Table of Contents

Introduction

This report covers the initiatives taken to develop a Smart Energy Monitoring System (SEMS), with emphasis on residential homes. It is important that end users gain understanding on efforts they can make to reduce electricity consumption, since it is a collective effort by everyone. Furthermore, users are incentivised to save energy as they can save money and cost of usage as well. As technologies like Internet-of-Things (IoT) and machine learning (ML) become more widespread, they become easier to be introduced and implemented in homes everywhere.

The system contains three main components: Meter, Controller, and Server. The meter component encompasses smart meters and IoT sensors for monitoring energy usage in the household intelligently and in real-time. The Controller component is a newly developed mobile application for the end user to interact with, to learn about their personal electricity usage at home, and control appliances (specifically air conditioners) remotely to manage their energy usage. The mobile application is the main focus of this project, with the goal of providing users a friendly way to interact with their smart devices and optimise their personal electricity usage. The Server component is a cloud-based platform that analyses the users' electricity usage to identify inefficiencies and ways to conserve electricity.

The software process model that is chosen is the incremental development model, which relies on feedback with stakeholders more often. Hence, ensuring that the mobile application grows with technological improvements and user demands. This SEMS mobile application seeks to be a user-friendly and informative tool that enables users to actively engage in community electricity saving initiatives.

Project Objectives and Scope

The main objective of the SEMS is to optimise electricity usage in residential buildings. The goal is for the homeowner to be able to monitor their energy usage and adjust their usage habits with the suggestions provided that are determined by algorithms in the cloud server based on their real-time usage.

The scope of this project is optimizing how users can manage energy in residential areas and commercial buildings. This project requires the installation of smart meters and IoT sensors throughout the interior of buildings or homes which allows tracking of electricity consumption in real-time. The data that are collected by these sensors will be sent to a cloud platform which in turn utilize certain algorithms to be able to detect users' electricity usage patterns, anomalies, and even detect the users' future needs. In addition, users will be able to access and view the data that is being collected by the installed sensors with the help of a mobile application - this application will also give the user the ability to control and monitor their appliances, particularly air conditioners, in real time. Furthermore, the user will also be notified periodically with automated alerts and recommendations on how to reduce energy waste and costs and promote sustainable practices via the application. Ultimately, the smart energy monitoring system will be able to provide an efficient, cost-effective, and eco-friendly energy management solution.

The timeline of this project began on 10 June 2024 with the release of the project requirements document. The predicted timeline lasts for 12 weeks, ending with a submission on 19 August 2024. However, the project extended beyond week 12 with updates and fixes to support the evolution of the software. A visualisation of the timeline is shown in Table 1 and Figure 1.

Task	*Start week*	*Weeks for completion*
Project Preliminary Stage		
Introduction	0	1
Objective and Scope	0	1
Software Process Model	1	1
Project Analysis and Design		
Requirements Plan	1	2
System Functionality	2	1
Prototype	3	2
UML Diagrams		
Use Case Diagram	3	1
Class Diagram	3	1
Sequence Diagram	4	1
Activity Diagram	4	1
Software Design and Implementation		
Mobile App Development	5	5
Server Development	5	5
System Testing	6	5
Project Publishing		
Quality Assurance Checking	9	2
Submitting Deliverables	11	1
Updates and Fixes	12	2

Table 1. Detailed timeline of project tasks.

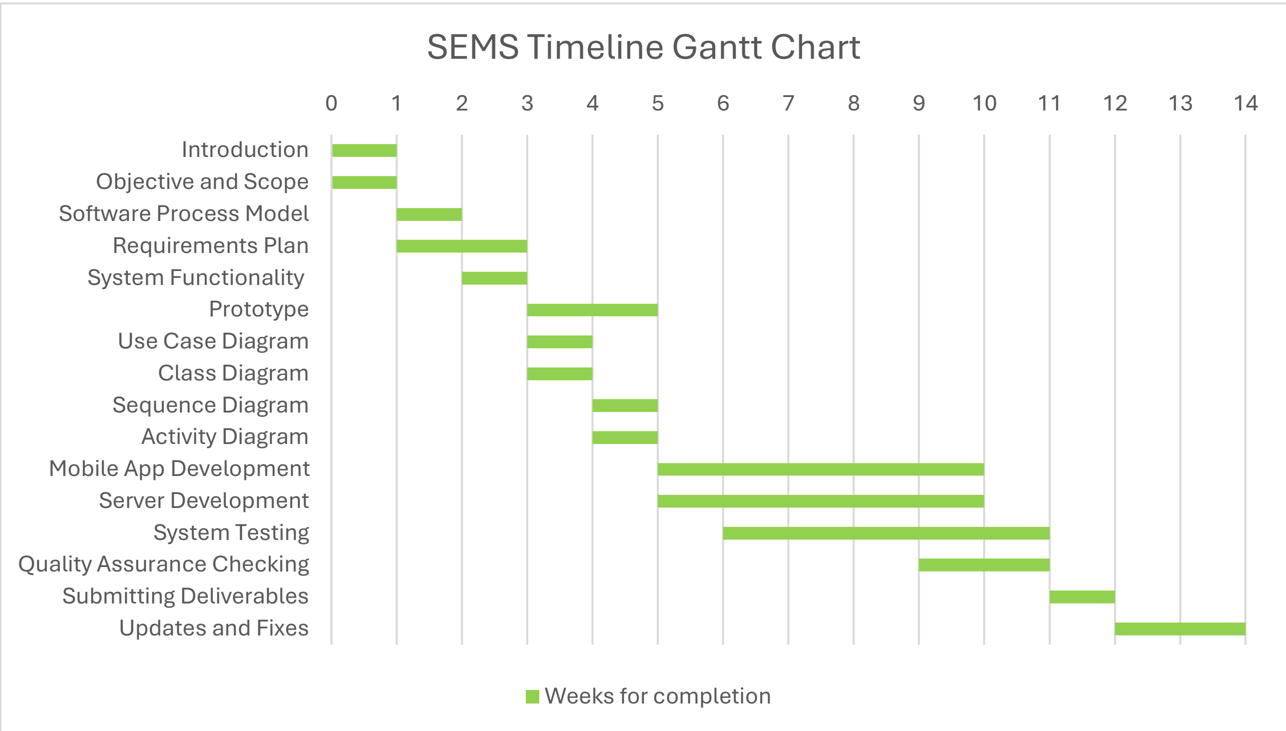


Figure 1. Gantt chart for project tasks.

The budget of this project can be broken down into five sections - hardware, development, cloud, personnel, and operational. The estimated budget of the hardware, which includes the smart meters and IoT sensors, can cost RM302 and RM1.79 per unit respectively. If 100 units for each of these hardware are used, the total would be RM30,379. For the development of the software used in the smart energy monitoring system, the average estimated cost of development would lie between RM80,000 – RM470,000 approximately, depending on the complexity of the system. As for the cloud services, the average monthly cost would be roughly RM1,080.50 which will make the yearly cost around RM13,000. For personnel, it can be separated into software developers and data analysts. The average annual salary for software engineers is around RM60,000 and the average annual salary for data analysts in Malaysia is close to RM50,000. There will be five of each being hired so the annual cost for personnel will be RM550,000. To keep the system up and running, maintenance must be conducted regularly, so a maintenance engineer must be hired as well. The average annual salary for a maintenance engineer is RM48,000. Overall, the first year's estimated budget will be around RM917,000.

The project deliverables are threefold: a report that details the specifications and development plan of the SEMS project; a prototype of the mobile application designed using Figma; and 4 UML diagrams showcasing the processes of the system and their relationships.

Software Process Model

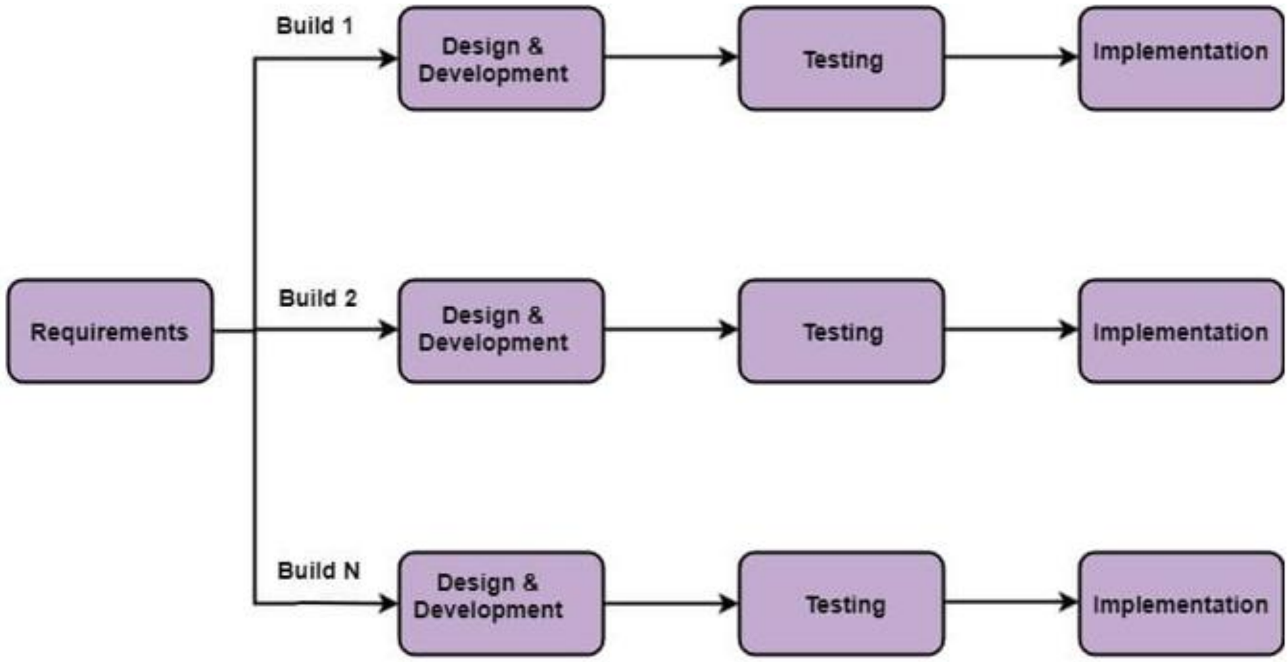


Figure 2: Incremental Model

The software process model chosen for this Smart Energy Monitoring System (SEMS) for residential applications is the incremental development model. This decision is influenced by numerous major criteria that are consistent with the project's objectives and characteristics. The incremental model is a software development approach in which requirements are separated into several isolated modules during the software production cycle. In this paradigm, every component undergoes the requirements, design, implementation, and testing stages. Each succeeding version of the module extends functionality to the prior one. The procedure continues until the entire system is realised (Javatpoint, 2011).

The SEMS development is guided by the incremental development model through iterative cycles, with each cycle concentrating on delivering a particular set of functions. In order to gather user requirements and specify functionalities, a comprehensive requirement study is first conducted using use case analysis, surveys, and user interviews. Document analysis ensures regulatory compliance. These criteria are converted into a functional software increment throughout development and implementation. To guarantee smooth integration with current features, the development team carefully develops the functionalities, tests each component individually, and runs integration tests. In order to ensure that the capabilities are functioning as intended and offer a user-friendly experience, testing and deployment also includes thorough functional testing and user acceptability testing (UAT). UAT stands for User Acceptance Testing as this is require before delivering the software and to make any necessary adjustments (Elazar, 2018). To improve the system and prepare for the next stage of development, this iterative cycle consistently takes into account insightful customer feedback, guaranteeing that the finished product of SEMS meets user needs (Sommerville, 2016).

Given that the model is iterative, it is an appealing approach to construct the SEMS. Through the implementation of smaller capabilities in stages, the approach facilitates ongoing user feedback during the development process. Some of the benefits of this feedback loop are early detection of usability problems, user needs-driven feature development prioritisation, and flexibility in response to changing market demands.

On the other hand, the disadvantage is scope creep. This unchecked feature expansion can lead to longer development times and higher costs, as well as additional complexity and future maintenance challenges for SEMS. A clearly defined project scope with explicit impact and feasibility-based prioritisation criteria is essential to minimize the risk (GeeksforGeeks, 2024).

Ultimately, the incremental model is most suitable for creating a user-centric and successful SEMS for residential energy management because of its emphasis on risk reduction, adaptability, and user feedback.

Requirements Plan

Our requirements plan defines how requirements will be gathered, analysed, documented, and managed throughout the lifecycle of the electrical optimization system. We have gathered the key stakeholders below, categorised into three main groups.

Stakeholders:

End Users:

These users will use the developed system to monitor and optimize electrical energy in their facilities. They are key stakeholders who will provide insights into the system's features and functionalities.

- Facility Managers:
  - Homeowners
  - Renters
  - Small business owners
- Environmentally conscious individuals:
  - Individuals that are concerned about their electrical energy usage.

Development Team:

These are the shareholders that are tasked with developing the system by providing their technical solutions.

- Project Manager
  - Individuals that watch over the development process, ensuring projects are completed on time, within budget, and meet user requirements.
- UX/UI Designers
  - Individuals that create intuitive, user-friendly interfaces that enhance user satisfaction.
- Backend Developers
  - Individuals that handle data storage, security, and integration with external services

Requirements Elicitation

To understand the needs and challenges of the energy-efficient system, we chose the interviewing method. We conducted an **open discussion** with the stakeholders, specifically the end users and development team, to gather their insights on energy efficiency requirements. Furthermore, with **document analysis** our team managed to extract information from existing systems and documents as a guide for our system development. (Tiwari, n.d.)

Requirements discovery:

Interviewing

During this process, our stakeholders either set too high requirements or don't really know the vague idea of what they want. The approach we took is clarifying the project scope to them and held a lot a lot of discussions. We host individual interviews, making sure that every stakeholder involved knew what requirement they want individually, and their points go across. Then we held group interviews for discussion and gather collective insights. Which helped in identifying common requirements and resolving conflicting needs.

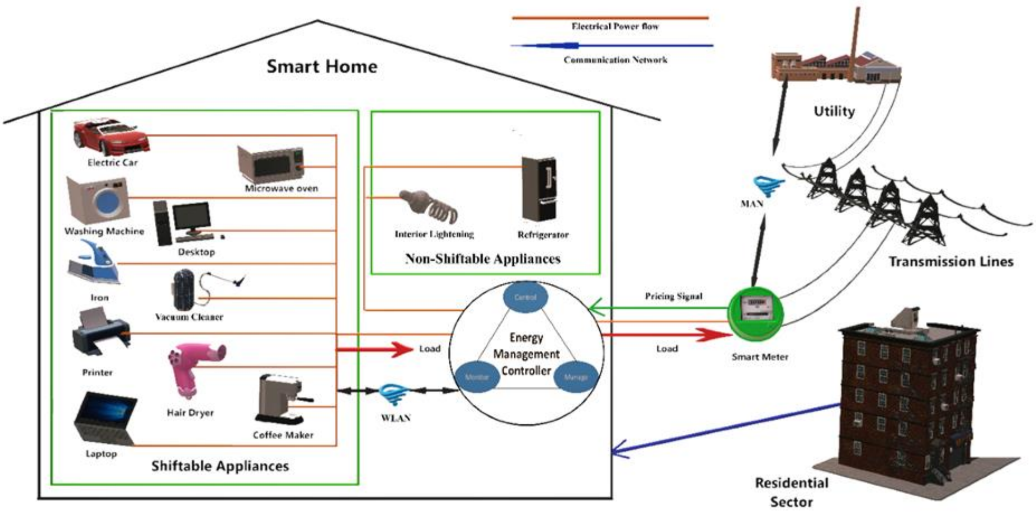


Figure 3: Smart Home (Hussain, 2020)

Document elicitation

After some documentation research, we have research on different SEMS and decided to write the requirements based on the smart home figure showed above. We also read through technical manuals and research articles to gather in-depth knowledge about the technologies and methodologies employed in existing SEMS.

Additionally, by gathering insights and feedback from these stakeholders, we ensure that the system not only meets technical specifications but also aligns closely with user needs and business goals. This collaborative approach helps to prioritize features effectively, resolve potential conflicts in requirements, and ultimately deliver a solution that adds significant value to all stakeholders involved.

Requirements classification and organisation

We've categorized and structured our requirements according to their specific functionalities. For instance, we have defined categories such as the login page, smart meter monitoring, home management, appliance control, alerts and notifications, automation, sharing control, and help page. Each category is dedicated to achieving distinct system goals, which will be elaborated on in detail in our requirements specification section.

Prioritisation and negotiation

Initially, we emphasized IoT management and its associated features but recognized the need to prioritize monitoring features for the data collection of electrical energy instead. During discussions, we identified several conflicts in requirements that we are currently addressing.

Firstly, there was a huge range of appliances varied in a smart home. Thus, rather than managing multiple appliances, we have decided to focus exclusively on the AC (Air Conditioning) system.

For instance, we wanted the integration of sharing control between different users. To not overcomplex this system, the result was removing it as it is a minor feature. The amended features can be observed in the diagrams below. Figure Features Model 1,2 & 3 shown in Requirements Validation.

Requirements Analysis

After gathering all the information regarding the Smart Energy Management System (SEMS), we analyzed the requirements thoroughly.

Initially, we conducted several interviews, individual and in groups with stakeholders. During this process we managed to capture their specific needs and expectations. By analyzing it, we addressed multiple challenges involving too high stakeholder expectations and vague requirements. This analyzing process helped to refine and clarify the system through discussions.

We conducted an ethnographic study, by observing how stakeholders interact with existing IoT management apps. For example, Smart Things by Samsung and The Home app by Apple. The information we observed provided insights into actual user workflows, focusing on a user's activities.

However, it is not enough as this is a SEMS. Thus, we also observed other systems like TNB smart meter application, and the Smart Home Diagram shown in requirement elicitation. Reviewing existing operation documentation to understand current workflows and identify integration points for the SEMS. Furthermore, document analysis further informed the integration of the IoT management system with existing energy control processes.

We synthesized and categorized data from these activities into a comprehensive list of requirements.

Requirements Validation

We synthesized and categorized the data from these activities into a comprehensive list of requirements, ensuring that the final system meets user needs and expectations effectively.

Prototyping

We took the incremental development model approach. Basically, we were monitoring the development of requirements through iterative cycles of prototyping. Which are subsequent versions of the SEMS prototype are refined based on validated requirements and stakeholder input, ensuring alignment with user needs and regulatory standards. By little incrementation of development we can evaluate and verify if the requirement before proceeding to the development of the next increment. This iterative approach accelerates the development by enabling quick adjustments but also enhances the system's responsiveness to evolving requirements, ultimately delivering a robust and user-centric SEMS solution.

Furthermore, we also created simple diagrams showcasing the operational flow for the stakeholders. Thus, the change of requirements can be observed and documented after every discussion. The diagrams shown below in Figure 3, 4 and 5 illustrates feature models 1,2 and 3.

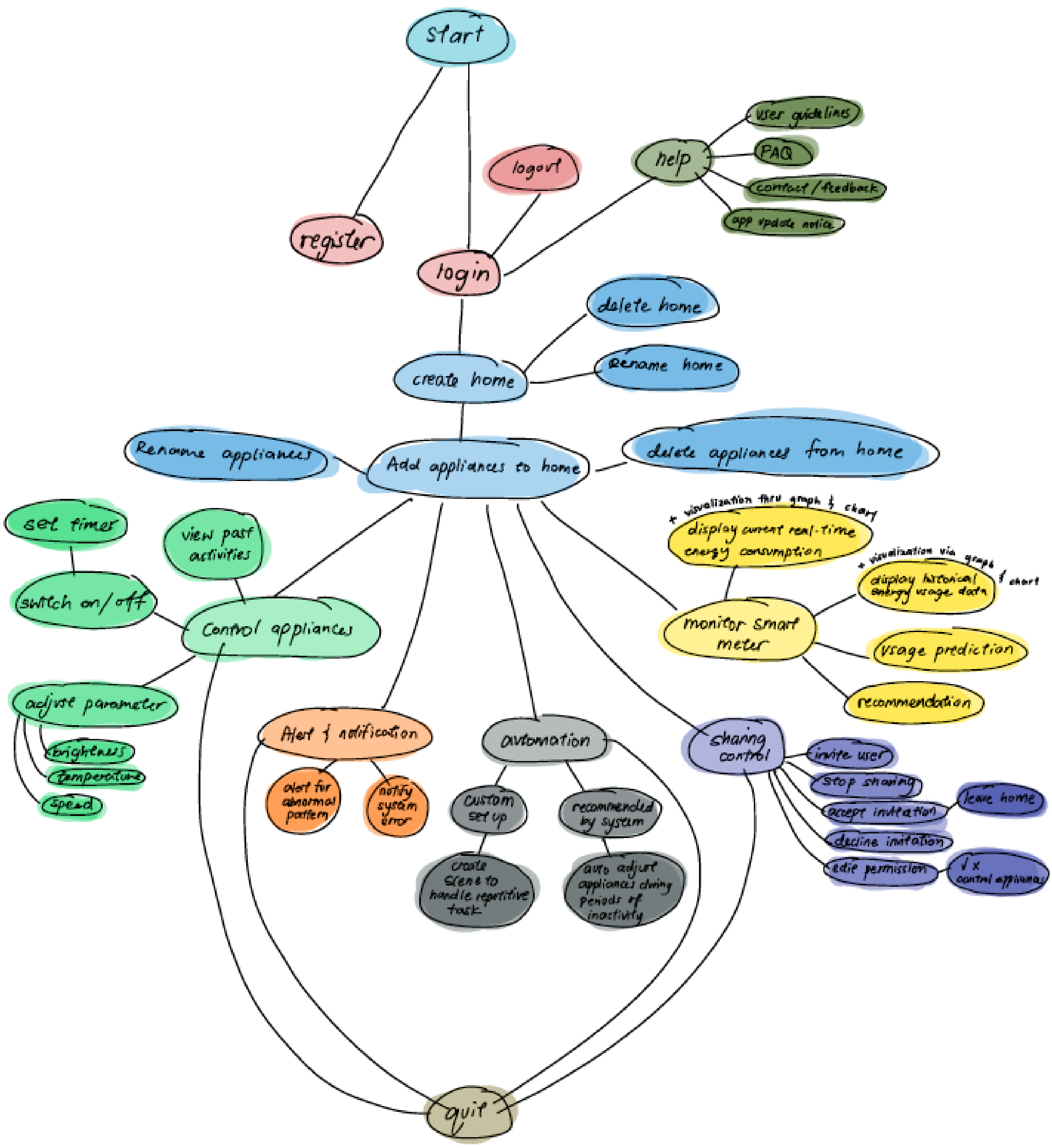


Figure 4: Features Model 1

Figure 5: Features Model 2

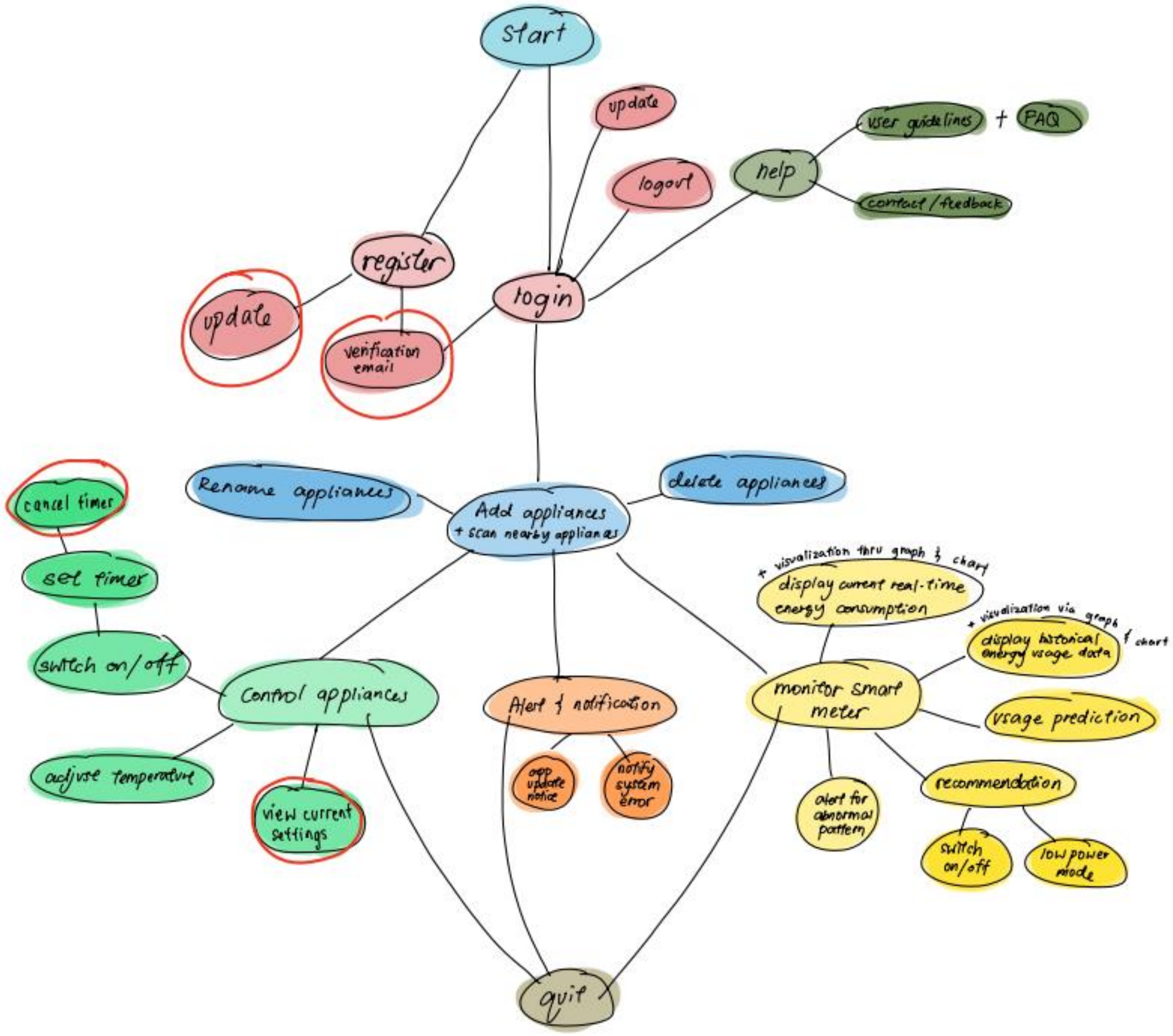


Figure 6: Features Model 3

Reviews

Discussions were conducted among all stakeholders to review the ongoing development of requirements. These sessions include participation from both end users and development teams. First and foremost, at the beginning of every meeting, the origin document of the requirement will be clearly stated and repeat if there are any uncertainties. If uncertainties remain, we will verify them through system testing to ensure data output and system actions align with expectations. However, if there are any issues, it will be resolved in the most rationale way and without having too much on the requirements suggested by both parties. Lastly, changes will be documented and added to the feature models at the end of the meeting.

Requirements Management

Requirements management for SEMS is a systematic process to keep track of the changing requirements throughout its development lifecycle. Initially, requirements are elicited through stakeholder interviews and document analysis, ensuring a comprehensive understanding of user needs and system capabilities. These requirements are then classified, organized, and prioritized based on their importance. Thus, making it easier to address potential conflicts early on.

Throughout development, the incremental development model is utilized, allowing SEMS to refine its requirements through prototyping and stakeholder feedback. Each incrementation focuses on delivering specific functionalities while ensuring alignment with validated requirements and stakeholder expectations. Regular reviews and discussions among stakeholders further validate and refine requirements, addressing any emerging issues promptly. During this stage, the conflict or the change in requirements are properly analysed to verified. Decisions made also can be traced back during each increment of the development process.

Tools like diagrams and clear requirement specification aid in visualizing and testing requirements, facilitating clear communication and alignment across development teams and stakeholders. Therefore, ensuring that the requirements of SEMS are clear and concise to the stakeholders during the process.

By integrating these principles into requirements management, SEMS targets to deliver a robust, user-centric solution that optimizes electrical energy usage effectively in residential settings, thereby meeting both customer expectations and business objectives during the entire development process.

User Requirements

1. Users shall be able to create an account, and log into their accounts with the required information. (*Justification: Users need to have an account to monitor their electricity usage.)*
2. Users' identity shall be verified via email after creating an account. *(Justification: To ensure the email provided is valid and belongs to the user.)*
3. Users shall be able to modify their account details. *(Justification: Users might want to change their details over time and the user data stored in the system must be up-to-date.)*
4. Users shall be able to log out securely. *(Justification: The system needs to maintain user privacy and security.)*
5. The application shall provide basic supports to users. (*Justification: Users might need some simple guidelines to learn how to navigate the application, especially for new users.)*
6. Users shall be able to contact the support team when they encounter issues. (*Justification: Users might encounter problems when using the application, contacting the support team can help users to fix their issues immediately.)*
7. Users shall be able to provide feedback for the application. (*Justification: The application would need constant improvements after implementation; therefore users' feedback is essential.)*
8. Users shall be able to manage only one home within a single account. (*Justification: Generally, each home can only have one smart meter, and each smart meter can only link to one account.)*
9. Users shall be able to manage every air conditioner in a home. (*Justification: Users need to identify and organize the air conditioners in a home.)*
10. Users shall be able to control air conditioners from anywhere and at any time, after connecting them to the system. *(Justification: To ensure users can manage their electrical consumption directly and conveniently.)*

11. Users shall be notified for any application updates. *(Justification: Users need to know the new features and improvements for each update, so that they have a better understanding in using the application.)*

12. Alerts shall be prompted to user if error occurs. *(Justification: If error occurs when performing any functions in the application, users should know what is the error and how to respond to the error.)*

13. Users shall be able to monitor their current and historical electricity usage. *(Justification: Users need to know how much electricity is being used.)*

14. The application shall predict future electricity usage based on users' historical electricity usage pattern. *(Justification: Prediction of future usage can help users in planning and managing their future electricity usage.)*

15. The application shall provide recommendations to control air conditioners. *(Justification: The recommendations could help users improve energy efficiency and reduce costs conveniently.)*

16. Users shall be notified if the system detects abnormal energy consumption. (*Justification: Users need to be notified so that they could detect inefficiencies and other potential issues.)*

System Requirements

Functional Requirements

1. **Account Management**

    1.1. New users shall be able to create an account with a valid username, email, password (must be at least 12 characters long with a combination of uppercase letters, lowercase letters, numbers and symbols) , phone number, first name and last name. *(Justification: User information is essential to recover account and enhance security.)*

    1.2. The system shall send a verification email to newly registered users after they create an account. *(Justification: To ensure the user is the owner of the email account.)*

    1.3. Users shall be able to log into their account after clicking the link provided in the verification email. *(Justification: Users need to be redirected back to the application to log into their account after their email is verified.)*

    1.4. Users shall enter their username and password to log into their accounts. *(Justification: The ensure user data is protected and to prevent unauthorized access.)*

    1.5. The system shall allow users to update their account details including username and password. *(Justification: Users might want to change their username and password after using the application for some time.)*

    1.6. The application shall allow users to log out of their account when desired. *(Justification: To ensure that that user interactions with the application are securely terminated and prevent unauthorized access.)*

2. **Support**

    2.1. The application shall provide simple user guidelines and a list of frequently asked questions and answers about the application. (*Justification: To allow users to fix simple and common issues independently if they cannot reach out to the support team.)*

    2.2. Users shall be able to contact the support team by filling a form that requires users' first name, last name, email, and their message to the support team. (*Justification: Support team needs to know the reasons why users are contacting them and the email of users so they can reach back to them.*)

3. Air Conditioners Management

3.1. Users shall be able to scan for nearby air conditioners that are compatible (compatible air conditioners are listed in *Table 2* under appendix), and add them to the system via wireless connection (Bluetooth and Wi-Fi). *(Justification: To simplify the process of adding new air conditioners, and ensure they are compatible with the system.)*

3.2. Users shall be able to delete air conditioners. *(Justification: To allow users to remove air conditioners that are no longer in use or have been replaced.)*

3.3. Users shall be able to rename air conditioners. *(Justification: To allow users to easily identify and manage the air conditioners.)*

4. Controlling Air Conditioners

4.1. Users shall be able to switch the air conditioners on or off remotely. *(Justification: Users should have direct control over the air conditioners to optimise energy usage and they might want to control the air conditioners outside their home.)*

4.2. Users shall be able to set up a timer to automatically switch the air conditioners on or off remotely at a specific time within a 24-hour period. *(Justification: To help users save electricity and provide a personalized user experience.)*

4.3. Users shall be able to cancel the timer that will automatically switch on or off the air conditioners. *(Justification: Users need to have the flexibility to control air conditioners based on their changing needs.)*

4.4. Users shall be able to adjust the temperature of the air conditioners. *(Justification: Users need the ability to adjust the air conditioners to warm up or cool down the environment so that a comfortable interior temperature is maintained. By doing so, electrical energy usage could also be optimised.)*

4.5. Users shall be able to view the current settings (power on/ off status, temperature, timer on/ off status) of the air conditioners. *(Justification: Users can monitor the status of the air conditioners clearly and adjust them according to their needs.)*

5. Alerts and Notifications

5.1. The application will send an alert to users when application error occurs. The possible application errors are stated below. *(Justification: Users need to be informed about application errors so that they can take prompt actions to fix the issues.)*

5.1.1. Failed to update account details
5.1.2. Failed to detect compatible air conditioners
5.1.3. Failed to add/ delete/ rename air conditioners
5.1.4. Failed to switch air conditioners on or off
5.1.5. Failed to adjust air conditioners' temperatures
5.1.6. Failed to set air conditioners to low power mode
5.1.7. Air conditioners are disconnected
5.1.8. Failed to set up/ cancel a timer
5.1.9. Failed to display current/ historical energy usage
5.1.10. Failed to display electricity usage prediction

5.2. Users shall be notified should there be any application updates. *(Justification: Users need to be constantly aware of the application's new features, improvements and bug fixes so that they can make full use of the application.)*

6. Smart Meter Monitoring

6.1. The system shall be able to send data (electricity usage) collected by the smart meter to a cloud-based platform for analysis. *(Justification: Analysis is used to transform the raw data into meaningful information and graphical representations so users can gain insights into their electricity usage pattern.)*

6.2. The application shall display the current and historical daily and monthly electricity usage in kilowatt-hour (kWh) with graphical representations. *(Justification: To help users to visualize their electricity usage.)*

6.3. The system shall only display historical electricity usage over the past six months. *(Justification: To ensure the information displayed to users is relevant and manageable.)*

6.4. The system shall calculate the estimated cost of electricity usage per month in Ringgit Malaysia (RM), based on the users' monthly electricity usage. The formulas and conditions to calculate the cost are shown below. *(Justification: To help users in managing their budgets and learning the financial impacts of their electricity consumption.)*

6.4.1. The estimated cost includes a 8% service tax , which is charged in the Tenaga Nasional Berhad (largest electricity utility in Malaysia) electricity bills for residential customer when electricity consumption exceeds 600kWh.

6.4.2. The calculation is based on a normal billing cycle, which is 28 days.

6.4.3. The calculation does not take into account rebates, discounts, or special tariff incentives.

6.4.4. The formulas to calculate the estimated cost are shown below.

- Calculation of energy consumption *(in RM)* per month:

Monthly energy consumption *(in RM)* =
(First 200kWh of total consumption * 21.8)/100 +
(Next 100kWh of total consumption * 33.4)/100 +
(Next 300kWh of total consumption * 51.6)/100 +
(Next 300kWh of total consumption * 54.6)/100 +
(Next 901kWh onwards of total consumption * 57.1)/100
Note: RM = Ringgit Malaysia; kWh = kilowatt-hour

- 8% Service Tax *(in RM):*

8% Service Tax =
Consumption beyond 600kWh *(in RM)* * 8%
Note: RM = Ringgit Malaysia; kWh = kilowatt-hour

- Total estimated cost *(in RM):*

Total estimated cost =
Current month total consumption *(in RM)* + 8% service tax
Note: RM = Ringgit Malaysia

6.5. The system shall predict future electricity consumption (in kilowatt-hour, kWh) based on the pattern exhibited in the historical electricity consumption. *(Justification: To help users plan and reduce their electricity usage.)*

6.6. The application shall provide recommendations to control the air conditioners based on historical electricity consumption patterns. The descriptions of the recommendations are stated below. *(Justification: To provide convenience to users where they can optimize their electricity usage with the preset recommendations.)*

6.6.1. Automatically switch the air conditioner on/ off during periods of inactivity.

6.6.2. Set the air conditioner to low power mode, in which the temperature of the air conditioner will be raised by 0.5 to 1 degree Celsius every hour, for a maximum of 3 degree Celsius per hour.

6.7. The system shall alert users if the current monthly electricity consumption exceeds the average monthly electricity consumption. *(Justification: To help users identify potential inefficiencies of electricity usage.)*

Non-functional Requirements

1. **Usability Requirements**

   1.1. The user interface of the application shall be user-friendly and attractive. *(Justification: To ensure users can navigate the application easily and intuitively.)*

2. **Performance Requirements**

   2.1. The system shall be able to react to all users' request within 2 seconds. *(Justification: To ensure user interactions with the system is seamless.)*

   2.2. The system shall display the data of current electricity consumption within 3 seconds. *(Justification: Users require instant feedback on their electricity usage to make timely decisions and to ensure seamless user experience.)*

   2.3. The time for scanning nearby available air conditioners should not exceed 1 minute. *(Justification: To ensure users can setup their air conditioners quickly and conveniently.)*

   2.4. The verification email sent to users upon registration of an account should be sent within a latency of no greater than 1 minute. *(Justification: Users need to be able to start using the application immediately after registering an account.)*

3. **Space Requirements**

   3.1. The system shall have sufficient server storage to store up to 50,000 users' data. *(Justification: To ensure performance of the application is not compromised when user data increases substantially as number of users increases.)*

   3.2. The system shall be able to store past 6 months electricity consumption data of every users. *(Justification: The historical data will be used in some of the system features including usage prediction and recommendations.)*

   3.3. The system shall have adequate storage for daily backups of user data. *(Justification: Sufficient storage will ensure backups are stored safely, this is necessary during data recovery.)*

4. Dependability Requirements

4.1. The system shall maintain an uptime of 99.9% annually. *(Justification: The 99.9% uptime is equivalent to a maximum downtime of 8.76 hours per year, this ensures minimal disruption and enhance user satisfaction.)*

4.2. The system shall automatically backup user data daily. *(Justification: Regular backups can reduce the risk of data loss when system failure occurs, so that user data can be recovered when needed.)*

4.3. The system shall have a disaster recovery strategy that guarantee recovery in 4 hours when a significant system failure occurs. *(Justification: To minimize the impact of disastrous events and maintain user trust.)*

5. Scalability Requirements

5.1. The system shall accommodate up to 50,000 users without experiencing any malfunction in performance. *(Justification: To ensure the system is able to handle large number of users, particularly during high usage period, while maintaining the system reliability and preventing downtime.)*

6. Compatibility Requirements

6.1. The application shall be able to operate on multiple mobile operating systems, including Android, iOS, EMUI and HarmonyOS with their respective versions stated below. *(Justification: To encourage users to use the application by providing different types and versions of compatible operating systems.)*

6.1.1. The application shall be able to function smoothly with:

- Android smartphone running on Android OS versions 12, 12.1, 13 and 14.
- iPhone devices running on iOS versions 14, 15, 16, 17 and 18.
- Huawei smartphone running on:
  - EMUI versions 10, 11, 12 and 13.
  - HarmonyOS versions 2.0, 3.0, 3.1, 4.0, 4.2.

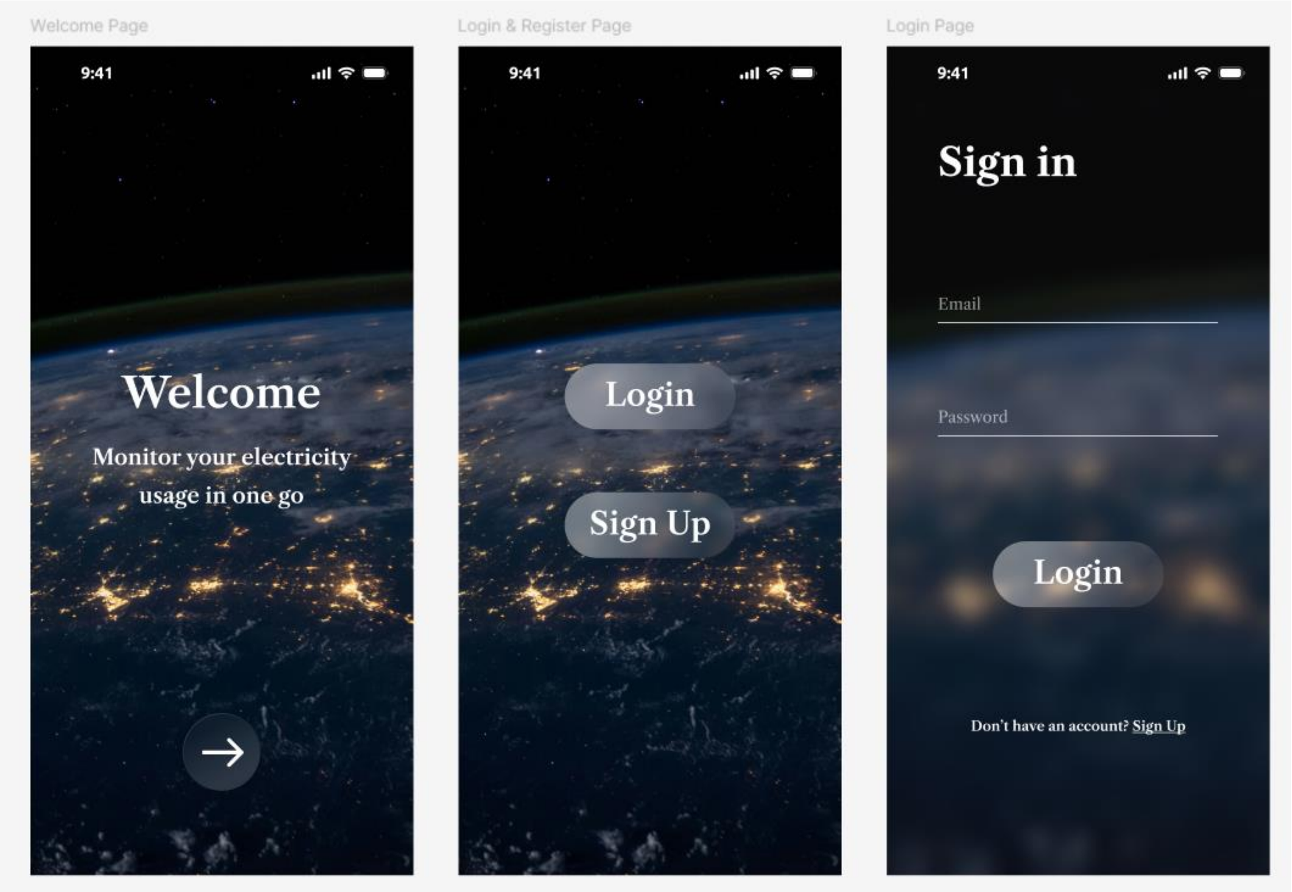
7. Security Requirements

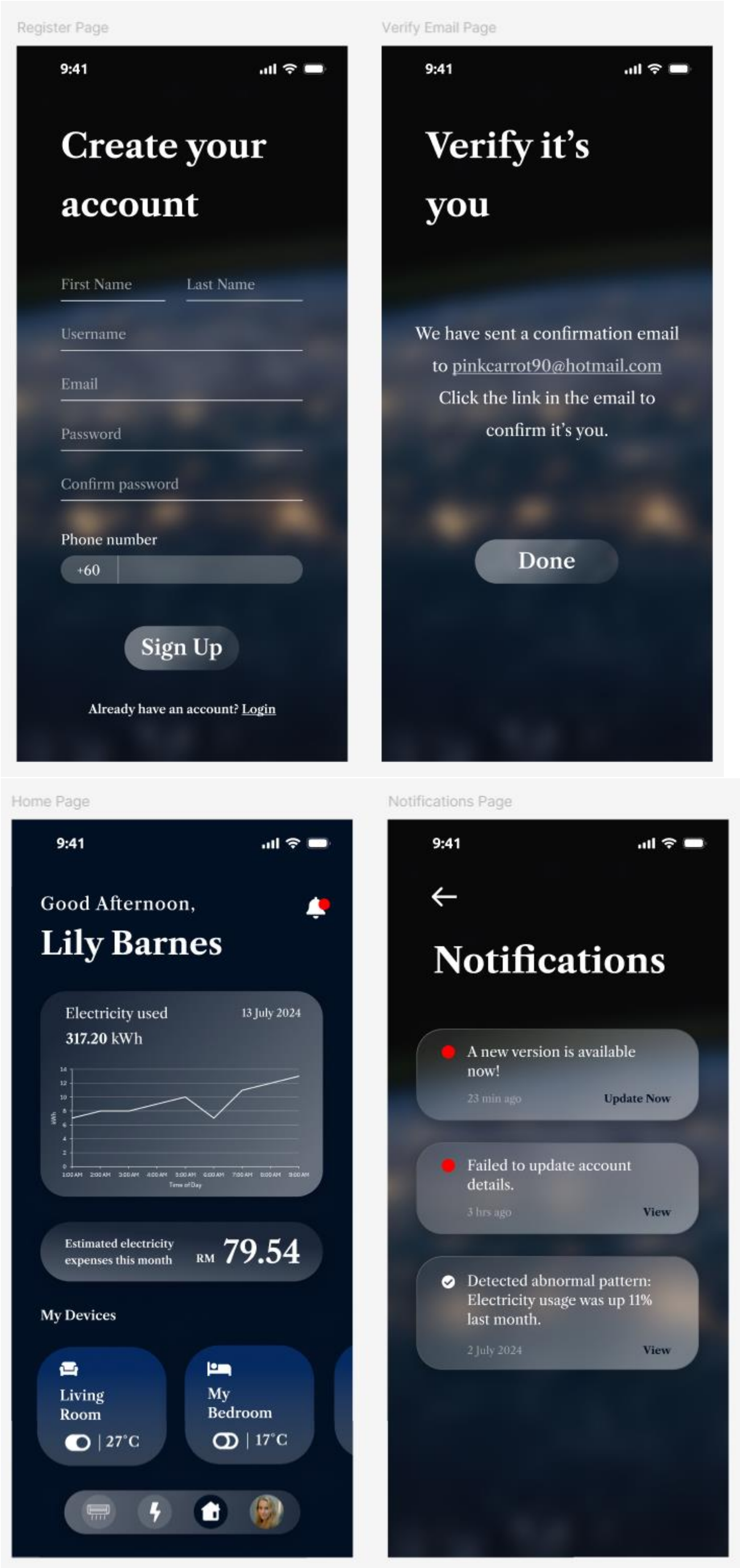
7.1. The system shall ensure the user data stored and being transferred over the network is encrypted using Advanced Encryption Standard (AES), an industry-standard encryption protocol. *(Justification: The users' privacy shall be protected and the utilization of user data shall comply to data protection regulations.)*

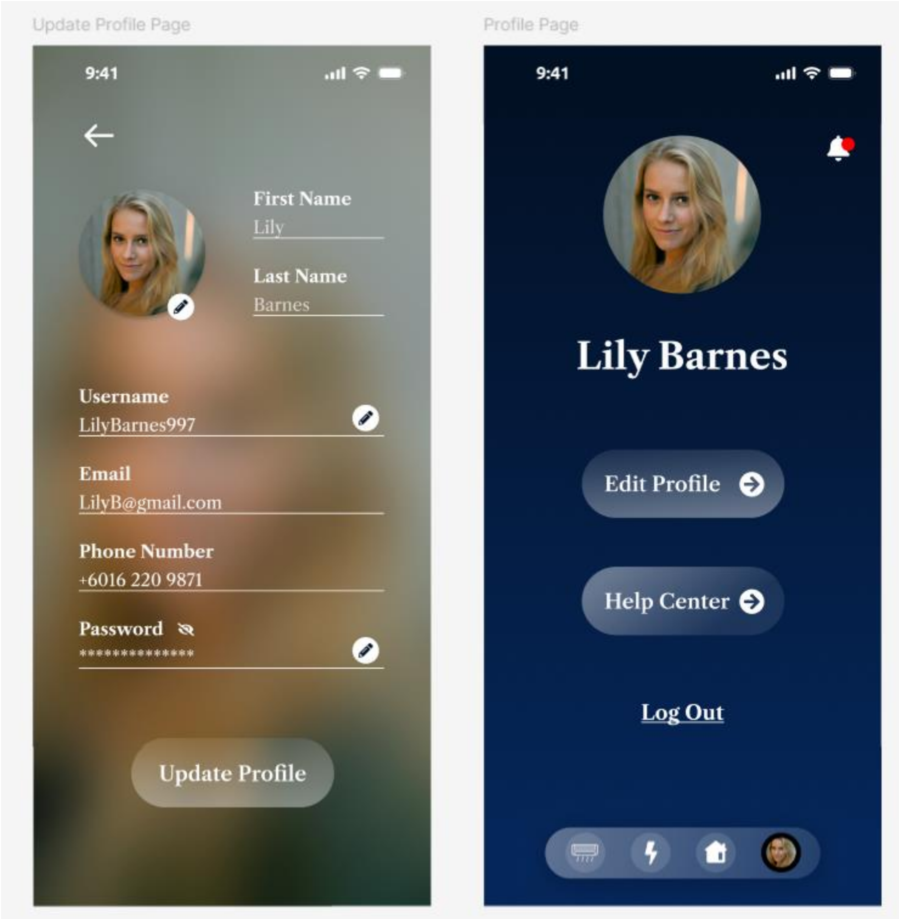
Prototype Design

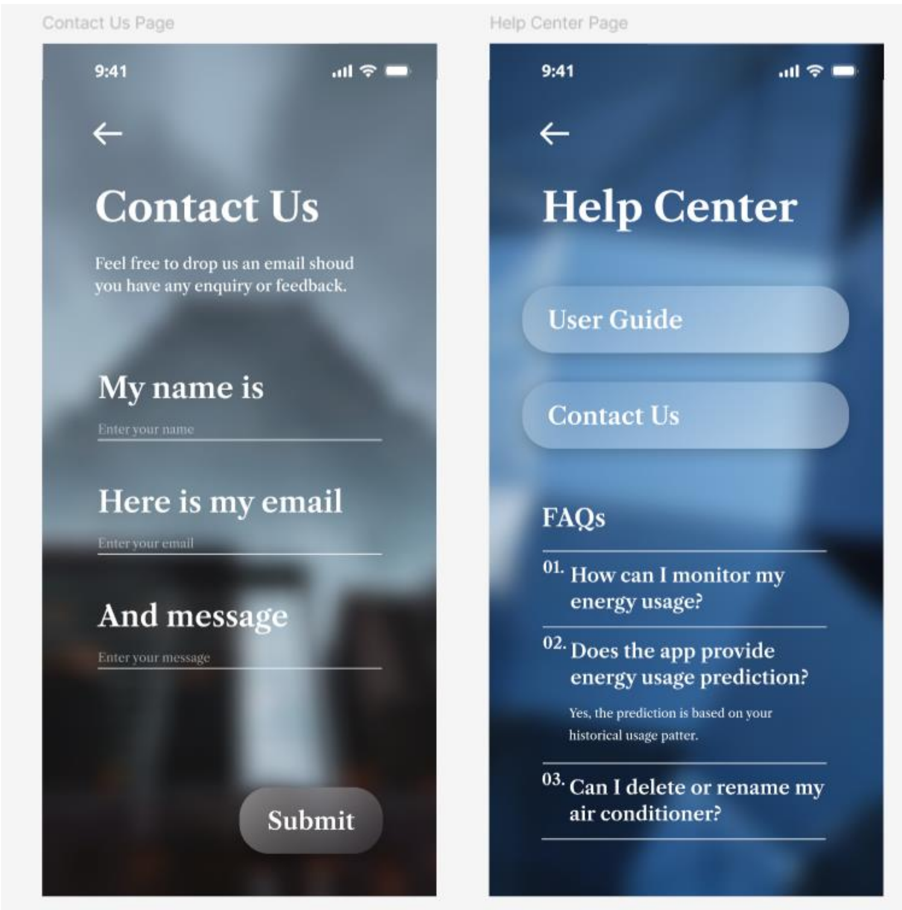
Figma file link: https://www.figma.com/design/H3TGLCFTtMBM87CA6rUgzs/SEMS-Mobile-App?node-id=1-809&t=zMu0SG8BBwTuaXaP-1

Figma prototype link: https://www.figma.com/proto/H3TGLCFTtMBM87CA6rUgzs/SEMS-Mobile-App?t=zMu0SG8BBwTuaXaP-1

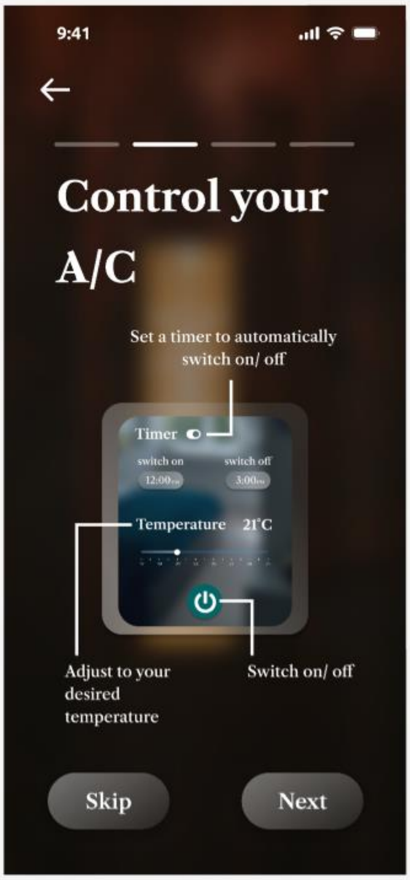



Register Page
9:41
Create your account
First Name
Last Name
Username
Email
Password
Confirm password
Phone number
+60
Sign Up
Already have an account? Login
Verify Email Page
9:41
Verify it's you
We have sent a confirmation email to pinkcarrot90@hotmail.com Click the link in the email to confirm it's you.
Done
Home Page
9:41
Good Afternoon,
Lily Barnes
Electricity used
13 July 2024
317.20 kWh
Estimated electricity expenses this month
RM 79.54
My Devices
Living Room
27°C
My Bedroom
17°C
Notifications Page
9:41
Notifications
A new version is available now!
23 min ago
Update Now
Failed to update account details.
3 hrs ago
View
Detected abnormal pattern: Electricity usage was up 11% last month.
2 July 2024
View


Update Profile Page
9:41
First Name
Lily
Last Name
Barnes
Username
LilyBarnes997
Email
LilyB@gmail.com
Phone Number
+6016 220 9871
Password
****
Update Profile
Profile Page
9:41
Lily Barnes
Edit Profile
Help Center
Log Out

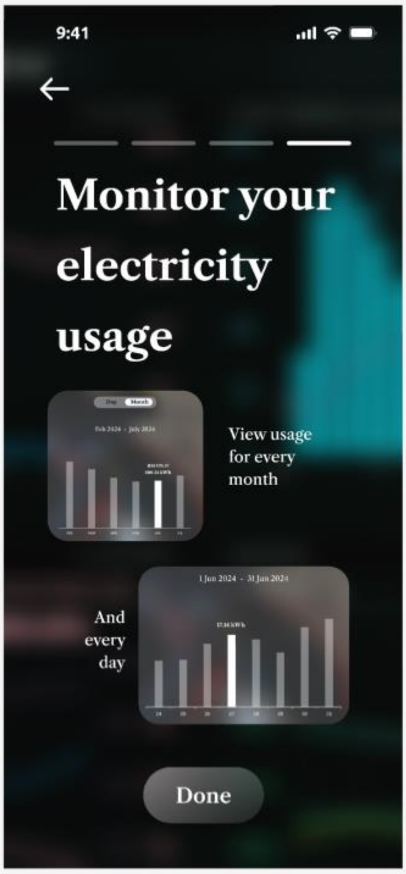

Contact Us Page
9:41
Contact Us
Feel free to drop us an email shoud you have any enquiry or feedback.
My name is
Enter your name
Here is my email
Enter your email
And message
Enter your message
Submit
Help Center Page
9:41
Help Center
User Guide
Contact Us
FAQs
01. How can I monitor my energy usage?
02. Does the app provide energy usage prediction?
Yes, the prediction is based on your historical usage patter.
03. Can I delete or rename my air conditioner?

User Guide Page 2 - Control A/C

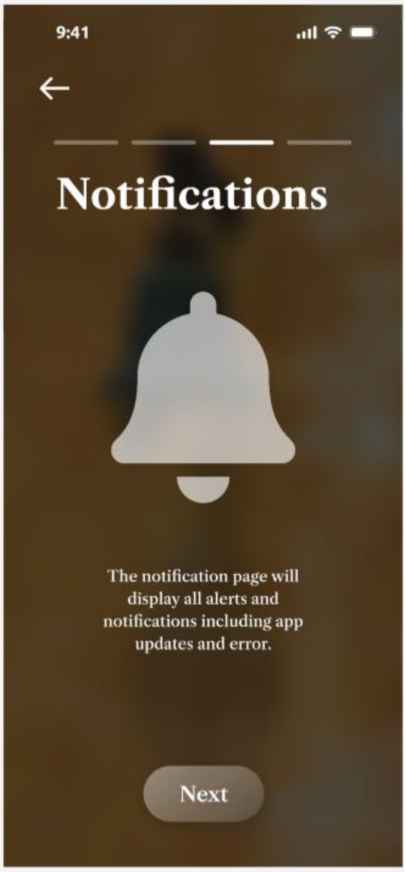


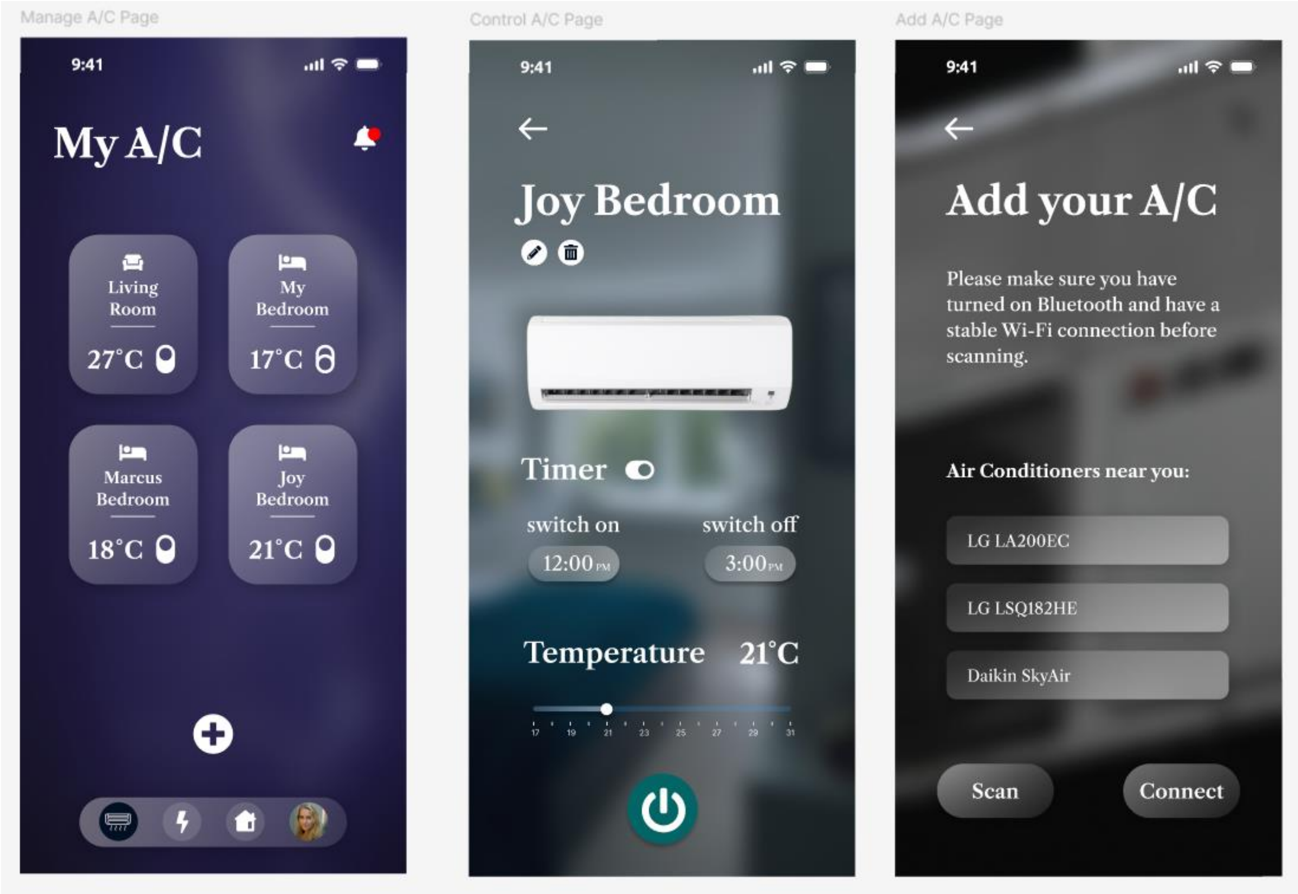
User Guide Page 1 - Add A/C

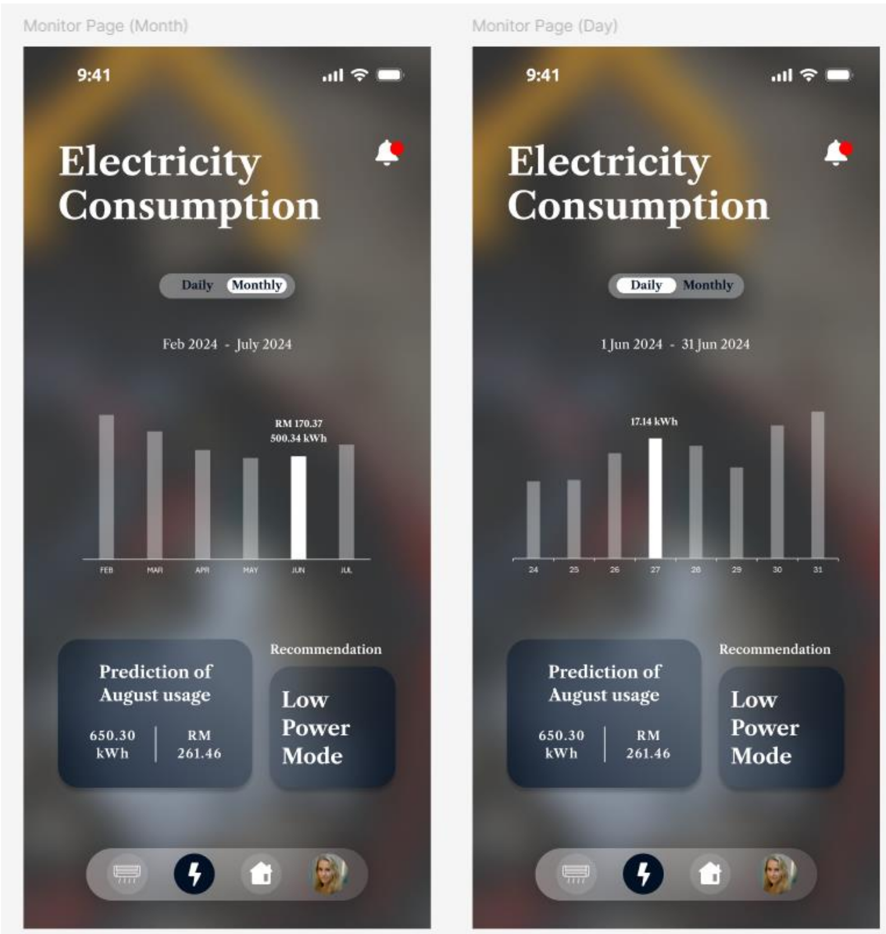
User Guide Page 4 - Monitor



User Guide Page 3 - Notifications




Manage A/C Page
9:41
My A/C
Living Room
27°C
My Bedroom
17°C
Marcus Bedroom
18°C
Joy Bedroom
21°C
Control A/C Page
9:41
Joy Bedroom
Timer
switch on
12:00 PM
switch off
3:00 PM
Temperature 21°C
17 19 21 23 25 27 29 31
Add A/C Page
9:41
Add your A/C
Please make sure you have turned on Bluetooth and have a stable Wi-Fi connection before scanning.
Air Conditioners near you:
LG LA200EC
LG LSQ182HE
Daikin SkyAir
Scan
Connect


Monitor Page (Month)
9:41
Electricity Consumption
Daily Monthly
Feb 2024 - July 2024
RM 170.37
500.34 kWh
FEB MAR APR MAY JUN JUL
Prediction of August usage
650.30 kWh
RM 261.46
Recommendation
Low Power Mode
Monitor Page (Day)
9:41
Electricity Consumption
Daily Monthly
1 Jun 2024 - 31 Jun 2024
17.14 kWh
24 25 26 27 28 29 30 31
Prediction of August usage
650.30 kWh
RM 261.46
Recommendation
Low Power Mode

UML Diagrams

Use case diagram

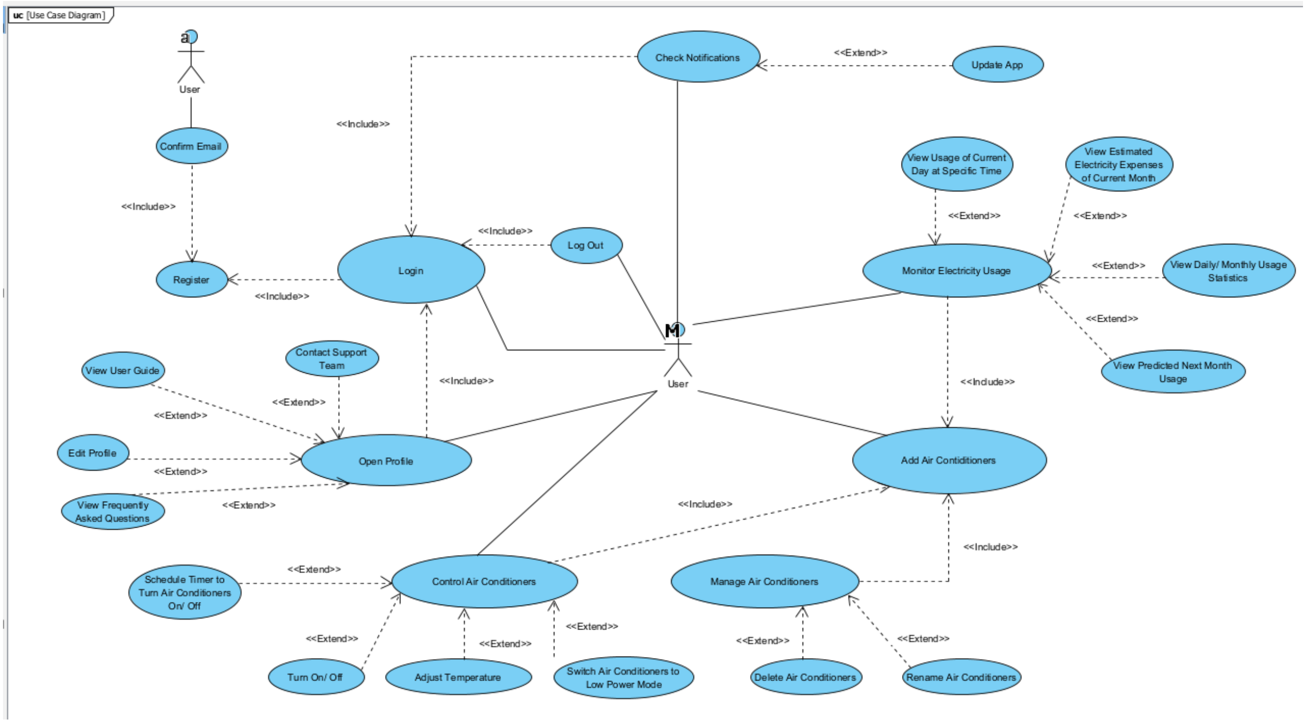


Figure 7: Use case diagram

The use case diagram shown in figure 7 includes 1 actor which is the user of the mobile application. One the user is labelled **M**, which indicates it is a master view and is the existing user of the application. On the other hand, another user with the label **a**, indicates that it is the auxiliary view and is the new user. Some of the major functions that a user can perform are register, confirm email, login, log out, check notification, monitor electricity usage, add air conditioners, manage air conditioners, control air conditioners and open profile. Their relationships and other extend functions are further illustrated in figure 7.

Class diagram

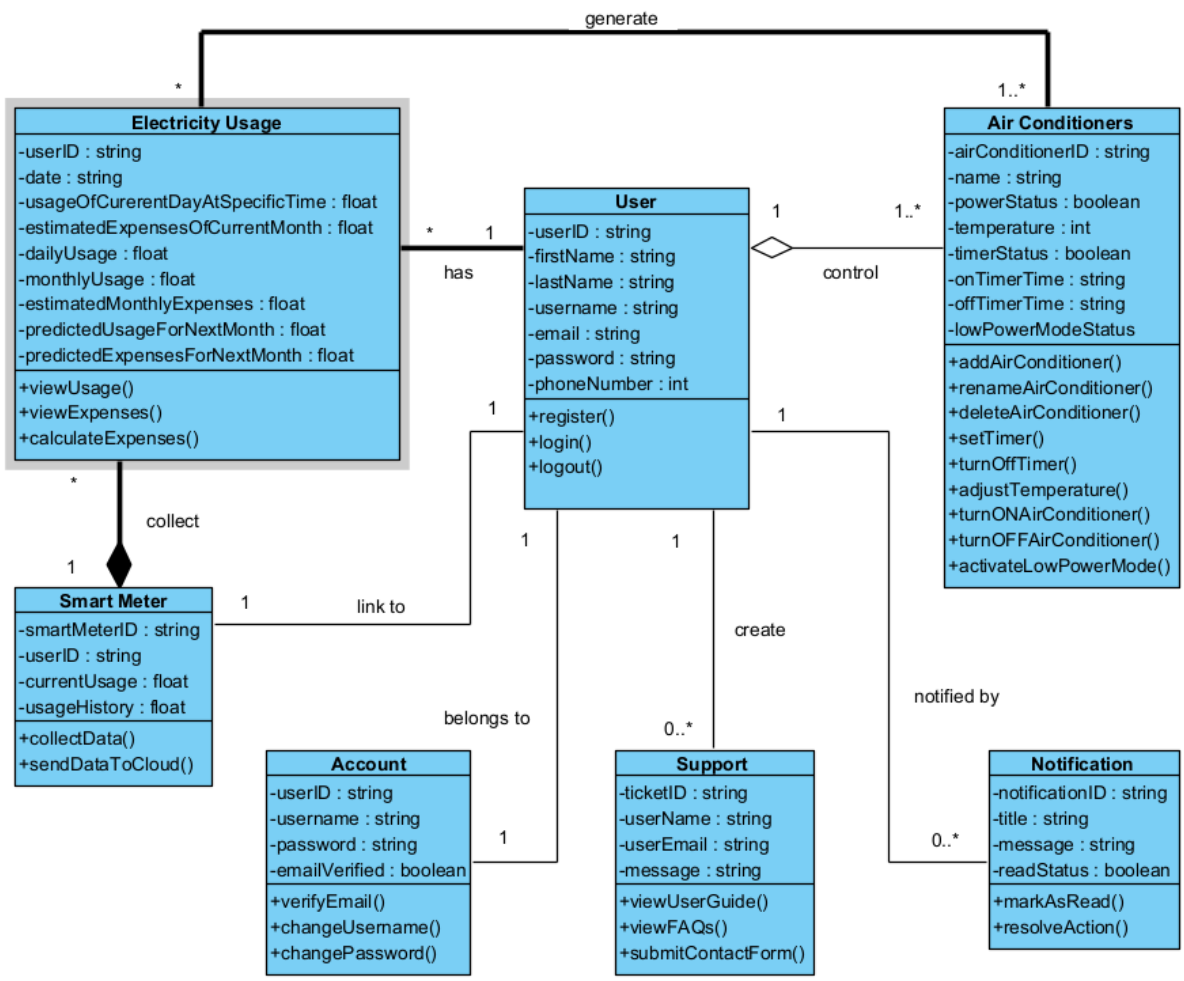


Figure 8: Class diagram

Figure 8 shows the class diagram of the SEMS mobile application. Some of the classes defined are user, electricity usage, air conditioners, smart meter, account, support and notification. Their attributes, operations, and relationships be seen in figure 8.

Sequence diagram

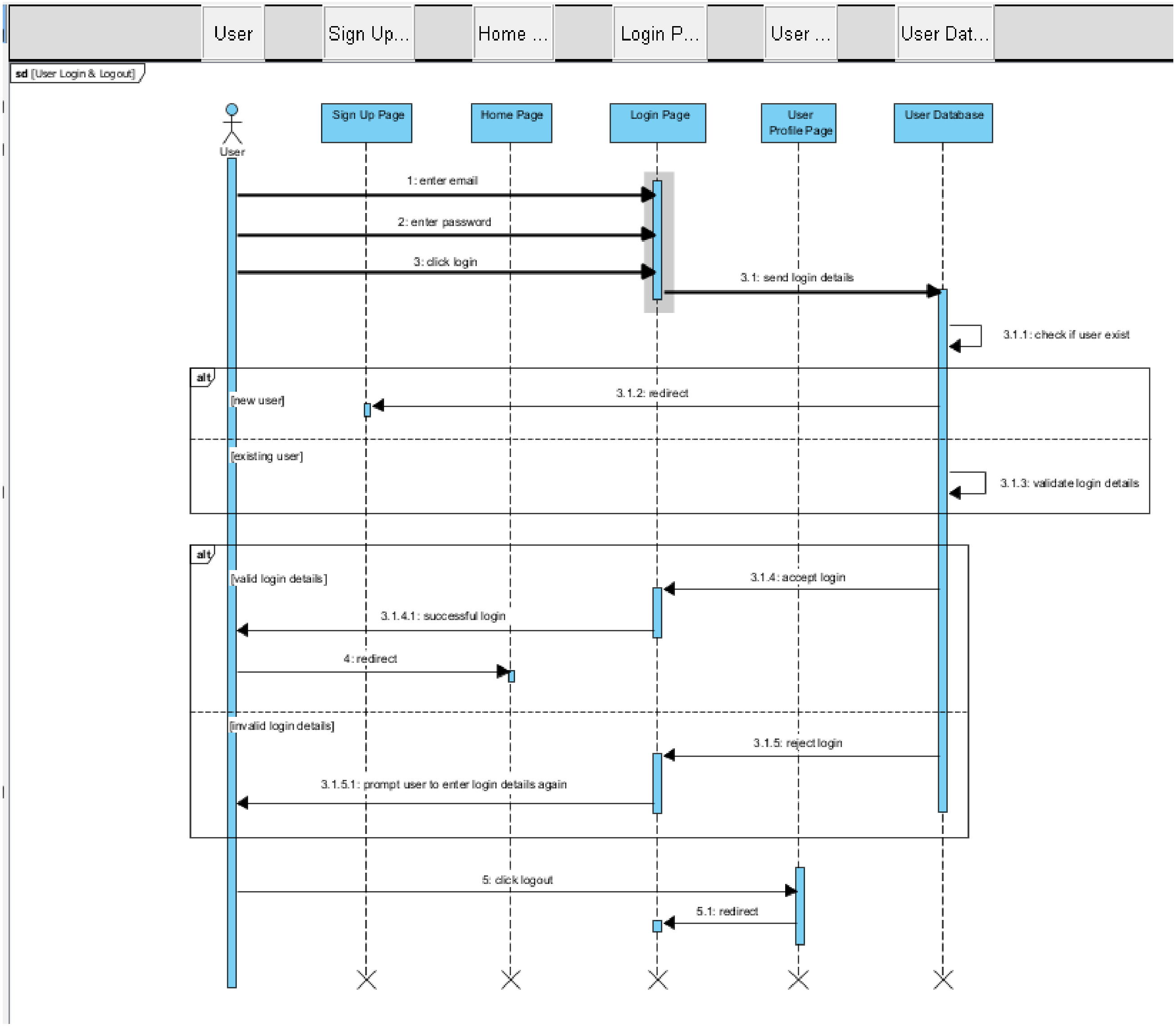


Figure 9.1: Sequence diagram for user login and logout

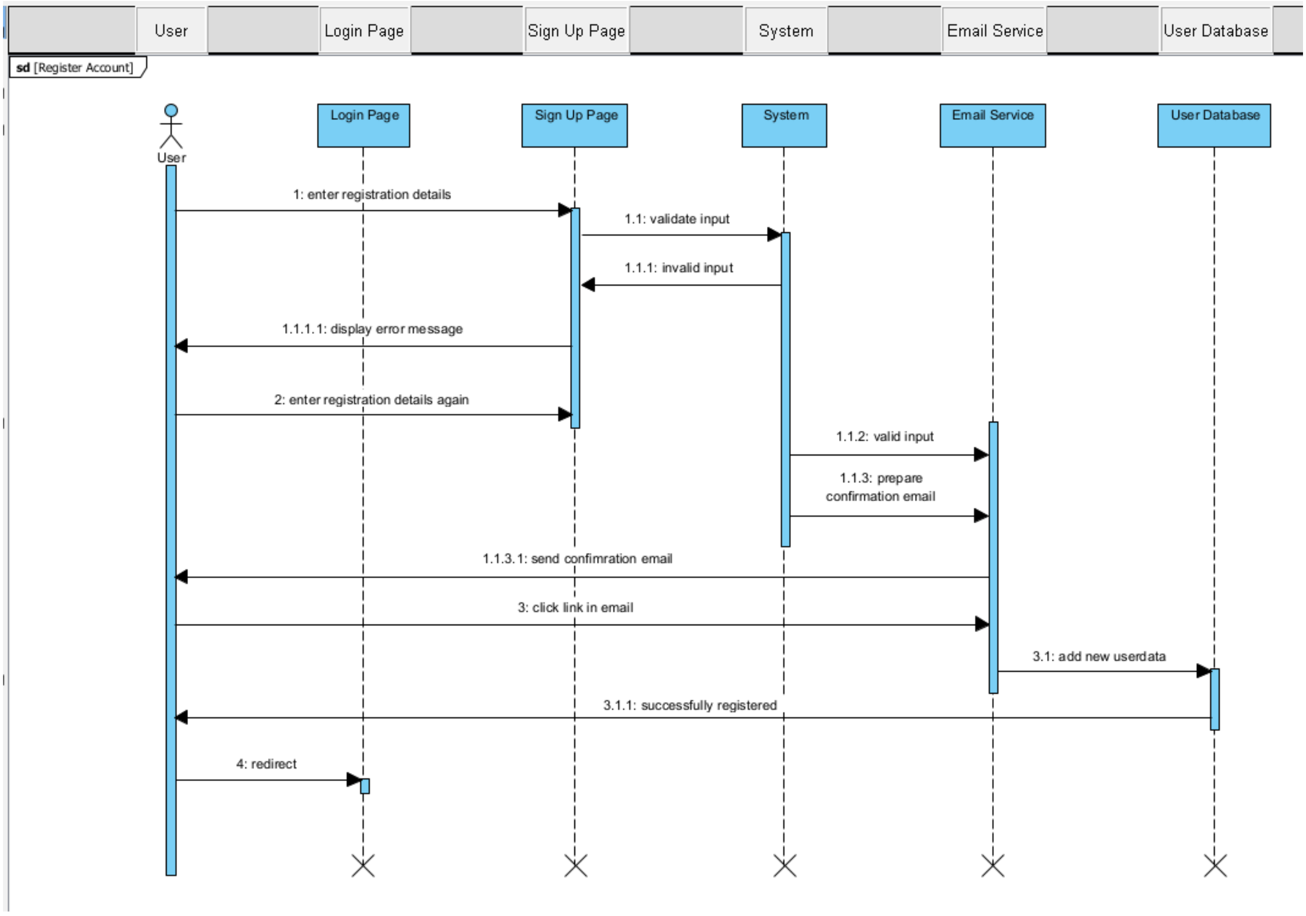


Figure 9.2: Sequence diagram for register account

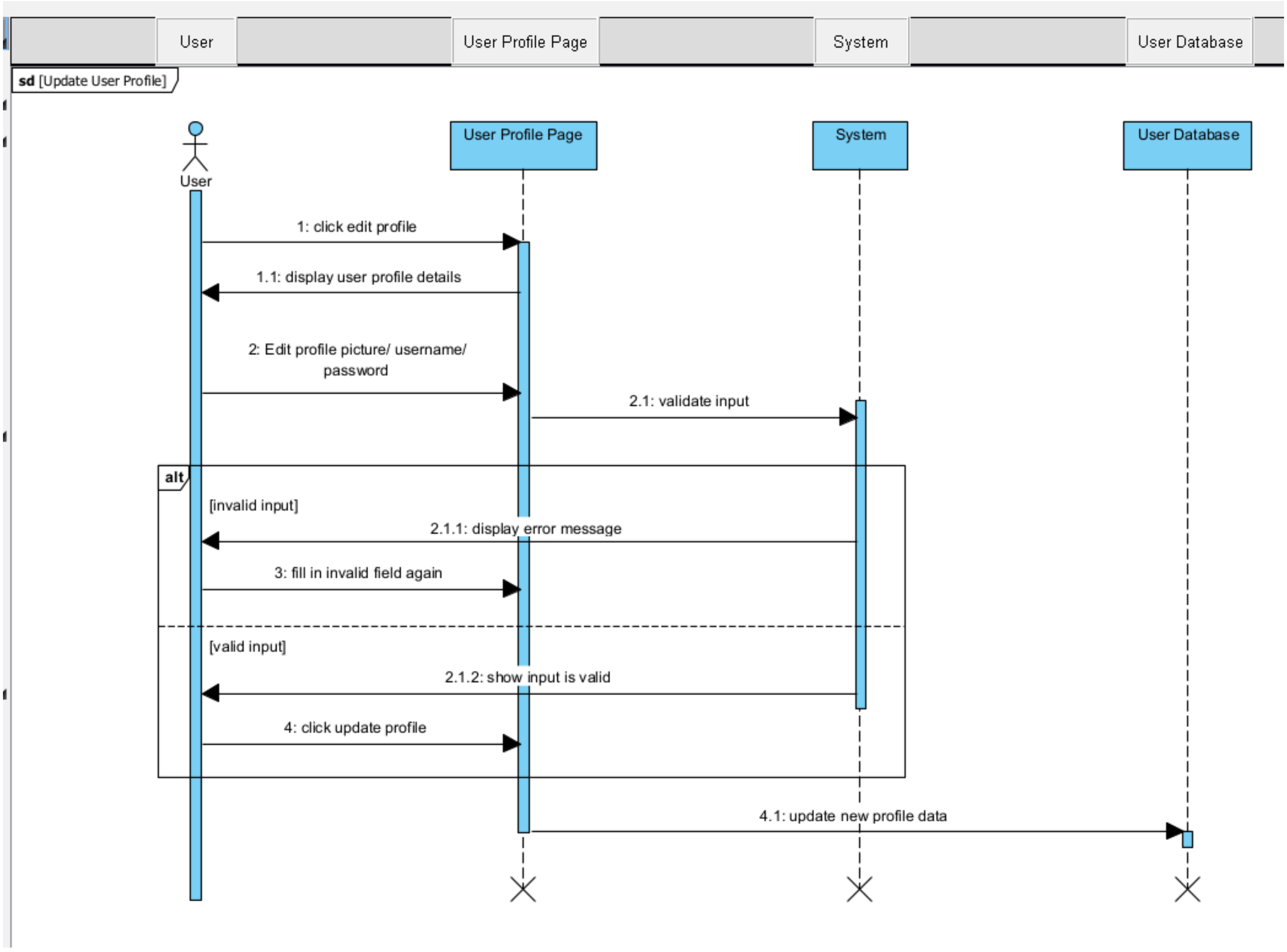


Figure 9.3: Sequence diagram for update user profile

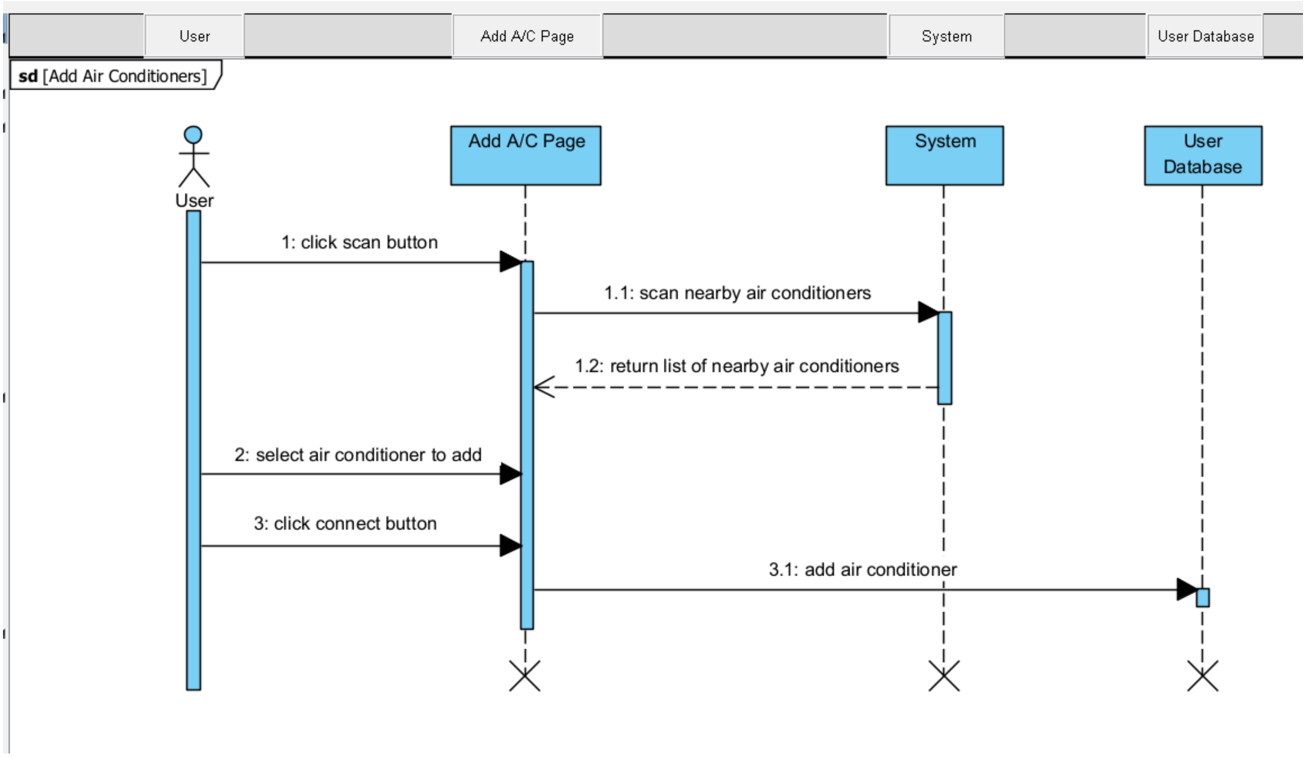


Figure 9.4: Sequence diagram for add air conditioners

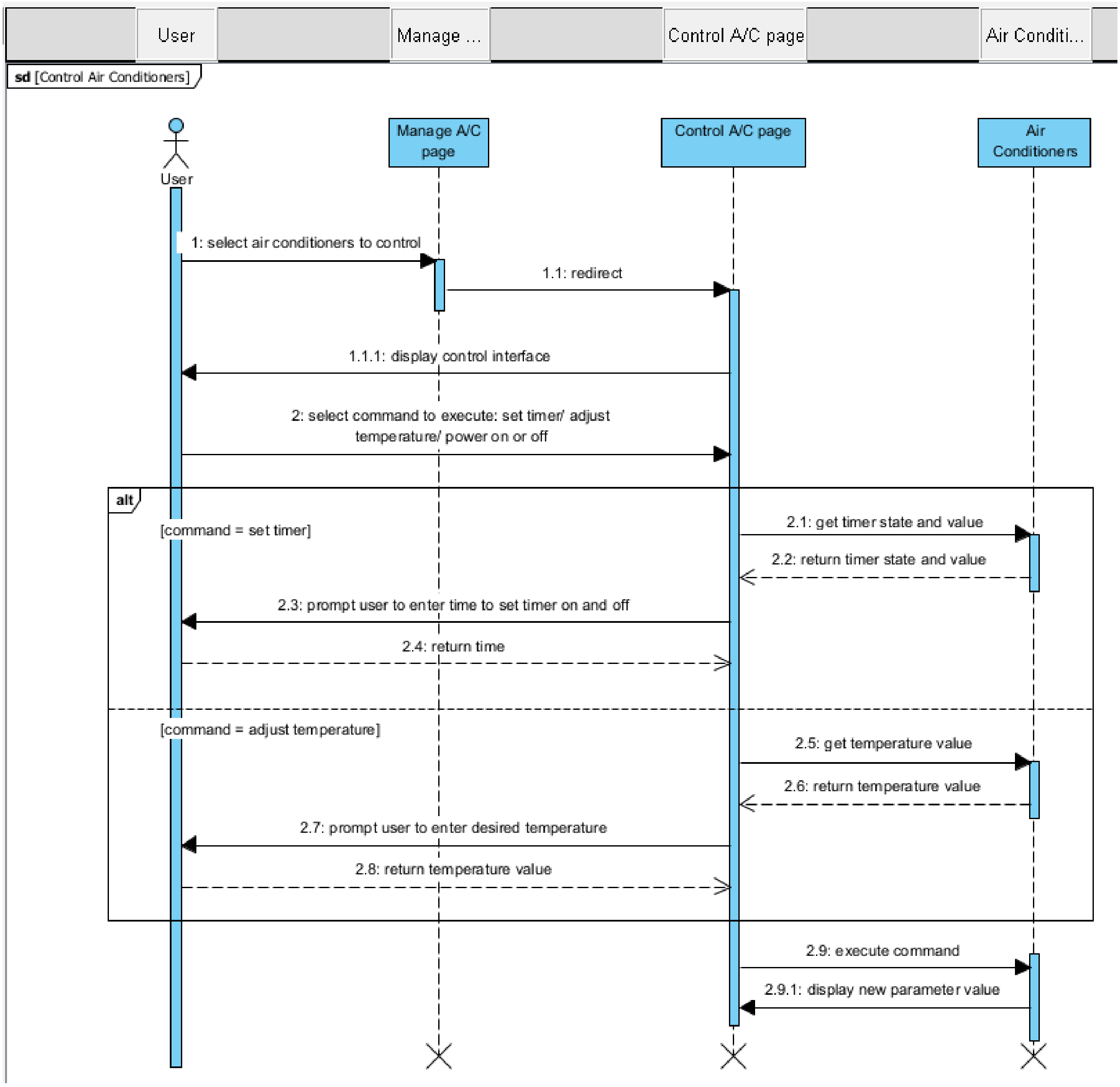


Figure 9.5: Sequence diagram for control air conditioners

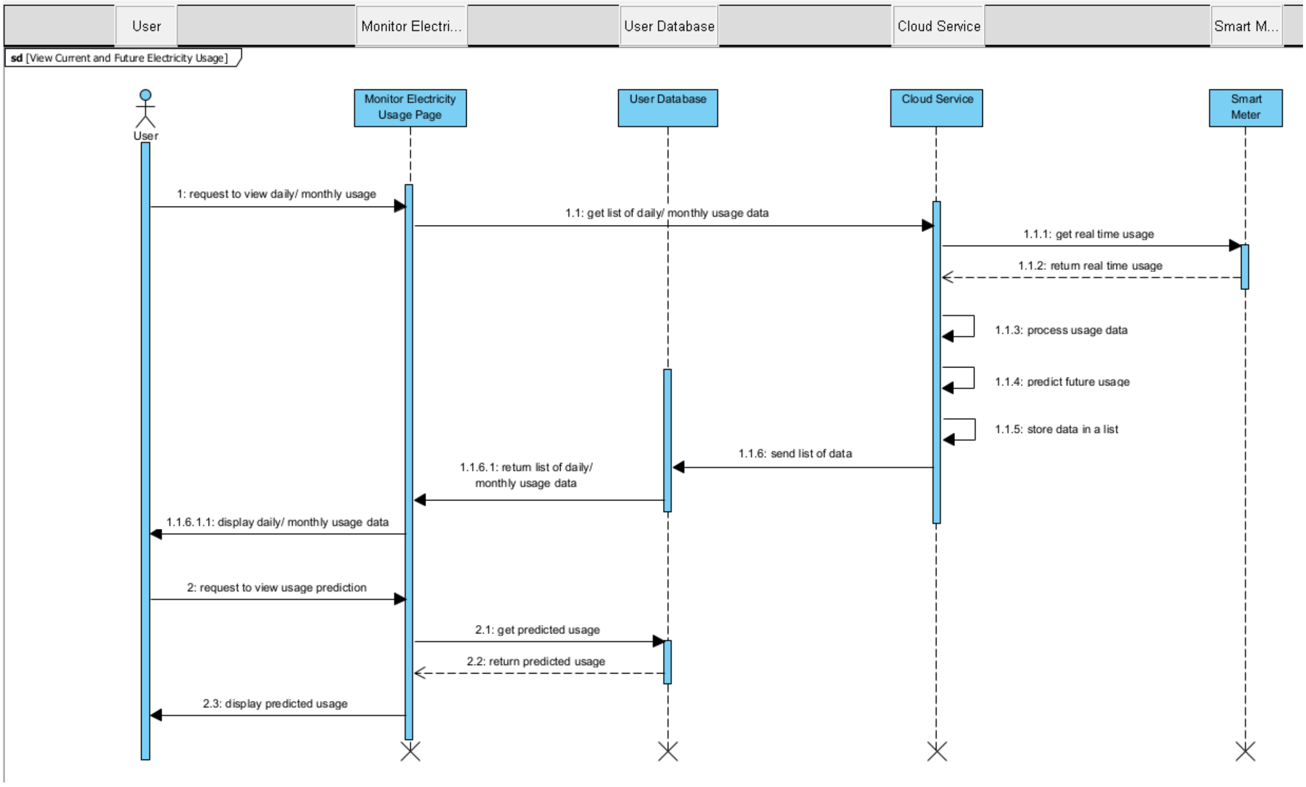


Figure 9.6: Sequence diagram for view current and future electricity usage

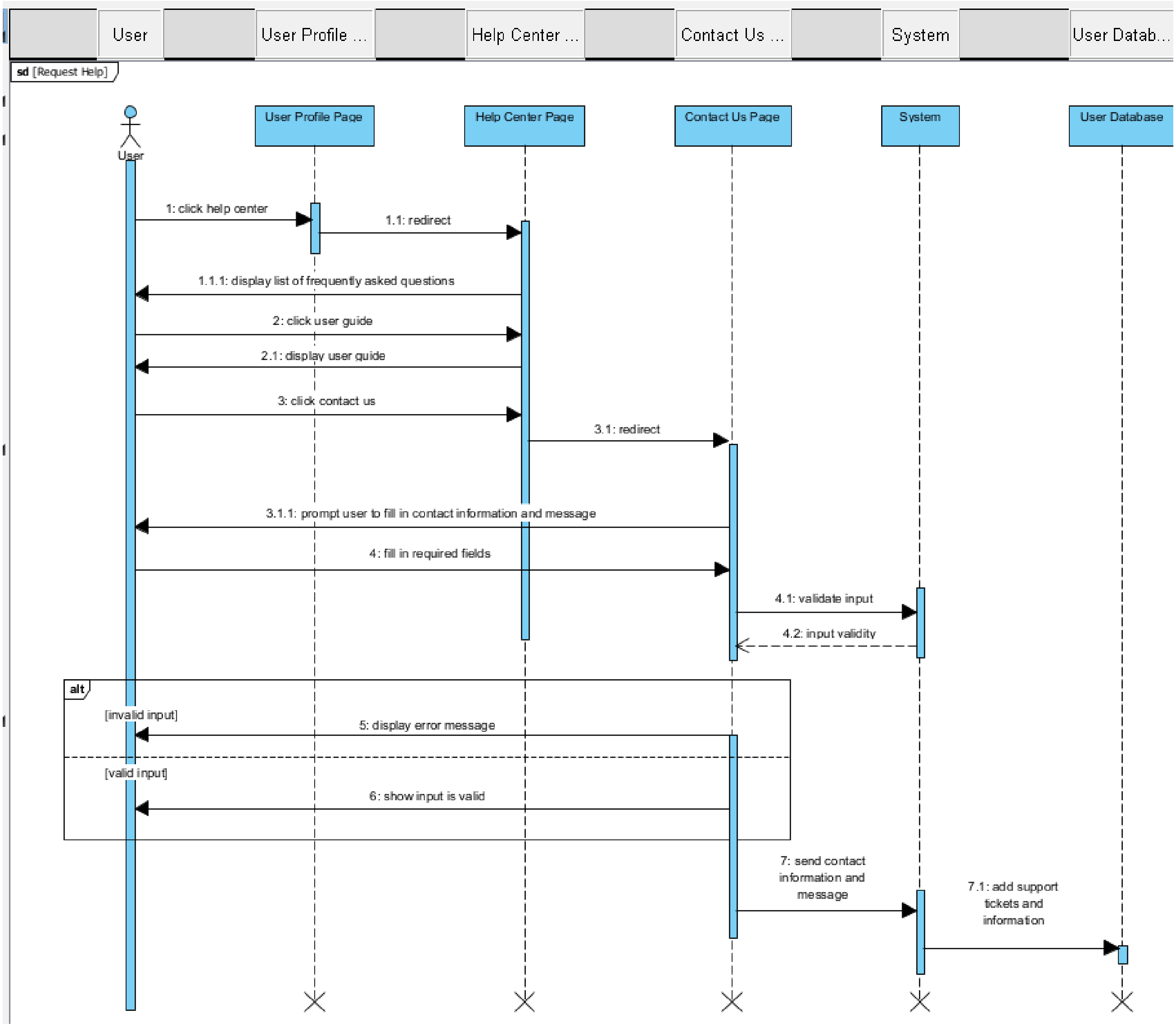


Figure 9.7: Sequence diagram for request help

Figure 9.1 to figure 9.7 show the sequence diagrams for main functions of the SEMS mobile application. The main functions are login & logout, register account, update user profile, add air conditioners, control air conditioners, view current and future electricity usage and request help. Each of the sequence diagram has only one actor and have at least three objects depending on the type of functions.

Activity diagram

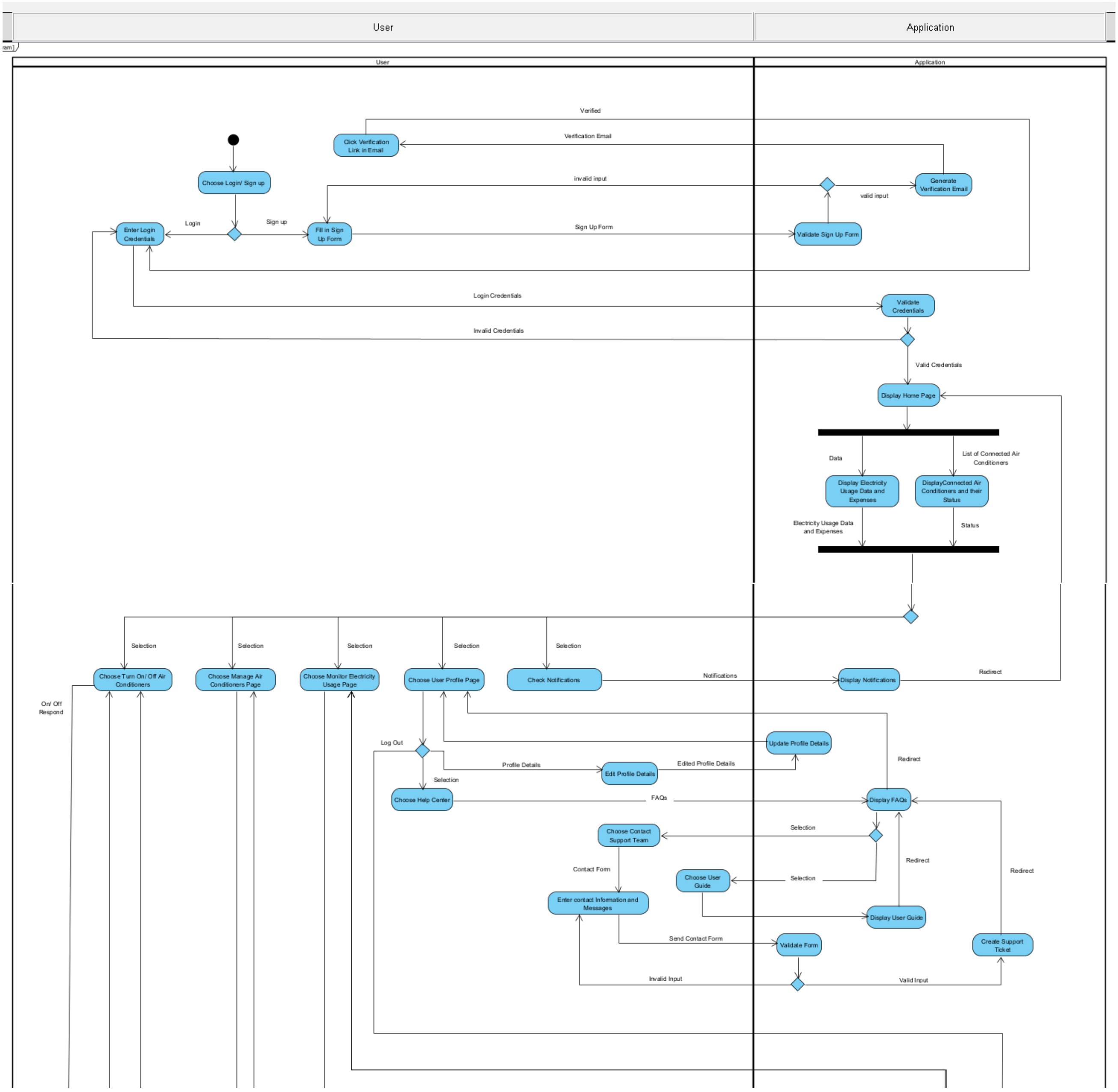


Figure 10.1: Activity diagram part 1

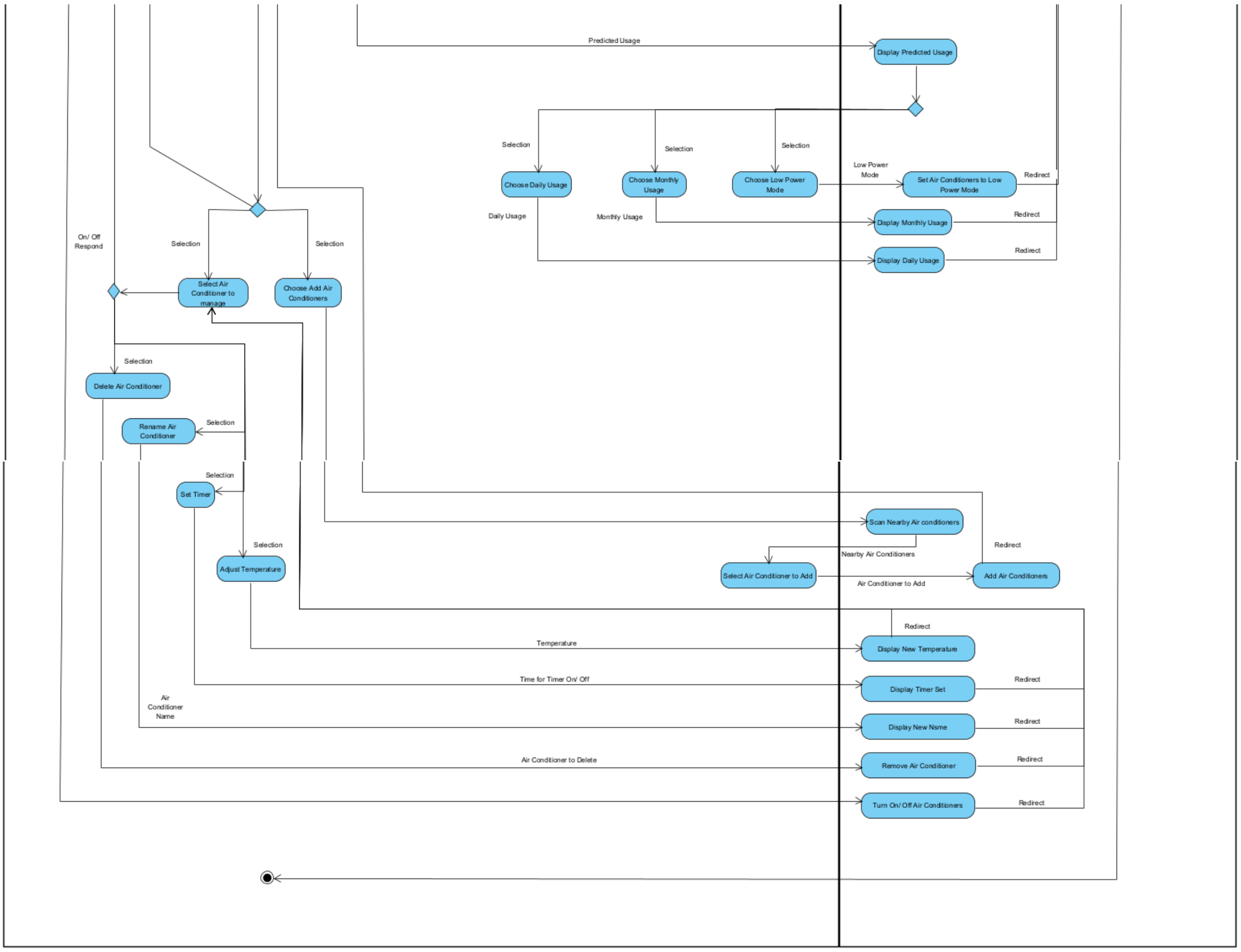


Figure 10.2: Activity diagram part 2

Figure 10.1 and figure 10.2 illustrate the activity diagram of the SEMS mobile application. The activity diagram has two swimlanes, namely user and application. In the activity diagram, there is a start node to indicate the beginning of the application flow, and an end node to show the ending of the application execution flow. The flow of the functions are drawn in between the two nodes.

References

Elazar, E. (2018, April 23). *Panaya*. Retrieved from https://www.panaya.com/blog/testing/what-is-uat-testing/#:~:text=User%20Acceptance%20Testing%20(UAT)%2C: https://www.panaya.com/blog/testing/what-is-uat-testing/#:~:text=User%20Acceptance%20Testing%20(UAT)%2C

GeeksforGeeks. (2024, March 15). *GeeksforGeeks*. Retrieved from Incremental Process Model – Software Engineering: https://www.geeksforgeeks.org/software-engineering-incremental-process-model/

Hussain, I. (2020, Febuary 28). *Optimizing Energy Consumprion in the Home Energy Management System via a Bio-Inpired Dragonfly Algorithm and the Genetic Algortihm*. Retrieved from MDPI: https://www.mdpi.com/2079-9292/9/3/406

Javatpoint. (2011). *Javatpoint*. Retrieved from Incremental Model: https://www.javatpoint.com/software-engineering-incremental-model

Sommerville, I. (2016). Software Engineering . *Pearson Education*, 52-53.

Tiwari, S. (n.d.). *A Methodology for the Selection of Requirement Elicitation Techniques*. Retrieved from ar5iv.labs: https://ar5iv.labs.arxiv.org/html/1709.08481

Appendix

1. Compatible air conditioners:

1.	Acson Avory Premium Inverter A3WMY10APF
2.	Acson Moveo
3.	Acson Slim Duct
4.	Amana ASX13
5.	Aux ASW-H12B4/FAR1DI-US
6.	Blue Star 5HW18MA1
7.	Bosch Climate 5000 RAC
8.	Btu AZAI12EAC
9.	Carrier 42QHC012
10.	Carrier 42VZL025
11.	Daikin FTKF Series
12.	Daikin Inverter Smile FTKQ
13.	Daikin SkyAir
14.	Daikin Urusara 7 FTXZ-N
15.	Electrolux ESV12CRR-C4
16.	Friedrich Chill Premier
17.	Friedrich Kuhl
18.	Fujitsu ASTG18KMCA
19.	Fujitsu General ASYG12LMCE
20.	Gree Amber 9k BTU
21.	Gree Bora
22.	Gree Sapphire 24k BTU
23.	Haier HSU-12LEK03
24.	Haier HSU-19CIF03T
25.	Haier Tundra
26.	Hisense AN09DJ1G
27.	Hisense AP1419CW1G
28.	Hisense AS-18TR4RBBTV00
29.	Hisense Inverter AI10KAGS
30.	Hitachi RAS-X18
31.	Hitachi RPK-5.0FSN1Q
32.	LG Dual Inverter Air Conditioner S3-Q12JA2PA
33.	LG LA200EC
34.	LG LSQ182HE
35.	Midea MAW10R1BWT
36.	Midea MSXS19
37.	Mitsubishi Electric MSY-GL09NA
38.	Mitsubishi SRK50ZSX-W
39.	Panasonic CS/CU-PU9VKZ
40.	Panasonic CS/CU-Z12XKR
41.	Panasonic CS-PN9WKH
42.	Panasonic Inverter Deluxe Aero Series
43.	Samsung AR12TXFCAWK
44.	Samsung AR24TYGCDWKNTC

45. Samsung Wind-Free™ AR9500T
46. Sharp AH-XP13SHV
47. Sharp AH-XP18UHD
48. Sharp J-Tech Inverter
49. TCL Elite Series
50. Toshiba RAS-10BKCV-A
51. Toshiba RAS-24UKCV-E
52. Toshiba Seiya RAS-B10J2KVG-E
53. Trane XR16
54. Voltas 185V JZJ
55. Whirlpool SPIW418L
56. Xiaomi Mi 1.5HP Inverter AC
57. York YHFE09ZE
58. York YVKC09B17
59. Zephyr ZAC20
60. AUX Freedom
61. Bosch Climate Class 6000i
62. Beko BBLPG121
63. Beuhring Brilliant
64. CHIGO CS-25V3A-M2H2
65. CLIVET CRF1
66. Convair Magicool Platinum
67. Dometic Harrier Inverter
68. Electra EAC-LD12
69. Eurom AC2400
70. GE Appliances APER05LY
71. Godrej GSC 18
72. Goodman GSX16
73. Green Dot GD1500
74. Haier HSU-10LEK03T3
75. Kelvinator KSD25HWJ
76. Kenwood KW09
77. Klimaire KSIF012-H115-S
78. LLOYD LS18I32AF
79. Olympia Splendid Unico
80. Onida IR183RHO
81. Pioneer WYS012-17
82. Polaris PAC-11-010
83. Qlima S5291
84. Rowa RAC-10
85. Sansui SFS0921
86. Severin SC7145
87. Symphony Jumbo
88. TCL TAC-12CHSA/XA71
89. Tecnogas TAS-12HR
90. Toyotomi TAD-T40DW
91. Vestfrost WN12T

92.	Voltas SAC 153 CY
93.	Whirlpool PACW29COL
94.	White-Westinghouse WWB12A4G
95.	Whynter ARC-12S
96.	Winia WNAC-20
97.	Xtreme XAC-10
98.	York YHGE14ZE
99.	Zamil ZSD48H
100.	Zenith ZTH-09

Table 2. List of compatible air conditioners