Home Automation System

Final Year Project

Session 2020-2024

A project submitted in partial fulfillment of the degree of

BS in Computer Science

Department of Computer Science

Faculty of Computer Science & Information Technology

Superior University Lahore

Spring 2024

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R**&**D**
Area of specialization	Development & Integration
FYP ID	FYP-BCSM-F23-015

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	**BCSM-F20-167**	**Hafiz Mian Muhammad Raza Fareed**	bcsm-f20-167@superior.edu.pk	
(ii)	**BCSM-F20-243**	**Abdullah Asad Javaid**	bcsm-f20-243@superior.edu.pk	
(iii)	**BCSM-F20-213**	**Usama Bin Zaman**	bcsm-f20-213@superior.edu.pk	

Plagiarism Free Certificate

This is to certify that, I am Hafiz Mian M. Raza Fareed S/D of Mian Muhammad Fareed, group leader of FYP under registration no FYP-BCSM-F23-015 at Computer Science Department, The Superior University Gold Campus, Lahore. I declare that my FYP report is checked by my supervisor.

Date: _____ Name of Group Leader: Hafiz Mian M. Raza Fareed Signature: _____

Name of Supervisor: Mr. Saleem Mustafa
Designation: Lecturer
Signature: _____

Co-Supervisor: Mr. Saleem Mustafa
Designation: Lecturer
Signature: _____

HOD: Dr. Irfan
Signature: _____

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

This Home Automation System project is dedicated to Mr. Saleem Mustafa, my amazing instructor, whose support and direction have been like a beacon of light throughout this journey. This project would not have been feasible without your help, and your lessons have stoked my interest in development. We appreciate your passion in sharing your knowledge and your role as an inspirational mentor. This project is an expression of everything you have taught me and the abilities. I have developed with your help. I appreciate your continuous guidance and support.

Acknowledgements

First I express my heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete the final year project successfully.

I sincerely appreciate and want to express my gratitude to Supervisor Mr. Saleem Mustafa, Lecturer, Department of CS Superior University Gold Campus, Lahore. Deep Knowledge & keen interest of my supervisor in the field of “Home Automation System” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

My sincere appreciation goes out to Mr. Jawad Ahmed Butt and Head of the Computer Science Department, as well as other faculty members of Superior University Gold Campus, Lahore, for their helpful assistance in completing our research.

I want to thank everyone of my Superior University classmates who participated in this discussion while finishing the material. Finally, I must respectfully appreciate our parents unwavering patience and support.

Executive Summary

In today's fast-paced world, the demand for convenience, efficiency, and security in our homes has never been greater. The Home Automation System, herein referred to as the "System," is poised to transform the way people interact with and manage their living spaces. The Home Automation System is an integrated smart home solution designed to enhance the quality of life for homeowners by providing seamless control over various aspects of their homes, including lighting, temperature and entertainment.

Features & Benefits:

- Centralized Control
- Energy Efficiency
- Customization
- Convenience
- Energy Savings
- Peace of Mind
- Increased Property Value
- Customized Living

The Home Automation System is well-positioned to tap into this expanding market, offering a comprehensive solution with a focus on user-friendliness and energy efficiency. This System represents a game changing solution for modern homeowners seeking convenience, energy savings and customization. With the potential to increase property value and improve the overall quality of life, this system is set to redefine the way we live in and interact with our homes. By investing in this innovative technology, homeowners can enjoy a smarter, more efficient, and safer living environment.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Chapter 1: Introduction

A home automation system is a technological solution that enables automating the bulk of electronic, electrical and technology based tasks within a home. It uses a combination of hardware and software technologies that enable control and management over appliances and devices within a home. A home automation system is like a smart assistant for your house that controls things for you. We can control our home devices by using the simple and user friendly application with the feature of eco-system. Home automation system making our lives more efficient, comfortable, and environmental friendly.

1.1. Background

Home automation is all about making our homes smarter and more convenient using technology. At first, it could only perform basic functions like setting the thermostat, but as technology advanced, it became possible for the devices to communicate with one another. The concept of getting everything in your house online expanded along with technology. You can now manage lights, security cameras, and more from anywhere at any time with only your phone. The goal is to create houses that are practical, manageable, and individualized to your preferences, making living easier and more comfortable.

1.2. Motivations and Challenges

The core motivation lies in the opportunity to improve the quality of daily living. Imagine the satisfaction of creating a system that genuinely makes life more convenient, efficient, and enjoyable for homeowners. The project offers a chance to explore innovative solutions and unleash creativity. It allows for the integration of cutting-edge technologies, making the project an exciting playground for inventive ideas. The project provides a platform for significant learning and skill development. From programming and hardware integration to problem-solving, every step of the project presents an opportunity to enhance technical skills and gain valuable hands-on experience. One of the challenges is integrating various devices and technologies seamlessly. Ensuring that different components communicate effectively and work together. Designing the

user-friendly interface can be challenging. Managing costs and resources efficiently can be a challenge. Keeping the project within budget and achieving the intended functionality.

1.3. Goals and Objectives

The main objectives of our object are as follows:

1. To remotely control home appliances and monitor them.
2. To save the time and utilize the energy efficiently.

1.4. Gap Analysis

In today's busy world, managing various tasks in our homes can be challenging and time-consuming. We often forget to turn off lights and fans, or control appliances, leading to energy waste and also enhance our electricity bills. Specially, the disable person can feel trouble to manage the lights and other home appliance, due to their disability the electrical equipment remain on which can cause an accident and also the wastage of electricity and it leads to power shortage which is not suitable.

1.5. Proposed Solution

Home automation system aims to solve these issues by allowing us to control the electronic devices like fans, lights and socket’s by using our smart phone through technology. We can also provide the feature of eco system, which prove the beneficial for the disable persons. By using the home automation system, we can control our devices smartly or remotely and also save the electricity from the wastage. Home automation system making our lives more efficient, comfortable, and environmentally friendly.

1.6. Project Plan

Our project has been decomposed into the following phases:

ENVISIONED PHASE	• Information Gathering • Project Initiation
PLANNING PHASE	• Hardware & software estimation • Scheduling and distribution of task
MODELLING PHASE	• Analysis of project • Design and architechture of project
CONSTRUCTION PHASE	• Coding and hardware assembly • Testing and Debugging
DEPLYOMENT PHASE	• Delivery of project • Feedback

PP Table No 1.

1.6.1. Experimentation:

This phase involved discussions regarding necessary equipment regarding the project. The study of related already existing projects, gathering required theoretical learning. It also included figuring out the coding part, by developing simple algorithms and flowcharts to design the whole process.

1.6.2. Design:

This phase is designing layout of the application, and the necessary features to be included. This involved the complete hardware assembly and installing the code to Node.

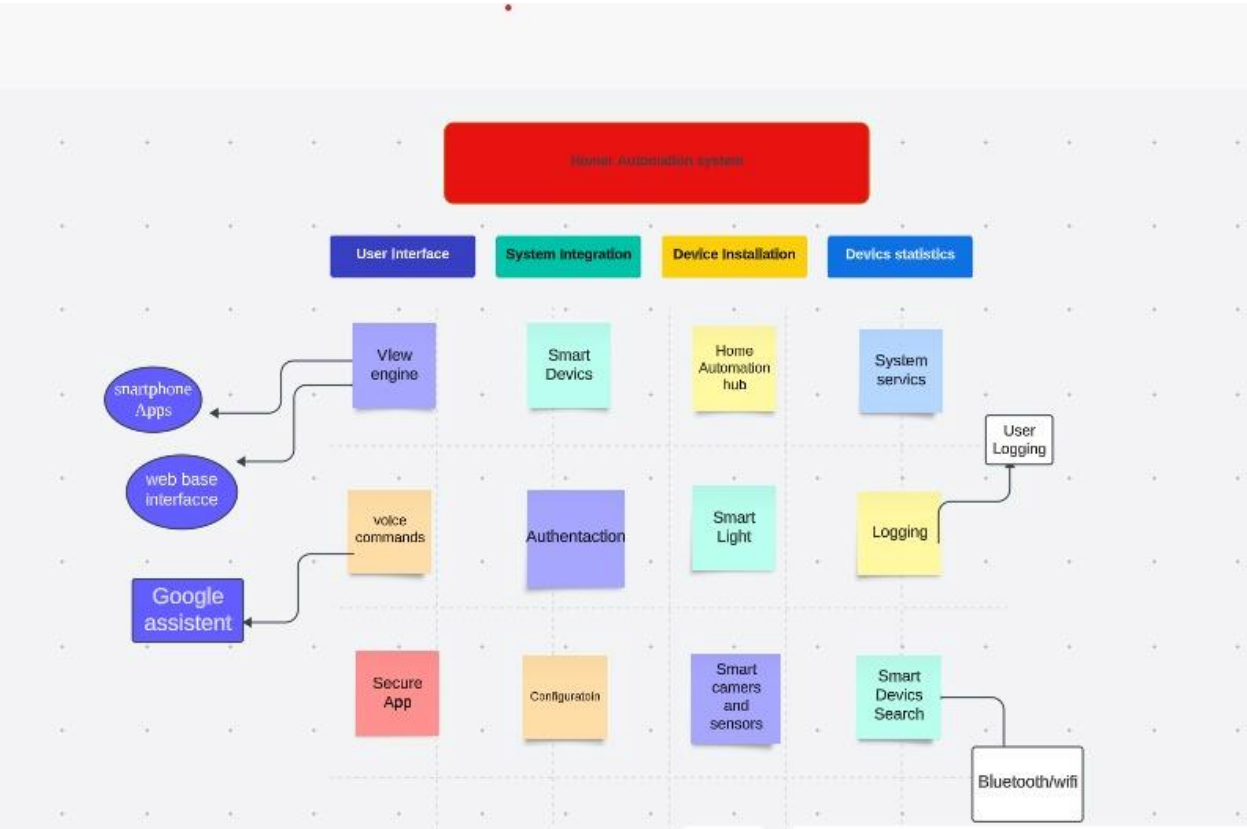
1.6.3. Development and testing:

This phase had the development of the application. The android device was connected to the Node via wireless network and the whole prototype was tested for identification and removal of bugs.

1.6.4. Real world testing:

The prototype was ready to be tested into the real world and integrated with various real time electrical appliances.

1.6.5. Work Breakdown Structure



WBS Figure No 1.

Sample WBS

1. Project Initiation
    - 1.1.1. Define project objectives and goals
    - 1.1.2. Establish project scope and boundaries
    - 1.1.3. Identify key stakeholders

2. Planning and Design
    - 2.1.1. Create a system design and architecture
    - 2.1.2. Develop a project plan, including timelines and budgets
    - 2.1.3. Identify hardware and software requirements

3. Procurement
    - 3.1.1. Research and purchase necessary components
    - 3.1.2. Select and acquire a central home automation hub

4. Infrastructure Setup
    - 4.1.1. Ensure a stable power supply and electrical circuits
    - 4.1.2. Set up a home network with Wi-Fi or Ethernet connections
    - 4.1.3. Install any necessary wiring for smart devices

5. Device Installation
    - 5.1.1. Install and configure the central hub
    - 5.1.2. Install and connect smart devices
    - 5.1.3. Smart lighting system
    - 5.1.4. Configure device settings and connectivity

6. System Integration
    - 6.1.1. Integrate smart devices and components
    - 6.1.2. Configure automation routines and scenarios
    - 6.1.3. Set up voice control
    - 6.1.4. Integrate Google Assistant
    - 6.1.5. Test device interoperability and communication

7. User Interface
    - 7.1.1. Set up user interfaces
    - 7.1.2. Smartphone apps
    - 7.1.3. Web-based interfaces
    - 7.1.4. Set up voice commands for Google Assistant

8. Testing and Quality Assurance
    - 8.1.1. Perform system testing
    - 8.1.2. Test voice control
    - 8.1.3. Identify and address any issues or bugs
    - 8.1.4. User acceptance testing (UAT)

9. Project Closeout
    - 9.1.1. Final system inspection
    - 9.1.2. Quality check

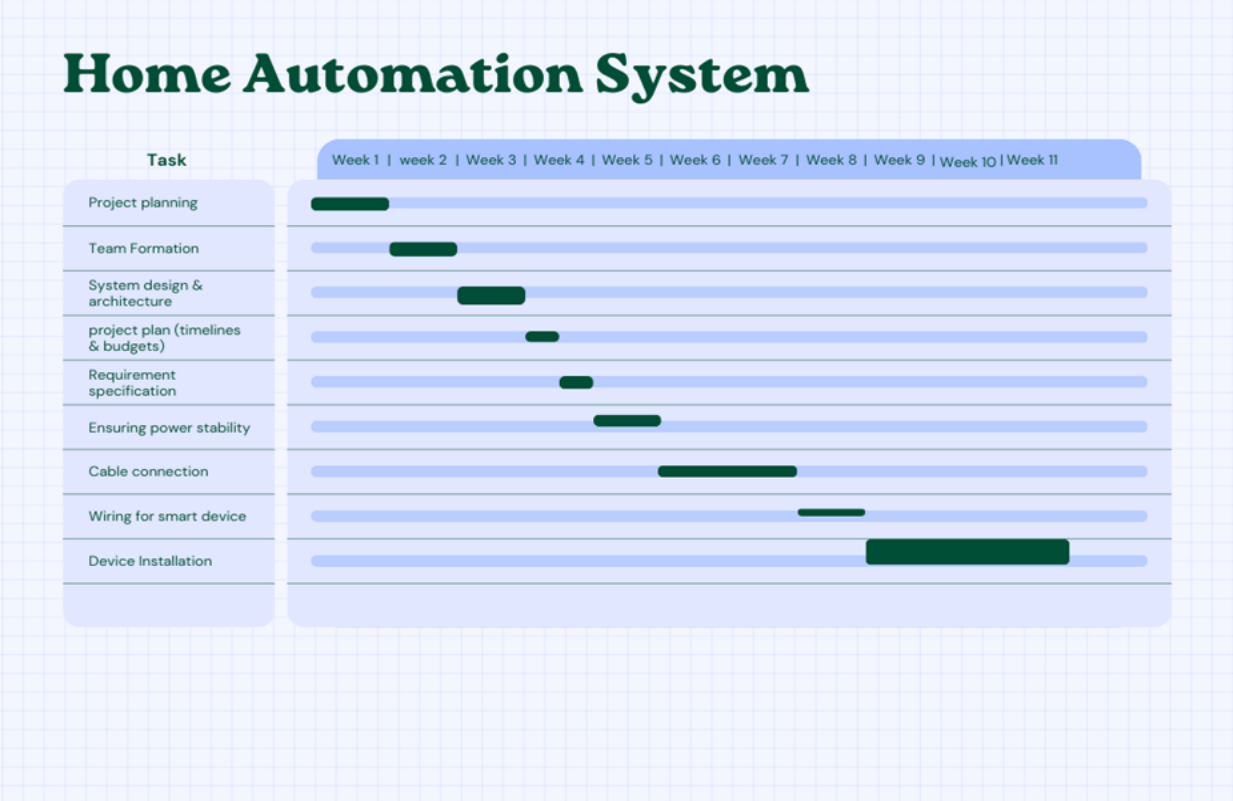
1.6.6. Roles & Responsibility Matrix

WBS #	WBS Deliverable	Activity	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Project Management	1	Project Planning Team Formation	7 days	Mian Raza
2	Planning and design	2	Create a system design and architecture. Develop a project plan, including timelines and budgets Identify hardware and software requirements	7 days	Mian Raza M Abdullah
3	Infrastructure Setup	3	Ensure a stable power supply and electrical Circuits. Wi-Fi or Ethernet connections. Wiring for smart devices	25 days	Mian Raza M Abdullah
4	Device Installation	4	Install and connect devices. lighting system, sensors and access control	20 days	Usama Zaman
5	System Integration	5	Set up voice control Integrate Google Assistant	60 days	Mian Raza Abdullah Asad

6	User Interface	6	Web App Design Set up user interfaces for controlling the system Smartphone app Tablets Web-based interfaces	60 days	Mian Raza M Abdullah
7	Authentication	7	Device Auth & App	15 days	Mian Raza

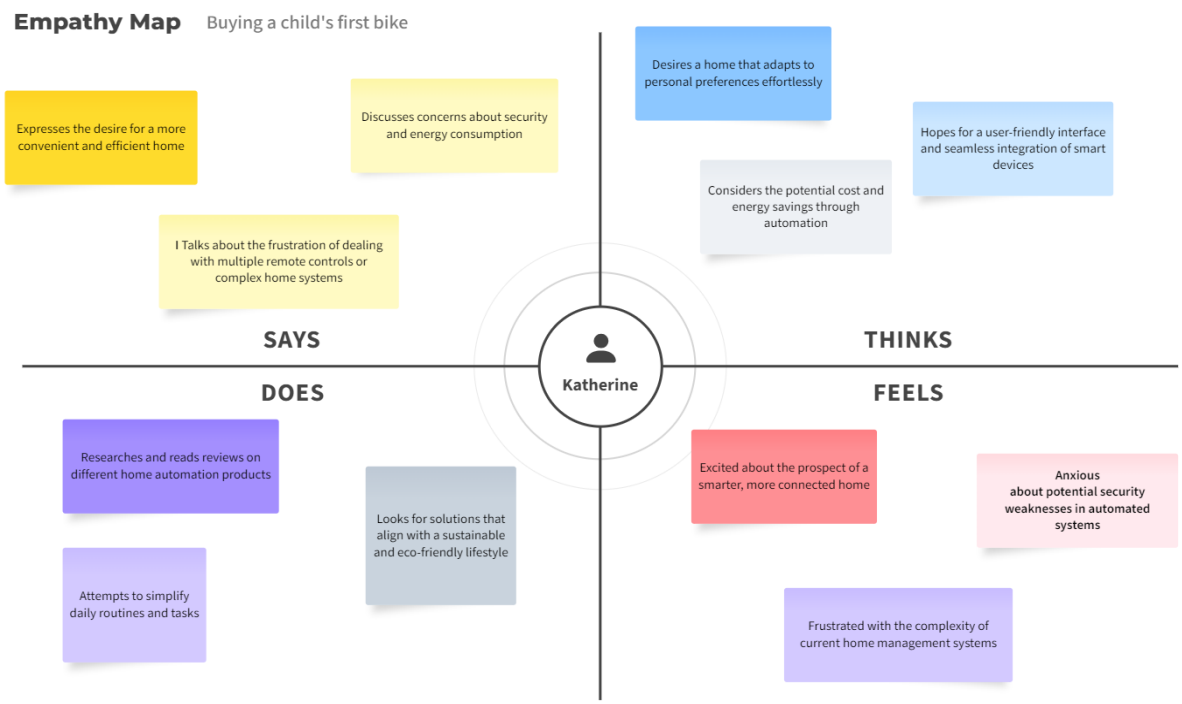
RRM Table No 2.

1.6.7. Gantt Chart



GC Figure No 2.

1.7. Empathy Map



EM Figure No 3.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

Our idea has different approach from others, instead of implementing on HTTP server and using Bluetooth and relay we are going to implement this automated system by using ZigBee device communicating with Arduino device through ZigBee protocol.

ZigBee Device features compared with WIFI

- Low battery consumption
- It works on a long range network
- It is non-rechargeable
- ZigBee devices are extendable

2.1.2. Document Conventions

For project title, advisor name and group members we have used Times New Roman bold type font and for page header we have used normal Times New Roman type font. Other than these, all the text has been typed in normal Calibri font.

2.1.3. Intended Audience and Reading Suggestions

This document is specially designed for seeking the attention of Gold Campus Project Office department and Campus Faculty members and also for those who wants to get the knowledge of Home automation system.

2.1.4. Product Scope

Home automation is building automation for a home, called a smart home or smart house. We are doing opposite to traditional home automation. Our system will reduce the human effort to control the home appliances.

2.1.5. References

Smart home technology, also often referred to as home automation, provides homeowners comfort, and convenience and energy efficiency by allowing them to control smart devices, often by a smart home app on their smart phone or other networked device.

2.2. Overall Description

2.2.1. Product Perspective

Home Automation system allows users to interact with their home appliances automatically with no human effort involved. This can be done simply by touching the option from tablet touch screen.

PP Figure No 4.

2.2.2. User Classes and Characteristics

This product will target the audience of any class this would be neither cheap nor expensive. Our main idea is to target our national citizens who are not familiar to this innovation. This would be beneficial for older people who face problems by doing these jobs manually, disabled people as well either security issues are concerned or non-automated issues which needed to be tackled.

2.2.3. Operating Environment

Our home automation system is likely to be operated on a test board first for simulation. We will use Arduino as a hardware platform and IOT protocols will perform their operations to gather instructions from user and send it to the following devices. Mobile application is used as software which helps the user to have to entertain the instructions easily.

2.2.4. Design and Implementation Constraints

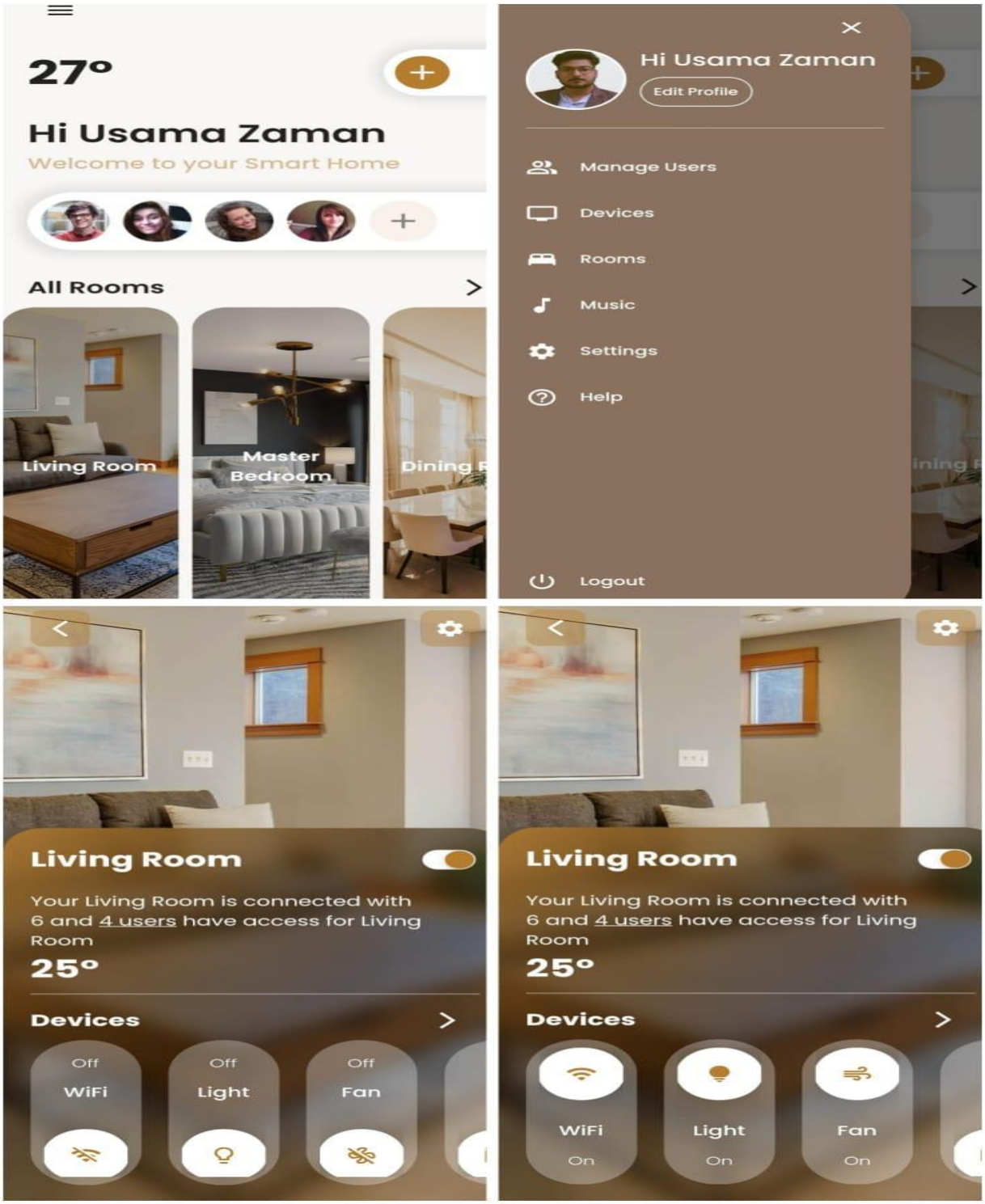
Hardware limitations could be an obstacle for us, as the hardware is not provided by our institution we have to find it from market. It could be possible that some hardware needs to be imported from outside the city. Other than that, as we are not professionals such problems can be arise while designing and implementation phase like design conventions, communication protocols and programming standards.

2.2.5. Assumptions and Dependencies

It is assumed to be completed the project by the end of next semester depending upon the member's performance and skill set. A test-based simulator would be implemented which would function all the elements which are required. This project would be dependent upon our regular courses which are also going on simultaneously. We would manage and divide our time in between all the following activities.

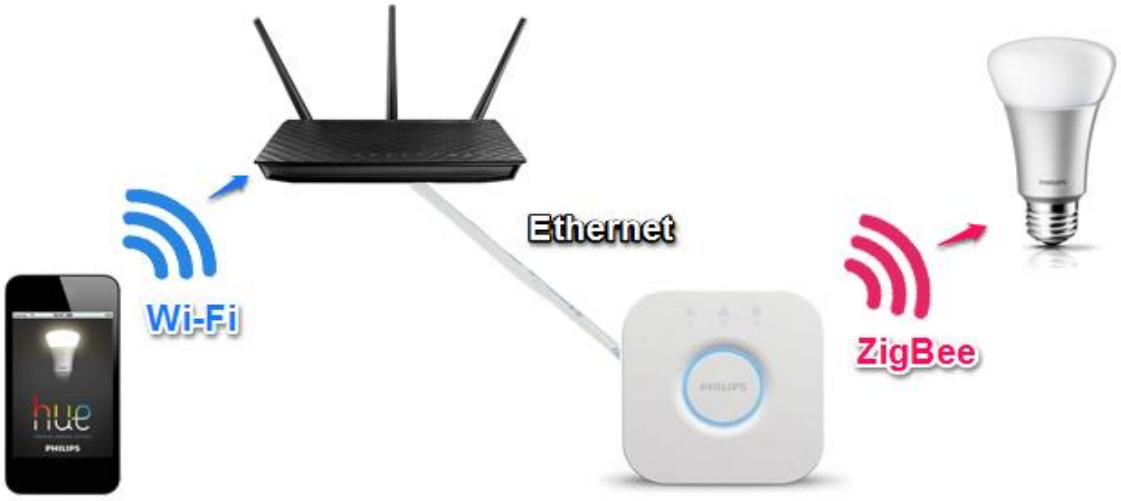
2.3. External Interface Requirements

2.3.1. User Interfaces

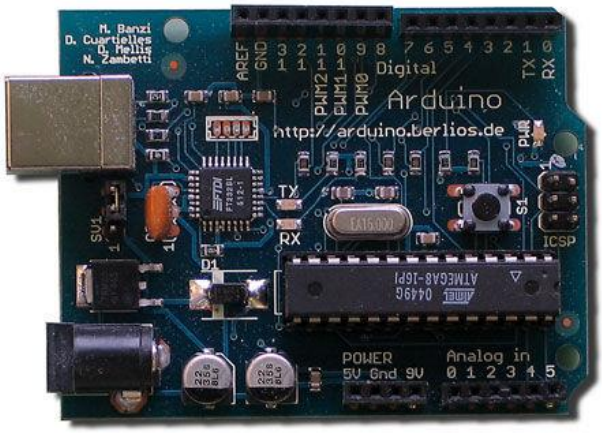


UI Figure No 5.

2.3.2. Hardware Interfaces



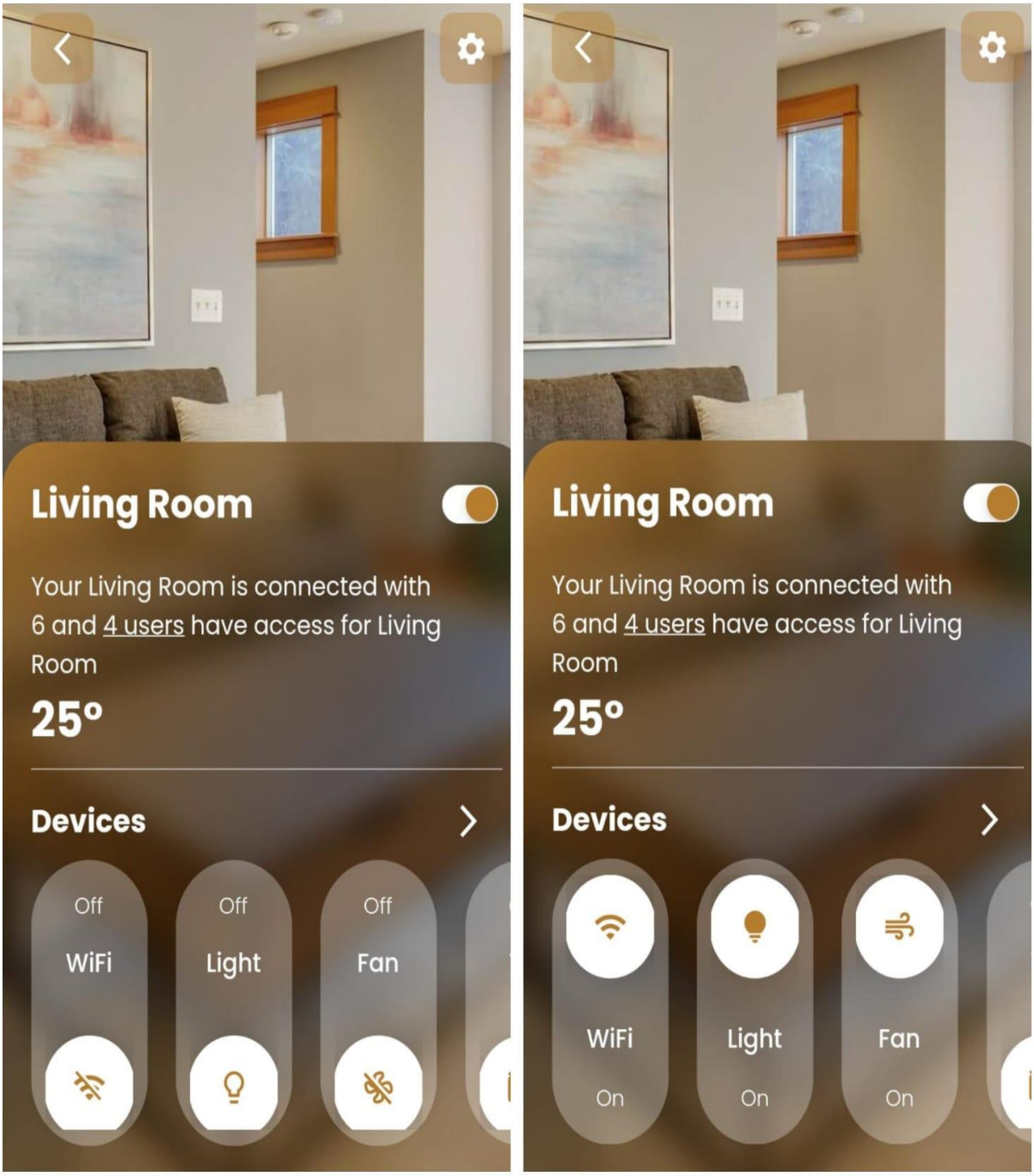
HI Figure No 6.



AR Figure No 7.

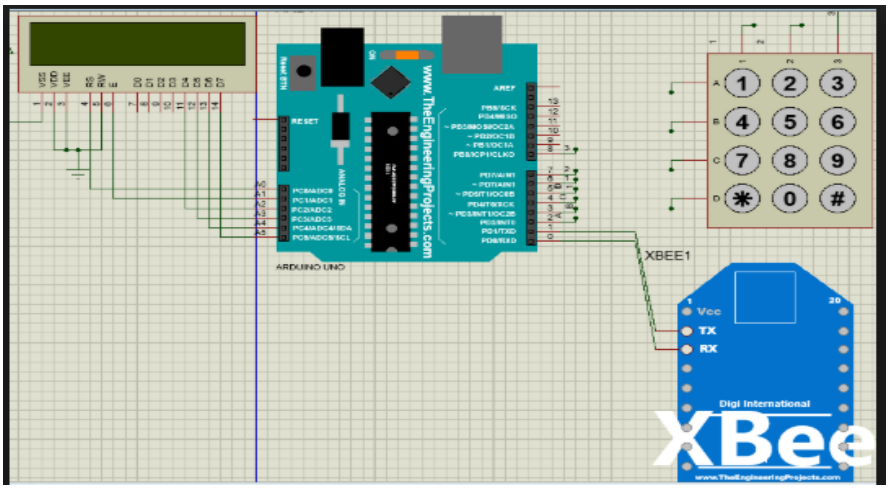
Arduino would be used as a hardware interface. This device is attached to the central server and the following appliances (fan, AC and lights)

2.3.3. Software Interfaces



SI Figure No 8.

2.3.4. Communications Interfaces

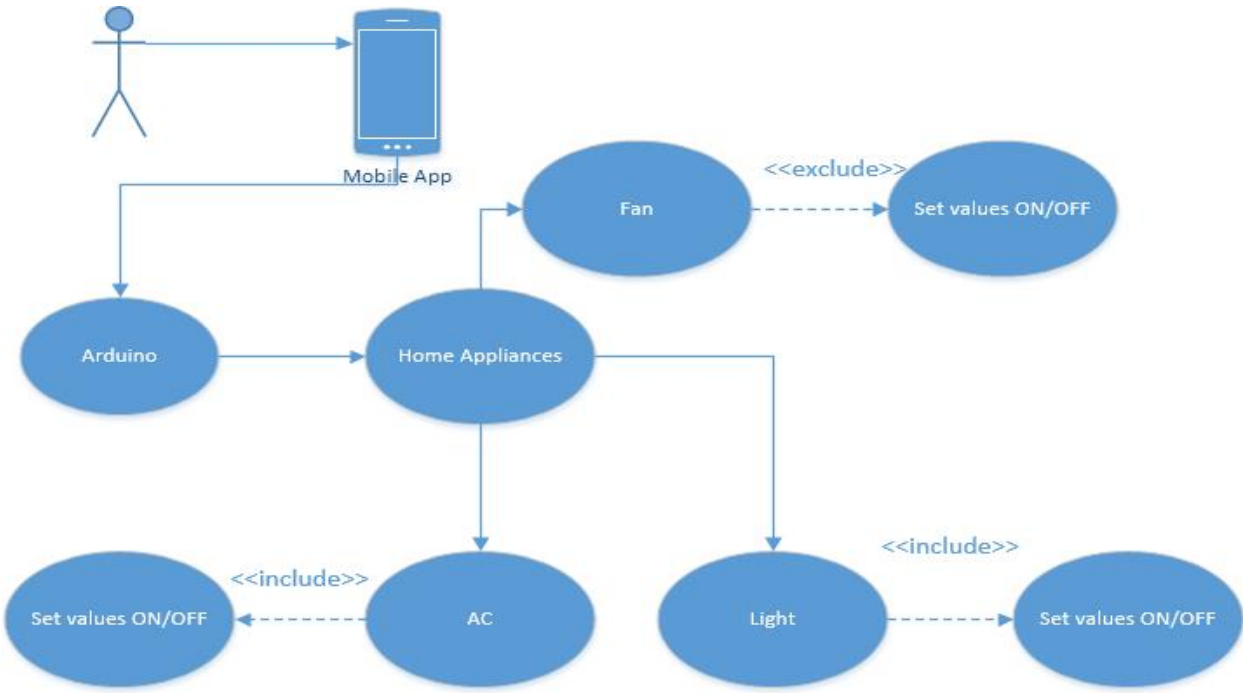


CI Figure No 9.

This is the interface through which ZigBee device communicate with Arduino to establish connection.

2.4. Product Features / Functional Requirements

"Home Automation System"



PF Figure No 10.

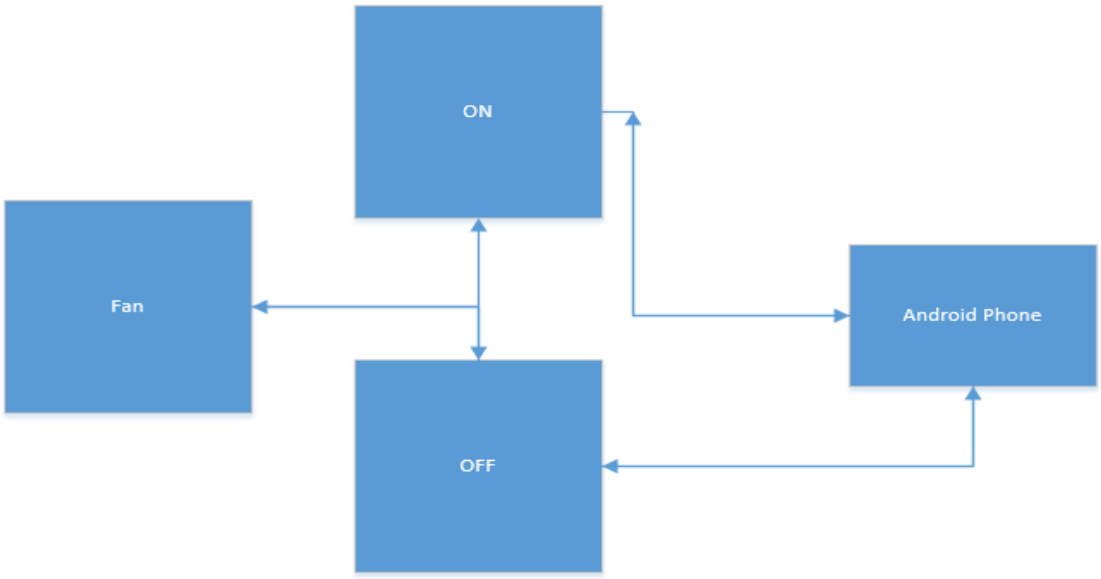
2.4.1. Home Automation System

Identifier	Room controlling	
Purpose	Controlling home appliances, Lights and infotainment system automatically	
Priority	Medium	
Pre-conditions	No appliance have dependencies	
Post-conditions	All appliances or devices functions automatically simultaneously	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Selects the desired module from Mobile Application and Arduino refer the specific appliance to perform functionality	Arduino allocates signals to specific sensor and operates that module
2	Increase/Decrease brightness of lights	The light sensor analyze it to increase the intensity
3	Turn ON AC while turning OFF Fan	Following temperature has been pre-set for both Fan and AC, if temperature rises AC turn ON while decreasing the speed of Fan
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	No action entertained	No response
2	Stay with the same brightness or turn the lights Off	Light sensor again indicates the brightness and requests to either stay up with same brightness or turns the lights Off
3	Fan turns ON or turns Off both AC and Fan	Depending upon the temperature, temperature sensor indicates what to do either turning ON the Fan or turns Off both AC and Fan
…		

HAS Table No 3.

2.4.2. Feature 2

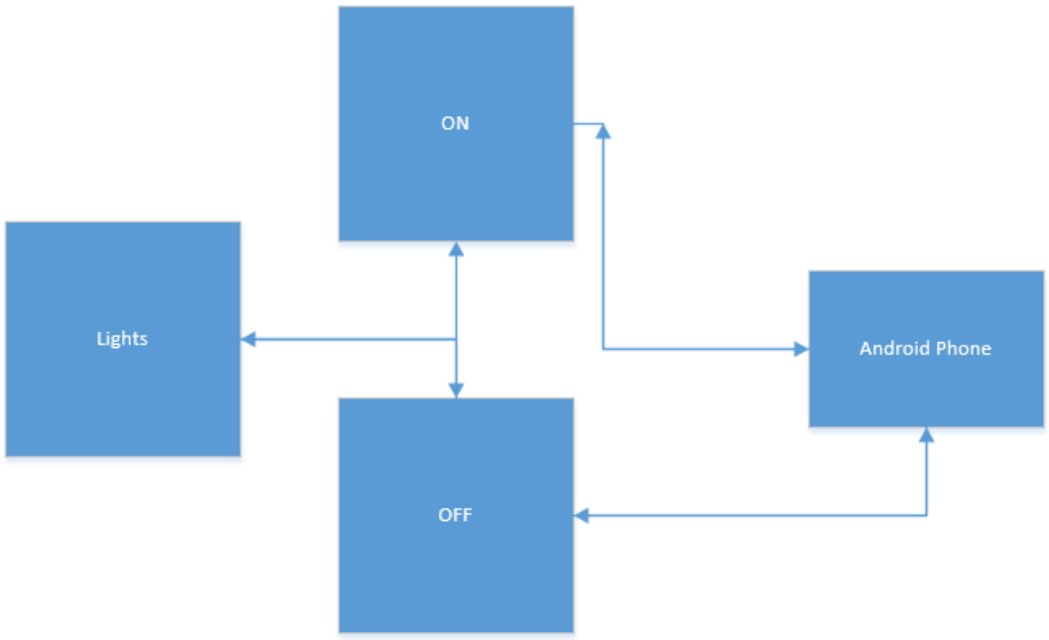
2.4.2.1. Fan Module



FM Figure No 11.

2.4.3. Feature 3

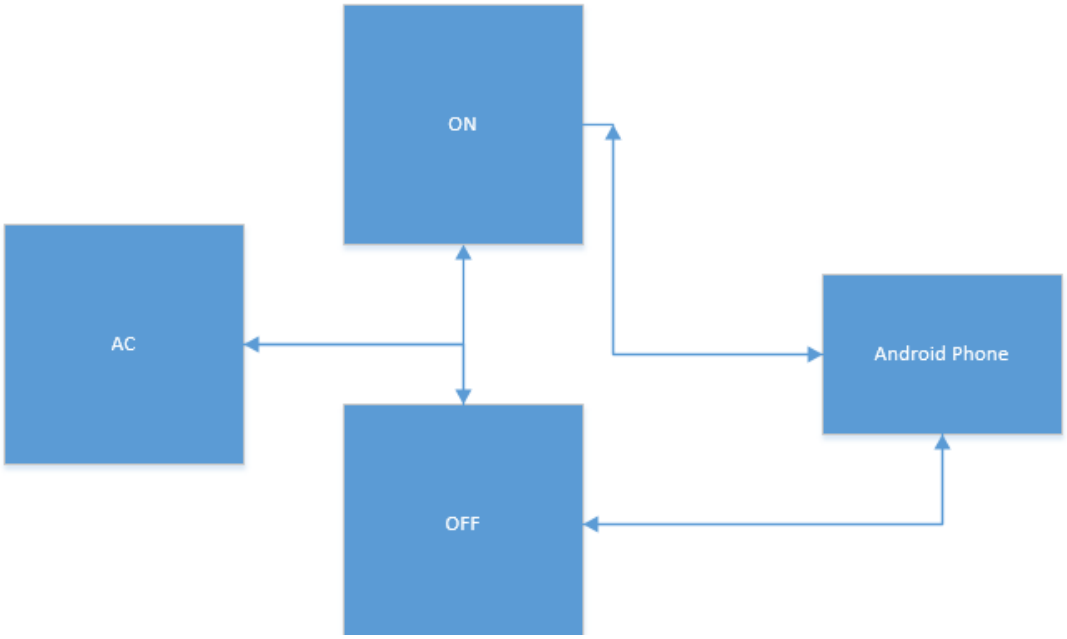
2.4.3.1. Lights Module



LM Figure No 12.

2.4.4. Feature 4

2.4.4.1. AC Module



AM Figure No 13.

2.5. Nonfunctional Requirements

2.5.1. Performance Requirements

The system shall not add more than two seconds to the time required to perform an action if the system is not connected. For example, if it takes 3 seconds to turn the Microwave on normally, it will take no longer than 5 seconds for the microwave to turn on through the system.

2.5.2. Safety Requirements

- Smart thermostats can provide alerts if a home loses power or if the temperature in the home falls below or rises above a set threshold.
- Water sensors can detect unwanted water in the home, alerting the owner to potential leaks near washing machines, dishwashers, water heaters and other areas.

- Smoke, heat and carbon monoxide detectors that can interface with a central hub or mobile apps are available. This provides remote monitoring and control capabilities beyond the classic alarm system.

2.5.3. Security Requirements

As mentioned earlier, devices can pair with each other, thus a link is created between them. To secure the link level, four entities are used; the device address which is unique for each device, private authentication and encryption keys, both which are 8 to 128 bits in length, and finally a random number which changes frequently. These entities are used to generate a key or Personal Identification Number (PIN) which is then used between the devices to connect or transfer data.

- Smart light bulbs remotely control or program lights when you are away, in order to make a home appear occupied.
- Smart motion lights and exterior lighting allow for remote programming and monitoring.

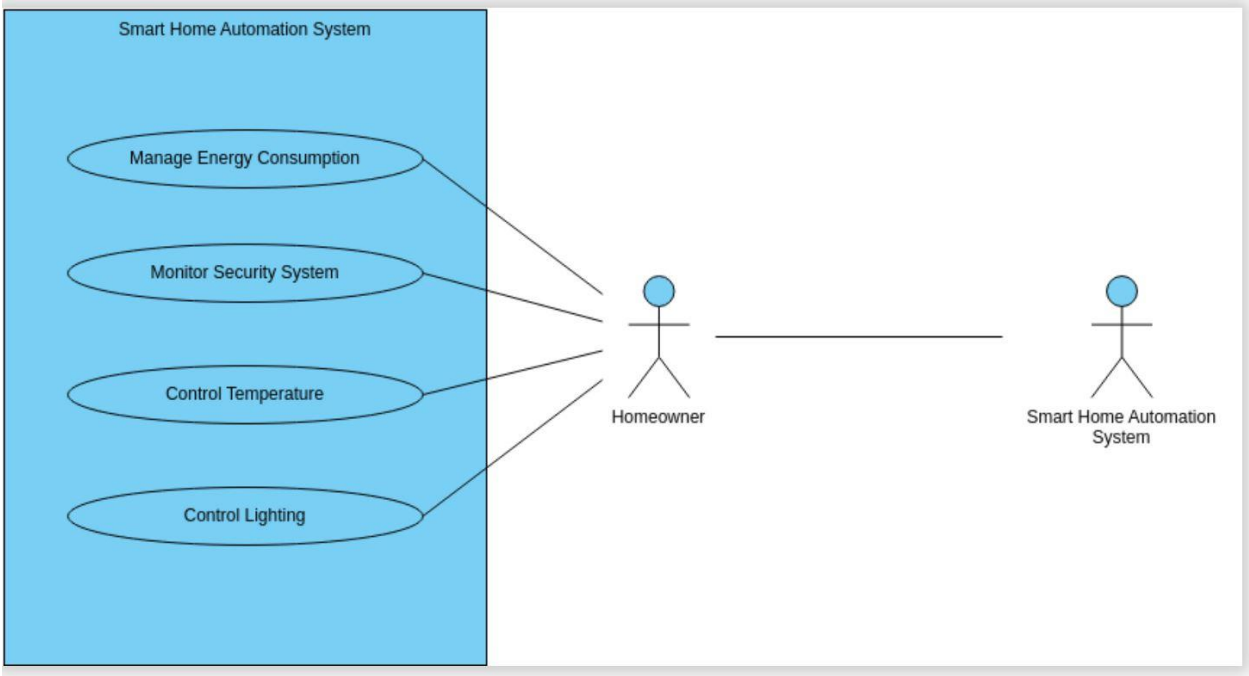
Chapter 3

Use Case Analysis

Chapter 3: Use Case Analysis

Use Case Analysis is all about how our home automation system works in real life. We're going to look at situations where people use the system and see how it makes their lives better, like turning off lights with a voice command or setting the temperature of a AC temperature before getting home. It's all about showing how our system is practical and makes everyday tasks easier and more enjoyable for people at home.

3.1. Use Case Model



UCM Figure No 14.

3.2. Use Cases Description

3.2.1. Use Case: Turn On/Off Lights

- **Actor:** Homeowner
- **Preconditions:** Home Automation System is online.

Flow of Events:

1. Homeowner initiates "Turn On/Off Lights" through the system interface or voice command.

2. System presents available lights and their current status.
3. Homeowner selects lights and specifies action: turn on or turn off.
4. System processes the command, sending signals to selected lights.

Post conditions: Selected lights are turned on or off based on homeowner's command.

3.2.2. Use Case: Adjust AC Temperature

- **Actor:** Homeowner
- **Preconditions:** Home Automation System is operational

Flow of Events:

1. Homeowner initiates "Adjust AC temperature" via system interface or voice command.
2. System retrieves and displays current temperature settings and mode.
3. Homeowner selects desired temperature and may adjust thermostat mode.
4. System sends new settings to the AC, implementing changes.

Post conditions: Home temperature is adjusted based on homeowner's preferences.

3.2.3. Use Case: Voice Commands

- **Actor:** Homeowner
- **Preconditions:** Home Automation System is online; voice command feature is enabled.

Flow of Events:

1. Homeowner initiates "Voice Commands" through a compatible device.
2. System listens for and interprets voice commands.
3. Homeowner issues a command related to home automation (e.g., controlling lights, Fan, or AC).
4. System processes the voice command and executes the corresponding action.

Post conditions: Home automation actions are performed based on the interpreted voice command.

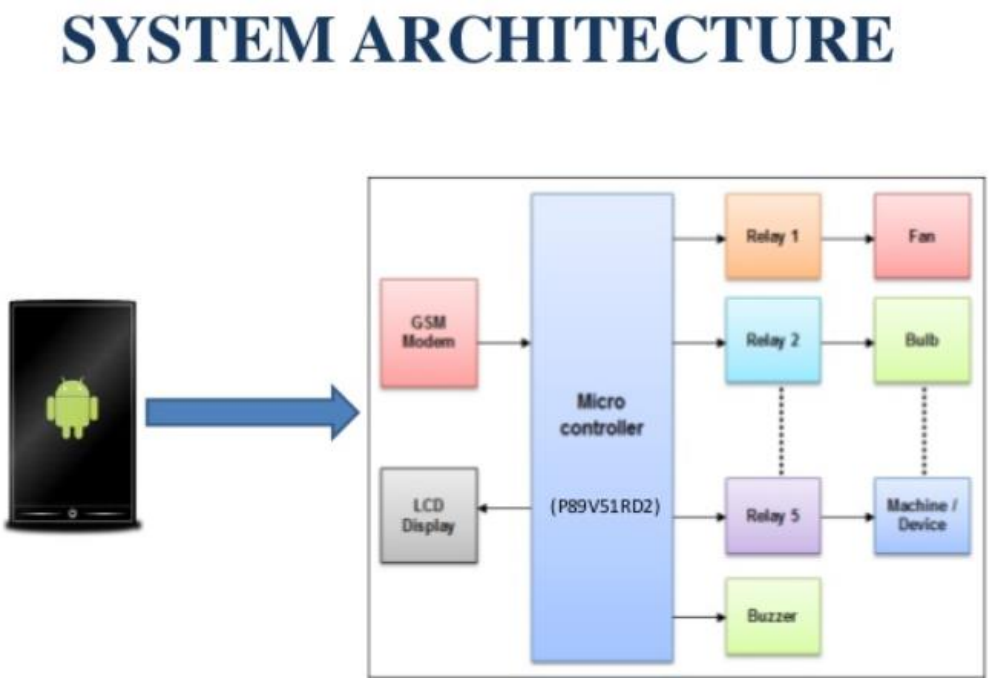
Chapter 4

System Design

Chapter 4: System Design

In terms of "Design System," we resemble architects creating a smart house plan. We determine how the many technological components, such as devices and software, will integrate to create an amazing house. Each component is examined in detail, their interoperability is considered, and the overall fit of the system is ensured. It is comparable to creating a puzzle with intricate details, where every component has a specific purpose. It should also be safe, expandable with additional features, and simple enough for all household members to utilize. Building a smart, safe, and incredibly simple-to-use house is made easier with the help of this chapter.

4.1. System Architecture Diagram

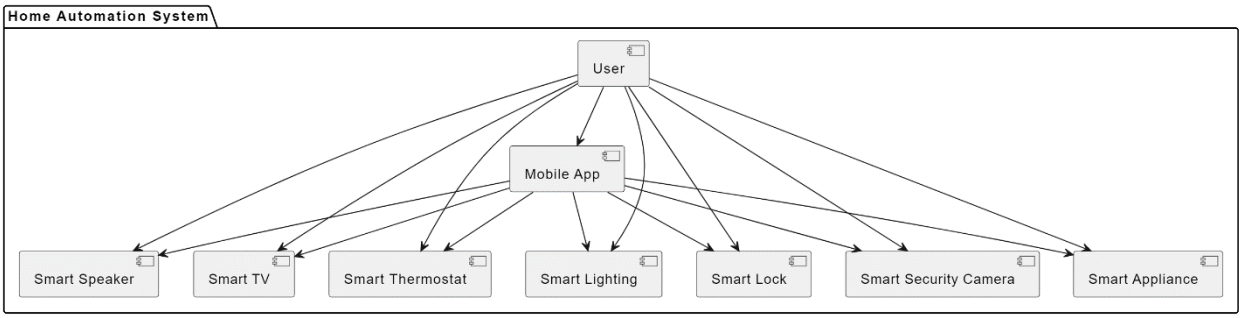


SAD Figure No 15.

4.2. Domain Model

Home automation system is like having a digital attendant for your house. There are smart devices like lights, Fan, and AC, each with its own job. Your phone becomes the remote control, letting you tell these devices what to do. For example, you can ask the lights to turn off or set the temperature of AC, all without leaving your couch. The system connects everything, making it work together smoothly. It's like magic and making your home comfortable, secure, and energy-

efficient, just the way you like it. So, the domain model is like a map that shows how all these smart parts talk to each other, making your home life easy.



DM Figure No 16.

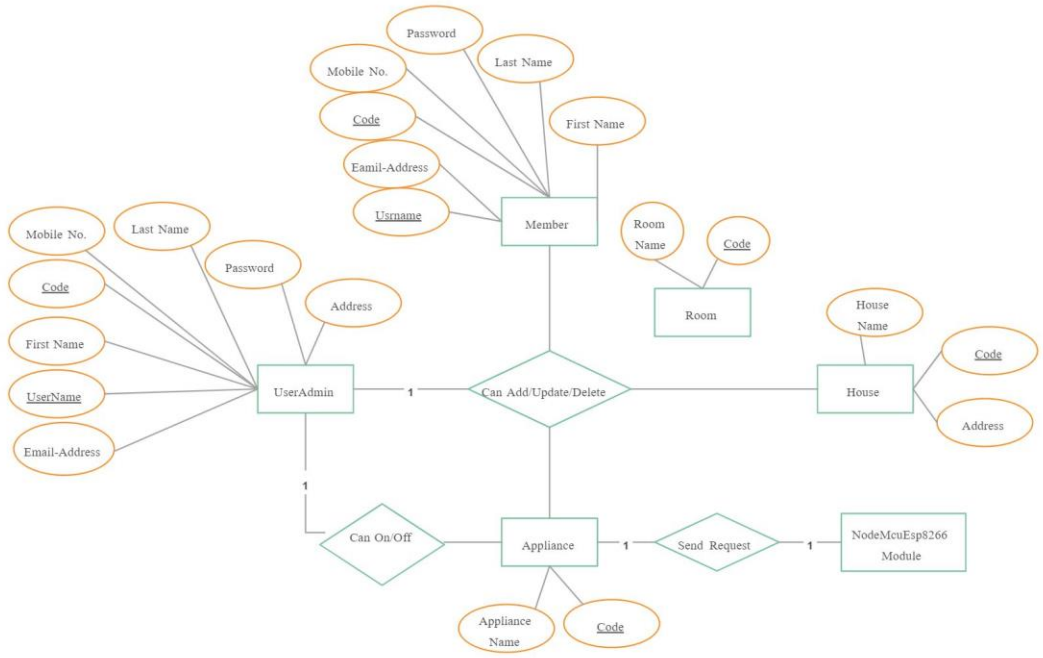
4.3. Entity Relationship Diagram with data dictionary

4.3.1. Data Dictionary

1. **Home**
   - Attributes: Home ID (Primary Key), Address, Owner, Size
2. **Room**
   - Attributes: Room ID (Primary Key), Type, Size
   - Relationships: Home ID (Foreign Key Referencing Home)
3. **Device:**
   - Attributes: Device ID (Primary Key), Type, Status
   - Relationships: Room ID (Foreign Key Referencing Room)
4. **Device Detail:**

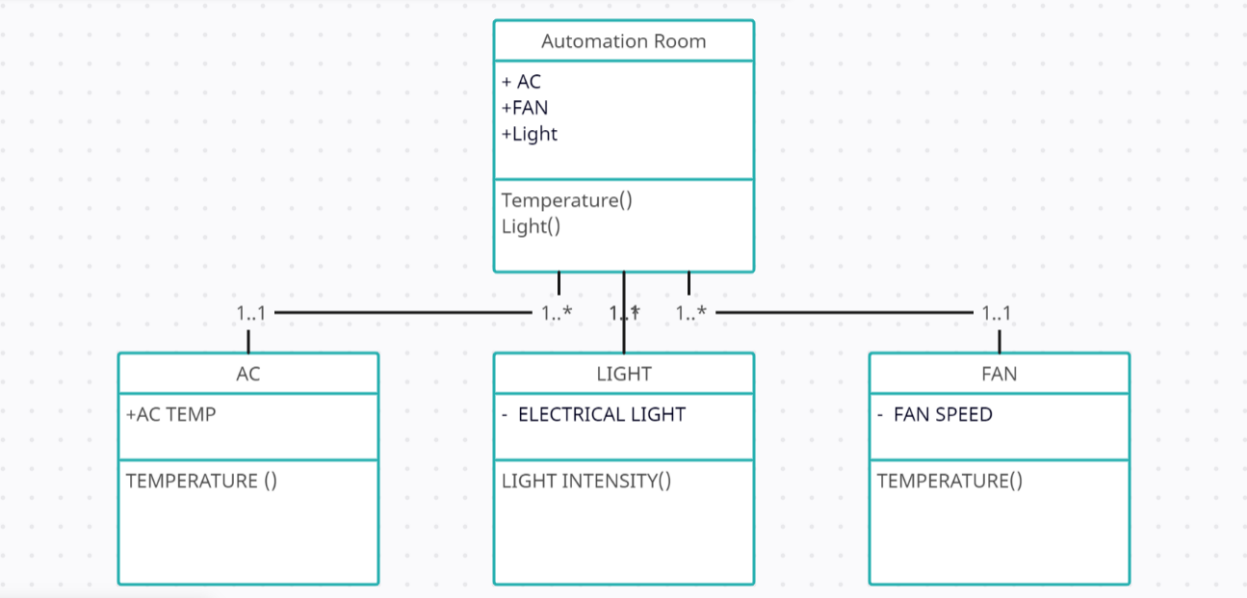
- Attributes: Device ID (Foreign Key Referencing Device), Detail Type, Detail Value
- Additional Information: Used to store specific details for each device, such as brightness level, temperature setting, etc.

4.3.2. Entity Relationship Diagram:



ERD Figure No 17.

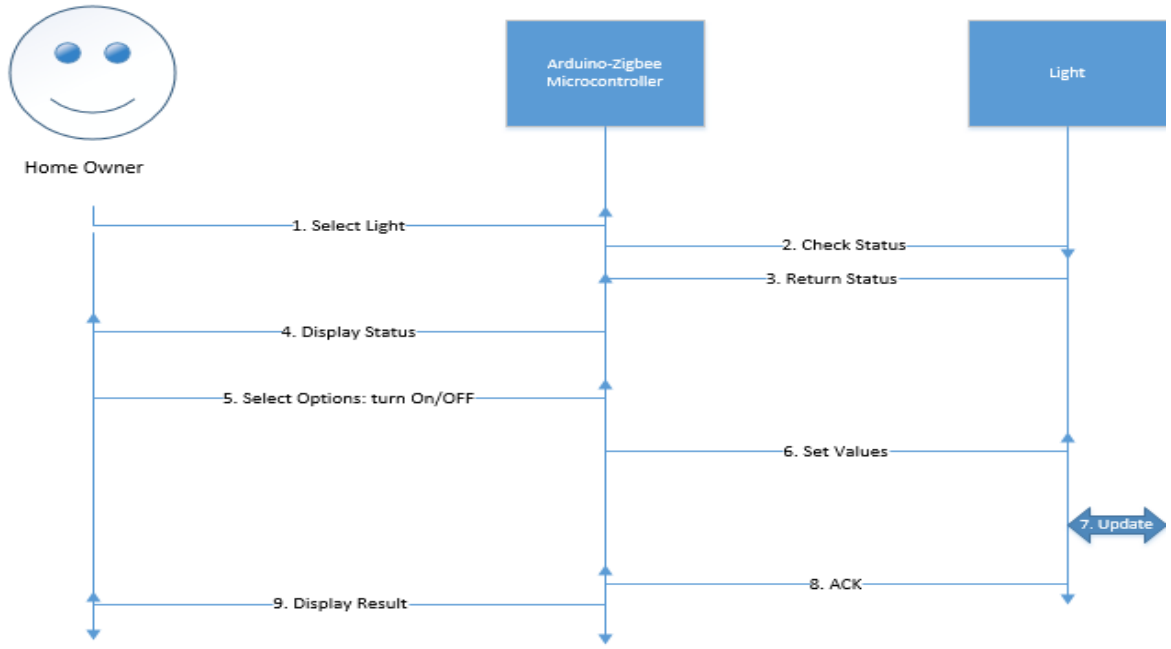
4.4. Class Diagram



CD Figure No 18.

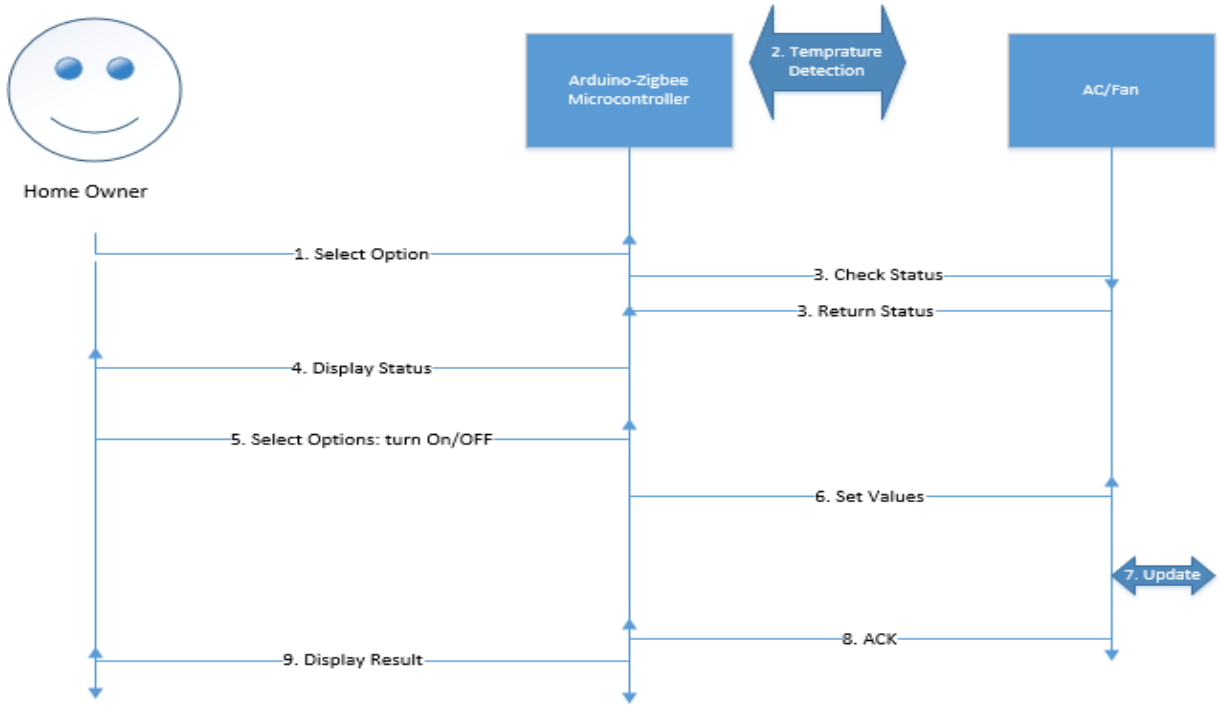
4.5. Sequence / Collaboration Diagram

4.5.1. Light Control



LC Figure No 19.

4.5.2. AC / FAN Control



AC/ Fan Figure No 20.

4.6. Operation contracts

4.6.1. Operation: Control Lights

Input:

- Home ID (string)
- Room ID (string)
- Light Action (enum: {Turn On, Turn Off, Adjust Brightness})

Output:

- Success (Boolean)
- Message (string)

Behavior:

- Controls lights in the specified room based on the provided action.
- Returns success as true if the operation is successful; otherwise, returns false with an appropriate message.

4.6.2. Operation: AC Temperature

Input:

- Home ID (string)
- Room ID (string)
- Temperature (float)

Output:

- Success (Boolean)
- Message (string)

Behavior:

- Adjusts the thermostat in the specified room to the given temperature.
- Returns success as true if the operation is successful; otherwise, returns false with an appropriate message.

4.6.3. Operation: View Home Status

Input:

- Home ID (string)

Output:

- Home Status (object containing various details about the home's current state)

Behavior:

- Retrieves and returns the status of the specified home, including room temperatures, security status, and device statuses.

4.6.4. Operation: Set Room Preferences

Input:

- Home ID (string)
- Room ID (string)
- Preferences (object containing user preferences for the room)

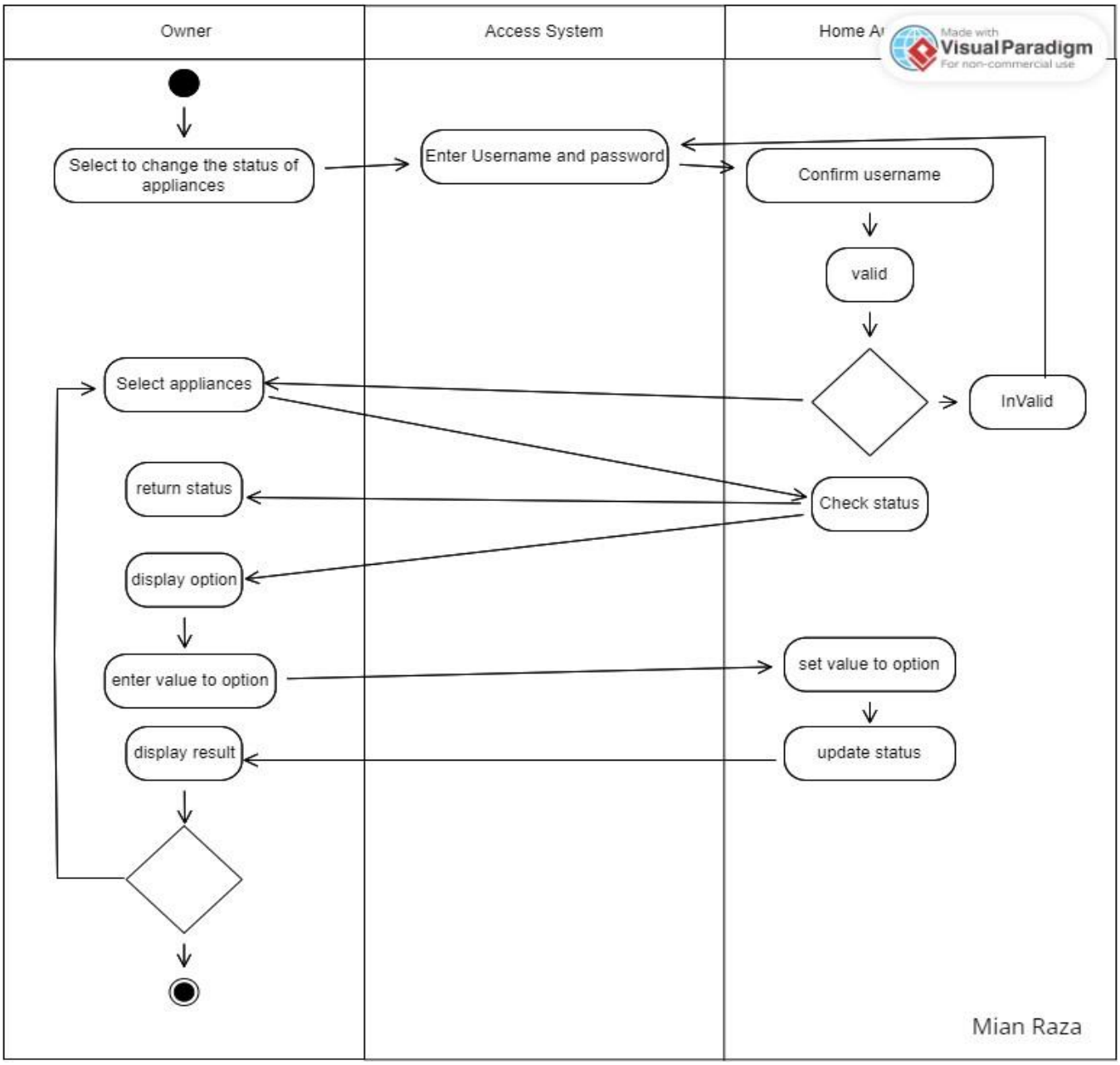
Output:

- Success (Boolean)
- Message (string)

Behavior:

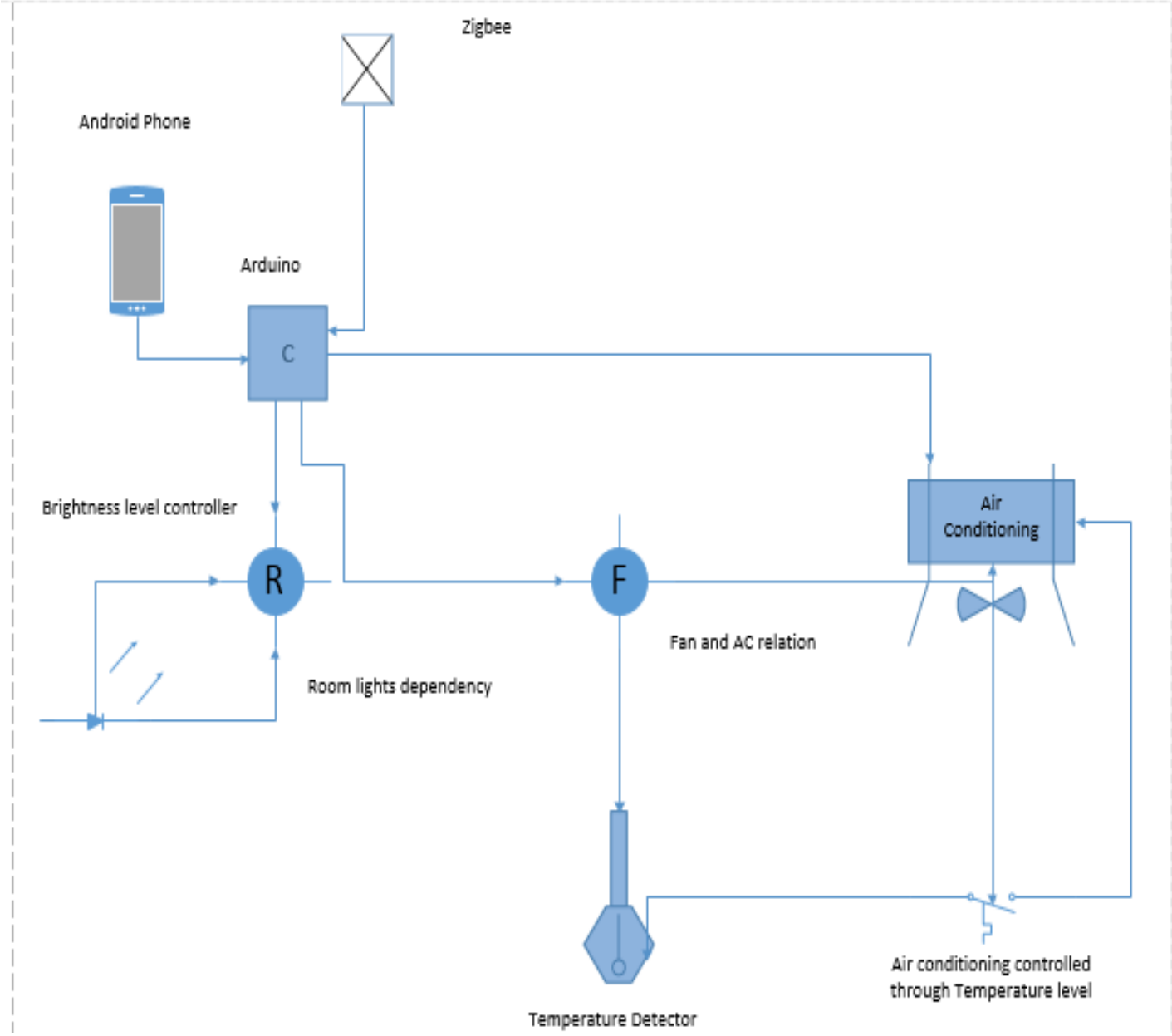
- Sets user preferences for the specified room, such as preferred lighting levels, temperature settings, etc.
- Returns success as true if the operation is successful; otherwise, returns false with an appropriate message.

4.7. Activity Diagram



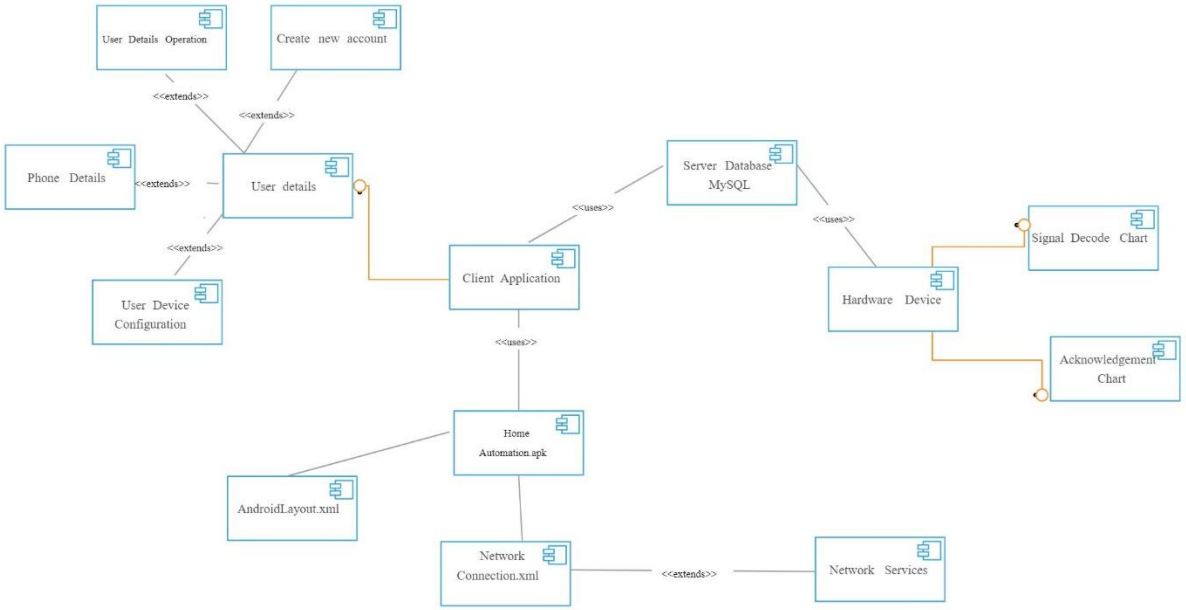
AD Figure No 21.

4.8. State Diagram



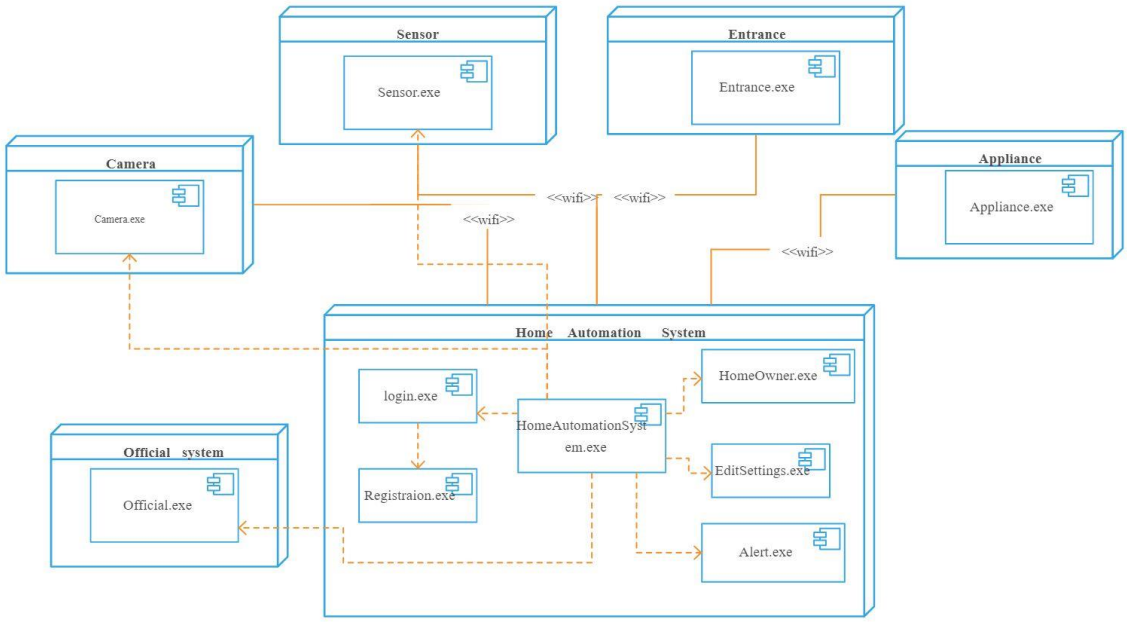
SD Figure No 22.

4.9. Component Diagram



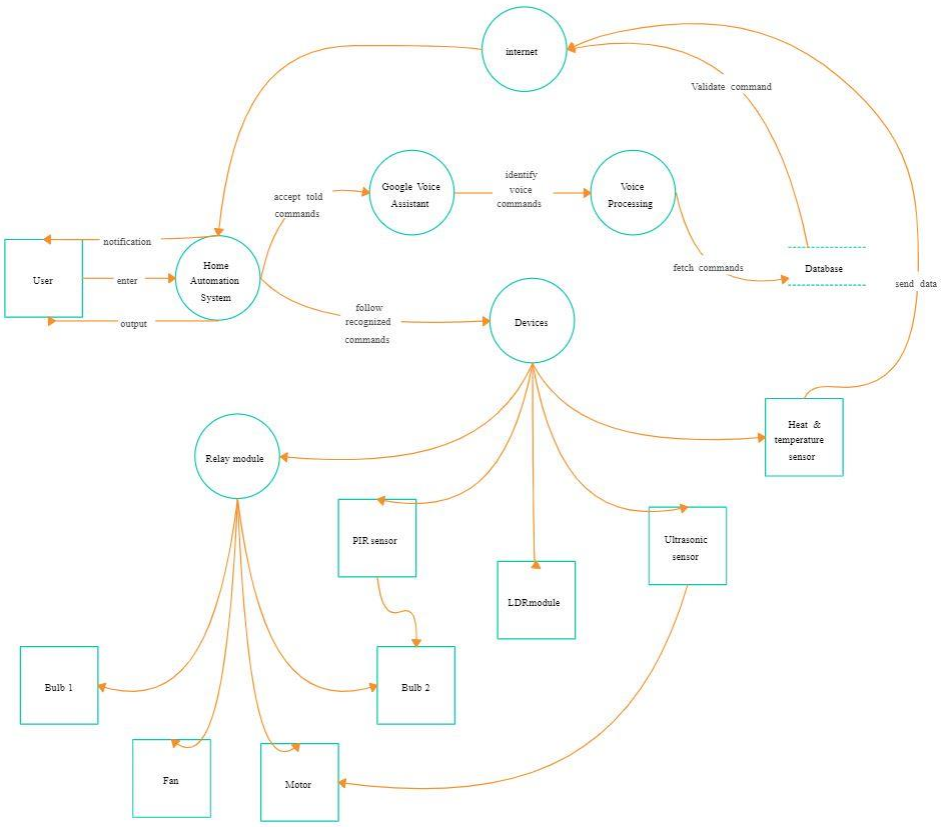
CD Figure No 23.

4.10. Deployment Diagram



DD Figure No 24.

4.11. Data Flow diagram



DFD Figure No 25.

Chapter 5

Implementation

Chapter 5: Implementation

Our home automation system is currently in the construction stage. This is the point at which the awesome features we discussed in previous chapters materialize as functional software. To make the system execute tasks we want it to, like turning on lights or regulating the temperature of the air conditioning system, developers employ their coding expertise. They assembled the bits of code to form our smart house in the same way as blocks are built. Make sure everything functions properly by testing and resolving any problems, much like making sure all the parts fit together perfectly.

5.1. Important Flow Control/Pseudo codes

Start Home Automation System

Homeowner

1. Initialize Homeowner

Turn On/Off Lights

2. Homeowner initiates action to Turn On/Off Lights

3. System receives command from Homeowner

4. System verifies Homeowner's credentials

If credentials are valid:

- Prompt Home Automation System to control lights
- Execute action to turn on/off lights
- Display success message

If credentials are invalid:

- Display error message

Adjust AC Temperature

5. Homeowner initiates action to Adjust AC temperature

6. System receives command from Homeowner

7. System verifies Homeowner's credentials

If credentials are valid:

- Prompt Home Automation System to adjust AC temperature
- Execute action to adjust AC temperature
- Display success message

If credentials are invalid:

- Display error message

Voice Commands

8. Homeowner issues voice command

9. System processes and interprets voice command

10. System identifies the corresponding action (e.g., lights, Fan, AC)

11. Execute the action based on the interpreted command

12. Display success message

End Home Automation System

5.2. Components, Libraries, Web Services and stubs

5.2.1. Components

1. **Home Automation Controller**

   Description: Central component managing the overall system. It processes commands from users, communicates with devices, and ensures seamless operation.

2. **Device Interface**

   Description: Interface connecting the controller with various smart devices such as lights, thermostats, and security systems. It facilitates communication and control.

3. **Voice Recognition Module:**

   Description: Component responsible for processing voice commands from users. It converts spoken words into actionable instructions for the home automation system.

5.2.2. Libraries

1. **Smart Device Integration Library**

Description: A library that provides standardized interfaces and protocols for integrating a variety of smart devices. It ensures compatibility and ease of integration.

2. **Security Algorithms Library**

Description: Library containing algorithms for security-related functionalities, such as encryption, authentication, and authorization. It enhances the system's security.

5.2.3. Web Services

1. **Weather Information Service**

Description: External web service providing real-time weather information. It enables the system to make informed decisions, such as adjusting thermostat settings based on current weather conditions.

2. **User Authentication Service**

Description: External service responsible for authenticating users. It ensures that only authorized users can control the home automation system.

5.2.4. Stubs

1. **Device Emulator Stub**

Description: A stub emulating the behavior of smart devices during testing and development. It allows the system to simulate interactions with devices before the actual devices are integrated.

2. **Voice Command Stub**

Description: A stub simulating voice command inputs for testing the voice recognition module. It helps ensure that the system accurately interprets and processes voice commands.

5.3. Deployment Environment

5.3.1. Hardware

- Home Automation Controller: A dedicated physical or virtual server that hosts the core logic and functionalities of the home automation system.
- Smart Devices: Devices such as smart bulbs, thermostats, and security cameras that are part of the home automation ecosystem.

5.3.2. Software

- Home Automation System Software: The core software responsible for managing and controlling various aspects of the home automation system. This software includes the controller's logic, user interface components, and integration with devices.
- Device Firmware: Software embedded in smart devices that enables communication and interaction with the home automation system.

5.3.3. Communication Protocols

- Wireless Protocols: Technologies like Wi-Fi, ZigBee, or Z-Wave used for communication between the home automation controller and smart devices.
- Voice Recognition Service: If using cloud-based voice recognition, the deployment involves communication with external voice recognition services.

5.3.4. Network Infrastructure

- Local Area Network (LAN): The network connecting the home automation controller and smart devices within the home.
- Internet Connection: Required for remote access and cloud-based services.

5.3.5. Security Measures

- Firewall and Intrusion Detection Systems: Implemented to safeguard the home automation system from unauthorized access.

- Encryption: Ensures secure communication between the home automation controller and smart devices.

5.3.6. User Interfaces

- Mobile Apps/Web Interfaces: User interfaces allowing homeowners to interact with and control the home automation system remotely.

5.3.7. External Services

- Weather Information Service: If the system relies on real-time weather data, it connects to an external weather service over the internet.
- User Authentication Service: For verifying user credentials during remote access.

5.3.8. Testing and Development Tools

- Device Emulators: Tools used during development and testing to emulate the behavior of smart devices without physical devices.
- Voice Command Simulation Tools: Used for testing the voice recognition module.

5.3.9. Logs and Monitoring Tools

- Logging System: Captures system events and errors for troubleshooting and analysis.
- Monitoring Tools: Keep track of system performance, device status, and security events.

5.4. Tools and Techniques

In the implementation phase of a home automation system, various tools and techniques are employed to ensure effective development, testing, and deployment. Development often relies on integrated development environments (IDEs) such as Visual Studio or Eclipse, facilitating code writing, debugging, and version control. Collaboration tools like GitHub enhance team coordination and code sharing. Testing employs tools like JUnit for unit testing and Selenium for automated testing of web interfaces. Continuous Integration (CI) tools like Jenkins ensure regular

integration and testing of code changes. Deployment is often managed using containerization tools like Docker for consistent and scalable deployment across different environments. Monitoring tools such as Nagios or Prometheus assist in tracking system performance, ensuring optimal functionality. Additionally, project management tools like Jira or Trello aid in organizing tasks and timelines, ensuring a smooth and well-coordinated development process.

5.5. Best Practices / Coding Standards

1. Consistent Style: Imagine everyone using the same cookbook. Consistent coding style makes it easier for all developers to understand and work together.
2. Clear Naming: Think of variable names as ingredients. Just like you wouldn't mix up sugar with salt, clear and descriptive names prevent confusion in the code.
3. Documentation: Think of comments and README files as cooking instructions. Good documentation helps others (and your future self) understand how to use and modify the code.
4. Version Control (Git): Imagine saving different versions of your recipe. Git allows developers to track changes, making it easier to collaborate without messing up the original recipe.
5. Unit Testing: Picture tasting your dish at different stages. Unit tests check if each part of your code works well, ensuring the final product is tasty and bug-free.
6. Code Reviews: Think of it like having friends taste your dish before serving. Code reviews let other developers check your work, provide feedback, and make sure everything is just right.

5.6. Version Control

5.6.1. State Management

Provider: A simple yet powerful state management solution for Flutter applications.

5.6.2. Local Database

Hive: A lightweight and fast NoSQL database for Flutter, which can be used to store local data.

5.6.3. HTTP Requests

http: A package for making HTTP requests, useful for interacting with a server-side application or a version control system.

5.6.4. Logging

logger: A flexible logging package that can help you keep track of events and errors in your application.

5.6.5. Permissions

Permission Handler: A plugin for handling runtime permissions in Flutter applications, which may be needed for accessing certain features of the device.

5.6.6. Background Jobs (Optional)

Flutter background service: If your home automation system requires background tasks or periodic updates, this package might be useful.

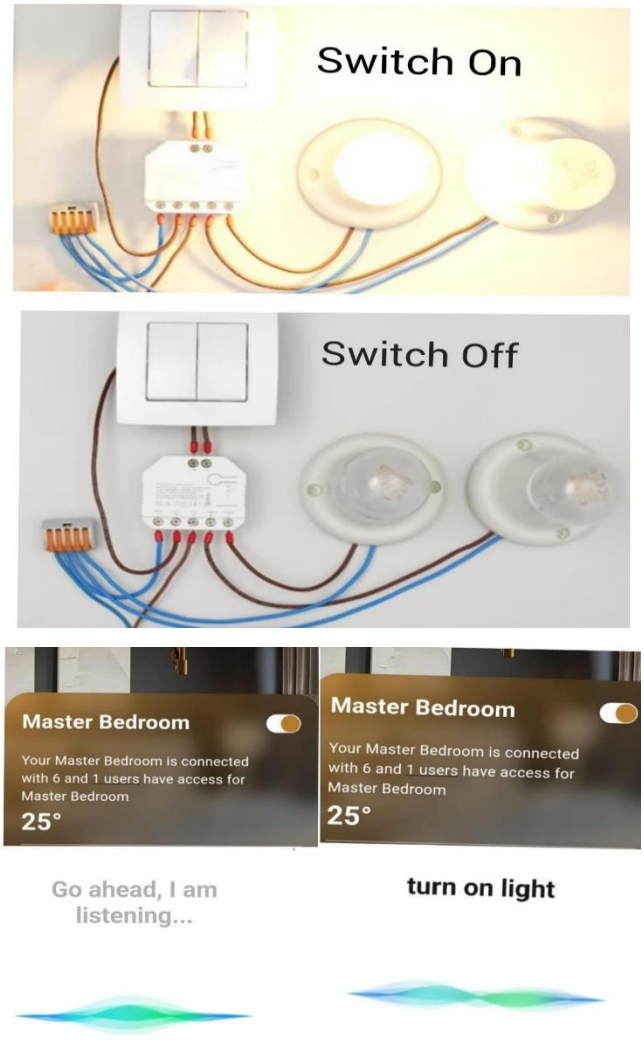
Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

In testing and evaluating the home automation system we can check that if all the smart devices and features work correctly and smoothly. This process includes setting up the system, making sure each device responds to commands, and ensuring the system as a whole operates smoothly without any glitches and can be controlled easily through apps or voice assistants. It also looks for testing the system's reliability over time, and making sure it's secure from unauthorized access.

6.1. Use Case Testing



UCT Figure No 26.

6.2. Equivalence partitioning

6.2.1. Light Control

- Valid Inputs:
    i. Turning lights on
    ii. Turning lights off
    iii. Adjusting brightness levels (e.g., 0% to 100%)
- Invalid Inputs:
    i. Negative brightness values (e.g., -10%)
    ii. Brightness values over 100% (e.g., 150%)
    iii. Non-numeric values (e.g., "bright")

6.2.2. Fan Control

- Valid Inputs:
    i. Turning fan on
    ii. Turning fan off
    iii. Adjusting fan speed (e.g., 1 to 5)
- Invalid Inputs:
    i. Speeds below the valid range (e.g., 0)
    ii. Speeds above the valid range (e.g., 6)
    iii. Non-numeric values (e.g., "fast")

6.2.3. Socket Control

- Valid Inputs:
    i. Turning socket on
    ii. Turning socket off
- Invalid Inputs:
    i. Commands other than on/off (e.g., "activate")
    ii. Non-Boolean values (e.g., 123)

6.2.4. Voice Control Module

- Valid Inputs:
    i. Recognizable voice commands for lights (e.g., "Turn on the lights")
    ii. Recognizable voice commands for fan (e.g., "Set fan speed to 3")
    iii. Recognizable voice commands for socket (e.g., "Turn off the socket")
- Invalid Inputs:
    i. Unrecognizable commands (e.g., mumbled speech)
    ii. Commands in unsupported languages
    iii. Background noise interpreted as commands

6.3. Boundary value analysis

6.3.1. Light Control System

Light Intensity Levels:

- Lower Boundary: Minimum intensity, e.g., 0% (off)
- Upper Boundary: Maximum intensity, e.g., 100% (full brightness)
- Test Cases:
    - Just below the lower boundary (-1%)
    - At the lower boundary (0%)
    - Just above the lower boundary (1%)
    - Just below the upper boundary (99%)
    - At the upper boundary (100%)
    - Just above the upper boundary (101%)

6.3.2. Fan Speed Control

Fan Speed Levels:

- Lower Boundary: Minimum speed level, e.g., 0 (off)
- Upper Boundary: Maximum speed level, e.g., 5 (highest speed)

- Test Cases:
    - Just below the lower boundary (-1)
    - At the lower boundary (0)
    - Just above the lower boundary (1)
    - Just below the upper boundary (4)
    - At the upper boundary (5)
    - Just above the upper boundary (6)

6.3.3. Socket Control

Socket Power State:

- States: Off (0) and On (1)
- Test Cases:
    - Just below the lower boundary (-1)
    - At the lower boundary (0)
    - Just above the lower boundary (1)
    - Just below the upper boundary (0)
    - At the upper boundary (1)
    - Just above the upper boundary (2)

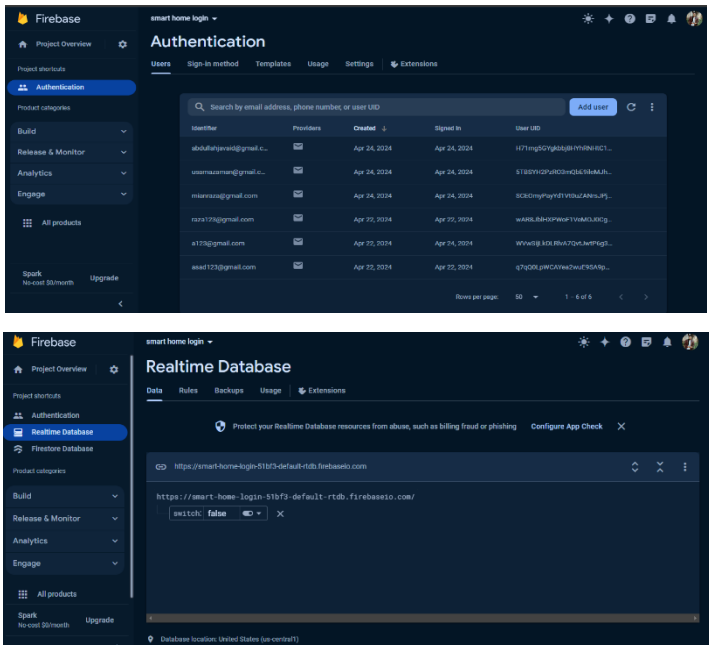
6.3.4. Voice Assistant Command Length

Voice Command Length:

- Lower Boundary: Minimum length of command, e.g., 1 word
- Upper Boundary: Maximum length of command, e.g., 10 words
- Test Cases:
    - Just below the lower boundary (0 words)
    - At the lower boundary (1 word)

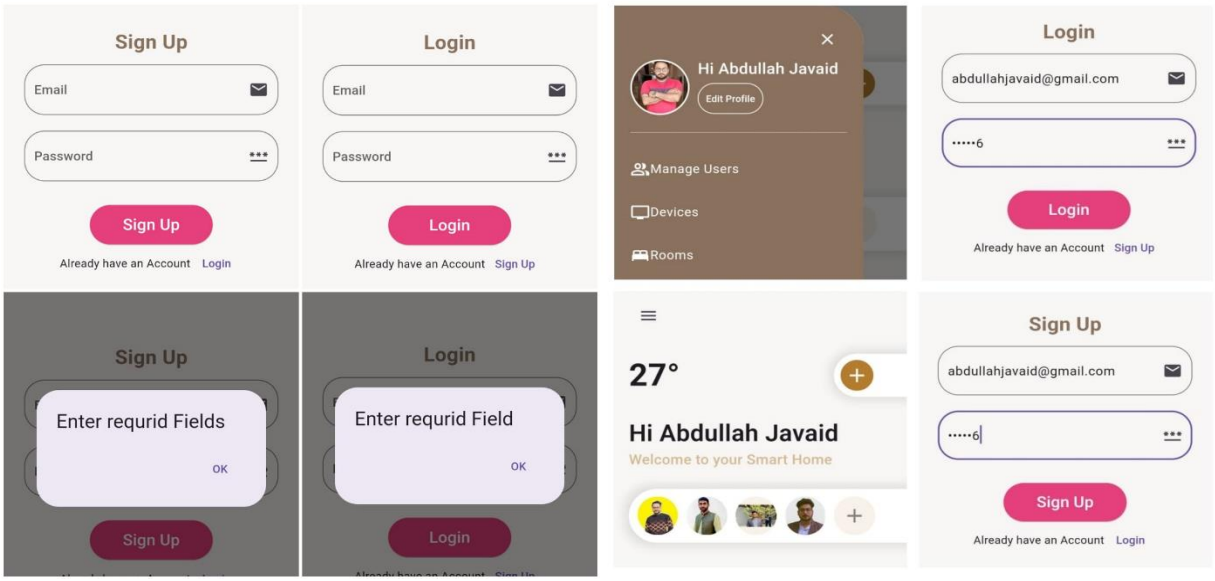
- Just above the lower boundary (2 words)
- Just below the upper boundary (9 words)
- At the upper boundary (10 words)
- Just above the upper boundary (11 words)

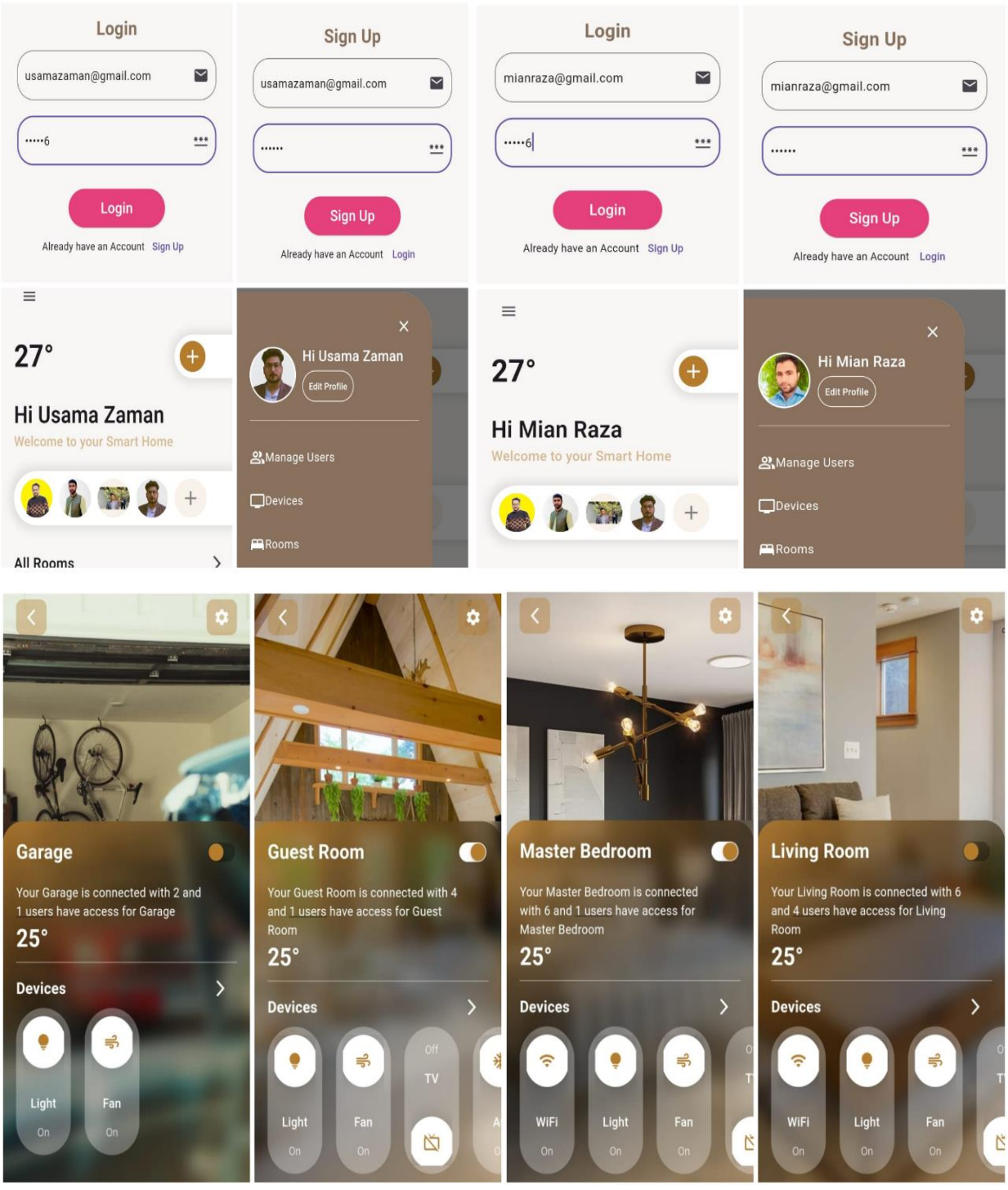
6.4. Data flow testing



DFT Figure No 27.

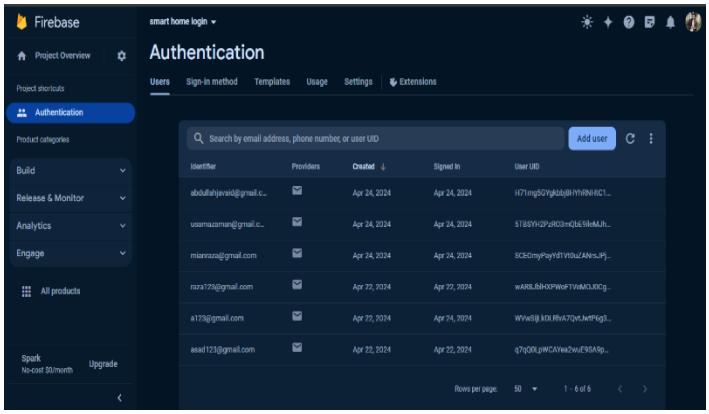
6.5. Unit testing

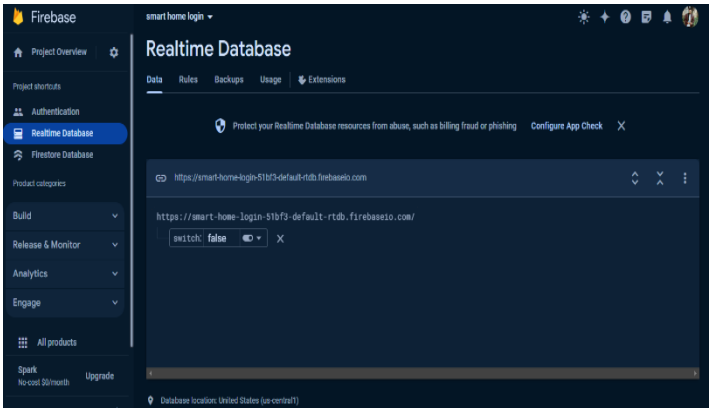




UT Figure No 28.

6.6. Integration testing






```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>
#include<DHT.h>

#define FIREBASE_HOST "https://console.firebase.google.com/project/smart-home-login-51bf3/database/smart-home-login-51bf3-default-rtdb/data/~2F"                                                    //
database URL
#define FIREBASE_AUTH "qOl4gs8Fr5NYeyn7vOvrMXw3ppRZSPuR2KCcOz2Q"
#define WIFI_SSID "sweet home"
#define WIFI_PASSWORD "Sweet@home"
#define DH
#define DHTTYPE DHT11

  DHT dht(DHTPIN,DHTTYPE);

  String myString;


 bool light;
 unsigned long previousMillis=0;
 const long interval=10000;
```

```
void setup()
{

  pinMode(5,OUTPUT);
  Serial.begin(115200);
  delay(500);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED)
     {
    Serial.print(".");
    delay(500);
     }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  dht.begin();

   Serial.println("DHT11 Humidity & temperature Sensor\n\n");
  delay(1000);
}
 void readDatas(){
```

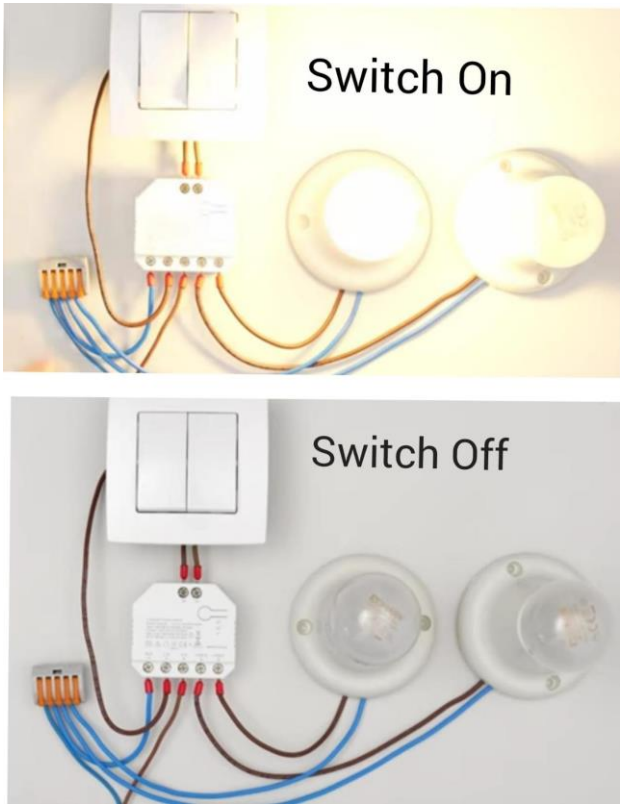
```
float h=dht.readHumidity();
float t=dht.readTemperature();
Serial.println(t);
Serial.println(h);
Firebase.setFloat("Data/Temperature:",t);
Firebase.setFloat("Data/Humidity:",h);
    }

void loop()
{

 light=Firebase.getBool("LightState/switch");
 Serial.println(light);
 if(light==true){
 digitalWrite(5,HIGH);
 }
 if(light==false){
   digitalWrite(5,LOW);
 }
 unsigned long currentMillis=millis();
 if(currentMillis- previousMillis>=interval){
 readDatas();
 previousMillis=currentMillis;
 }
```

IT Figure No 29.

6.7. Performance testing



PT Figure No 30.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

In today's busy world, managing various tasks in our homes can be challenging and time consuming. We often forget to turn off lights and fans, or control appliances, leading to energy waste and also enhance our electricity bills. Specially, the disable person can feel trouble to manage the lights and other home appliance, due to their disability the electrical equipment remain on which can cause an accident and also the wastage of electricity and it leads to power shortage which is not suitable.

Home automation system aims to solve these issues by allowing us to control the electronic devices like fans, lights and socket's by using our smart phone through technology. We can also provide the feature of eco system, which prove the beneficial for the disable persons. By using the home automation system, we can control our devices smartly or remotely and also save the electricity from the wastage.

A home automation system is a technological solution that enables automating the bulk of electronic, electrical and technology-based tasks within a home. It uses a combination of hardware and software technologies that enable control and management over appliances and devices within a home. A home automation system is like a smart assistant for your house that controls things for you. We can control our home devices by using the simple and user friendly application with the feature of eco-system. Home automation system making our lives more efficient, comfortable, and environmentally friendly.

7.2. Achievements and Improvements

7.2.1. Achievements

7.2.1.1. Integration of Multiple Devices

Successfully integrated lights, fans, and sockets into a combined home automation system, allowing for centralized control.

7.2.1.2. Voice Assistant Module

Implemented a voice assistant module that enhances user interaction by enabling voice commands to control various devices.

7.2.1.3. Remote Access

Developed functionality for remote access, allowing users to control their home devices via a smartphone app.

7.2.1.4. User-Friendly Interface

Designed an intuitive and user-friendly interface that simplifies the process of controlling and monitoring home devices.

7.2.1.5. Security Measures

Implemented security protocols to ensure the system is protected against unauthorized access.

7.2.2. Improvement

7.2.2.1. Enhanced Voice Recognition

Improve the accuracy and responsiveness of the voice assistant module to better understand and execute user commands, especially in noisy environments.

7.2.2.2. Data Analytics and Insights

Implement advanced data analytics to provide users with insights into their energy consumption patterns and suggest ways to optimize usage.

7.2.2.3. Scalability

Ensure the system is scalable to accommodate additional devices and more complex automation scenarios as user needs evolve.

7.2.2.4. Enhanced Security Features

Continuously update security protocols to stay ahead of potential vulnerabilities and ensure robust protection against emerging threats.

7.2.2.4. Environmental Sensors

Integrate additional environmental sensors to provide a more comprehensive smart home experience and enable more advanced automation routines.

7.2.2.5. Seamless Automation

Created automation routines that allow for seamless interaction between devices, such as turning on the lights when motion is detected or adjusting the fan speed based on room temperature.

7.3. Critical Review

The home automation system project is good at integrating important household appliances such as lights, fans and sockets in addition to having voice assistant that helps control them without any physical interaction. Remote access, user-friendly interface, energy efficiency features, security measures and customization options are among the qualities it possesses. However, there are challenges like voice recognition accuracy; limited compatibility with other service providers and need for advanced data analytics. Other areas such as scalability, user support, continuous release of security patches, offline functionality as well as environmental sensor integration should not be ignored. If these areas can be improved upon then the system will be more reliable functional and provide more satisfaction to users by leveraging on its strengths already in place.

7.4. Lessons Learnt

The analysis of the home automation system project had several lessons. Merging many devices such as bulbs, fans and plugs in one network needs a lot of planning and testing to ensure that everything works well together. This is especially needed when we are working with highly variable voice commands for different languages spoken in noisy environments. A user-friendly interface coupled with a range of customizable options like skin themes can greatly increase user satisfaction making technology more accessible and attractive.

Creating scalability from the start ensures that it continues growing to meet future needs without affecting its performance or user experience. Its remote access capability gives users control even when they are not near it as well as peace of mind knowing it will work properly always Integrating energy saving functionality deals with environmental concerns while at the same time reducing utility bills for consumers.

Thus, important documents and reliable customer support are essential for helping users understand how to use all features of this application and fix any bugs which may occur. The

rapidly changing technology world demands that security and functionality of the system should be kept up to date through continuous updates. Feedback from users is critical in pinpointing areas for improvement as well as making certain that the system meets real life requirements and expectations. Reliability as well as user confidence demands that there is offline functionality even if there is power outage or no internet coverage due to various reasons.

The smart home experience can be enhanced and more comprehensive automation capabilities achieved by introducing other environmental sensors like those for humidity and air quality. These are some of the key lessons learnt on user-centric design, strong security, scalability, continuous improvement and the importance of feedback in developing efficient and reliable home automation systems.

7.5. Future Enhancements/Recommendations

7.5.1. Advanced Voice Recognition

Improve better precision and responsiveness by implementing the cutting-edge voice recognition technology, which would make it possible for users to use it even in loud noise and varied accents.

7.5.2. Elevated Third-Party Integration

This will allow users to integrate more third-party smart devices with their smart home systems, thus giving them flexibility and wider choice of products.

7.5.3. Scalability Improvements

To enhance scalability, system architecture has been improved in order to enable addition of more devices by users while maintaining a high level of automation sophistication without degrading system performance.

7.5.4. Regular Security Updates

Establish a routine for regular security updates and audits to protect against emerging threats. Implementing features such as two-factor authentication and encrypted communication can further enhance system security.

7.5.5. Environmental Sensors Integration

Incorporate additional sensors for monitoring humidity, air quality, and other environmental factors. This will enable more comprehensive automation routines and improve the overall smart home experience.

7.5.6. Machine Learning for Automation

Utilize machine learning algorithms to analyze user behavior and automatically adjust settings for lights, fans, and other devices to optimize comfort and energy efficiency.

7.5.7. Geo-Fencing Capabilities

Implement geo-fencing to automate actions based on the user's location. For instance, lights and climate control systems can be adjusted when the user is approaching home or leaving.

7.5.8. User Training and Support

Develop comprehensive user guides and video tutorials to assist users in setup and utilization of the home automation system.

7.5.9. Smart Alerts and Notifications

Implement a system for smart alerts and notifications to inform users of important events, such as security breaches, unusual energy consumption, or maintenance reminders.

Appendices

Appendix A: Glossary

A.1. Automation

You should not confuse “home automation” with the ability to control one aspect of your home like your speakers or your lights. Home automation is about integrating different devices in your home to work together in unison while controlling them with one central controller. Things that can work together in unison include smart thermostats, lights, cameras, and other appliances and electronics.

A.2. Lighting control system

This is a system where lights both in and out of your home are controlled from one central source instead of a variety of switches.

A.3. Smart Device

The difference between your regular light bulb and a smart light bulb is that the smart light bulb is wireless and connected to the internet. Smart devices can be controlled by using an app remotely as well as scheduled using software to do certain actions at certain times. A Nest Thermostat and the Philips Hue Light Bulbs are both examples of smart devices.

A.4. ZigBee

ZigBee is a low-power communication network that allows for low latency device communication. Devices that are ZigBee compatible do not need a central access point because they are capable of sending and receiving simple signals to and from each other.

Reference and Bibliography

Reference and Bibliography

[1] Al-Ali, Al-Rousan, “Java based home automation system”, IEEE transaction on consumer electronics, vol.50, no2, pp.498-504,May 2004

[2] Byenogkwan Kang, Sunghoi Park et.al “IoT Based monitoring systems using Tri-level context making model for Smart Home Services”, 2015 IEEE International conference on consumer Electronics9(ICCE), 2015.

[3] Dhiraj Sunehra, M.Veena, “Implementation of interactive home automation systems based on Email and Bluetooth technologies”, 2015 International Conference On Information Processing”, Vishwakarma Institute of Technology, Dec 16-19,2015.

[4] Rakesh k. Deore et.al (2015), “Internet of Things Based Home Appliances Control”, International Conference on Intelligence Communication Networks, 2015.

[5] Shariq Suhail Md et.al, ”Multi-Functional Secured Smart Home”,2016 International conference on advances in computing communication and information,2016.

[6] Muhammad Izhar Ramli, Mohd Helmy Abd Wahab, Nabihah, “TOWARDS SMART HOME: CONTROL ELECTRICAL DEVICES ONLINE”, Nornabihah Ahmad International.

[7] Shariq Suhail Md et.al, ”Multi-Functional Secured Smart Home”,2016 International conference on advances in computing communication and information,2016.

[8] M. A. Hoque and C. Davidson, “Design and implementationof an IoT-based smart home security system,” InternationalJournal of Networked and Distributed Computing, vol. 7, no. 2,pp. 85–92, 2019.

[9] A. S. Abdulraheem, A. A. Salih, A. I. Abdulla et al., “Homeautomation system based on IoT,” 2020.

[10] Khan, M.Y.; Javaid, N.; Khan, M.A.; Javaid, A.; Khan, Z.A.; Qasim, U. Hybrid DEEC: Towards efficient energy utilization in wireless sensor networks. World Appl. Sci. J. 2013.

[11] D. D. Swetanjal Murati Dutta, Pitanjal Murati Dutta and D. Roy, “Home automation,” International Journal of Scientific Engineering Research, 2015.