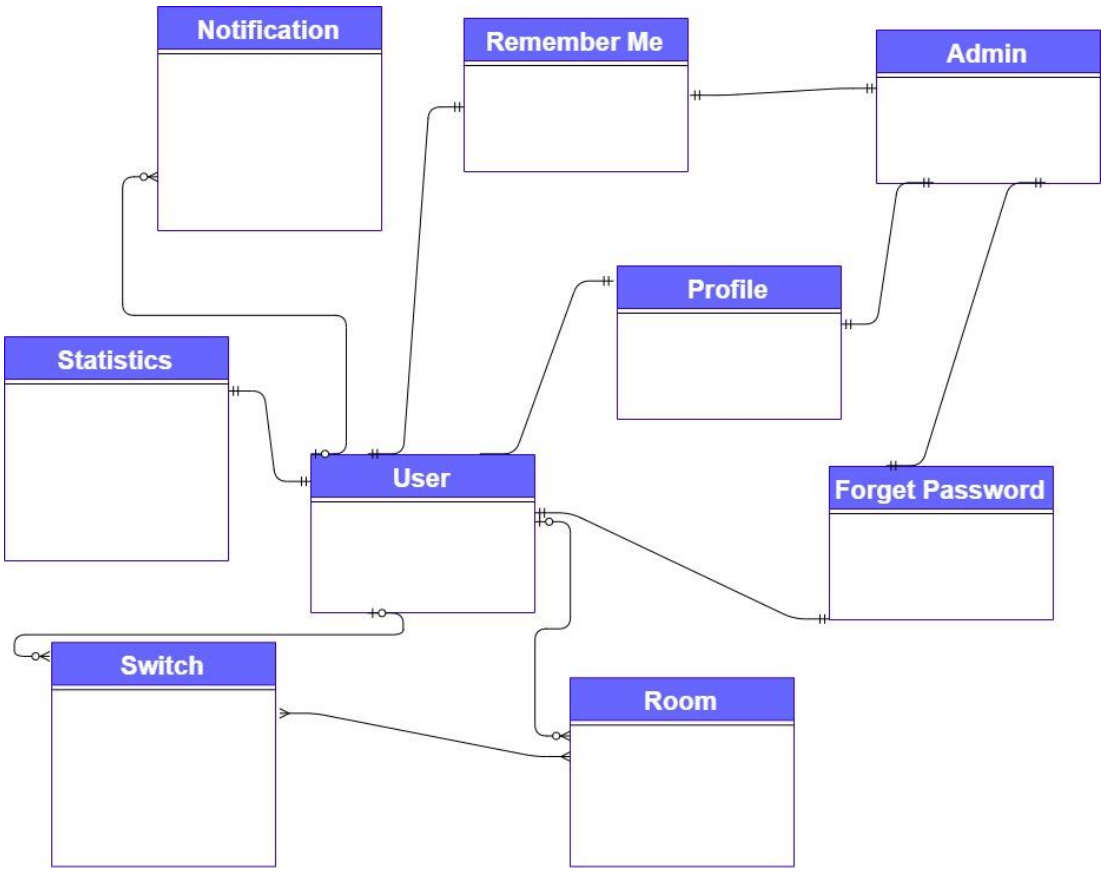


Figure 3-11: Domain Model of HTS

3.6 Collaboration Diagram

The communication diagram illustrates the relationships and interactions among objects.

3.6.1 Collaboration diagram of Login

Following the diagram shows the collaboration between the admin system to log in to the system.

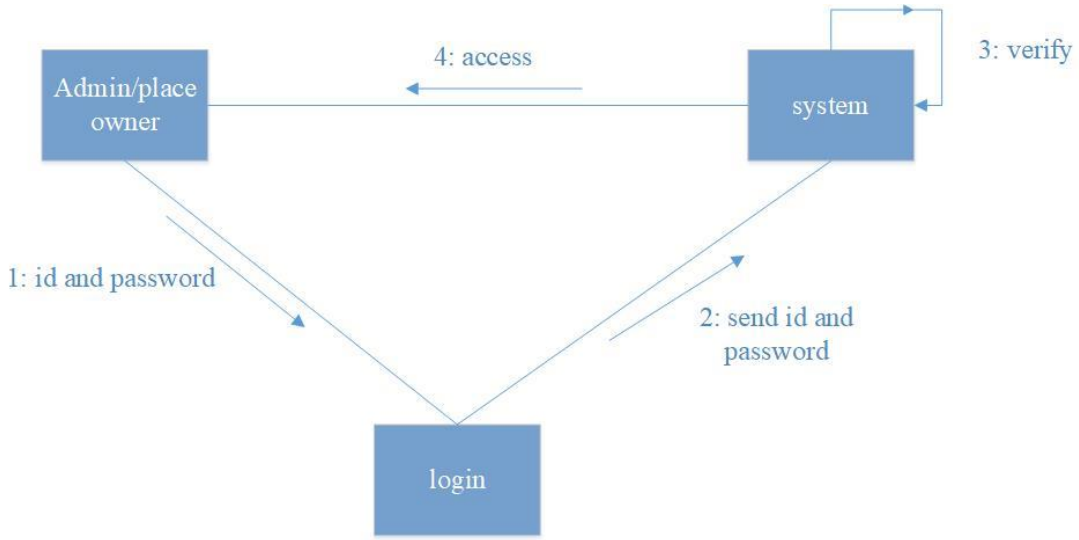


Figure 3-12: Collaboration diagram of Login

3.6.2 Collaboration diagram of New Registration

Following the diagram shows the collaboration between the admin system to register a new member into the system.

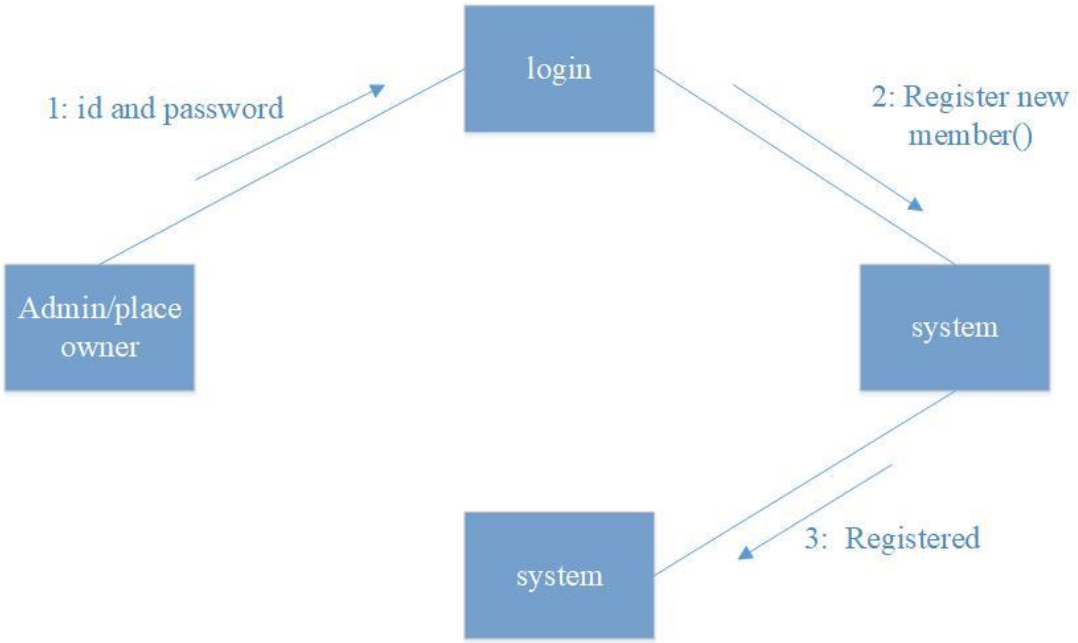


Figure 3-13: Collaboration diagram of New Registration

3.6.3 Collaboration diagram of Forget Password

Following the diagram shows the collaboration between the admin system to reset the password if the user forgets his password.

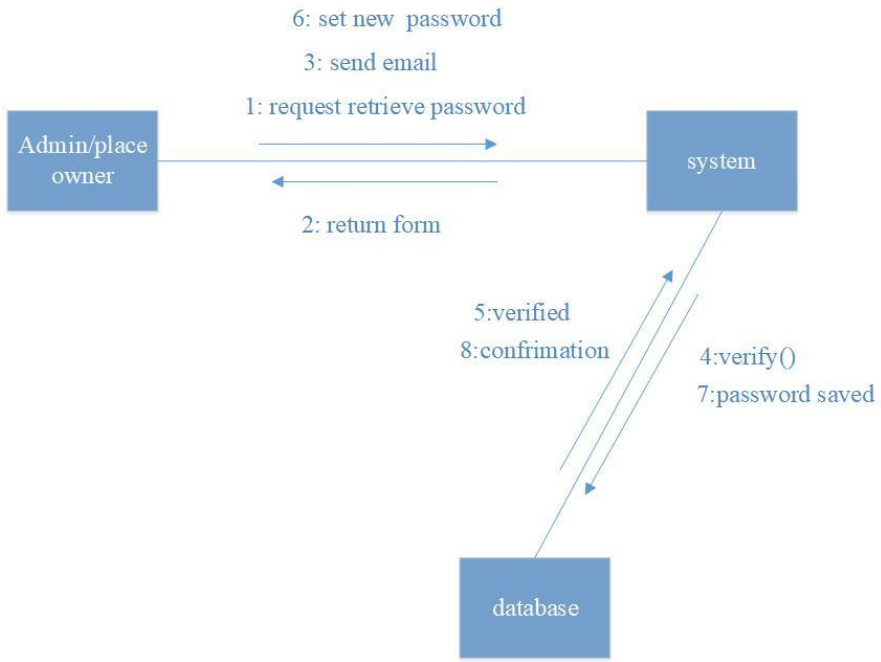


Figure 3-14: Collaboration diagram of Forget Password

3.6.4 Collaboration diagram of Register New Room

Following the diagram shows the collaboration between the admin system to register a new room into the application.

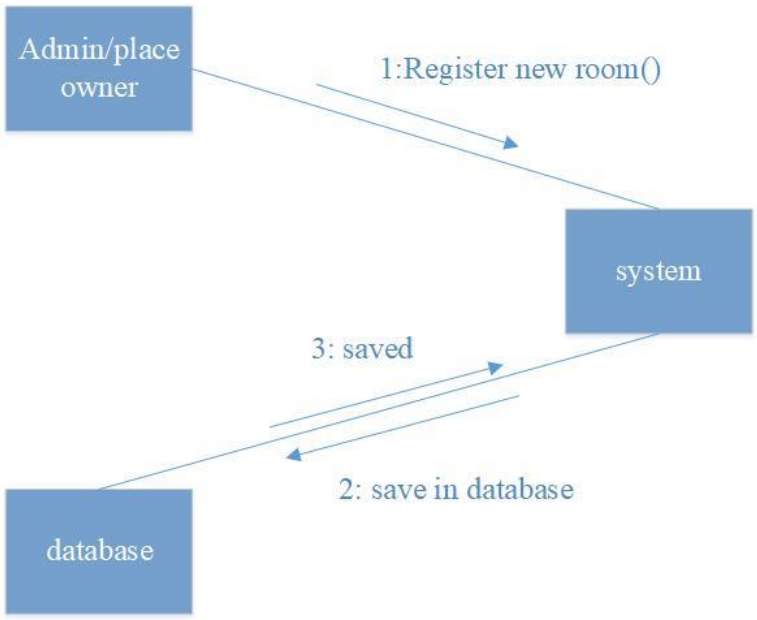


Figure 3-15: Collaboration diagram of Register New Room

3.6.5 Collaboration diagram of Register New Switch

Following the diagram shows the collaboration between the admin system to register a new switch into the room.

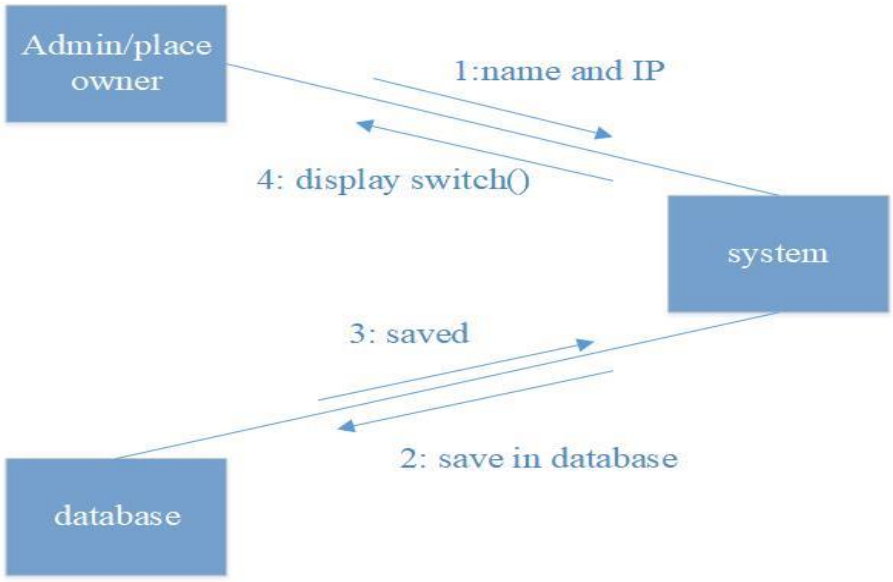


Figure 3-16: Collaboration diagram of Register New Switch

3.7 Class Diagram

The class diagram of the Home telemetry system showing the system's classes, their attributes, operations, and the relationships among objects.

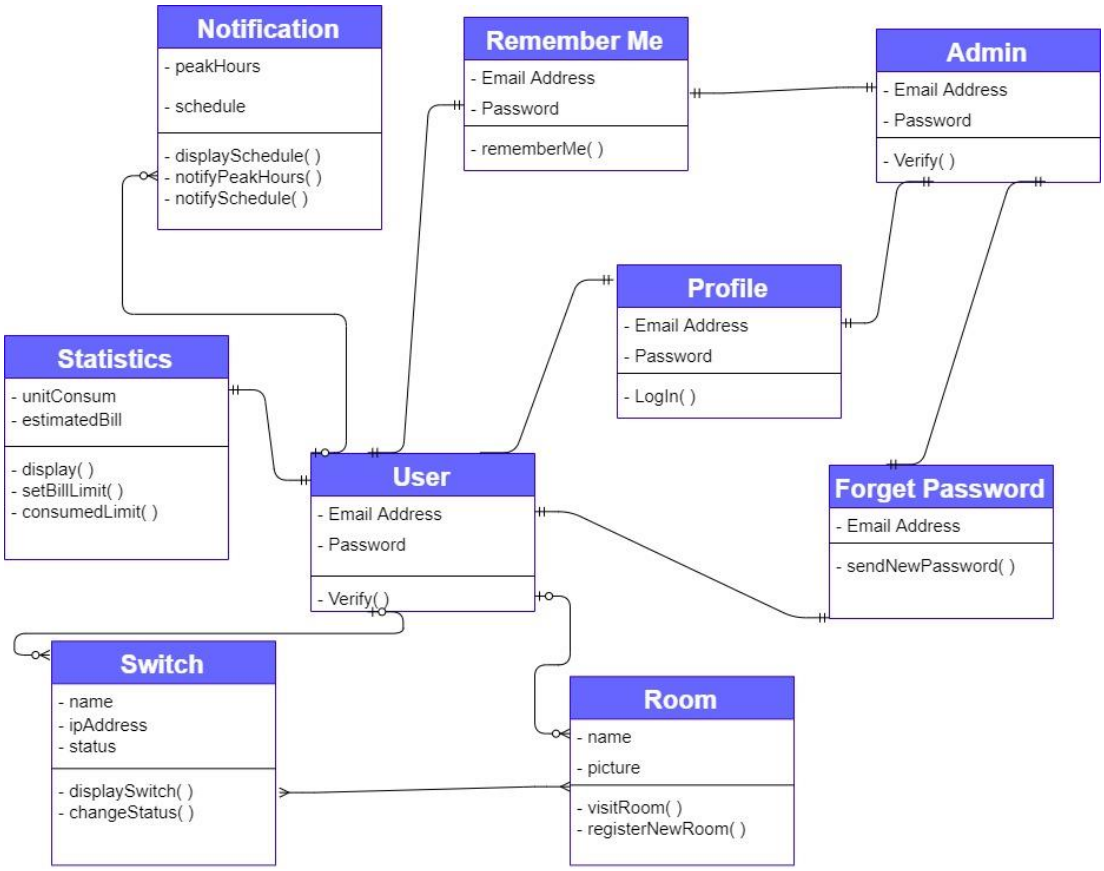


Figure 3-17: Class Diagram of HTS

3.8 Data Model

The relational model of HTS shows the relationship between entities and produce database design to use in database creation, management, and maintenance. An ER model of HTS also provides a means for communication

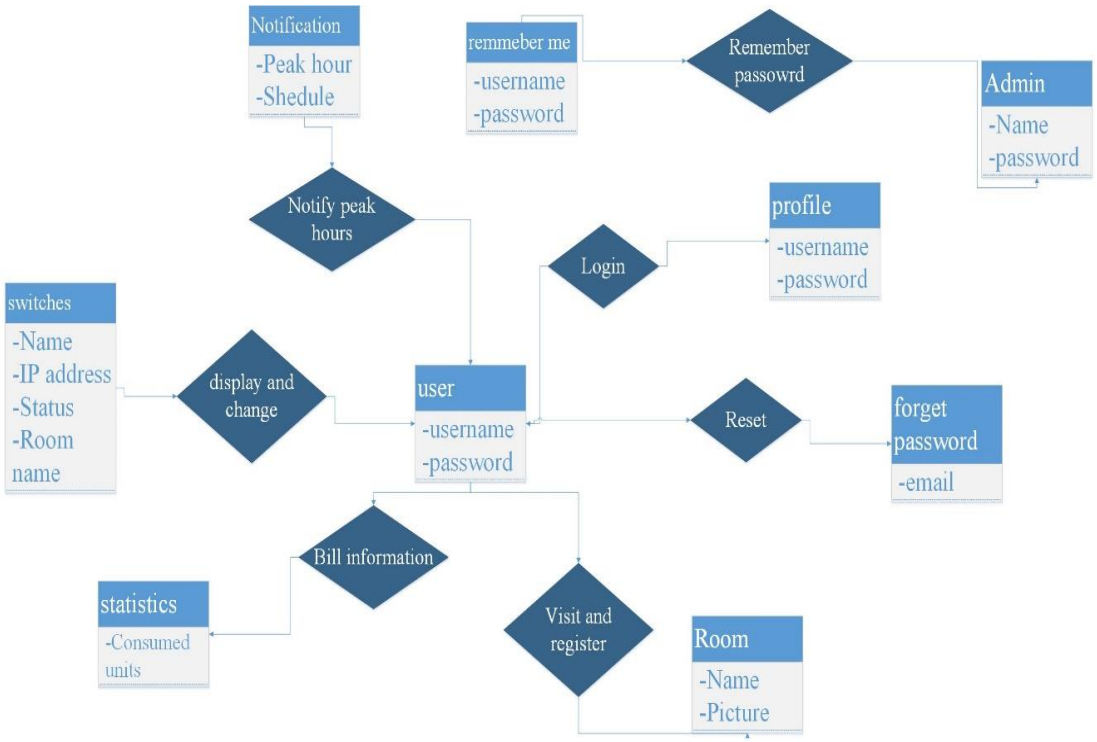


Figure 3-18: Data Model of HTS

3.9 Operation Contracts

The contract identifies system state changes when an HTS operation happens. An operation is taken from a system sequence diagram. A domain model of HTS can be used to help generate an operation contract.

3.9.1 Contract CO1: login to the application

Operation: login()

Cross References: use case - login

Preconditions: User must register before login

Postconditions: -user provide the correct credentials
-logged into the application

3.9.2 Contract CO2: reset the password

Operation: sendnewPassword()

Cross References: use case – forget password

Preconditions: user should reset password through a valid email

Postconditions: -user had set new password
-logged into the application

3.9.3 Contract CO3: consumption of units

Operation: consumedLimit()

Cross References: use case – display consumed units

Preconditions: Logged in with the valid electric meter

Postconditions: -user had set bill limit
-user had received notifications through app

3.9.4 Contract CO4: set bill limit

Operation: SetBillLimit(()

Cross References: use case – set bill limit

Preconditions: Logged in with the valid electric meter

Postconditions: -user had set bill limit and peak hours
-user had received notifications through application
-displayed expected bills

3.9.5 Contract CO5: change status

Operation: changeStatus()

Cross References: use case – turn ON/OFF a switch

Preconditions: switch should be registered with a valid IP address

Postconditions: **-the** user had to change the status of a switch
-status of switch changed ON to OFF and vice versa

3.10 Methodology

The methodology used in this project regarding software engineering is “Agile Scrum methodology”. The agile scrum methodology for mobile apps is a type of agile development methodology in which scrum is an agile structure that breaks the process of app development into smaller chunks or sprints. Each chunk is called scrum. By working in short sprints, this iterative cycle can be repeated until enough work items have been completed.

Agile Scrum methodology has several benefits. First, it encourages products to be built faster, since each set of goals must be completed within each sprint's time frame. It also requires frequent planning and goal setting, which helps the scrum team focus on the current sprint's objectives and increase productivity.

The main activity in Scrum project management is the Sprint, a time-boxed iteration that usually lasts between 1-4 weeks, with the most common sprint length being 2 weeks.

- **Sprint Planning Meeting:**

  At the start of each sprint, a planning meeting is held to discuss the work that is to be done.

- **Daily scrum or daily stand-up:**

  Each day during the sprint team members share what they worked on the prior day, will work on today, and identify any impediments. These meetings are time-boxed to no more than 15 minutes.

- **Sprint Review:**

  At the end of a sprint, the team demonstrates the functionality added during the sprint.

- **Sprint Retrospective:**

At the end of each sprint, the team participates in a retrospective meeting to reflect on the sprint that is ending and identify opportunities to improve in the newsprint.

SCRUM and there are a number of other rules and procedures that are followed [19].

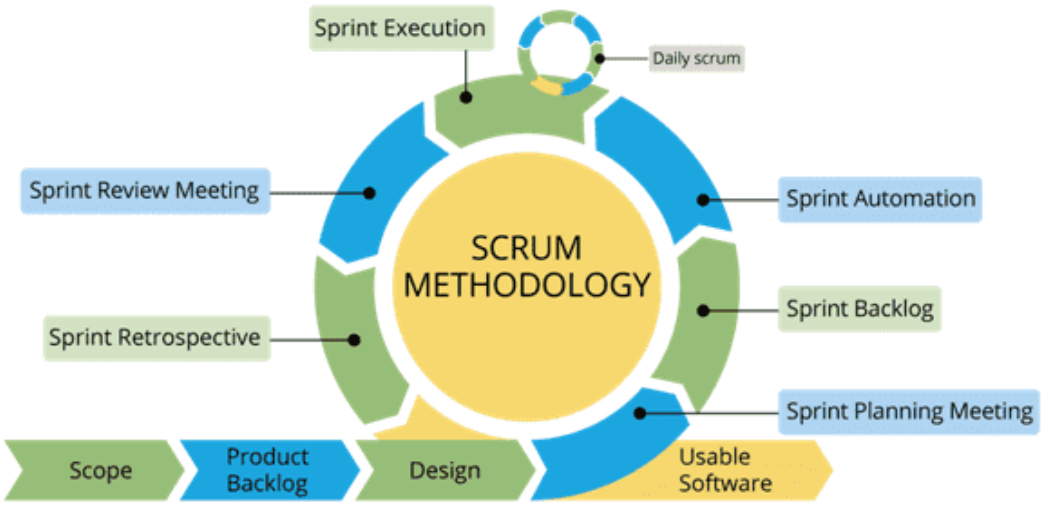


Figure 3-19: Scrum Methodology [19]

CHAPTER 4

DATA AND EXPERIMENTS

4.1 Languages used for Implementation

The following are the details of Language to develop the home telemetry.

4.1.1 JAVA

Android applications are developed using the Java language. Android itself is built on Java, there are plenty of Java libraries to our aid. Java apps are lighter and more compact. Java yields a faster build process. It's easy to learn and understand. It's designed to be platform-independent and secure, using virtual machines and object-oriented

4.2 Tools for Implementation

The following are the details of the tool that is used to develop the home telemetry.

4.2.1 Android Studio

Android Studio is an integrated development environment designed specifically for Android development.

- We use Android Studio 3.2 and a system will at least with 8 GB ram and i5 processor.
- The testing device with minimum API Level 25.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 User Manual

- The user firstly has to sign in the account.
- Login the account to access the home appliance and control the functionalities.
- Users can also reset the password by using forget functionality.
- Users can save passwords by using remember me functionality, don't need to enter the password every time he opens the application.
- The home screen will appear after the login screen.
- Home Screen consists of
    - Rooms
        - Users can see all registered rooms in this window and can add a new room and rename it.
        - On room screen, there are all registered rooms.
        - The user can add more room and rename according to his need.
        - Users can add switches in rooms.

    - Statics

      It will display the estimated bill and consumed units also display the percentage of set bill limit and all the adjustments regarding the bill.
        - Notify the electric fluctuation
        - Load shedding and shutdown notification
        - Notify about the peak hours
    - Schedule

This window displays the load shedding schedules

- Profile
    - User can update his email and username.
    - He also updates his display picture.
    - He can change or update his password.

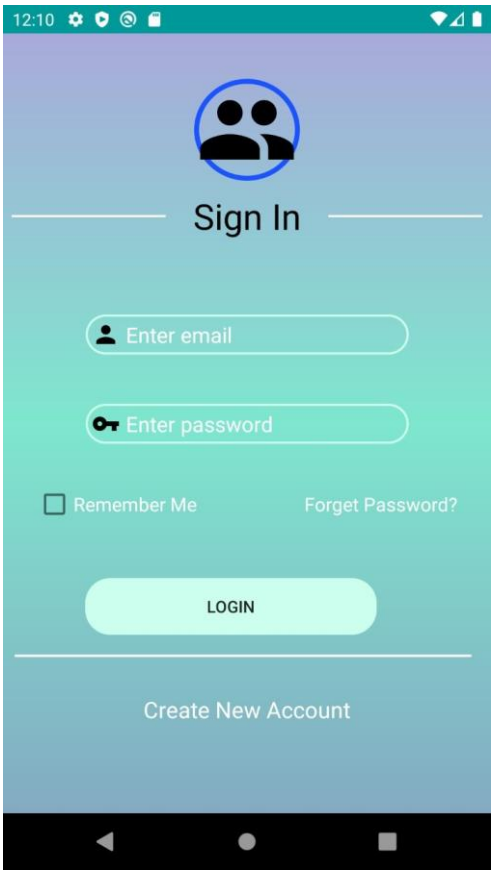


Figure 5-1: Sign-in Screen

Users can sign in an application through email and password.

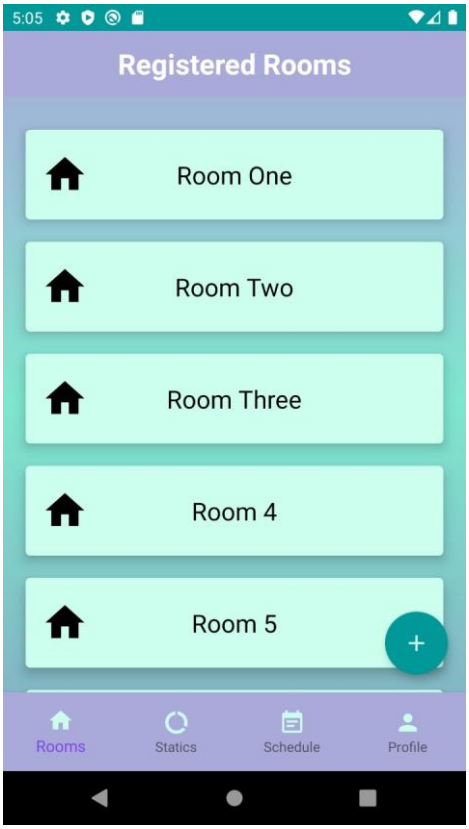


Figure 5-2: Registered Room Screen

Users can see all registered rooms here visit and can add new room.

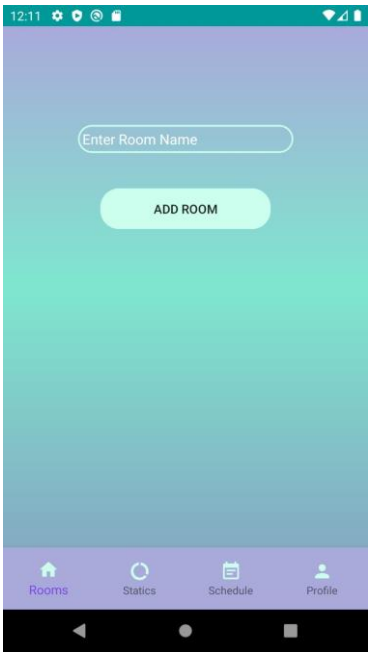


Figure 5-3: Add Room Screen

Users can add a new room from this screen.

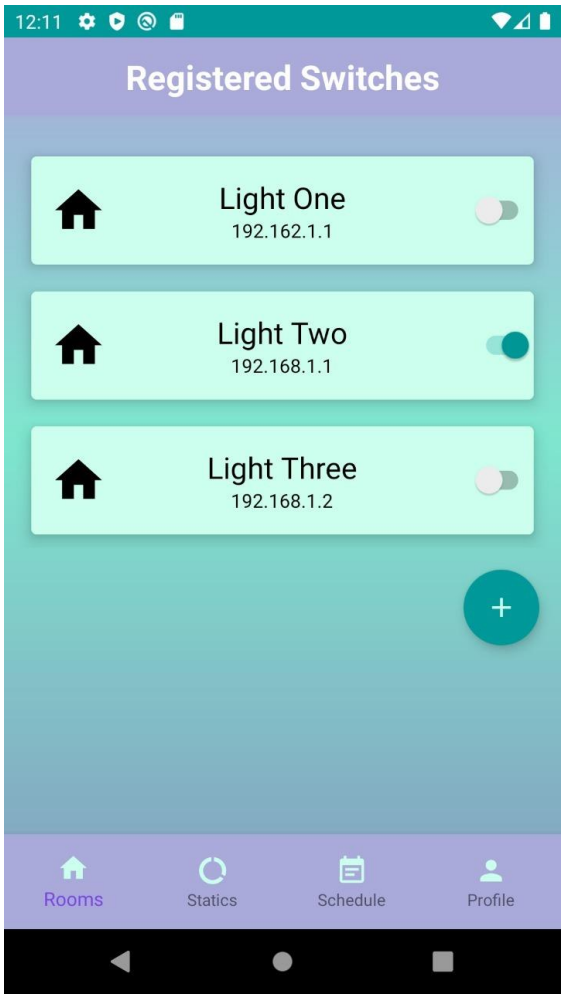


Figure 5-4: Registered Switches Screen

Users can control the switch (fan, light, bulb, etc.) from here. Turn on/off when need.

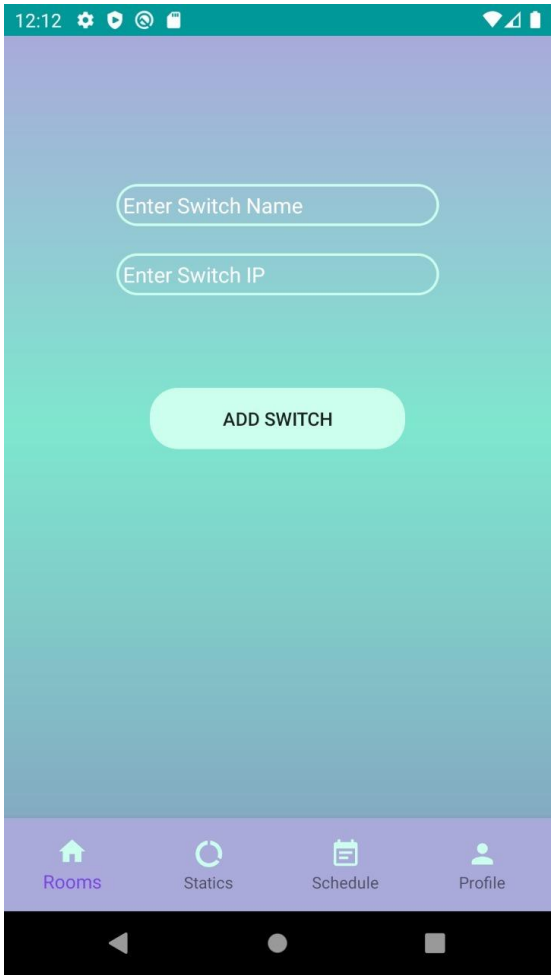


Figure 5-5: Add Switch Screen

Users can add a switch.

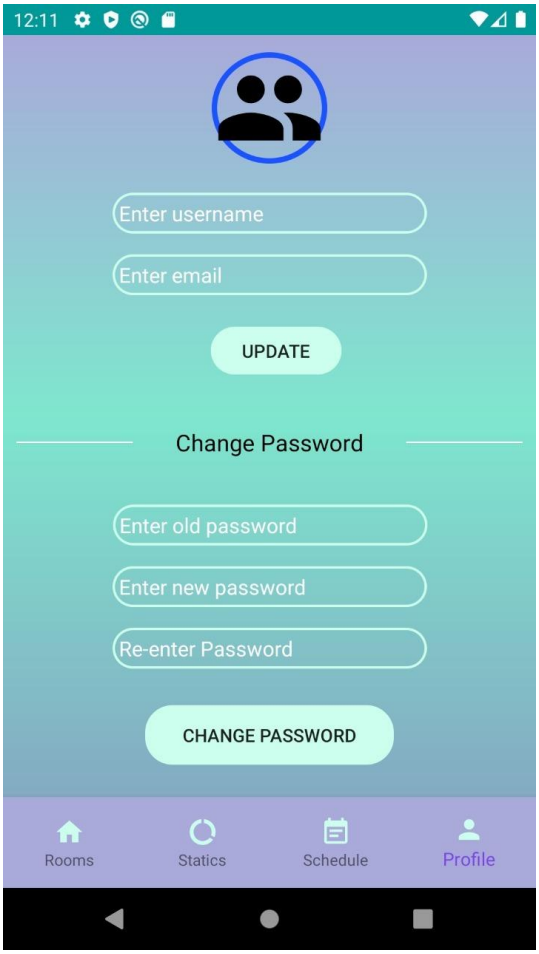


Figure 5-6: User Profile Screen

User can update his profile information. He can also update his password.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

Home telemetry is a resource that can make an automated environment people can set up controlling action through smartphones. HTS using the Internet of things work satisfactorily by successfully controlled appliances, monitor and control unnecessary energy consumption remotely through the internet.

The project illustrates the way of monitoring and tracking through IP addresses of all switches. This project will use the Android platform to monitor and control an automated home using a web server, Database, and APIs to provide better security and cost management. The user can easily touch on the screen of the phone to control the home appliances. A friendly interface that not only for the younger and educated consumers but people for every age. They can run the household more smoothly.

This application is beneficial for people to save the household's waste of electricity. In the future, this product has the potential for marketing because it has a solution to the current problem to reduce overall cost and energy consumption. It will help the user to analyse the condition of various parameters of home anywhere anytime. We are so keen to help the community through this.

6.2 Recommendations

To better suits customers need, manufacturers are creating innovative products., like, window blinds, coffeemakers, etc. and turned into automated devices to capture information from the environment.

- A lot of sensors available that can be used to control the home.
- Also, mobile phone-based inputs can be implemented in the future.
- The technologies may also increase the real state value of the home.
- Implementation of voice commands that may increase the comfort level.
- Improves accessibility to appliances through a natural interface, i.e., human voice [17].

REFERENCES

Conference paper:

[1] T.Malche, “System,” pp. 65–70, 2017.

[2] P. S. Nagendra Reddy, K. T. Kumar Reddy, P. A. Kumar Reddy, G. N. Kodanda Ramaiah, and S. N. Kishor, “An IoT based home automation using an android application,” *Int. Conf. Signal Process. Commun. Power Embed. Syst. SCOPES 2016 - Proc.*, pp. 285–290, 2017.

[3] S. A. I. Quadri and P. Sathish, “IoT based home automation and surveillance system,” *Proc. 2017 Int. Conf. Intell. Comput. Control Syst. ICICCS 2017*, vol. 2018-January, pp. 861–866, 2017.

[4] D. Pavithra and R. Balakrishnan, “IoT based monitoring and control system for home automation,” *Glob. Conf. Commun. Technol. GCCT 2015*, no. Gcct, pp. 169–173, 2015.

[5] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, “A survey based on Smart Homes system using Internet-of-Things,” *4th IEEE Spons. Int. Conf. Comput. Power, Energy, Inf. Commun. ICCPEIC 2015*, pp. 330–335, 2015.

[6] Kumar, Shiu. "Ubiquitous smart home system using android application." arXiv preprint arXiv:1402.2114 (2014).

[7] Mowad, Mohamed Abd El-Latif, Ahmed Fathy, and Ahmed Hafez. "Smart home automated control system using android application and microcontroller." *International Journal of Scientific & Engineering Research* 5.5 (2014): 935-939.

[8] Piyare, Rajeev. "Internet of things: ubiquitous home control and monitoring system using an android based smartphone." *International Journal of Internet of Things* 2.1 (2013): 5-11.

[9] Gunge, Vaishnavi S., and Pratibha S. Yalagi. "Smart home automation: a literature review." *International Journal of Computer Applications* 975 (2016): 8887.

[10] Panth, Sharon, and Mahesh Jivani. "Home automation system (HAS) using android for mobile phone." *International Journal of Electronics and Computer Science Engineering (IJECSE)* 3.1 (2013): 1-11.

[11] Hao, Shuai, et al. "Estimating mobile application energy consumption using program analysis." *2013 35th international conference on software engineering (ICSE)*. IEEE, 2013.

[12] Tsou, Yu-Ping, et al. "Building a remote supervisory control network system for smart home applications." *2006 IEEE International Conference on Systems, Man and Cybernetics*. Vol. 3. IEEE, 2006.

[13] Jie, Yin, et al. "Smart home system based on iot technologies." *2013 International Conference on Computational and Information Sciences*. IEEE, 2013.

[14]. Soliman, Moataz, et al. "Smart home: Integrating internet of things with web services and cloud computing." *2013 IEEE 5th international conference on cloud computing technology and science*. Vol. 2. IEEE, 2013.

[15]. Khan, Murad, Bhagya Nathali Silva, and Kijun Han. "Internet of things based energy aware smart home control system." *Ieee Access* 4 (2016): 7556-7566.

[16]. Asadullah, Muhammad, and Khalil Ullah. "Smart home automation system using Bluetooth technology." *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*. IEEE, 2017.

[17]. Mittal, Yash, et al. "A voice-controlled multi-functional Smart Home Automation System." *2015 Annual IEEE India Conference (INDICON)*. IEEE, 2015.

[18]. Bhide, Vishwajeet Hari, and Sanjeev Wagh. "i-learning IoT: An intelligent self learning system for home automation using IoT." *2015 International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 2015.

[19] z. zahra, “SCRUM Methodology,” 2017. [Online]. Available: https://zaynabzahrablog.wordpress.com/2017/10/07/scrum-methodology/.

APPENDICES

APPENDIX A: Computer Programme Listing

This is the code of registration button on click functionality.

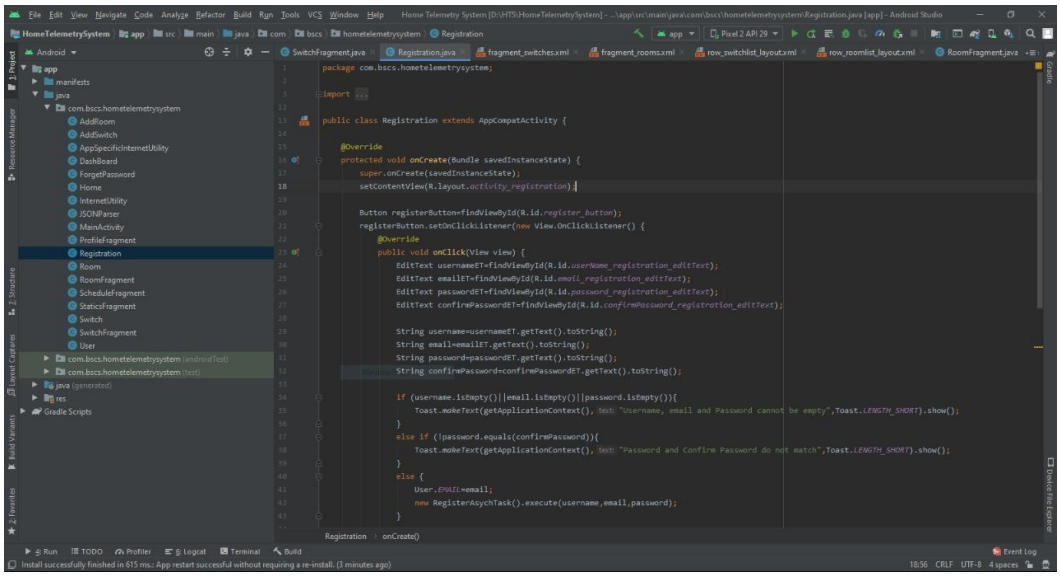


Figure A.1: Registration

This code doing all the background task of registration (i.e. communicating server)

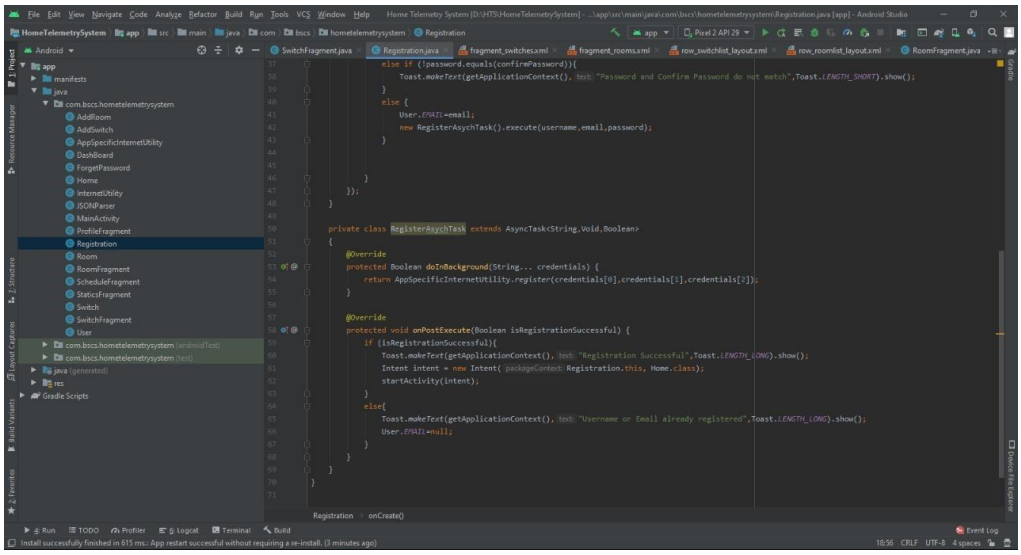


Figure A.2: Registration panel

These are the initialization of main activity.

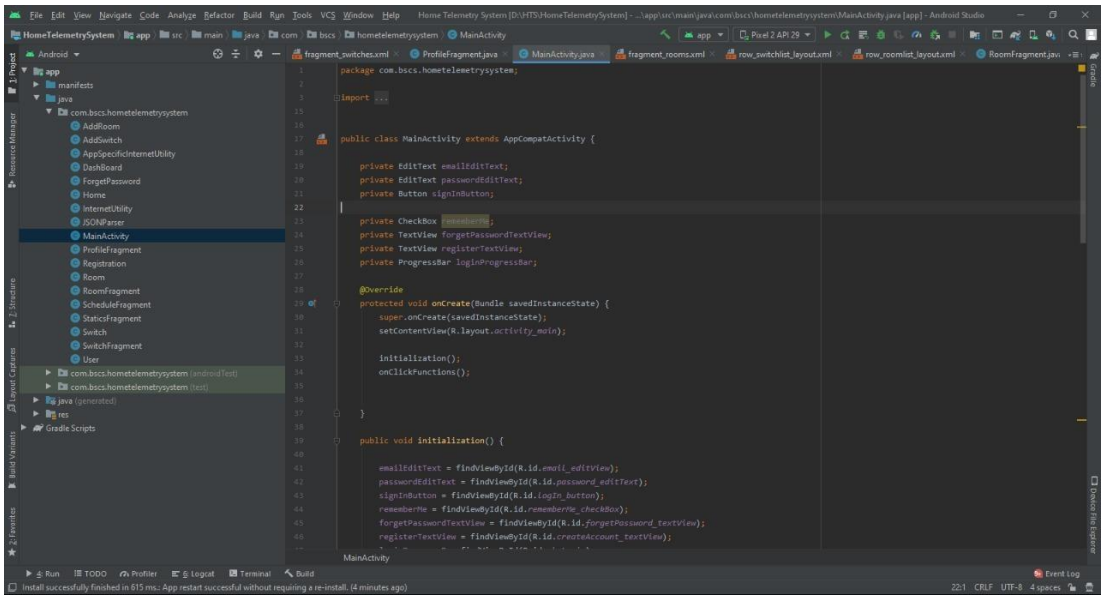


Figure A.3: Main activity panel

This code is responsible for transferring the main screen to dashboard, registration or forget password.

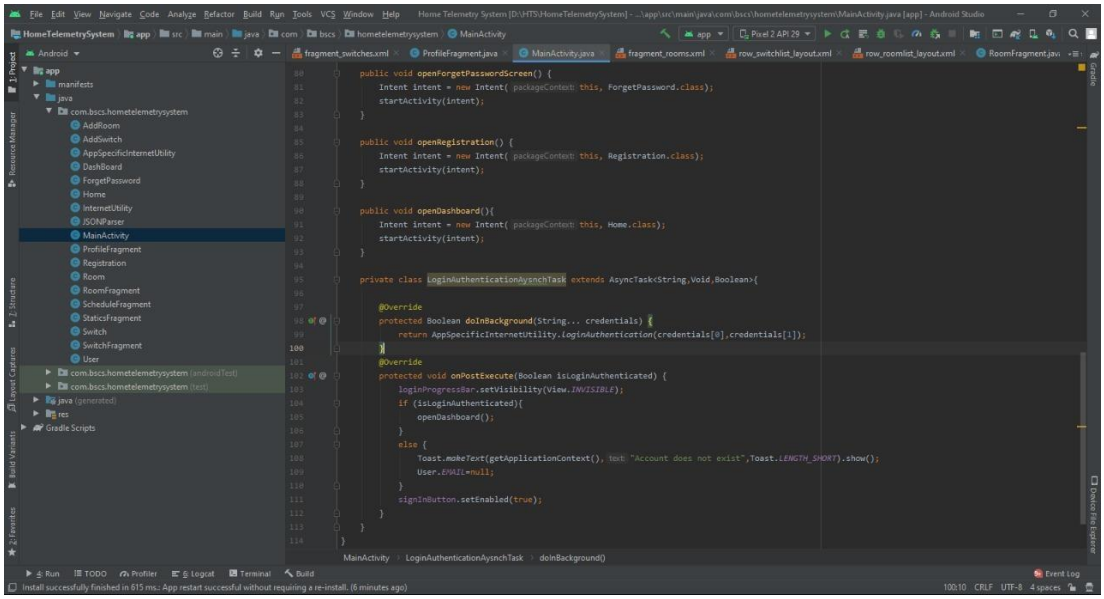


Figure A.4: Main activity panel

Connect the xml object with the java code in add room.

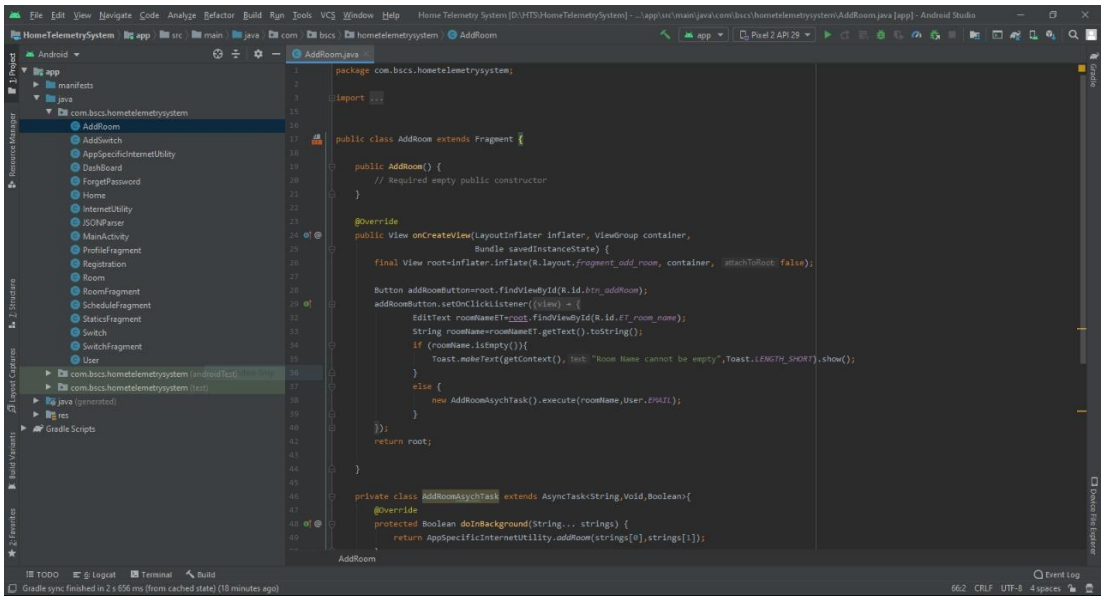


Figure A.5: Add Room

Functionality implementation of the add room button.

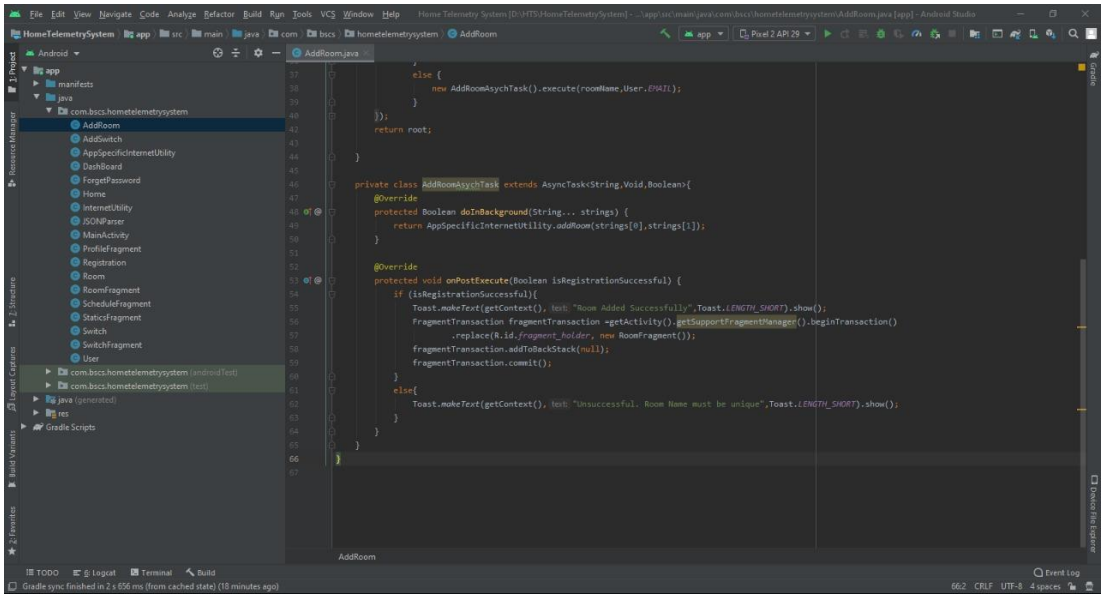


Figure A.6: Add Room Panel

Connect the xml object with the java code in add switch.

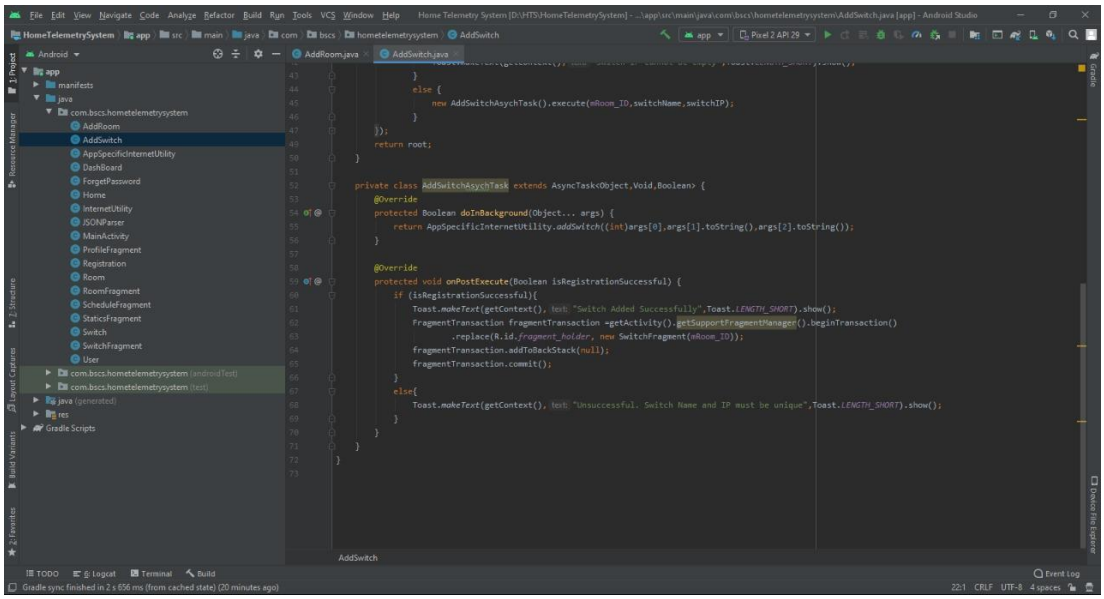


Figure A.7: Add Switch

Functionality implementation of the add room button.

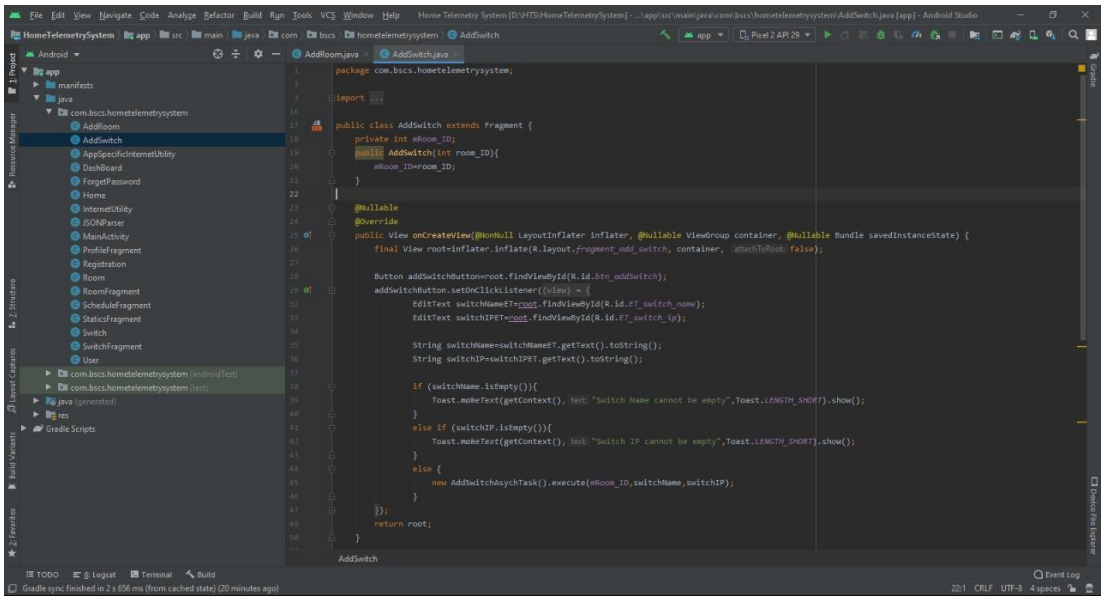


Figure A.8: Add Switch Panel

Change the switch status on the server and delete any switch.

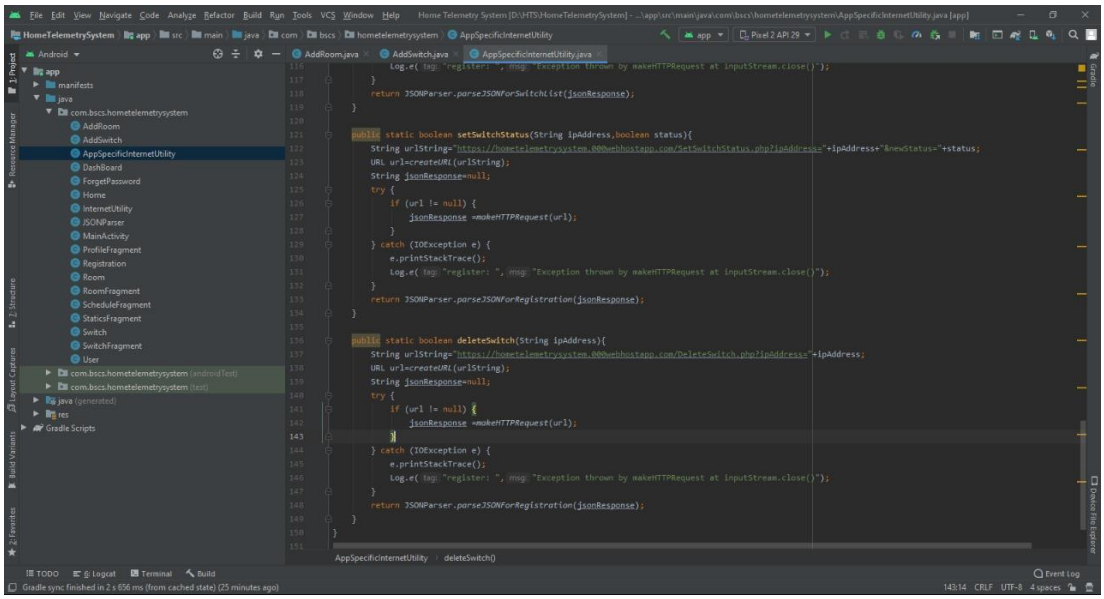


Figure A.9: App Specific Internet Utility (a)

Change password on the server.

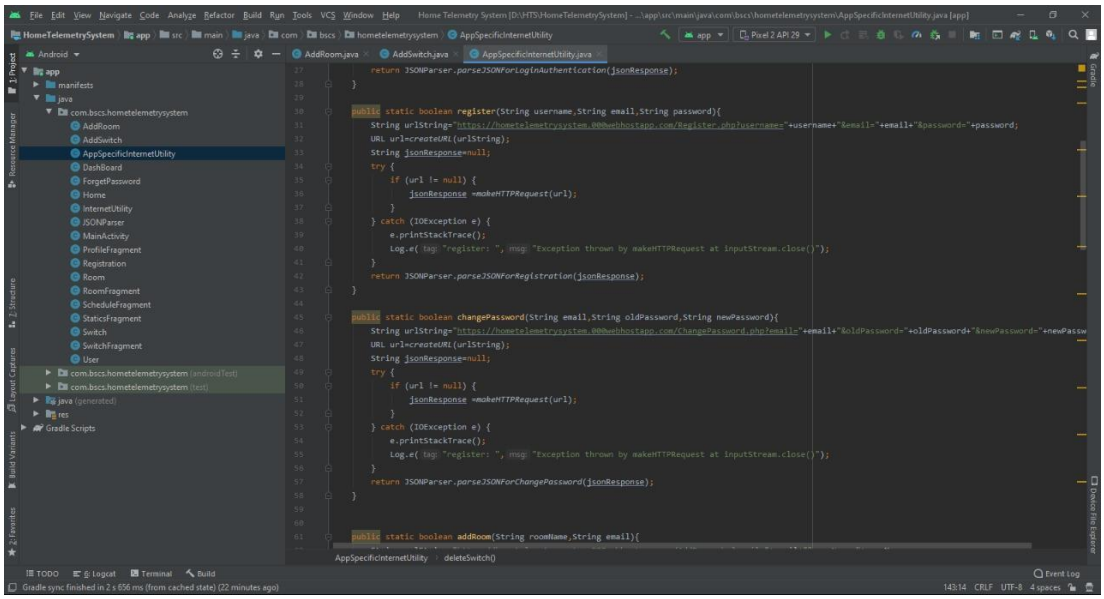


Figure A.10: App Specific Internet Utility (b)

Authenticating the user during login from server.

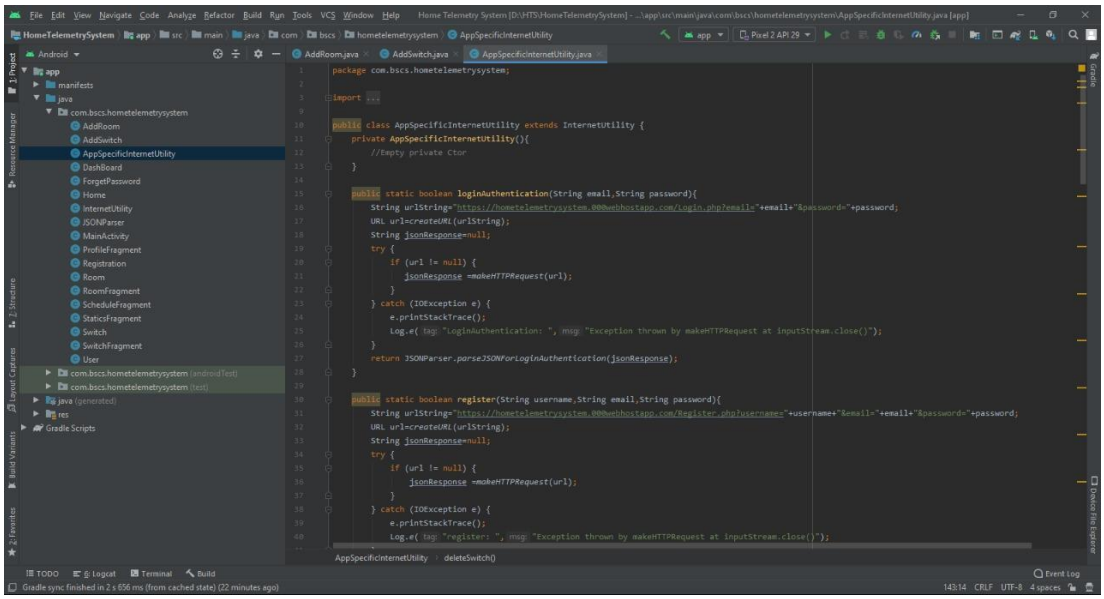


Figure A.11: App Specific Internet Utility (c)

Connect the xml object with the java code in profile fragment.

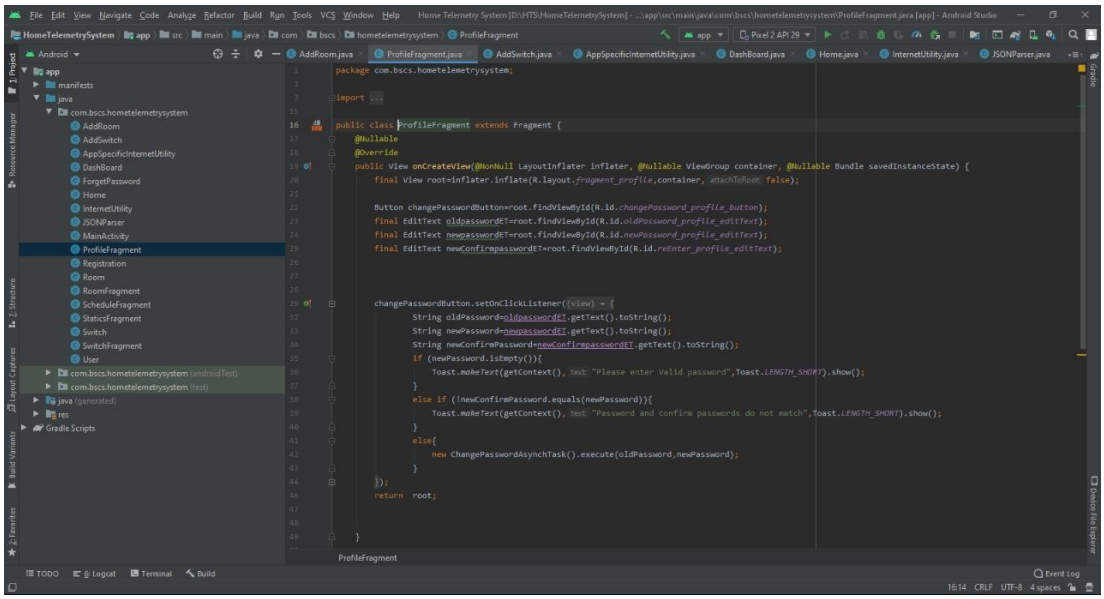
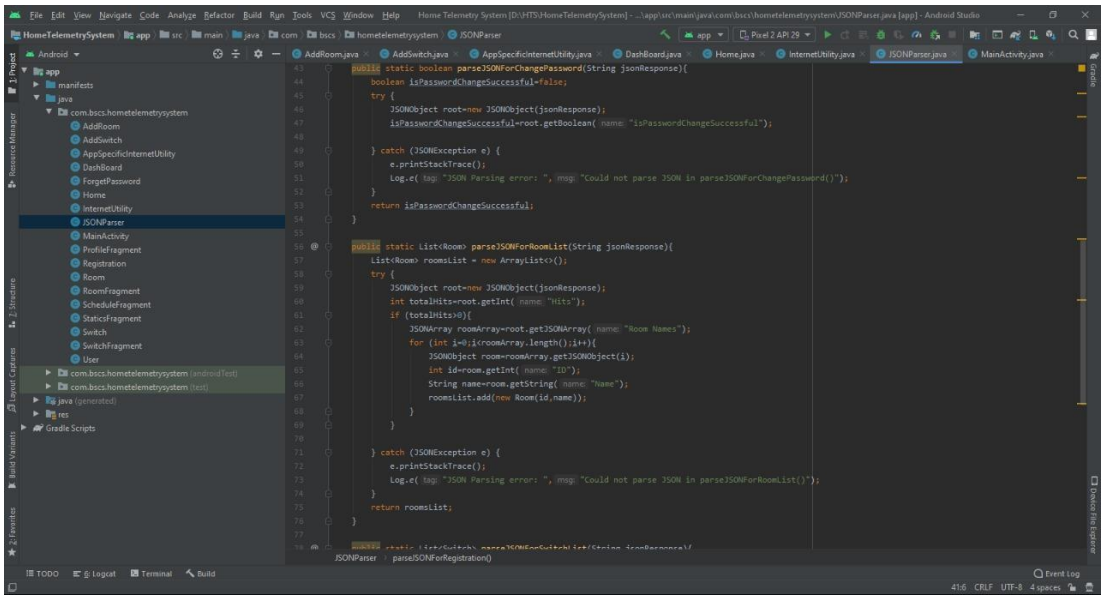


Figure A.12: Profile Fragment

Converting Json response into java.



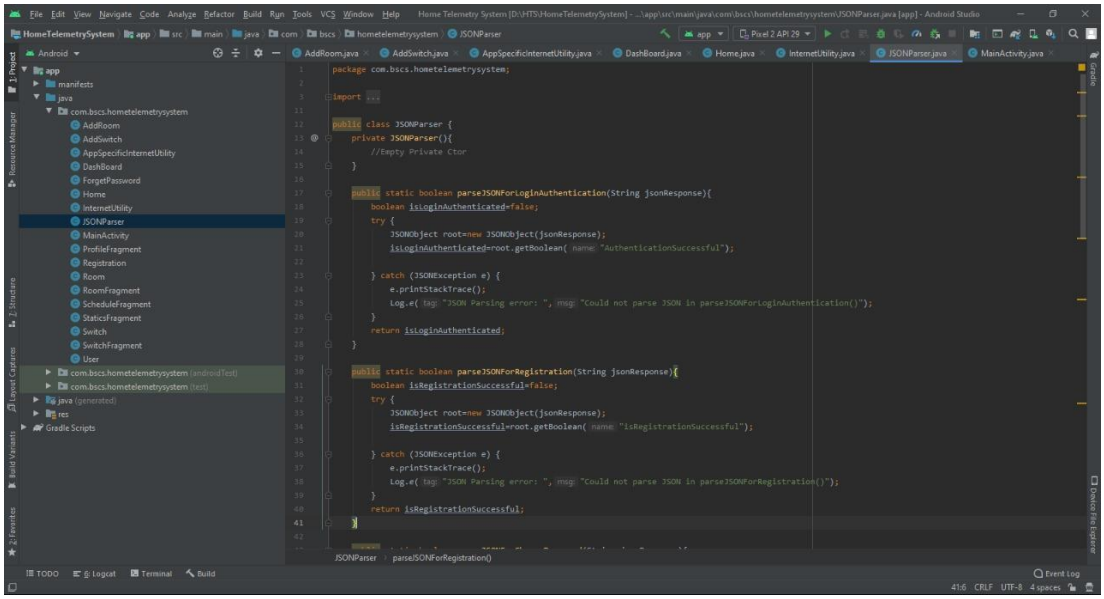
```
public static boolean parseJSONForChangePassword(String jsonResponse){
    boolean isPasswordChangeSuccessful=false;
    try {
        JSONObject root=new JSONObject(jsonResponse);
        isPasswordChangeSuccessful=root.getBoolean( name: "isPasswordChangeSuccessful");

    } catch (JSONException e) {
        e.printStackTrace();
        Log.e( tag: "JSON Parsing error: ", msg: "Could not parse JSON in parseJSONForChangePassword()");
    }
    return isPasswordChangeSuccessful;
}

public static List<Room> parseJSONForRoomList(String jsonResponse){
    List<Room> roomsList = new ArrayList<>();
    try {
        JSONObject root=new JSONObject(jsonResponse);
        int totalHits=root.getInt( name: "Hits");
        if (totalHits>0){
            JSONArray roomArray=root.getJSONArray( name: "Room Names");
            for (int i=0;i<roomArray.length();i++){
                JSONObject room=roomArray.getJSONObject(i);
                int id=room.getInt( name: "ID");
                String name=room.getString( name: "Name");
                roomsList.add(new Room(id,name));
            }
        }

    } catch (JSONException e) {
        e.printStackTrace();
        Log.e( tag: "JSON Parsing error: ", msg: "Could not parse JSON in parseJSONForRoomList()");
    }
    return roomsList;
}
```

Figure A.13: Json Parsing (a)



```
package com.bscs.hometelemetrysystem;

import ...

public class JSONParser {
    private JSONParser(){
        //Empty Private Ctor
    }

    public static boolean parseJSONForLoginAuthentication(String jsonResponse){
        boolean isLoginAuthenticated=false;
        try {
            JSONObject root=new JSONObject(jsonResponse);
            isLoginAuthenticated=root.getBoolean( name: "AuthenticationSuccessful");

        } catch (JSONException e) {
            e.printStackTrace();
            Log.e( tag: "JSON Parsing error: ", msg: "Could not parse JSON in parseJSONForLoginAuthentication()");
        }
        return isLoginAuthenticated;
    }

    public static boolean parseJSONForRegistration(String jsonResponse){
        boolean isRegistrationSuccessful=false;
        try {
            JSONObject root=new JSONObject(jsonResponse);
            isRegistrationSuccessful=root.getBoolean( name: "isRegistrationSuccessful");

        } catch (JSONException e) {
            e.printStackTrace();
            Log.e( tag: "JSON Parsing error: ", msg: "Could not parse JSON in parseJSONForRegistration()");
        }
        return isRegistrationSuccessful;
    }
```

Figure A.14: Json Parsing (b)

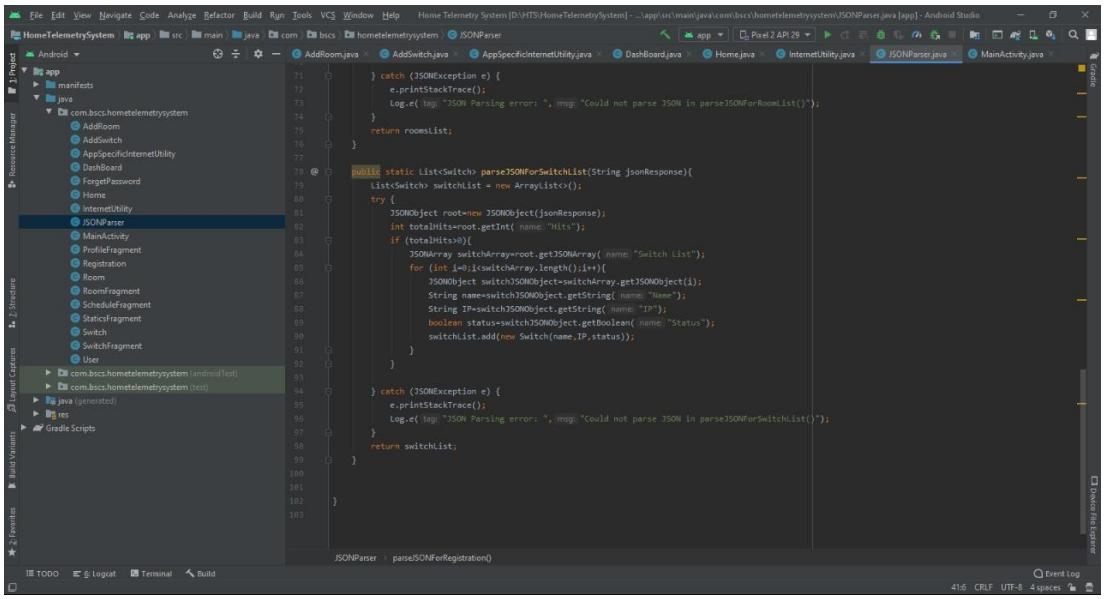


Figure A.15: Json Parsing (c)