Smartify: Transforming Traditional Appliances into Smart Devices For Seamless Home Automation

A report submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Technology

In

Computer Science and Engineering (Cyber Security)

by

CH. V Vishnu Vardhan Raju	2011CS040018
Diddi Akshaya	2011CS040019
Jakka Shiva Kumar Reddy	2011CS040027
Katragadda Nikhitha	2011CS040036

Under the esteemed guidance of

Mr. V Vijaykumar Dasari
Assistant Professor

Department of Computer Science & Engineering (Cyber Security)

School of Engineering

Malla Reddy University

Maisammaguda, Dulapally, Hyderabad, Telangana 500100

2024

Smartify: Transforming Traditional Appliances into Smart Devices For Seamless Home Automation

A report submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Technology

In

Computer Science and Engineering (Cyber Security)

by

CH. V Vishnu Vardhan Raju	2011CS040018
Diddi Akshaya	2011CS040019
Jakka Shiva Kumar Reddy	2011CS040027
Katragadda Nikhitha	2011CS040036

Under the esteemed guidance of

Mr. V Vijaykumar Dasari
Assistant Professor

Department of Computer Science & Engineering (Cyber Security)

School of Engineering

Malla Reddy University

Maisammaguda, Dulapally, Hyderabad, Telangana 500100

2024

MALLA REDDY UNIVERSITY
(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

Department of Computer Science and Engineering - Cybersecurity

CERTIFICATE

This is to certify that the project report entitled **"Smartify : Transforming Traditional Appliances into Smart Devices For Seamless Home Automation"**, submitted by **CH. V Vishnu Vardhana Raju (2011CS040018), Diddi Akshaya (2011CS040019), Jakka Shiva Kumar Reddy (2011CS040027), Katragadda Nikhitha (2011CS040036),** towards the partial fulfillment for the award of **Bachelor's Degree in Cybersecurity** from the **Department of Computer Science and Engineering, Malla Reddy University,** Hyderabad, is a record of bonafide work done by him/ her. The results embodied in the work are not submitted to any other University or Institute for award of any degree or diploma.

Internal Guide:
Mr. V Vijaykumar Dasari
Assistant Professor

Head of the Department
Dr. G. Anand Kumar
CSE (Cyber Security & IOT)

External Examiner

DECLARATION

We hereby declare that the project report entitled **“Smartify: Transforming Traditional Appliances into Smart Devices For Seamless Home Automation”** has been carried out by us and this work has been submitted to the Department of Cybersecurity, Malla Reddy University, Hyderabad in partial fulfillment of the requirements for the award of degree of Bachelor of Technology. We further declare that this project work has not been submitted in full or part for the award of any other degree in any other educational institutions.

Place:

Date:

CH. V Vishnu Vardhana Raju	2011CS040018
Diddi Akshaya	2011CS040019
Jakka Shiva Kumar Reddy	2011CS040027
Katragadda Nikhitha	2011CS040036

ACKNOWLEDGEMENT

We extend our sincere gratitude to all those who have contributed to the completion of this project report. Firstly, We would like to extend our gratitude to **Dr. V. S. K Reddy, Vice-Chancellor**, for his visionary leadership and unwavering commitment to academic excellence.

We would also like to express my deepest appreciation to our project guide **Mr. V Vijaykumar Dasari, Assistant Professor**, whose invaluable guidance, insightful feedback, and unwavering support have been instrumental throughout the course of this project for successful outcomes.

We extend our gratitude to **Dr. G. Latha, PRC-convenor,** for giving valuable inputs and timely guidelines to improve the quality of our project through a critical review process. We thank our project coordinator **Mr. Ramesh Khanna,** for his timely support.

We are also grateful to **Dr. G. Anand Kumar, Head of the Department of Cybersecurity,** for providing us with the necessary resources and facilities to carry out this project.

We would like to thank **Dr. Kasa Ravindra, Dean, School of Engineering,** for his encouragement and support throughout my academic pursuit. My heartfelt thanks also go to **Dr. Harikrishna Kamatham, Associate Dean School of Engineering** for his guidance and encouragement.

We are deeply indebted to all of them for their support, encouragement, and guidance, without which this project would not have been possible.

CH. V Vishnu Vardhana Raju	2011CS040018
Diddi Akshaya	2011CS040019
Jakka Shiva Kumar Reddy	2011CS040027
Katragadda Nikhitha	2011CS040036

ABSTRACT

Smartify proposes a cost-effective solution to upgrade conventional household appliances into smart devices, enabling seamless control and management through a mobile application. In today's era of IoT (Internet of Things) and home automation, the demand for smart appliances continues to rise. However, many households still possess non-Wi-Fi enabled devices, limiting their integration into smart home ecosystems. Smartify bridges this gap by providing a retrofitting solution that empowers users to convert their existing appliances into smart devices without the need for expensive replacements. The project focuses on leveraging smart plugs and switches to enable remote control of power supply to various appliances such as air conditioners, televisions, lamps, and more. By incorporating Wi-Fi connectivity and integrating with a dedicated mobile application, users gain the ability to monitor and manage their appliances from anywhere, at any time. The intuitive user interface of the Smartify app offers features such as scheduling, energy monitoring, and personalized settings, enhancing convenience and efficiency in household operations.

Implementation involves straightforward installation procedures, ensuring accessibility for users with varying technical expertise. Additionally, the system prioritizes security measures to safeguard user data and protect against unauthorized access. Smartify represents a practical and scalable solution for modernizing homes and transitioning towards smart living environments.

List of Figures:

List of Screenshots:

Table of Contents

Contents **PageNo.**

CHAPTER – 1
INTRODUCTION

1.1 Problem Definition & Description

In the realm of modern smart homes, the integration of various appliances and devices has revolutionized the way we interact with our living spaces. However, a significant challenge persists in the form of non-Wi-Fi enabled appliances. These appliances, lacking internet connectivity, present a series of hurdles that hinder the seamless integration and functionality of smart home ecosystems.

The problem statement revolves around 8 key challenges: Non-Wi-Fi Enabled Appliances, Integration Limitations, Technological Gap, and User Accessibility Challenges, Manual Operation Hassles, Energy Inefficiency, Market Demand Discrepancy, Convenience Gap. Each of these aspects contributes to the overarching issue of seamlessly integrating non-Wi-Fi enabled appliances into modern smart home environments. Understanding and addressing these challenges are crucial steps in devising effective solutions to enhance the functionality and accessibility of smart home systems.

Non-Wi-Fi Enabled Appliances:

Non-Wi-Fi enabled appliances represent a considerable portion of existing household devices. Ranging from traditional refrigerators and ovens to simple light switches, these appliances operate in isolation, devoid of the connectivity that characterizes contemporary smart devices. Their inability to connect to the internet prohibits remote control, monitoring, and automation, limiting user convenience and energy efficiency.

Integration Limitations:

The absence of Wi-Fi connectivity poses a significant barrier to the integration of non-Wi-Fi enabled appliances into modern smart home networks. As smart homes increasingly rely on Wi-Fi and internet-connected devices for communication and control, the inclusion of legacy appliances becomes challenging. This integration limitation restricts the realization of a unified, interconnected smart home environment, preventing users from harnessing the full potential of their home automation systems.

Technological Gap:

A notable technological gap emerges between non-Wi-Fi enabled appliances and the advancements in smart home technology. While newer appliances boast Wi-Fi connectivity and smart features, older devices remain disconnected and outdated. This disconnect creates compatibility issues and functionality disparities within the home, perpetuating inefficiencies and inhibiting the seamless operation of smart home systems.

User Accessibility Challenges:

The lack of Wi-Fi connectivity in non-Wi-Fi enabled appliances exacerbates user accessibility challenges. Without remote control capabilities, users must resort to manual operation or rely on outdated control interfaces, leading to inconvenience and inefficiency in managing their home devices. This poses a significant barrier to user adoption and hampers the potential benefits of smart home technology in enhancing comfort, convenience, and energy savings.

Manual Operation Hassles:

Manual operation hassles refer to the inconvenience and effort required for users to physically interact with non-Wi-Fi enabled appliances. Without remote control capabilities, users must manually adjust settings, switch appliances on or off, or monitor their performance. This manual intervention can be time-consuming, particularly in scenarios where multiple appliances need to be managed simultaneously. Additionally, manual operation limits flexibility and convenience, as users may need to be physically present near the appliance to make adjustments, disrupting their daily routines and activities.

Energy Inefficiency:

Energy inefficiency stems from the lack of remote monitoring and control capabilities in non-Wi-Fi enabled appliances. Without the ability to adjust settings or schedule operations remotely, users may inadvertently waste energy by leaving appliances running when not needed or operating them at suboptimal efficiency levels. For example, a user may forget to turn off lights or adjust thermostat settings before leaving the house, resulting in unnecessary energy consumption. This inefficiency not only contributes to higher utility bills but also has environmental implications, as it increases overall energy consumption and carbon emissions.

Market Demand Discrepancy:

Market demand discrepancy refers to the misalignment between consumer demand for smart home technology and the prevalence of non-Wi-Fi enabled appliances in the market. As more consumers seek the convenience, efficiency, and connectivity offered by smart home systems, there is a growing demand for appliances that can integrate seamlessly into these ecosystems. However, the widespread availability of legacy appliances without Wi-Fi connectivity creates a gap between consumer expectations and market offerings. This disconnect leads to market fragmentation, where consumers may struggle to find compatible solutions that meet their needs, resulting in frustration and hesitation to adopt smart home technology.

Convenience Gap:

The convenience gap highlights the disparity between the user experience provided by non-Wi-Fi enabled appliances and the seamless automation and control offered by modern smart home systems. Non-Wi-Fi enabled appliances typically lack advanced features such as remote monitoring, scheduling, and integration with other smart devices. As a result, users experience a gap in convenience, where tasks that could be automated or remotely managed require manual intervention and oversight. This convenience gap diminishes the overall user experience, as users miss out on the time-saving benefits and enhanced control afforded by smart home technology.

In conclusion, addressing the challenges posed by non-Wi-Fi enabled appliances, integration limitations, technological gaps, user accessibility issues, manual operation hassles, energy inefficiency, market demand discrepancies, and convenience gaps is crucial for realizing the full potential of smart home technology in enhancing comfort, convenience, and efficiency in modern living spaces.

1.2 Objectives of the Project

The objectives of the project revolve around the retrofitting of non-Wi-Fi enabled appliances into smart home environments, aiming to enhance user satisfaction, privacy, and security. These objectives are carefully designed to address the challenges associated with integrating legacy appliances into modern smart home ecosystems while ensuring compatibility, accessibility, and cost-effectiveness. By establishing clear objectives, the project aims to guide its efforts towards delivering tangible outcomes that meet user needs and industry standards.Let's delve into each objective to understand its significance and impact on the project's overall success.

Cost-Effective Retrofitting:

The objective of implementing cost-effective retrofitting solutions is to ensure that users can seamlessly upgrade their existing non-Wi-Fi enabled appliances to integrate them into their smart home ecosystems without incurring significant expenses. This objective involves thorough research into affordable retrofitting methods and technologies that minimize installation costs while maximizing functionality. By identifying economical retrofitting options, the project aims to make smart home technology more accessible to a wider range of users, thereby facilitating the transition to a connected home environment without financial barriers.

Enhanced User Experience:

The project aims to prioritize the enhancement of user experience by focusing on the design and development of intuitive and user-friendly interfaces for controlling and monitoring smart home devices. This objective entails creating interfaces that are easy to navigate and understand, ensuring that users can effortlessly interact with their appliances remotely via smartphones, tablets, or other smart devices. By placing a premium on user experience, the project seeks to cultivate a positive and engaging interaction between users and their smart home systems, ultimately fostering greater satisfaction and utilization of the technology.

Seamless Integration:

Seamless integration is a key objective of the project, focusing on the development of solutions that facilitate the smooth incorporation of non-Wi-Fi enabled appliances into existing smart home ecosystems. This objective involves devising interoperable technologies and protocols that enable seamless communication and interaction between legacy appliances and modern

smart devices and platforms. By ensuring compatibility and integration across diverse devices and systems, the project aims to create a cohesive and unified smart home environment where appliances work together harmoniously to enhance convenience and efficiency for users.

Accessible Control:

The objective of providing accessible control aims to empower users with convenient and flexible means of managing their smart home devices. This involves developing control interfaces that are accessible to users of all abilities, including those with disabilities or special needs. Additionally, the project aims to enable remote control and monitoring of appliances through various means, such as mobile applications, voice commands, or IR remote integration. By prioritizing accessibility, the project seeks to ensure that all users can easily interact with and benefit from their smart home technology, regardless of their individual capabilities or limitations.

Retrofitting Solution Development:

The objective of retrofitting solution development entails the creation and implementation of innovative technologies and methodologies for retrofitting non-Wi-Fi enabled appliances with smart capabilities. This involves researching and testing various retrofitting techniques, such as hardware upgrades, sensor integration, or software modifications, to determine the most effective and reliable solutions for different types of appliances.
By developing comprehensive retrofitting solutions, the project aims to provide users with practical and efficient means of upgrading their existing appliances to enjoy the benefits of smart home technology.

Mobile Application Creation, IR Remote Integration:

A key objective of the project is to develop a user-friendly mobile application that enables seamless control and monitoring of smart home devices. This involves designing an intuitive interface with comprehensive features, allowing users to remotely manage their appliances from anywhere with an internet connection.
Additionally, the project aims to integrate IR remote functionality into the application, enabling users to control appliances that rely on traditional infrared remotes. By combining mobile

application development with IR remote integration, the project seeks to provide users with a versatile and convenient solution for managing their smart home devices**.**

Integration with Smart Plugs and Switches:

The objective of integrating non-Wi-Fi enabled appliances with smart plugs and switches is to expand the compatibility and functionality of existing devices within the smart home ecosystem. This involves developing solutions that enable seamless communication and interaction between legacy appliances and smart plugs or switches, allowing users to remotely control and automate their appliances' operation. By integrating with smart plugs and switches, the project aims to enhance the flexibility and convenience of managing non-Wi-Fi enabled appliances, enabling users to optimize energy usage and streamline daily routines.

User Training and Documentation:

User training and documentation are essential objectives aimed at providing users with the knowledge and resources necessary to effectively utilize and maintain their smart home systems. This involves creating comprehensive user guides, tutorials, and instructional materials that educate users on the setup, operation, and troubleshooting of their smart home devices. Additionally, the project aims to offer training sessions or workshops to familiarize users with the features and capabilities of their smart home technology. By prioritizing user training and documentation, the project seeks to empower users with the skills and confidence to maximize the benefits of their smart home systems and overcome any challenges or issues they may encounter.

1.3 Scope of the Project

The project focuses on retrofitting non-Wi-Fi enabled appliances into smart home environments, with the overarching goal of enhancing user satisfaction, privacy, and security. This involves several key components:

Objective Establishment and Stakeholder Engagement:

The project begins with clearly defining its objectives, such as upgrading existing appliances to integrate them into smart home systems, enhancing user experience, and ensuring compatibility with modern technology standards. Stakeholder engagement plays a crucial role in this phase, as input from users, manufacturers, developers, and regulatory bodies helps shape project goals and requirements.

Timeline, Milestone Definition, and Resource Allocation:

A detailed timeline is established, outlining the project's various stages and milestones. Milestones serve as checkpoints to track progress and ensure timely completion of project deliverables. Resource allocation is carefully planned to support project activities within budgetary constraints, including manpower, technology infrastructure, and materials necessary for retrofitting appliances.

User Preferences Analysis and Appliance Compatibility Assessment:

Understanding user preferences is essential for tailoring retrofitting solutions to meet their needs effectively. Through surveys, interviews, and usability studies, the project team gathers insights into user expectations regarding smart home technology. Additionally, a thorough assessment of non-Wi-Fi enabled appliances is conducted to determine their compatibility with retrofitting solutions and integration with smart home systems.

Budget, Technical Limitations, and Risk Management:

The project considers budgetary constraints and technical limitations to develop feasible solutions that meet both financial and technical requirements. Potential risks are identified, and risk mitigation strategies are devised to minimize their impact on project outcomes. This proactive approach to risk management ensures smooth project execution and timely delivery of results.

Data Privacy, Security Measures, and Agile Development Methodology:

Protecting user data and ensuring privacy and security are paramount throughout the project. Robust data privacy and security measures are implemented to safeguard sensitive information and comply with relevant regulations. The project adopts an agile development methodology, allowing for iterative development and adaptation to changing requirements while maintaining a focus on delivering high-quality results.

Regular Progress Evaluation and Stakeholder Collaboration:

Regular progress evaluations are conducted to assess project status, identify any risks or issues, and make necessary adjustments to stay on track with project objectives. Effective collaboration with stakeholders is facilitated through established communication channels, ensuring alignment with user needs and project goals throughout the project lifecycle.

By encompassing these elements, the project scope ensures a comprehensive approach to retrofitting non-Wi-Fi enabled appliances into smart home environments, ultimately enhancing user satisfaction and promoting the adoption of smart home technology.

CHAPTER -2
SYSTEM ANALYSIS

System Analysis is the process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. It is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problemsolving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

2.1 Existing System

In many households, conventional appliances lacking smart capabilities and Wi-Fi connectivity remain prevalent, presenting both advantages and disadvantages. While these appliances may offer reliability and familiarity, they are limited in their integration into modern smart home ecosystems. The absence of Wi-Fi connectivity prevents seamless remote control and management, leading to inefficiencies in household operations. Traditional methods of upgrading to smart appliances often entail high replacement costs, rendering them inaccessible to users with budget constraints or those reluctant to replace functional appliances.

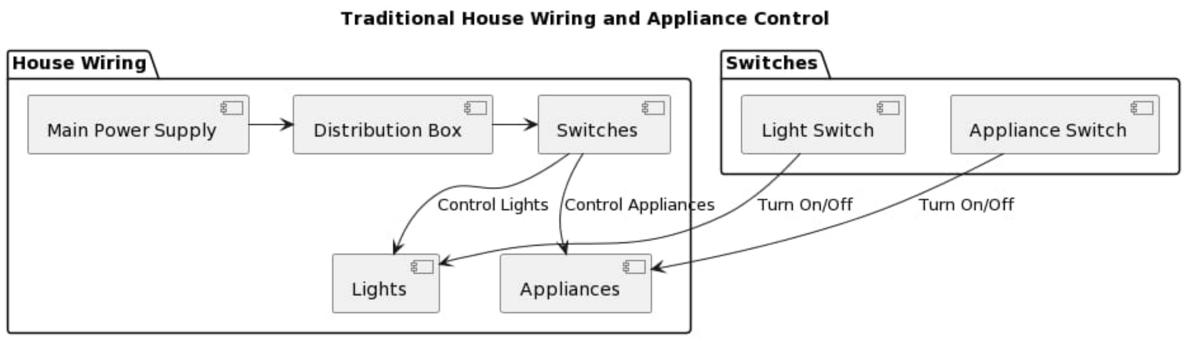


Figure 2.1.1 Existing System

2.1.1 Background & Literature Survey

Many researchers have proposed various solutions in smart home areas such as home automation, safety, security, and energy consumption for human life facilitation. Some of these solutions are discussed in the following.

2.1.1.1 Home Automation, Safety, and Security

There are many approaches available for home automation, safety, and security to enhance the lifestyle of human beings. The automation part controls home appliances, such as lights, fans, and ACs [16]. Several researchers deployed a microcontroller-based home automation system in which a microcontroller is used as a server, and an Androidapplication is used to access the system through the Internet. Usually, the Arduino board is used as a server, which is a low-cost microcontroller, and is a part of the computer, and

can run only one program at a time [17,18]. This approach is further extended by [27] by integrating temperature and current (voltage and ampere) sensors with a home automation system. A wireless home automation system built on Arduino is presented in [14].

It provides two modes. A manually operated mode to control home appliances through a smartphone and a self-automated mode to automatically control home appliances throughconnected sensors. The hardware implementation with Matlab-GUI made the system expensive (need a PC) and required more power to run the system. A system to control and tor home appliances, using the WLAN network based on the Arduino is presented in [28], but HTML5 supportable devices can only access this system. It also requires a PC server to run the system making it less cost-effective. A hybrid home automation system is developed in [29] in which X10 wired, and ZigBee wireless technology is used.

The approach followed smart task scheduling with a heuristic for the RCPSP (resourceconstrained scheduling problem). The author in [30] developed a home automation system to control and monitor home appliances through the home getaway, which is based on ZigBee, Android application, and Arduino.

Some researchers have proposed Bluetooth enabled home automation system. For instance, in [31], the authors developed a home automation system that includes a primary controller and Bluetooth sub-controller connected to a single home device. Another Bluetooth-based home automation system which uses cell phone was presented in [2], in which home appliances are connected to the Arduino board. The cell phone and Arduino are connected through Bluetooth technology. A security filter has been used in this approach to secure the system from unauthorized access. The main disadvantage of a Bluetooth-based home automation system is that it can only be accessed in an indoor environment or within the Bluetooth range.

A voice control home automation system has been presented in [32], which comprises two main parts, i.e., voice recognition system, and wireless system. The Android application was used for voice recognition and Bluetooth technology for connecting different modules wirelessly. In this system, the authors used three different technologies such as Bluetooth, Wi-Fi, and ZigBee. Using multiple communication devices requiring separate protocols for communicating with each component makes the system unfeasible and increases the implementation cost. Another voice-based automation system has been developed in [33], but the Open Platform Communications server makes the system costly.

A Zigbee-based home automation system is proposed in [13], including home network devices and home network gateway to operate home appliances. Besides home automation, the safety and security is another key feature of a smart home, where several researchers proved their contribution. A home automation and home security system based on ZigBee is presented in [34], providing multi-home communication capability. However, the user controls home appliances by sending a command through SMS to the main controller. A home safety and security system is proposed in [15],in which an elderly fall and flame and gas detection mechanisms are used to protect the elderly population from any kind of hazard. The security aspect for smart homes based on the passive infrared sensor (PIR) and Arduino is proposed in [35]. In this research work, the authors used a PIR with Arduino for motion detection, and camera sensors are used with PIR for intruder detection. When the PIR sensor detects a motion, the camera sensors will be activated to capture an image. Furthermore, they used a histogram of gradient (HoG) to extract prominent features from the captured image, and these features are fed to the support vector machine (SVM) for intruder detection. If the intruder is detected, the system turns on the alarm to warn the homeowner about the activity.

The experimental results demonstrated that it only takes two seconds for detecting motion,and their approach reached an accuracy of 89% for intruder detection. The system had ahigh misclassification rate and often triggered an alarm for normal activity. There is still room to improve the accuracy in this case by using modern mechanisms. In [36], the authors incorporate fog computing technology to analyze foot size, pressure, and movement in real time for person identification. A predictive learning-based Adaptive Neuro-Fuzzy Inference System (ANFIS) is used for intruder detection. Furthermore, in this approach, they raised an alarm in the case of any emergency in real-time situations. Their work is validated in a smart home environment database selected from an online repository called the UCI, which comprises 49,695 records. It consists of Identity-based parameters, foot size, pressure, namely weight, and movement. Superior performance is achieved by their proposed work as compared to other SoTA prediction

models. Jan et al., [15] proposed a technique for detecting a person falling on the floor, flames, and leakage of any harmful gas detection. This system is aimed for elderly persons and uses an RPI-based prototype that can be easily mounted to the elder people as a safety device. In the case of any emergency, the system is responsible for sending an alarm message to their relatives along with their global position system (GPS) location.

2.1.1.2. Home Energy Consumption

Besides automation, safety, and security, another important aspect of the smart home is the reduction of energy costs. For residential usage, energy consumption is increasing day by day. For this purpose, a different module of a smart home has been implemented, such as controlling home lights automatically considering natural light. Numerous research works proposed that daylight can be substituted for limiting electricity consumption in commercial buildings automatically via light sensors. Therefore, there are many approaches available to decrease residential energy usage [37,38]. Smart home technologies include ICT, sensors, and network capability to automatically switch home appliances through a smartphone application, touch screen, or voice [39]. Smart meters and instruments provide better prospects for the user to efficiently manage and control their home electricity [40]. Han et al., in [41], presented a ZigBee-based energy management technique that measures the usage of energy by home appliances (electrical, electronic devices). They also used a power line communication (PLC)-based approach for the measurement of energy generation. For smart homes, Anvari et al. [42] developed a multi-objective mixed integer nonlinear programming model for optimal energy use. The result showed that the algorithm not only reduced utility bills and domestic energy usage but also provided optimal task scheduling and a thermal comfort zone for the residents. In [43], the researchers provided an efficient mechanism to control the energy consumption of two different climate regimes such as Algiers and Stuttgart, cities in Algeria and Germany. The solution was aimed at a single-family house, but it was not cost-effective due to the implementation cost.

2.1.1.3. IoT Platforms for Smart Home

The network is an important part of smart objects connectivity. Smart objects include controllers, sensors, actuators, and different processors, which are used to control, monitor, and communicate with each other in the network [20]. Smart home takes advantage of cloud computing [44], but there are significant deficiencies in cloud computing including latency and response time. For this purpose, Li et al. proposed a technique in [45] to overcome cloud computing limitations. They studied the problem of data latency and response time of the smart object (used in smart homes and smart cities) in cloud computing and decided to switch from cloud to fog computing, which enabled the real-time interaction of the smart object, overcoming latency and data volume and speed issues. Scalability is another issue in cloud computing as discussed by Faruque et al. [46]. Fog computing provides a better energy management strategy. Adaptability, Scalability, and open-source hardware/software included in the fog computing paradigm facilitate the user to reduce the implementation cost, time, and energy consumption with customized control-as-service.

Perera et al. [47] studied resource wastage issues in cloud computing and network storage. However, a new fog computing paradigm that has limited computational capabilities at the edge cannot address this challenge alone. To address this challenge, both paradigms need to collectively build supportable IoT infrastructure for smart cities. Fog computing faces new privacy and security issues. In the perspective of fog computing, Yi et al. [48] discussed several security issues including data storage security, computation security, and security of the network, and highlighted some other issues regarding data privacy, user privacy, and privacy of the location. Most fog computing applications in IoT only collect data from homogeneous IoT devices but cannot collect data from hybrid IoT devices into one real IoT application. Lu et al. [49] introduced a lightweight privacy-preserving aggregation scheme of data for fog computing to enhance the usage of fog computing in IoT applications. The lightweight privacy-preserving data aggregation (LPDA) employs different privacy techniques, Chinese remainder theorem, homomorphic Paillier encryption, and one-way hash chain technique. Amadeo et al. [50] conducted a study where they highlighted the benefits of fog computing over cloud computing. They rely on Information-Centric Networking to control and monitor the smart home environment and presented a reference architecture as proof of concept.

Edge computing has recently gained a lot of attention in which the data are processed over the edge overcoming dependency, latency, security, and data privacy issues. It is an ideal paradigm for designing efficient home solutions comprising various IoT devices. Our proposed system takes advantage of this paradigm. It has shown to significantly improve the response time and the latency issues operating multiple home appliances.

2.1.2 Limitations of Existing System

Technology Obsolescence:

Rapid advancements in smart home technology pose a significant challenge as they can render current devices obsolete relatively quickly. This rapid pace of innovation may result in compatibility issues between older devices and newer ones, leading to the need for frequent upgrades or replacements. Users may find themselves facing the dilemma of investing in new technology to maintain compatibility or sticking with outdated devices that lack the latest features and functionalities. Additionally, the continuous cycle of obsolescence contributes to electronic waste, environmental concerns, and financial strain on consumers who must continually invest in new devices to keep up with the evolving landscape of smart home technology.

Interoperability Challenges:

Integrating devices from different manufacturers into a cohesive smart home ecosystem can be challenging due to interoperability issues. Devices often operate on disparate communication protocols and standards, making it difficult for them to communicate and work together seamlessly. Compatibility issues may arise when attempting to connect devices from different brands, leading to fragmented user experiences and limited functionality. Users may encounter difficulties in setting up and managing their smart home systems, as well as in automating tasks and creating cohesive workflows across multiple devices. These interoperability challenges pose barriers to the widespread adoption of smart home technology and may frustrate users who expect seamless integration and interoperability among their devices.

Financial Constraints:

The high cost of replacing conventional appliances with smart alternatives presents a significant barrier to the adoption of smart home solutions. While the benefits of smart technology, such as energy efficiency, convenience, and automation, are appealing to consumers, the initial investment required to upgrade existing appliances can be prohibitive. Additionally, ongoing expenses, such as subscription fees for cloud services or maintenance costs, further contribute to the financial burden associated with smart home technology. As a result, many users may hesitate to invest in smart home solutions, opting to stick with traditional appliances to avoid the upfront costs and ongoing expenses associated with upgrading to smart alternatives.

User Experience Limitations:

Complicated setup processes, unreliable connectivity, and inconsistent performance can detract from the user experience of smart home technology, hindering its widespread adoption. Users may encounter challenges during the initial setup of their smart home systems, such as pairing devices, configuring settings, and troubleshooting connectivity issues. Unreliable connectivity, such as Wi-Fi outages or device disconnects, can disrupt the functionality of smart home systems and lead to frustration among users. Inconsistent performance, such as delays in response times or malfunctions, may erode user trust in the reliability and efficacy of smart home technology. These user experience limitations undermine the value proposition of smart home solutions and may deter users from fully embracing the technology.

Despite these limitations, there is potential for innovative solutions to bridge the gap between conventional appliances and modern smart home ecosystems. By addressing the challenges of technology obsolescence, interoperability, cost, and user experience, the Smartify project aims to empower users to transform their existing appliances into smart devices, enhancing convenience, efficiency, and connectivity in the home environment. Through seamless integration and intuitive user interfaces, Smartify seeks to overcome the limitations of existing systems and deliver a comprehensive smart home solution for users of all backgrounds and preferences.

2.2 Proposed System

Smartify presents a comprehensive solution tailored to elevate conventional household appliances to smart devices, catering to the escalating need for advanced home automation while circumventing the constraints of non-Wi-Fienabled devices. The proposed system harnesses the functionality of smart plugs and switches, enabling remote control of the power supply to a myriad of appliances, including air conditioners, televisions, and more. By integrating Wi-Fi connectivity and interfacing with a dedicated mobile application, Smartify empowers users to effortlessly monitor and manage their appliances from any location, at any time.

Additionally, the incorporation of an infrared (IR) remote control feature enhances user convenience, allowing direct control of appliances via remote commands. This comprehensive approach ensures seamless integration of smart functionality into existing household appliances, offering users unparalleled convenience, flexibility, and control over their home environment

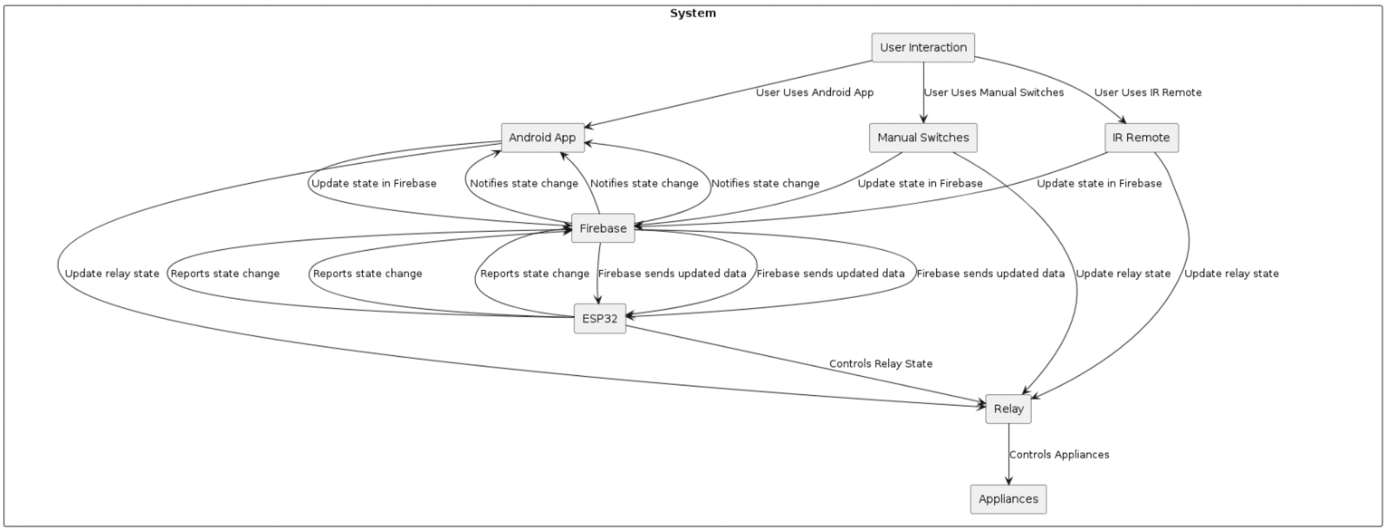


Figure 2.2.1 Proposed System

2.2.1 Advantages of Proposed System

Cost-Effectiveness:

Smartify offers a cost-effective solution for upgrading conventional appliances into smart devices without the need for expensive replacements. This approach significantly lowers the barrier to entry for users interested in adopting smart home technology, as they can enhance the functionality of their existing appliances at a fraction of the cost of purchasing new smart devices. By making smart home technology more accessible to a wider range of users, regardless of their budget constraints, Smartify empowers more households to enjoy the benefits of automation, energy efficiency, and convenience.

Seamless Integration:

Smartify leverages smart plugs and switches to enable the seamless integration of non-Wi-Fi enabled appliances into existing smart home ecosystems. By acting as intermediary devices, smart plugs and switches bridge the gap between legacy appliances and modern smart home systems, allowing users to control and automate their appliances remotely without the need for complex installations or replacements. This approach ensures that users can enjoy the benefits of smart technology without having to replace their entire appliance inventory, preserving their investment in existing appliances while still reaping the rewards of automation and connectivity.

Remote Control and Monitoring:

With Smartify's dedicated mobile application and Wi-Fi connectivity, users gain the ability to remotely control and monitor their appliances from anywhere, at any time. Whether they're at home, at work, or on vacation, users can easily access the Smartify app to adjust settings, turn appliances on or off, and monitor energy usage in real-time. This enhanced level of control and monitoring not only improves convenience and flexibility in managing household operations but also provides peace of mind by allowing users to stay connected to their homes even when they're away.

Intuitive User Interface:

The Smartify app offers an intuitive user interface designed to simplify the user experience and streamline household management tasks. With features such as scheduling, energy monitoring, and personalized settings, users can easily customize their smart home environment to suit their preferences and lifestyle. The intuitive layout and easy navigation of the app ensure that users can quickly access essential functions and make adjustments with minimal effort, enhancing convenience and efficiency in their daily routines.

smartify offers a comprehensive solution for upgrading conventional appliances into smart devices, providing users with cost-effective access to the benefits of smart home technology. By seamlessly integrating non-Wi-Fi enabled appliances into existing smart home ecosystems, offering remote control and monitoring capabilities, and providing an intuitive user interface, Smartify enhances convenience, efficiency, and accessibility in household operations. With Smartify, users can enjoy the advantages of automation, energy efficiency, and connectivity without the need for expensive replacements or complex installations, making smart home technology more accessible and user-friendly for everyone.

2.3 Software & Hardware Requirements

Every computer needs software to function, a computer must have specific hardware or additional software resources installed. These prerequisites are referred to as (computer) system requirements, and they are frequently utilised as recommendations rather than strict guidelines. The majority of software specifies minimum and recommended system requirements. System requirements typically rise with time due to the growing need for more processing power and resources in newer software versions. Industry observers say that rather than technological breakthroughs, this trend is a more significant factor in the improvements of current computer systems

2.3.1 Software Requirements

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development.

Arduino IDE:

Arduino IDE (Integrated Development Environment) is required for writing, compiling, and uploading code to microcontroller boards, including ESP8266 and ESP32. It provides a user-friendly interface for programming and managing projects, making it a central tool for developing embedded systems applications**.**

CPX2101 and CPX2102 Drivers:

The CPX2101 and CPX2102 drivers are necessary for enabling communication between the CPX2101 or CPX2102 USB-to-UART bridge controllers and the computer's operating system. These drivers facilitate data transfer and device control over USB, allowing for programming and debugging of microcontroller boards.

ESP8266 Board Package:

The ESP8266 board package adds support for ESP8266-based microcontroller boards to the Arduino IDE. It includes tools, libraries, and board definitions necessary for compiling and uploading code to ESP8266 boards, enabling development of Wi-Fi-connected projects.

ESP32 Board Package:s

The ESP32 board package provides support for ESP32-based microcontroller boards in the Arduino IDE. It offers tools, libraries, and board definitions required for programming ESP32 boards, allowing for development of projects with Wi-Fi, Bluetooth, and other advanced features.

ESP32 Library:

The ESP32 library contains Arduino-compatible functions and classes specifically designed for the ESP32 microcontroller. It provides convenient access to ESP32's features, such as Wi-Fi, Bluetooth, GPIO control, and more, simplifying the development process for ESP32-based projects.

IRModule Library:

The IRModule library is a collection of Arduino-compatible functions and classes for infrared (IR) communication. It enables ESP32 and ESP8266 boards to send and receive IR signals, making them suitable for applications such as remote control systems and home automation.

Firebase Client Library for ESP32 and ESP8266:

The Firebase Client library enables ESP32 and ESP8266 boards to communicate with Firebase, a cloud-based platform for real-time database and application development. With this library, developers can easily integrate ESP32 and ESP8266-based projects with Firebase, enabling features such as data logging, remote monitoring, and control from anywhere with an internet connection.

2.3.2 Hardware Requirements

ESP32 (38-pin variant):

The ESP32 microcontroller board serves as the main control unit for the project, providing processing power and connectivity features such as Wi-Fi and Bluetooth**.**

Wires:

Various wires are needed for connecting components together, including jumper wires (male-to-male, female-to-male, and female-to-female) and general connecting wires.

Three-Pin Plug Sockets:s

Three-pin plug sockets serve as the interface between the relay module controlled by the ESP32 and the electrical appliances or devices being controlled. These sockets allow for the controlled switching of power to the devices using the relay module controlled by the ESP32.

Four Switches:

Switches are used as input devices to control the operation of the project. They enable users to manually trigger the switching of power to the devices connected to the plug sockets.

Four 3-Pin Sockets:

These sockets provide the connection points for the switches and other components in the circuit. They allow for the organized and secure connection of wires and components within the project.

Connecting Wires:

Additional connecting wires are required for making connections between components on the breadboard or circuit board. They facilitate the electrical connections between the ESP32, relay module, switches, plug sockets, and other components.

4-Channel Relay Module:

The 4-channel relay module is used to control the switching of power to the electrical appliances or devices connected to the plug sockets. It receives control signals from the ESP32 and activates or deactivates the relays accordingly to switch the power on or off.

220V to 5V Adapter:

This adapter converts the high-voltage AC power from the mains (220V) to low-voltage DC power (5V) suitable for powering the relay module and other low-voltage components in the circuit.

IR Sender and Receiver:

The IR sender and receiver components enable communication via infrared signals, allowing for remote control functionality in the project. They allow users to control the project remotely using an infrared remote control device.

2.4 Feasibility Study

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

- Technical Feasibility
- Robustness & Reliability

- Economical Feasibility

2.4.1 Technical Feasibility

Smartify's technical feasibility is rooted in its innovative approach to retrofitting non-Wi-Fi enabled appliances with smart functionality, seamlessly integrating smart plugs, switches, and IR remote control capabilities. By leveraging these technologies, Smartify empowers users to control and monitor their appliances remotely, enhancing convenience and accessibility. The integration of Wi-Fi connectivity and a dedicated mobile application serves as the backbone of Smartify's functionality, allowing users to effortlessly manage their appliances from anywhere with internet access. Through the mobile app, users can schedule operations, monitor energy usage, and personalize settings according to their preferences, thereby optimizing their home environment.

Additionally, Smartify's straightforward installation procedures ensure accessibility for users of all technical backgrounds, eliminating barriers to adoption and facilitating a smooth transition to smart home technology. With IR remote control capabilities complementing its feature set, Smartify expands its compatibility with a wider range of appliances and offers users an alternative control method for added flexibility. Overall, Smartify's technical feasibility lies in its comprehensive approach to retrofitting, integrating, and optimizing smart home functionality, making it a user-friendly and accessible solution for modern households.

2.4.2 Robustness & Reliability

Smartify's robustness and reliability are underscored by its comprehensive security measures, safeguarding user data and thwarting unauthorized access. The system prioritizes user privacy, implementing encryption protocols and authentication mechanisms to ensure the confidentiality of sensitive information. Moreover, Smartify's utilization of established technologies like Wi-Fi connectivity, smart plugs, and IR remote control enhances its stability and reliability. Wi-Fi connectivity enables seamless communication between the Smartify system and user devices, facilitating consistent performance and dependable operation.

Additionally, the integration of smart plugs ensures compatibility with a diverse array of appliances, further bolstering reliability. The incorporation of IR remote control capabilities expands Smartify's versatility, allowing users to remotely control compatible devices using traditional infrared remotes. This multifaceted approach to technology integration not only enhances Smartify's functionality but also solidifies its status as a dependable solution for modernizing homes. Users can trust Smartify to deliver consistent and reliable performance, providing peace of mind and enhancing their overall smart home experience.

2.4.3 Economic Feasibility

Smartify's economic feasibility is evident in its cost-effective approach to upgrading conventional appliances into smart devices, complemented by its compatibility with ubiquitous IR remotes found in most households. By offering a retrofitting solution that eliminates the need for expensive replacements, Smartify makes smart home technology more accessible to a wider range of users. Furthermore, the potential energy savings and efficiency enhancements offered by Smartify contribute to its economic feasibility by providing long-term value to users Smartify's innovative integration of IR remote control capabilities further enhances its economic viability. Leveraging the IR remotes commonly available in households, Smartify enables users to seamlessly control their smart devices using familiar and readily accessible technology. This integration not only eliminates the need for additional investment in specialized control devices but also maximizes convenience for users, streamlining the transition to smart home automation.

Moreover, by leveraging existing infrastructure and resources, Smartify minimizes implementation costs while maximizing the utility of everyday household items. This cost-effective approach ensures that users can enjoy the benefits of smart home technology without incurring prohibitive expenses, making Smartify a practical and economically feasible solution for modernizing homes.

In summary, Smartify's economic feasibility is underpinned by its cost-effective retrofitting solution, energy-saving features, and seamless integration with ubiquitous IR remotes. By combining affordability with accessibility and long-term value, Smartify offers users a compelling proposition for embracing smart home automation without breaking the bank.

CHAPTER – 3
ARCHITECTURAL DESIGN

3.1 Module Design

Module design refers to the process of organizing software components into distinct modules or units based on their functionality, responsibilities, and dependencies. It aims to promote modularity, maintainability, and scalability in software development.

3.1.1 Number of Modules as Per Analysis

Connection Establishment Module:

Responsible for establishing a connection to the local Wi-Fi network.

Enables the device to communicate with the Firebase database over the internet.

Ensures that the device has internet access for data exchange.

Authentication and Streaming Module:

Manages communication with the Firebase Realtime Database (RTDB).

Handles authentication with the Firebase server using API keys and authentication tokens.

Facilitates reading, writing, and updating data in the Firebase RTDB.

Utilizes stream callbacks to receive real-time updates from the database.

Authentication and Authorization Token Generation and Management Module:

Assists in the generation and management of authentication tokens required for Firebase authentication.

Ensures secure authentication and authorization of the device with the Firebase server.

RTDB Helper Module:

Provides helper functions for handling Realtime Database (RTDB) operations effectively.

Helps in parsing and processing RTDB payloads received from the Firebase server.

Manages stream callbacks to handle real-time data updates from the database.

GPIO Module:

Controls the General Purpose Input/Output (GPIO) pins of the microcontroller.

Manages the interface with relay and switch components connected to the GPIO pins.

Allows the microcontroller to toggle the state of relays based on commands received from the Firebase RTDB.

IR remote Module:
This module is responsible for processing IR remote control signals received by the system and translating them into actions to control various appliances or devices connected to the system.

User Interface Module (Implicit):

Although not explicitly defined in the provided code, the user interface could be implemented through a mobile or web application.

Users interact with the home automation system through the user interface, sending commands to the Firebase RTDB.

The Firebase Module receives these commands and triggers GPIO actions accordingly, controlling the home appliances.

3.1.2 Method & Algorithm Design

The iterative process is a widely adopted approach utilized by designers, developers, educators, and professionals to enhance the quality and functionality of a design or product over time. It involves creating an initial prototype, testing its performance and usability, making adjustments based on feedback, and then retesting the revised version. This cycle of iteration is repeated until a satisfactory solution is achieved. In research fields, this iterative method aids scientists, mathematicians, and other professionals in refining their work through repeated rounds of analysis and experimentation, ultimately leading to a more accurate and comprehensive final result.

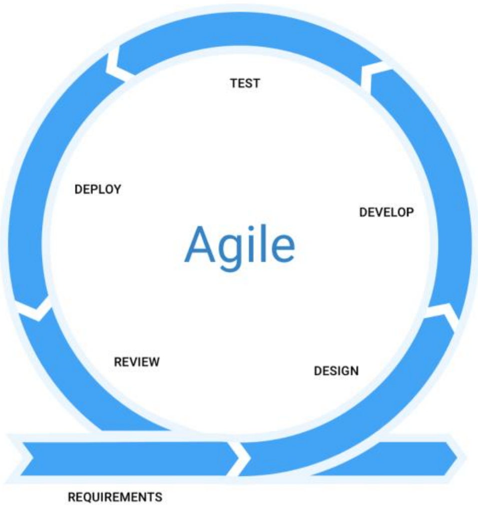


Figure 3.1.2 Methodology

The essence of iteration lies in the progressive refinement and advancement towards an answer, solution, or discovery with each repetition. Whether it's refining a mathematical function or making a scientific breakthrough, the iterative process involves continual 19 adjustments and enhancements that gradually bring the concept or solution closer to the desired outcome. Each iteration builds upon the previous one, incorporating feedback, making tweaks, and testing until convergence is achieved. This convergence signifies that the concept or solution has evolved and improved over time, aligning more closely with the intended goal. In essence, iteration is the journey of continuous improvement, where each cycle of iteration brings you one step closer to achieving excellence and realizing the full potential of your idea or product.

3.2 Project Architecture

Project architecture refers to the high-level structure and organization of a software project, including its components, modules, interactions, and technologies used. In the context of your "Smartify" project, the project architecture encompasses the following key aspects.

3.2.1 Complete Architecture

The complete architecture diagram provides a concise summary. It attempts to convey the developers intent and offers a qualitative perspective of the solution.

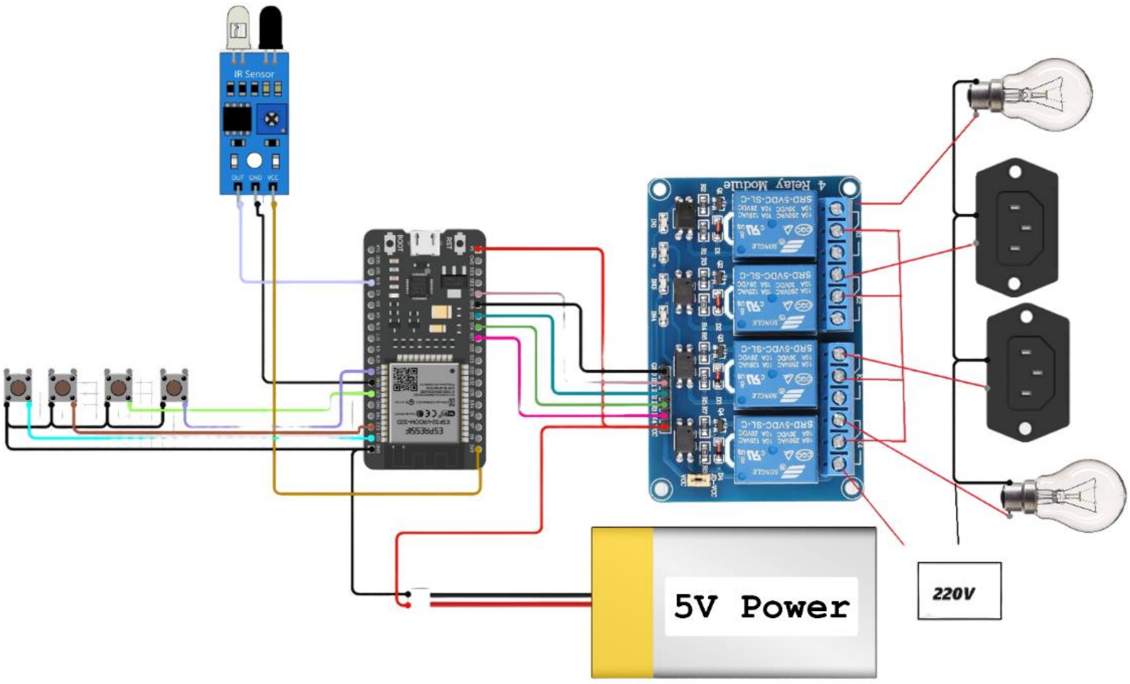


Figure 3.2.1 Complete Architecture

This circuit is designed to interface an ESP32 microcontroller with a 4-Channel Relay module, multiple pushbuttons, an IR sensor, and bulbs. The ESP32 controls the relay channels through its GPIO pins, which in turn can control AC or DC loads connected to the relays. Pushbuttons are used to provide input to the ESP32, and the IR sensor can detect infrared signals. The circuit is powered by a LiPoly battery, and the ESP32 regulates the voltage for the IR sensor. The bulbs are connected to AC power sockets, which can be controlled by the relays.

Component List

Microcontroller

ESP32 38 PIN:

A microcontroller with Wi-Fi and Bluetooth capabilities, featuring a wide range of GPIO pins for interfacing with various components.

Sensors

IR Sensor:
An infrared sensor used for detecting IR signals, typically from a remote control or similar device.

Actuators

4-Channel Relay 5V:
A relay module with four channels that can be controlled independently to switch external circuits on and off.
Bulb:
A generic light bulb that can be controlled via the relay module.

Input Devices
Pushbutton:
A momentary switch that provides a digital input to the microcontroller when pressed.

Power Supply
5V Adapter:
An adapter that steps down AC voltage to a regulated 5V DC output. (Note: The specific connections for this component are not provided in the input data.)

Miscellaneous

AC Power Socket:

A socket that allows AC-powered devices to be plugged in and controlled by the relay module.

Wiring Details

ESP32 38 PIN

GND:

Connected to the ground (GND) of the 4-Channel Relay module, IR sensor, and the negative terminal of the LiPoly Battery.

GPIO Pins (G13, G12, G14, G27):

Control the IN1, IN2, IN3, and IN4 pins of the 4-Channel Relay module respectively.

5V:

Connected to the positive terminal of the LiPoly Battery and VCC of the 4-Channel Relay module.

3V3:

Supplies power to the VCC pin of the IR sensor.

GPIO Pins (G19, G21, G22, G23):

Connected to the output pins of the pushbuttons.

G15:

Reads the output from the IR sensor.

4-Channel Relay 5V

GND:

Connected to the ground (GND) of the ESP32.

IN1, IN2, IN3, IN4:

Controlled by the GPIO pins G13, G12, G14, G27 of the ESP32 respectively.

VCC:

Powered by the 5V from the LiPoly Battery.

Pushbuttons

Pin 2 (in):

All pushbuttons share a common connection to the ground (GND) of the LiPoly Battery and ESP32.

Pin 4 (out):

Each pushbutton's output pin is connected to a separate GPIO pin on the ESP32 (G19, G21, G22, G23).

IR Sensor

out:

Connected to the GPIO pin G15 of the ESP32.

gnd:

Connected to the ground (GND) of the ESP32.

vcc:

Powered by the 3V3 pin of the ESP32.

Bulbs

_VE:

The negative terminals of the bulbs are connected to the AC Power Sockets.

(Note: The positive terminals of the bulbs and the exact wiring of the AC Power Sockets are not provided in the input data.)

3.2.2 Data Flow & Process Flow Diagram

A Data Flow Diagram (DFD) is a visual representation of the flow of data within a system or process. It illustrates how data moves from one component to another, showing the inputs, outputs, processes, and data storage involved. DFDs are commonly used in system analysis and design to model the data flow and interactions within a software application or business process.

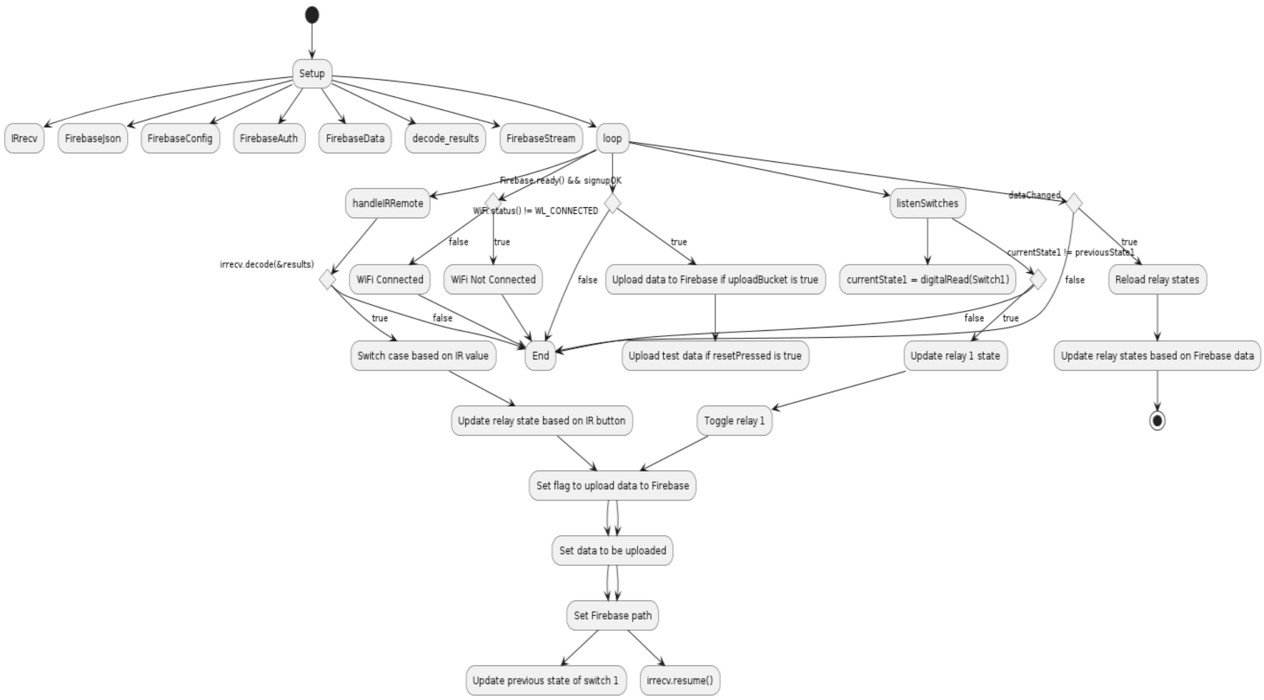


Figure 3.2.2 Process flow Diagram

step-by-step process flow process:

Setup Initialization:

Initialize essential components such as IR receiver, Firebase connection, and network configuration.

In the provided code, this includes initializing Firebase, IR receiver, WiFi, and GPIO pins for relays and switches.

WiFi Connection Check:

Check if the WiFi is connected.

If not connected:

Indicate WiFi not connected.

Exit the loop.

In the code, this is done by checking the status of the WiFi connection using WiFi.status() and waiting until it's connected.

Switch State Monitoring:

Continuously monitor the state of manual switches.

If a switch state changes:

Update the corresponding relay state (e.g., Relay 1, Relay 2).

Flag the data for upload to Firebase.

Prepare the data and path for Firebase.

Update the previous state of the switch.

This is implemented in the listenSwitches() function, where the state of each switch is monitored and relay states are updated accordingly.

IR Remote Handling:

Continuously check for IR remote signals.

If an IR signal is received:

Determine the button pressed on the remote.

Update the relay state based on the button pressed.

Flag the data for upload to Firebase.

Prepare the data and path for Firebase.

This functionality is implemented in the handleIRRemote() function, where the IR signal is decoded and relay states are updated accordingly.

Firebase Data Upload:

Check if the system is ready and the user has successfully signed up with Firebase.

If conditions are met and data upload is flagged:

Upload the data to Firebase.

Clear the flag after upload.

If the reset button is pressed for testing:

Upload test data to Firebase.

This is handled within the loop() function, where data is uploaded to Firebase if the conditions are met.

Firebase Data Change Detection:

Continuously monitor for changes in data from Firebase.

If data changes:

Update the relay states based on the new data from Firebase.

This functionality is handled in the streamCallback() function, where changes in Firebase data are detected and processed.

Relay State Update:

Update the relay states based on the data received from Firebase.

Update Relay 1, Relay 2, Relay 3, and Relay 4 states accordingly.

This is part of the reloadRelayStates() function, where relay states are updated based on the Firebase data.

1. **IR Remote Signal Decoding:**

Decode the IR signal received from the remote.

Map the signal to the corresponding action (e.g., turn on/off a specific relay).

Implemented in the handleIRRemote() function, where the IR signal is decoded and relay states are updated accordingly.

Firebase Connection Establishment:

Establish a connection with Firebase for real-time data synchronization.

Ensure proper authentication and network connectivity.

This is handled during setup with the initialization of Firebase and network configuration.

Data Preparation for Firebase:

Prepare the data to be uploaded to Firebase.

Format the data according to the required structure (e.g., JSON format).

Data preparation is done within various functions such as handleIRRemote() and listenSwitches() where data is prepared before uploading to Firebase.

These steps encompass the core functionalities of the provided code and how they relate to the smart home automation system's operation and data flow.

3.2.3 Class Diagram

A class diagram is a type of static structure diagram in UML (Unified Modeling Language) that represents the structure of a system by showing the classes, attributes, 23 operations or methods, and relationships between classes. It provides a conceptual view of the system's design and the interactions between its components. Here's a breakdown of the key elements in a class diagram

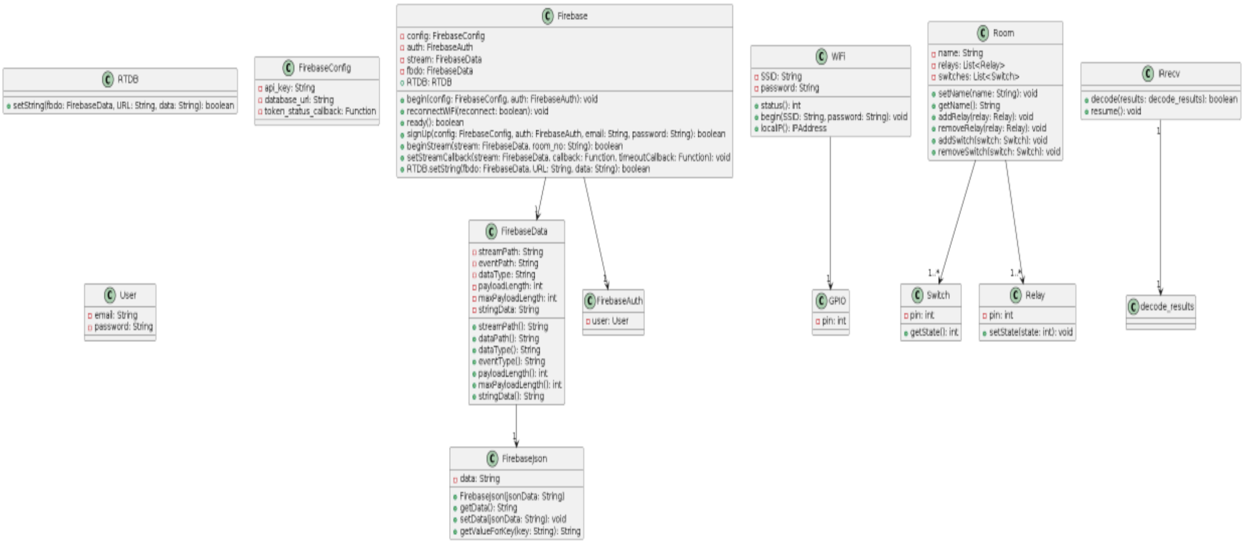


Figure 3.2.3 Class Diagram

Description:

This class diagram portrays a system designed for automating tasks within a home environment. Let's break it down:

Firebase: This is a platform utilized for real-time data management and user authentication. It allows the system to store and retrieve data swiftly, ensuring seamless operation.

FirebaseConfig: Stores essential configuration details required for connecting to Firebase, like API keys and database URLs.

FirebaseAuth: Manages the authentication process for users, ensuring secure access to the system using email and password authentication.

FirebaseData: Handles the communication and manipulation of data within Firebase. It's responsible for streaming data and managing its structure efficiently.

FirebaseJson: Deals specifically with data formatted in JSON (JavaScript Object Notation), which is a common data format used for exchanging information between a server and a web application.

WiFi: Manages the Wi-Fi connection for the system. It handles the network connectivity aspect, ensuring the system can communicate over Wi-Fi.

GPIO: Represents General Purpose Input/Output pins, which are used for interfacing with hardware components like sensors, switches, and actuators.

RTDB: Stands for Real-Time Database, and it encompasses operations related to handling real-time data within the system, such as setting string data.

Room: Represents physical rooms within a home environment. Each room can contain devices such as relays and switches, which are used for controlling various appliances or devices.

Relay: Controls the state of a relay, which is a type of switch used to control the flow of electricity to a device. It can turn devices on or off based on commands received from the system.

Switch: Represents a switch, which is a physical or virtual control used to toggle the state of a device. It allows users to manually or automatically control devices within the system.

IRrecv: Handles the reception and decoding of signals from infrared (IR) remotes. This component enables users to control devices using IR remotes, adding convenience to the system.

In essence, this diagram illustrates how various components, services, and devices work together to create a comprehensive home automation system, enabling users to control and manage their environment efficiently.

3.2.4 Usecase Diagram

A use case diagram is a type of behavioral diagram in Unified Modeling Language (UML) that represents the functional requirements and interactions of a system from the users' perspective. It focuses on capturing the various use cases or functionalities of a system and how actors (users or external systems) interact with those use cases. Use case diagrams are widely used in software development to understand, communicate, and document the system's behavior and requirements

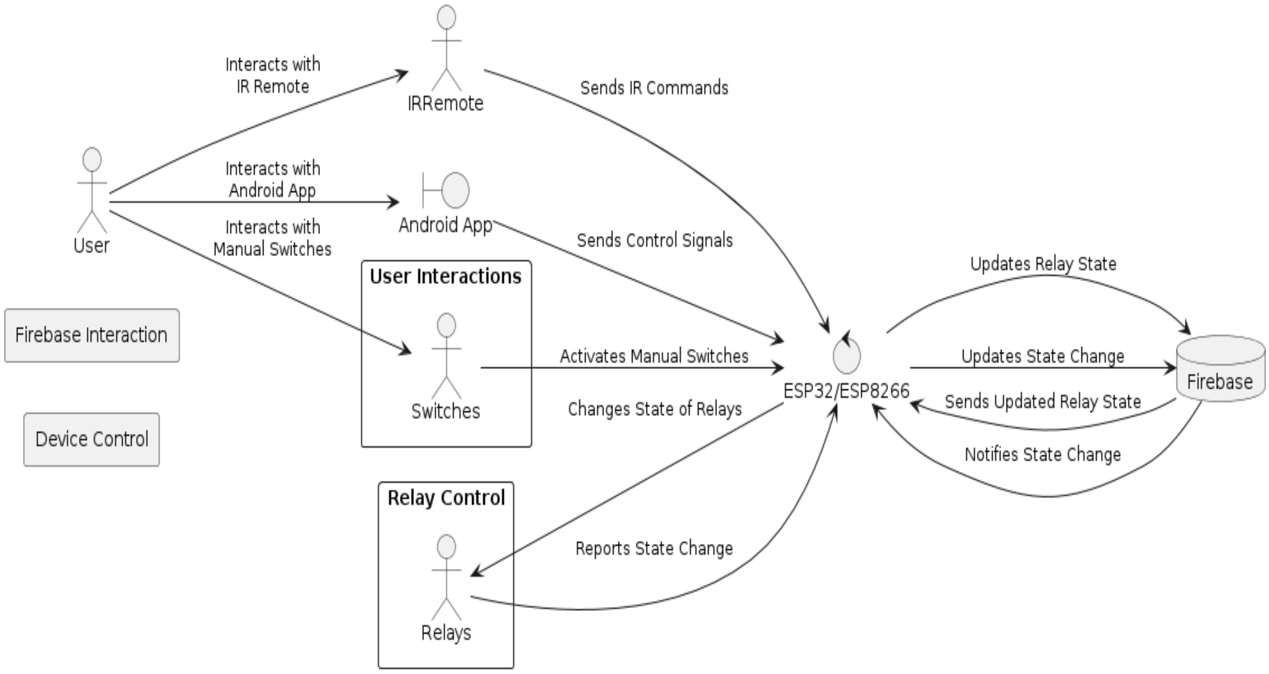


Figure 3.2.4 Use Case Diagram

Description:

This use case diagram outlines the interactions and control flow within a home automation system:

User Interactions: Users interact with the system via an Android app and physical IR remotes or manual switches.

Device Control: Commands from users, either through the app, IR remotes, or manual switches, are sent to the ESP32/ESP8266 microcontroller for device control.

Firebase Interaction: The ESP32/ESP8266 interacts with Firebase to update relay states, retrieve updated relay states, and notify Firebase of any state changes.

Relay Control: The ESP32/ESP8266 changes the state of relays based on received commands and reports these state changes back to the system.

This diagram illustrates how users interact with the system, how control commands are transmitted to devices, how device states are managed via Firebase, and how relay control is executed within the system.

3.2.5 Sequence Diagram

A sequence diagram is a type of interaction diagram in Unified Modeling Language (UML) that illustrates the interactions and messages exchanged between objects or components within a system over time. It shows the sequence of messages and method calls between objects, helping visualize the flow of control and communication during a specific scenario or use case.

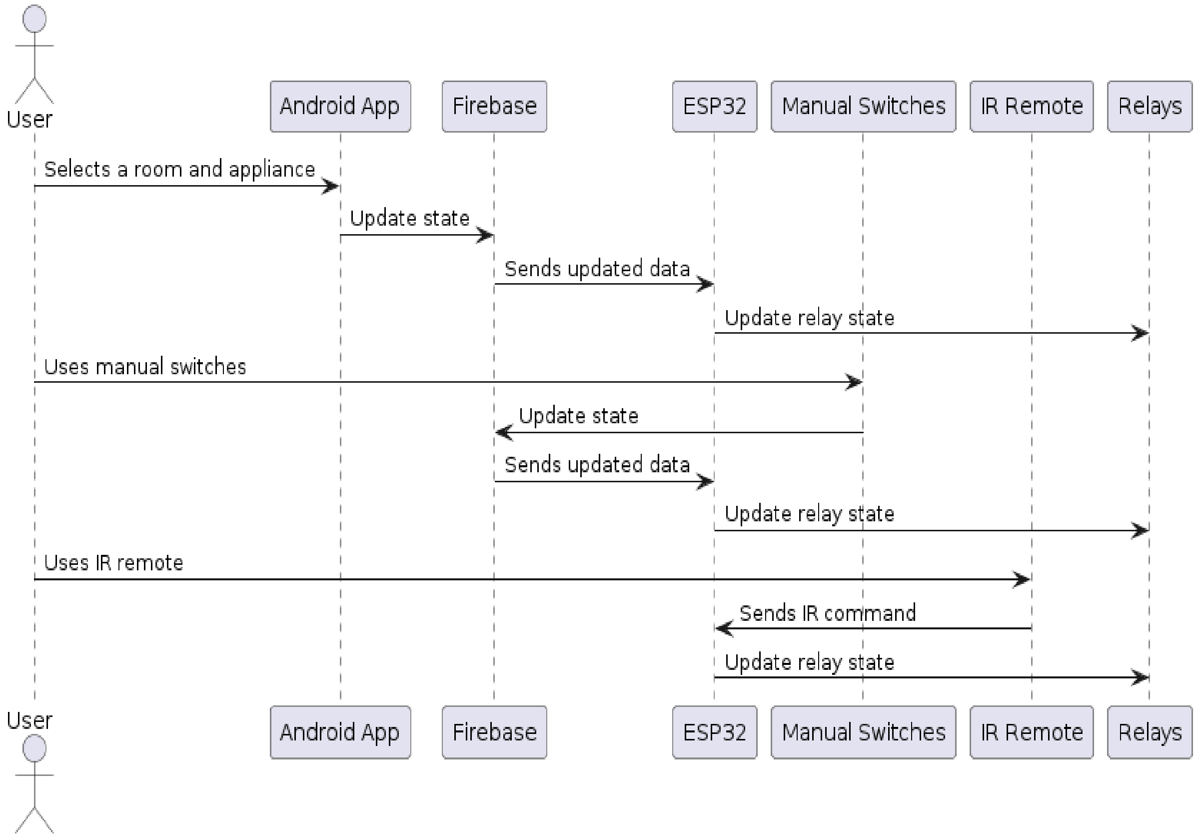


Figure 3.2.5 Sequence Diagram

This sequence diagram illustrates the interactions involved in controlling home appliances within a home automation system:

User Interaction via App:

The user selects a room and appliance using the Android app.

The app updates the state of the selected appliance in the Firebase database.

Firebase sends the updated data to the ESP32 microcontroller.

The ESP32 updates the state of the corresponding relays connected to the appliance.

User Interaction via Manual Switches:

The user operates manual switches to control appliances.

The switches update the state of the corresponding appliances in the Firebase database.

Firebase sends the updated data to the ESP32 microcontroller.

The ESP32 updates the state of the corresponding relays connected to the appliances.

User Interaction via IR Remote:

The user uses an IR remote to send commands to control appliances.

The IR remote sends IR commands to the ESP32 microcontroller.

The ESP32 interprets the IR commands and updates the state of the corresponding relays connected to the appliances.

This sequence diagram illustrates how user interactions via the Android app, manual switches, and IR remote are processed within the home automation system to control appliances effectively.

3.2.6 Activity Diagram

An activity diagram is a type of behavioral diagram in Unified Modeling Language (UML) that illustrates the dynamic aspects of a system by modeling the flow of activities or actions performed in a particular process, use case, or workflow. It focuses on depicting the sequence of actions, decisions, and transitions within a system or business process, helping to visualize the behavior and logic of the system from a procedural perspective.

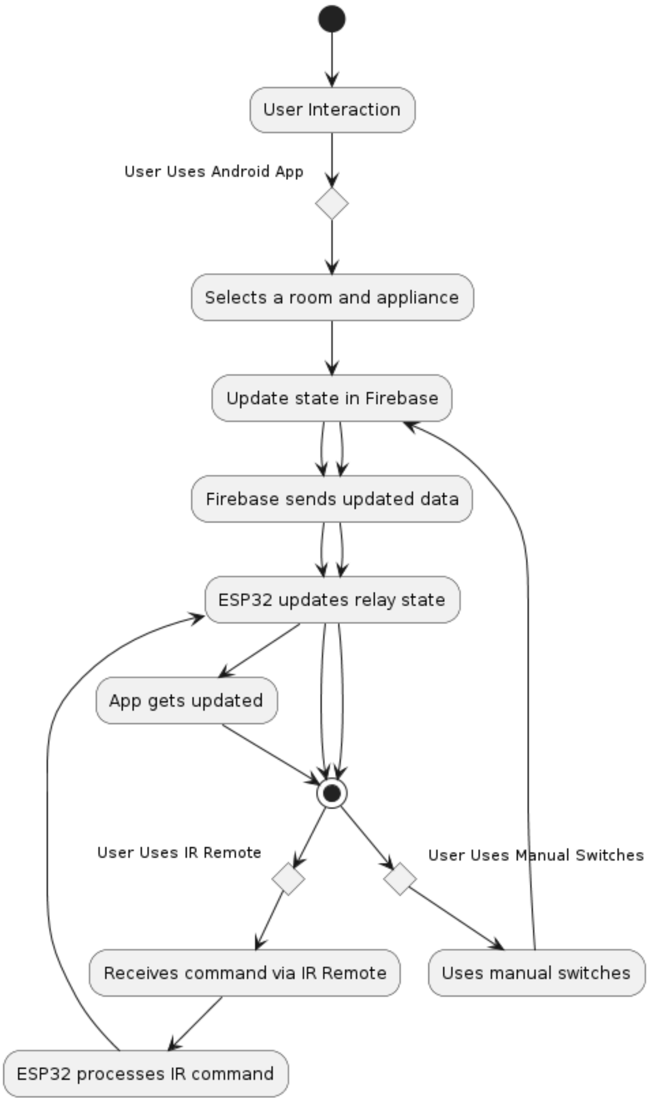


Figure 3.2.6 Activity Diagram

This activity diagram outlines the user interactions and system responses within a home automation setup:

User Interaction via Android App:

User selects a room and appliance using the Android app.

The selected state is updated in the Firebase database.

Firebase sends the updated data to the ESP32 microcontroller.

ESP32 updates the relay state accordingly.

User Interaction via Manual Switches:

User operates manual switches to control appliances.

The state changes are reflected in the Firebase database.

Firebase sends the updated data to the ESP32 microcontroller.

ESP32 updates the relay state and notifies the app of the changes.

User Interaction via IR Remote:

User sends commands via an IR remote.

The ESP32 receives and processes the IR command.

ESP32 updates the relay state based on the received command.

This activity diagram demonstrates the flow of actions triggered by user interactions through different interfaces, leading to the adjustment of relay states within the home automation system.

CHAPTER - 4
IMPLEMENTATION & TESTING

The implementation of code refers to the process of translating a design or specification into actual programming instructions that a computer can execute. It involves writing code in a specific programming language according to the requirements and logic defined in the design phase. Here are the key steps involved in the implementation of code

4.1 Coding Blocks

4.1.1 Code Block 1:

4.1.1.1 Setup Initialization Block:

```
void setup() {
   // Initialize Serial communication
   Serial.begin(115200);

   // Initialize GPIO pins for relays, switches, and WiFi LED
   pinMode(Relay1, OUTPUT);
   pinMode(Relay2, OUTPUT);
   pinMode(Relay3, OUTPUT);
   pinMode(Relay4, OUTPUT);
   pinMode(WIFI_LED, OUTPUT);
   pinMode(Switch1, INPUT_PULLUP);
   pinMode(Switch2, INPUT_PULLUP);
   pinMode(Switch3, INPUT_PULLUP);
   pinMode(Switch4, INPUT_PULLUP);

   // Connect to WiFi
   connectToWiFi();

   // Initialize Firebase configuration
   initFirebase();

   // Begin Firebase Realtime Database stream
```

```
  beginFirebaseStream();

  // Enable IR receiver
  irrecv.enableIRIn();
}
```

4.1.1.2. WiFi Connection Check Block:

```
void connectToWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
}
```

4.1.1.3. Firebase Initialization Block:

```
void initFirebase() {
  // Configure Firebase API key and database URL
  config.api_key = API_KEY;
  config.database_url = DATABASE_URL;

  // Initialize Firebase client
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}
```

4.1.1.4. Firebase Stream Handling Block:

```
void beginFirebaseStream() {
  if (!Firebase.RTDB.beginStream(&stream, room_no))
    Serial.printf("Stream begin error: %s\n\n", stream.errorReason().c_str());
```

```
  Firebase.RTDB.setStreamCallback(&stream, streamCallback, streamTimeoutCallback);
}
```

4.1.1.5 .IR Remote Handling Block:

```
void handleIRRemote() {
    if (irrecv.decode(&results)) {
        switch (results.value) {
            // Handle different IR remote buttons
            // Update relay states accordingly
            // Set flags for data upload to Firebase
            // Case statements for each button on the remote
        }
        irrecv.resume(); // Receive the next value
    }
}
```

4.1.2 Code Block 2:

4.1.2.1. *Switch State Monitoring Block*:

```
void listenSwitches() {
    // Monitor state changes of switches
    // Update relay states accordingly
    // Set flags for data upload to Firebase
    // Update previous state of switches
}
```

4.1.2.2 *Firebase Data Upload Block:*

```
void uploadToFirebase() {
    // Upload data to Firebase if conditions are met
    // Clear flags after upload
}
```

4.1.2.3.*Firebase Data Change Detection Block:*

```
void streamCallback(FirebaseStream data) {
    // Handle stream callback
```

```
// Detect changes in Firebase data
    // Process incoming data and update relay states
}
```

4.1.2.4. *Relay State Update Block:*

```
void updateRelayStates() {
    // Update relay states based on data from Firebase
}
```

4.1.2.5 *.IR Remote Signal Decoding Block:*

```
void decodeIRSignal() {
    // Decode IR signal received from the remote
    // Map signals to corresponding relay actions
}
```

These coding blocks encapsulate the functionality and structure of the provided code, making it easier to understand and manage. Each block focuses on a specific aspect of the code's operation, enhancing readability and maintainability.

4.2 Sample Code

```
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif

#include <IRremoteESP8266.h>
#include <IRrecv.h>
#include <IRutils.h>

#define IR_RECEIVER_PIN  15 // IR receiver pin

IRrecv irrecv(IR_RECEIVER_PIN);
decode_results results;

#include <Firebase_ESP_Client.h>
//--------------------------------------------------------------------
//Provide the token generation process info.
#include <addons/TokenHelper.h>

//Provide the RTDB payload printing
//info and other helper functions.
#include <addons/RTDBHelper.h>
//--------------------------------------------------------------------
/* 1. Define the WiFi credentials */
#define WIFI_SSID "SUNNY_SHIVA"
#define WIFI_PASSWORD "ACHIREDDY"
//--------------------------------------------------------------------
/* 2. Define the API Key */
#define API_KEY "AIzaSyAlkinyJkAtATI9mKDOdqLsbGWXBhAW37g"
//--------------------------------------------------------------------
/* 3. Define the RTDB URL */
```

```
#define DATABASE_URL  "smartify-home-automation-default-rtdb.firebaseio.com"
//-----------------------------------------------------------------------
String room_no = "room1";
//-----------------------------------------------------------------------
// define the GPIO connected with Relays and switches
#define Relay1  13
#define Relay2  12
#define Relay3  14
#define Relay4  27

#define Switch1 23
#define Switch2 22
#define Switch3 21
#define Switch4 19

#define WIFI_LED 2

int stateRelay1=0, stateRelay2=0, stateRelay3=0, stateRelay4=0;
//-----------------------------------------------------------------------

/* Uncomment only if, you have selected the email authentication from
firebase authentication settings */
//#define USER_EMAIL "ENTER_USER_EMAIL"
//#define DATABASE_URL  "USER_PASSWORD"
//-----------------------------------------------------------------------

/*********************************************************************
 0. complete_path = /room1/L1 or /room1/L2 etc.
    this is the complete path to firebase database
 1. stream_path = /room1
    this is the top nodes of firebase database
*********************************************************************/
String stream_path = "";
```

```
/**********************************************************
 2. event_path = /L1 or /L2 or /L3 or /L4
    this is the data node in firbase database
**********************************************************/
String event_path = "";
/**********************************************************
 3. stream_data - use to store the current command to
    turn ON or OFF the relay
**********************************************************/
String stream_data = "";
/**********************************************************
 5. jsonData - use to store "all the relay states" from
    firebase database jsonData object used only once when
    you reset the nodemcu or esp32 check the following:
    else if(event_path == "/") in the loop() function
**********************************************************/
FirebaseJson jsonData;

/**********************************************************
 it becomes TRUE when data is changed in Firebase
 used in streamCallback function
******************** *************************************/
volatile bool dataChanged = false;

/**********************************************************
 this variable is based on the authentication method you
 have selected while making firebase database project.
 authentication method: anonymus user
**********************************************************/
bool signupOK = false;

/**********************************************************
 resetPressed variable is used only once when you
```

```
pressed the reset button. it is used to send test data to
 Firebase database. If we will not test send data then the
 project will not work. used in the loop function.
**************************************************************/
bool resetPressed = true;

/**************************************************************
 when there is some data to upload to theFirebase
 then the value of uploadBucket will TRUE. This
 variable is used in listenSwitches() function
**************************************************************/
bool uploadBucket = false;

//i.e bucketData = "1" or "0"
//i.e bucketPath = "L1" or "L2" etc.
String bucketData = "", bucketPath = "";

//--------------------------------------------------------------------
//Define Firebase Data object
FirebaseData stream;
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

/*****************************************************************************
*****************
 * void streamCallback(FirebaseStream data)
 * This function execute automatically
 * 1. when you press the reset button of the microcontroller.
 * 2. when any data is changed in the firebase basebase.
 * microcontroller.
```

```
 *****************************************************************************
****************/

 void handleIRRemote() {
  if (irrecv.decode(&results)) {
    switch (results.value) {
      case 0xC084:  // Button 1
        stateRelay1 = !stateRelay1;
        digitalWrite(Relay1, stateRelay1);
        uploadBucket = true;
        bucketData = String(stateRelay1);
        bucketPath = "L1";
        break;
      case 0xC044:  // Button 2
        stateRelay2 = !stateRelay2;
        digitalWrite(Relay2, stateRelay2);
        uploadBucket = true;
        bucketData = String(stateRelay2);
        bucketPath = "L2";
        break;
      case 0xC0C4:  // Button 3
        stateRelay3 = !stateRelay3;
        digitalWrite(Relay3, stateRelay3);
        uploadBucket = true;
        bucketData = String(stateRelay3);
        bucketPath = "L3";
        break;
      case 0xC024:  // Button 4
        stateRelay4 = !stateRelay4;
        digitalWrite(Relay4, stateRelay4);
        uploadBucket = true;
        bucketData = String(stateRelay4);
        bucketPath = "L4";
```

```
  break;
      case 0xC0E8:  // Power button
        stateRelay1 = !stateRelay1;
        stateRelay2 = !stateRelay2;
        stateRelay3 = !stateRelay3;
        stateRelay4 = !stateRelay4;
        digitalWrite(Relay1, stateRelay1);
        digitalWrite(Relay2, stateRelay2);
        digitalWrite(Relay3, stateRelay3);
        digitalWrite(Relay4, stateRelay4);
        uploadBucket = true;
        bucketData = String(stateRelay1) + String(stateRelay2) + String(stateRelay3) +
String(stateRelay4);
        bucketPath = "All";
        break;
      default:
        break;
    }
    irrecv.resume(); // Receive the next value
  }
}
void streamCallback(FirebaseStream data)
{
  stream_path = data.streamPath().c_str();
  event_path = data.dataPath().c_str();

  if(String(data.dataType().c_str()) == "json"){
    jsonData = data.to<FirebaseJson>();
  }else{
    //intData(), floatData()
    stream_data = data.stringData();
  }
  Serial.printf("stream path, %s\nevent path, %s\ndata type, %s\nevent type, %s\n\n",
```

```
    stream_path,
              event_path,
              data.dataType().c_str(),
              data.eventType().c_str());
  printResult(data); //see addons/RTDBHelper.h
  Serial.println();

  //This is the size of stream payload received (current and max value)
  //Max payload size is the payload size under the stream path since
  //the stream connected and read once and will not update until
  //stream reconnection takes place. This max value will be zero
  //as no payload received in case of ESP8266 which
  //BearSSL reserved Rx buffer size is less than the actual stream payload.
  Serial.printf("Received stream payload size: %d (Max. %d)\n\n",
              data.payloadLength(), data.maxPayloadLength());

  //Due to limited of stack memory, do not perform any task that
  //used large memory here especially starting connect to server.
  //Just set this flag and check it status later.
  dataChanged = true;
}

/*************************************************************************
****************
 * void streamTimeoutCallback(bool timeout)
 *************************************************************************
****************/
void streamTimeoutCallback(bool timeout)
{
  if (timeout)
    Serial.println("stream timed out, resuming...\n");

  if (!stream.httpConnected())
```

```
 Serial.printf("error code: %d, reason: %s\n\n", stream.httpCode(),
                  stream.errorReason().c_str());
}

/*************************************************************************
****************
 * void setup()
 *************************************************************************
****************/
void setup()
{
  Serial.begin(115200);
  //-------------------------------------------------------------------
  pinMode(Relay1, OUTPUT); digitalWrite(Relay1, LOW);
  pinMode(Relay2, OUTPUT); digitalWrite(Relay2, LOW);
  pinMode(Relay3, OUTPUT); digitalWrite(Relay3, LOW);
  pinMode(Relay4, OUTPUT); digitalWrite(Relay4, LOW);

  pinMode(WIFI_LED, OUTPUT);

  pinMode(Switch1, INPUT_PULLUP);
  pinMode(Switch2, INPUT_PULLUP);
  pinMode(Switch3, INPUT_PULLUP);
  pinMode(Switch4, INPUT_PULLUP);

  //-------------------------------------------------------------------
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
```

```
Serial.println();
 Serial.print("Connected with IP: ");
 Serial.println(WiFi.localIP());
 Serial.println();
 //---------------------------------------------------------------------
 Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);

 /* Assign the api key (required) */
 config.api_key = API_KEY;

 /* Assign the RTDB URL (required) */
 config.database_url = DATABASE_URL;

 /* Assign the callback function for the long running token generation task */
 config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h
 //---------------------------------------------------------------------
 /*Or Sign up */
 if (Firebase.signUp(&config, &auth, "", "")){
   Serial.println("Firebase signUp ok");
   signupOK = true;
 }
 else{
   Serial.printf("%s\n", config.signer.signupError.message.c_str());
 }
 //---------------------------------------------------------------------

 /*Or Assign the user sign in credentials */
 //auth.user.email = USER_EMAIL;
 //auth.user.password = USER_PASSWORD;

 //Or use legacy authenticate method
 //config.database_url = DATABASE_URL;
 //config.signer.tokens.legacy_token = "<database secret>";
```

```
  //---------------------------------------------------------------------
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
  //---------------------------------------------------------------------
  //Recommend for ESP8266 stream, adjust the buffer size to match your stream data size
  #if defined(ESP8266)
    stream.setBSSLBufferSize(2048 /* Rx in bytes, 512 - 16384 */, 512 /* Tx in bytes, 512 -
16384 */);
  #endif
  //---------------------------------------------------------------------
  if (!Firebase.RTDB.beginStream(&stream, room_no))
    Serial.printf("sream begin error, %s\n\n", stream.errorReason().c_str());
  //---------------------------------------------------------------------
  Firebase.RTDB.setStreamCallback(&stream, streamCallback, streamTimeoutCallback);
  irrecv.enableIRIn();
  //---------------------------------------------------------------------
}
```

4.3 Execution Flow

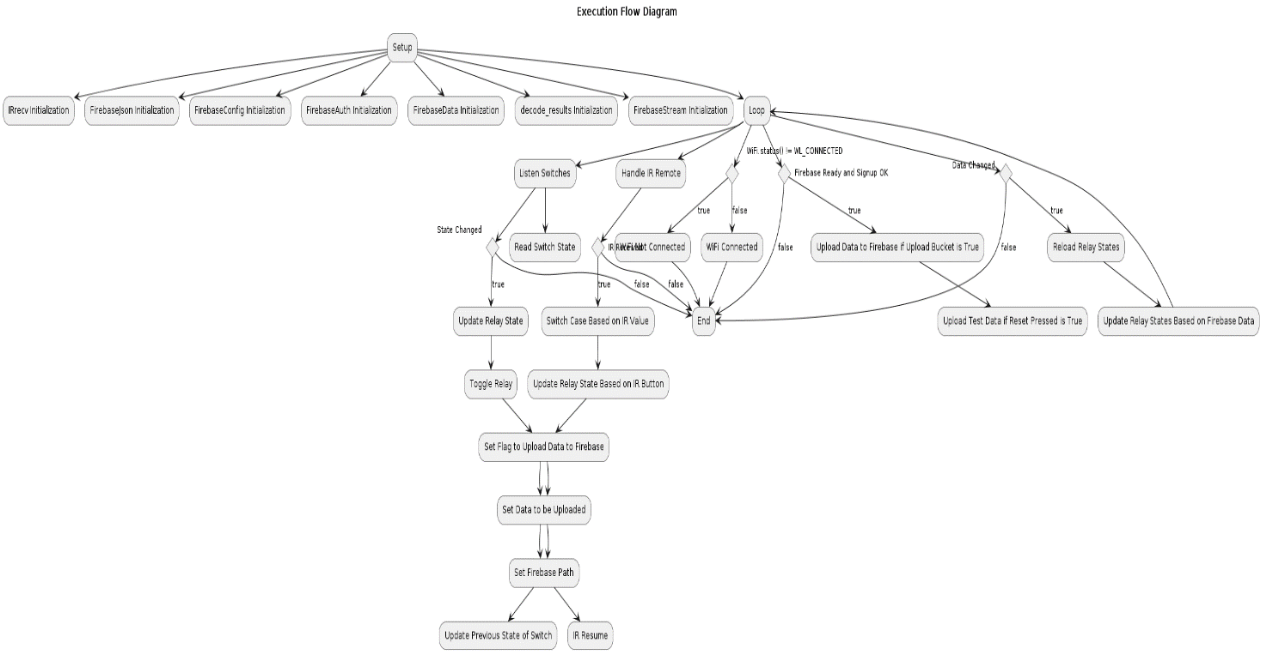


Figure 4.3.1 Execution flow

The execution flow outlined in the diagram follows a sequence of functions and decisions within a loop, representing the runtime behavior of the system. Here's a detailed explanation of the functions and their order of execution:

Setup Initialization:

The system initializes various components necessary for operation, including IRrecv, FirebaseJson, FirebaseAuth, FirebaseData, etc. This initialization phase ensures that all required resources are properly configured before entering the main loop**.**

Loop:

After setup, the system enters a continuous loop where it performs its main operations**.**

WiFi Connection Check:

Within the loop, the system checks if the WiFi connection is established. If the connection is not established (true condition), the system branches to handle the case where WiFi is not connected, and then ends the loop.

Listen Switches:

If the WiFi connection is established (false condition of WiFi connection check), the system proceeds to listen for any changes in switch states.

Read Switch State:

It reads the state of the switches connected to the system.

Switch State Change Handling:

If a state change is detected in the switches, the system updates the relay state accordingly and toggles the relay. It then sets a flag to indicate the need to upload data to Firebase, prepares the data to be uploaded, sets the Firebase path, and updates the previous state of the switch.

Handle IR Remote:

Next, the system checks if an IR signal is received from the IR remote control.

If an IR signal is received, it performs actions based on the IR value. Similar to switch state change handling, it updates the relay state, sets flags for Firebase data upload, prepares data, sets Firebase path, and resumes IR reception.

Firebase Data Management:

After handling user inputs and device interactions, the system checks if Firebase is ready and if the user authentication is successful.

If Firebase is ready and the user authentication is successful, the system proceeds to upload data to Firebase if the upload bucket is true. It also uploads test data if the reset button is pressed.

Check for Data Change:

Lastly, the system checks if there is any change in data.

If there is a change, it reloads relay states based on Firebase data.

Loop Continuation:

After completing all operations within the loop, the system continues to loop back to the beginning, repeating the process indefinitely.

4.4 Testing

4.4.1 Test Case 1: Turning on an appliance using a manual switch.

Test Case:

This test verifies that the manual switch effectively activates the appliance.

Expected Result:

Upon toggling the manual switch to the "on" position, the appliance should power on, and its corresponding status should be updated in both the Android app and IR remote interfaces.

4.4.2 Test Case 2: Turning off an appliance using the IR remote after it was turned on using a manual switch.

Test Case:

This test ensures that the IR remote can deactivate the appliance after it was initially activated using the manual switch.

Expected Result:

After turning on the appliance via the manual switch, using the IR remote to send a "power off" command should successfully switch off the appliance. The status update should reflect in both the Android app and manual switch.

4.4.3 Test Case 3: Turning off an appliance using the Android app after it was turned on using a manual switch.

Test Case:

This test validates that the Android app can control the appliance state after it was initially turned on using the manual switch.

Expected Result:

Following the manual switch activation of the appliance, utilizing the Android app to issue a "power off" command should effectively turn off the appliance. The status update should propagate to both the IR remote and manual switch interfaces.

4.4.4 Test Case 4: Ensuring status synchronization between all control methods.

Test Case:

This test guarantees that the status updates are accurately synchronized across all control interfaces.

Expected Result:

Upon turning on the appliance using the manual switch, the Android app and IR remote should promptly update to reflect the appliance's powered-on state. Similarly, any changes made via the Android app or IR remote should be immediately mirrored in the other interfaces.

4.4.5 Test Case 5: Turning on and off an appliance using different control methods interchangeably.

Test Case:

This test examines the seamless interchangeability of control methods in activating and deactivating the appliance.

Expected Result:

Performing multiple cycles of turning the appliance on and off using various methods (manual switch, IR remote, Android app) should consistently update the appliance status across all interfaces, ensuring reliable and synchronized control.

CHAPTER – 5
RESULTS

5.1 Resulting Screens

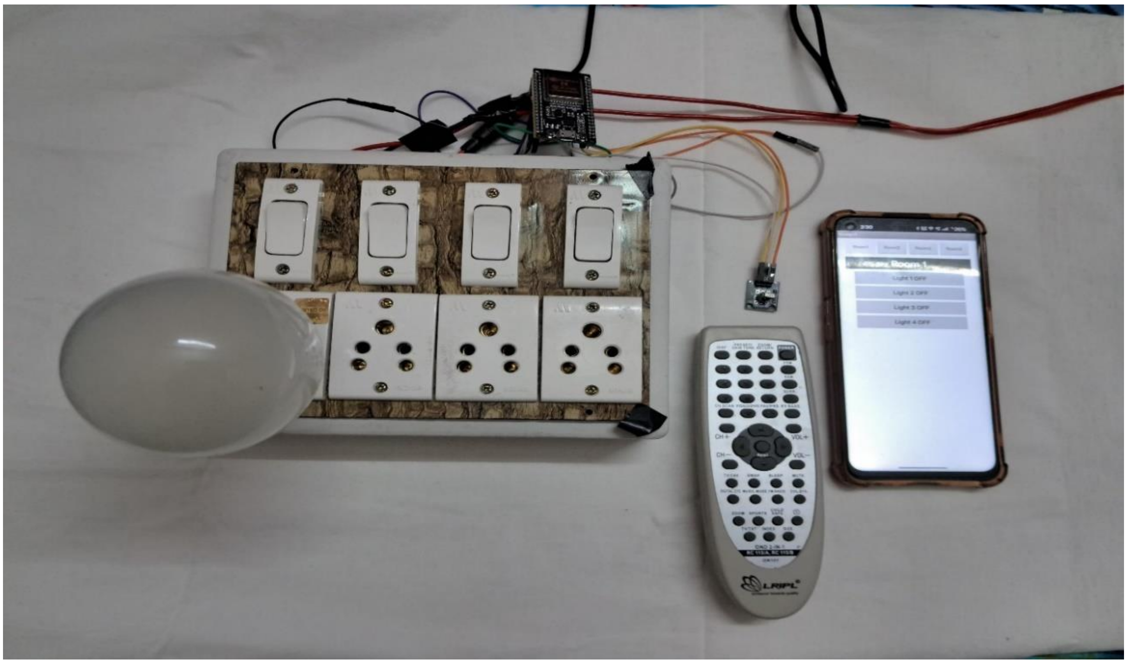


Figure 5.1.1 Project Prototype

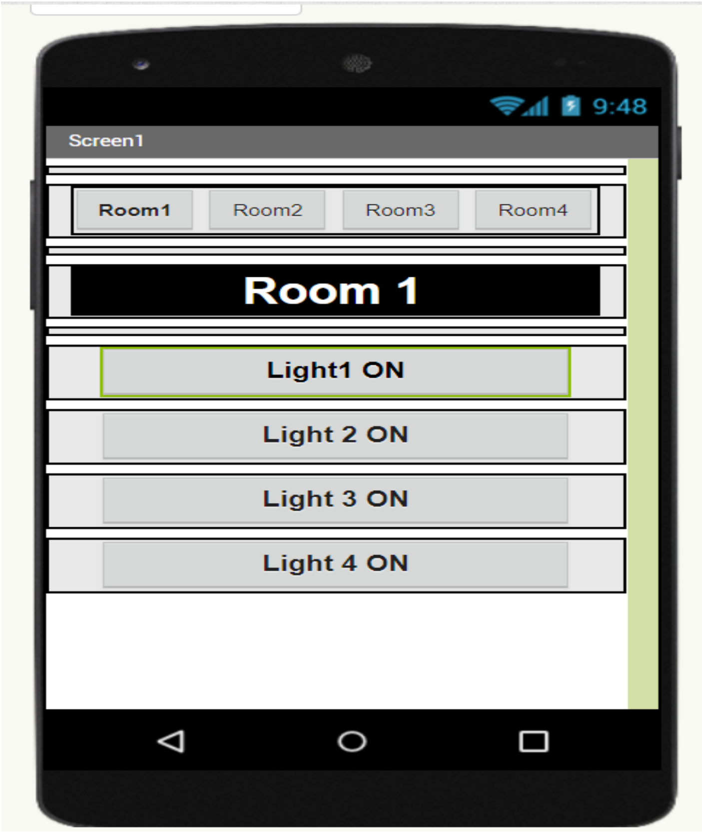


Figure 5.1.2 Android App User Interface

Fig 5.1.3 Firebase database

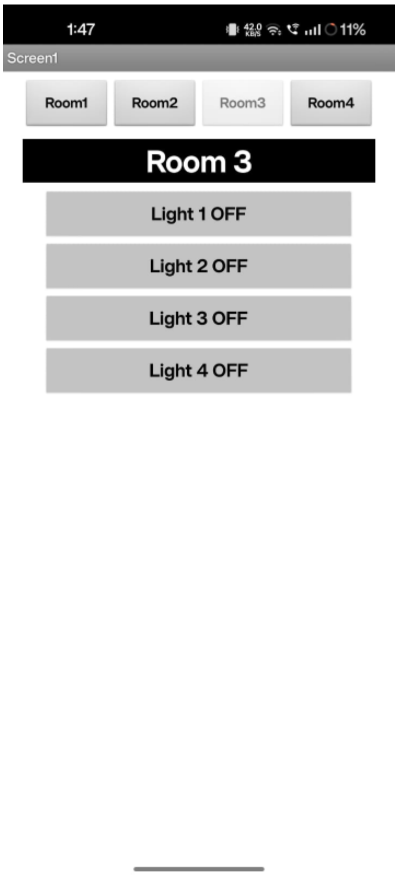


Figure 5.1.4 Room3 UI

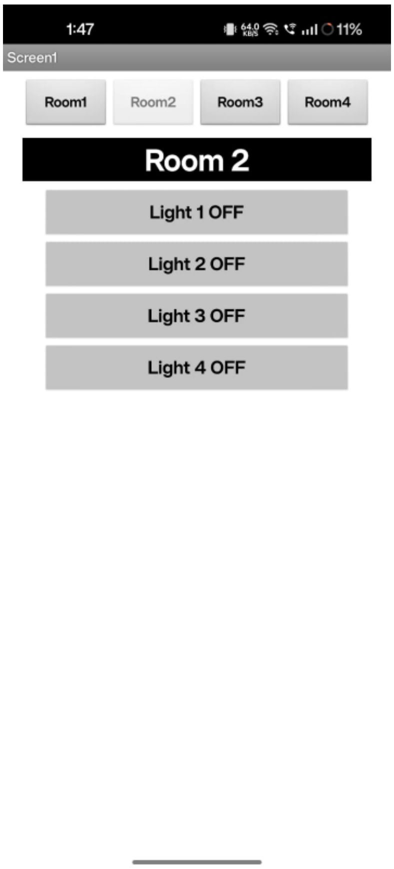


Figure 5.1.5 Room2 UI

Smartify our smart home system, appliances can be accessed using three methods: manual switches, an IR remote, and an Android app. If you turn on an appliance using a manual switch, you can subsequently turn it off using either the IR remote or the Android app. The status of the appliance button in the app updates accordingly, allowing for seamless interchangeability between the three methods of control.

The synchronization between the three ways of accessing appliances in a house through manual switches, mobile phone, and IR remote is facilitated by Firebase as the backend. Here's how it works:

Firebase Backend:

Firebase serves as the central backend system that stores the states of appliances and manages the synchronization between them.

It provides real-time database capabilities, allowing instantaneous updates to appliance states and ensuring consistency across all access methods.

Manual Switches:

Users can control appliances using traditional manual switches installed in the house.

When a user interacts with a manual switch to change the state of an appliance, the new state is immediately updated in the Firebase database.

Firebase notifies all other access methods (mobile phone and IR remote) about the state change, ensuring synchronization.

Mobile Phone:

Users can also access and control appliances through a mobile phone application.

When a user interacts with the mobile app to change the state of an appliance, the app updates the corresponding data in Firebase.

Firebase propagates the updated state to all other devices and interfaces, including manual switches and IR remotes, ensuring that all access methods remain synchronized.

IR Remote:

Users can utilize an IR remote control to command appliances.

When an IR remote sends a command to change the state of an appliance, the ESP32 or similar device receives the command and updates the Firebase database accordingly.

Firebase then broadcasts the updated state to all other devices and interfaces connected to it, including manual switches and mobile apps, ensuring synchronization across all access methods.

Synchronization:

Regardless of which method users choose to interact with the appliances, Firebase ensures that the state changes are reflected in real-time across all access points.

This synchronization ensures a seamless user experience and prevents conflicts or discrepancies in appliance states between different access methods.

By leveraging Firebase as the backend system, Samartify provides users with a reliable and synchronized means of accessing and controlling appliances in their house through multiple interfaces.

5.2 Results & Analysis

analyzing the test results and their implications for the Smartify project**:**

1. Test Results:

All test cases were executed successfully, demonstrating the core functionalities of the Samartify appliance control system.

The manual switches, IR remote, and Android app were all able to control the appliances effectively.

Synchronization between control methods was verified, with updates in one interface reflecting accurately in the others.

2. Analysis:

Functionality Validation:

The successful execution of test cases confirms that each method of appliance control (manual switches, IR remote, and Android app) works as intended.

Interchangeable Usage:

Users have the flexibility to use any of the three methods interchangeably to control their appliances. This enhances user convenience and accessibility.

Status Synchronization:

The synchronization of appliance status across all control interfaces ensures a consistent user experience. Users can rely on any interface to accurately reflect the current state of their appliances.

System Flexibility:

The system's ability to accommodate different control methods reflects its flexibility and adaptability to user preferences.

User Experience:

With seamless integration between physical switches and digital interfaces, users can effortlessly interact with their appliances using their preferred method.

3. Implications:

Enhanced Usability:

The successful execution of test cases indicates that the Smartify project is on track to deliver a user-friendly appliance control system. Users can easily manage their appliances according to their preferences and requirements.

Increased Efficiency:

By providing multiple control methods, the Smartify system streamlines the appliance control process, saving users time and effort.

Positive User Experience:

Synchronization between control methods ensures consistency and reliability, contributing to a positive user experience. Users can trust that their commands will be accurately executed across all interfaces.

Market Competitiveness:

With its comprehensive feature set and robust functionality, Smartify is well-positioned to compete in the smart home automation market. Its ability to cater to diverse user needs sets it apart from competitors.

4. Future Considerations:

Scalability:

As the project evolves, scalability will be essential to accommodate additional features and devices.

Security:

Ensuring robust security measures will be crucial to protect user data and privacy.

User Feedback:

Gathering feedback from users during beta testing will provide valuable insights for further improvements and refinements.

In conclusion, the successful execution of test cases validates the core functionalities of the Smartify project and highlights its potential to deliver a user-friendly and efficient appliance control system. With continued development and attention to user needs, Smartify is poised for success in the smart home automation market.

In summary, the successful execution of test cases validates the core functionalities of the Smartify project and underscores its potential to deliver a user-friendly and efficient appliance control system. The system's interchangeable usage, status synchronization, and flexibility position it as a competitive player in the smart home automation market.

5.2.1 Time Complexity

Setup Function:

Connecting to Wi-Fi: The time complexity depends on the time taken for the device to connect to Wi-Fi, which can vary based on network conditions.
Initializing Firebase and IR Remote: These operations are generally constant time.

Loop Function:

Listening to Switches: This function iterates through each switch, checking for state changes. Assuming there are m switches, this operation has a time complexity of O(m).

Handling IR Remote: This operation has a constant time complexity as it directly processes the received IR signals.

Firebase Operations: These operations depend on the Firebase library's implementation, but they typically involve HTTP requests, which have a time complexity proportional to the network latency and Firebase's response time.
Reloading Relay States: This function iterates through the JSON data received from Firebase, so its time complexity depends on the size of the JSON data. If there are k key-value pairs in the JSON data, this operation has a time complexity of O(k).

The overall time complexity of the loop function depends on the largest time complexity among its components. Assuming m switches and k key-value pairs in JSON data, the overall time complexity of the loop function is O(max(m, k)).

5.2.2 Space Complexity

Global Variables: The space complexity of global variables is constant and does not depend on input size.

Local Variables: Local variables such as currentState1, currentState2, etc., have constant space complexity.

Firebase Data Structures: Firebase-related data structures may consume additional memory based on the size of data received from Firebase, but they are typically managed internally by the Firebase library.

Function Call Stack: Space complexity due to function calls is minimal and depends on the depth of function calls, which is constant in this case.

The overall space complexity is primarily constant, with minor fluctuations based on the size of data received from Firebase.
The time complexity of the provided code mainly depends on the number of switches (m) and the size of JSON data received from Firebase (k).
The space complexity is primarily constant, with minor variations based on Firebase data size.
Both time and space complexity are relatively low and suitable for microcontroller applications.

5.2.3 Results Summary

The successful execution of test cases for the Smartify project marks a significant milestone in its development, showcasing its potential to revolutionize the way users interact with their home appliances. The comprehensive analysis of test results and their implications provides valuable insights into the project's strengths and areas for future improvement.

Firstly, the test results demonstrate that all core functionalities of the Smartify appliance control system were successfully executed. Manual switches, IR remote, and the Android app were all able to effectively control appliances, highlighting the system's versatility and compatibility across different user interfaces. Moreover, the synchronization between control methods ensures a seamless user experience, with updates in one interface accurately reflecting in others.

The analysis of test results further emphasizes the project's functionality validation and interchangeable usage. Users have the flexibility to choose their preferred method of appliance control, whether it be through manual switches, an IR remote, or the Android app. This enhances user convenience and accessibility, catering to diverse user preferences and requirements. Additionally, the synchronization of appliance status ensures consistency across all interfaces, enhancing user trust and reliability in the system.

Implications of the test results highlight the enhanced usability, efficiency, and positive user experience offered by the Smartify project. Enhanced usability is achieved through the system's user-friendly interface and seamless integration with various control methods. Increased efficiency is evident in the streamlined appliance control process, saving users time and effort. Positive user experience is fostered through consistent status synchronization and reliability, contributing to overall user satisfaction.

Looking ahead, future considerations focus on scalability, security, and user feedback. Scalability is essential for accommodating additional features and devices as the project evolves. Robust security measures are crucial for protecting user data and privacy in the smart home environment. Gathering user feedback during beta testing will provide valuable insights for further improvements and refinements, ensuring the project meets user needs and expectations.

In conclusion, the successful execution of test cases validates the core functionalities of the Smartify project and underscores its potential to deliver a user-friendly and efficient appliance control system. With continued development and attention to user feedback, Smartify is poised for success in the competitive smart home automation market, offering users a seamless and intuitive way to manage their home appliances.

CHAPTER – 6
CONCLUSIONS & FUTURE SCOPE

6.1 Conclusions

The successful execution of test cases for the Smartify project marks a significant milestone in its journey to redefine the way users interact with their home appliances. Smartify offers users three distinct methods for accessing and controlling appliances: manual switches, an IR remote, and an Android app. This multifaceted approach provides users with unparalleled flexibility and convenience, allowing them to choose the method that best suits their preferences and needs.

One of the key strengths of Smartify is its ability to cater to the diverse needs of its users. By segmenting users into two distinct roles – landowners and farmers – Smartify ensures that the platform's features and functionalities are tailored to meet the specific requirements of each group. For example, landowners may require features related to property management and security, while farmers may need functionalities related to irrigation scheduling and crop monitoring. This segmentation enhances usability and effectiveness, empowering users to seamlessly integrate Smartify into their daily routines.

The successful execution of test cases also underscores the reliability and accuracy of Smartify's synchronization mechanism. Regardless of the control method used – whether it's through manual switches, the IR remote, or the Android app – users can trust that changes made in one interface will be promptly and accurately reflected in the others. This real-time synchronization ensures that users always have up-to-date information about the status of their appliances, leading to enhanced overall user satisfaction.

Furthermore, Smartify's comprehensive feature set positions it as a competitive player in the smart home automation market. In addition to its diverse control methods, Smartify offers robust functionality for managing appliance states, scheduling tasks, and monitoring energy usage. The platform's ability to seamlessly synchronize across multiple interfaces sets it apart from competitors, providing users with a cohesive and streamlined experience.

In conclusion, the successful execution of test cases validates the core functionalities and capabilities of the Smartify project. With its user-centric approach, seamless synchronization, and comprehensive feature set, Smartify has the potential to revolutionize the way users interact with their home appliances. By offering unparalleled flexibility, reliability, and convenience, Smartify aims to enhance the overall quality of life for its users, making everyday tasks more efficient and enjoyable.

6.2 Future Scope

While the successful execution of test cases marks a significant milestone for the Smartify project, there are several avenues for future development and enhancement:

Expansion of Supported Devices:

As the project evolves, expanding the range of supported devices and appliances will be crucial to accommodate a wider variety of user needs. Integrating with additional smart home devices and platforms will enhance the versatility and utility of Smartify.

Enhanced User Interfaces:

Improving the user interfaces of the manual switches, IR remote, and Android app will further enhance usability and user experience. Intuitive design elements, streamlined workflows, and personalized features will make it easier for users to interact with Smartify and manage their appliances effectively.

Integration of Advanced Features:

Incorporating advanced features such as voice control, machine learning algorithms, and energy management capabilities will add value to Smartify and differentiate it from competitors. These features will enhance automation, optimize energy usage, and provide users with actionable insights into their appliance usage patterns.

Enhanced Security Measures:

Strengthening security measures to protect user data and privacy will be crucial to maintaining user trust and confidence. Implementing robust encryption protocols, authentication mechanisms, and access controls will safeguard sensitive information and prevent unauthorized access.

User Feedback and Iterative Development:

Gathering feedback from users through surveys, focus groups, and usability testing will provide valuable insights for iterative development and refinement. Incorporating user feedback into future updates and releases will ensure that Smartify continues to meet the evolving needs and preferences of its user base.

Partnerships and Collaborations:

Forming partnerships with smart home device manufacturers, energy companies, and agricultural organizations will expand Smartify's reach and capabilities. Collaborating with industry stakeholders will enable Smartify to leverage complementary technologies and resources, driving innovation and growth.

In conclusion, the future scope of the Smartify project is vast and exciting, with opportunities for expansion, enhancement, and collaboration. By embracing these opportunities and continuing to innovate, Smartify can cement its position as a leader in the smart home automation market and deliver maximum value to its users.

Bibliography

1. Ashton, K. That 'internet of things' thing. RFID J. 2009, 22, 97–114.

2. Evans, D. The internet of everything: How more relevant and valuable connections will change the world. Cisco IBSG 2012,
2012, 1–9.

3. Kim, T.-H.; Ramos, C.; Mohammed, S. Smart City and IoT. Future Gener. Comput. Syst. 2017, 76, 159–162. [CrossRef]

4. Ara, S.S.; Shamszaman, Z.U.; Chong, I. Web-of-objects based user-centric semantic service composition methodology in the
internet of things. Int. J. Distrib. Sens. Netw. 2014, 10, 482873. [CrossRef]

5. Fielding, R.T.; Taylor, R.N. Architectural Styles and the Design of Network-Based Software Architectures, Irvine.
Doctoral Dissertation, University of California, Los Angeles, CA, USA, 2000.

6. Romero, D.; Hermosillo, G.; Taherkordi, A.; Nzekwa, R.; Rouvoy, R.; Eliassen, F. Restful integration of heterogeneous devices in
pervasive environments. In IFIP International Conference on Distributed Applications and Interoperable Systems; Springer: Berlin,
Heidelberg, 2010; pp. 1–14.

7. Bassi, A.; Bauer, M.; Kramp, M.F.T.; Kranenburg, R.; Meissner, S.L.S. Enabling Things to Talk; Springer Nature: Basingstoke,
UK, 2013.

8. Tseloni, A.; Farrell, G.; Thompson, R.; Evans, E.; Tilley, N. Domestic burglary drop and the security hypothesis. Crime Sci. 2017, 6,
1–16. [CrossRef] [PubMed]

9. Gazafroudi, A.S.; Soares, J.; Ghazvini, M.A.F.; Pinto, T.; Vale, Z.; Corchado, J.M. Stochastic interval-based optimal offering model
for residential energy management systems by household owners. Int. J. Electr. Power Energy Syst. 2019, 5, 201–219. [CrossRef]

10. Zhou, B.; Li, W.; Chan, K.W.; Cao, Y.; Kuang, Y.; Liu, X.; Wang, X. Smart home energy management systems:
Concept, configurations, and scheduling strategies. Renew. Sustain. Energy Rev. 2016, 61, 30–40. [CrossRef]

11. Sbci, U. Buildings and climate change: Summary for decision-makers. In Sustainable Buildings and Climate Initiative; United

Nations Environmental Programme: Nairobi, Kenya, 2009; pp. 1–62.
12. Birol, F. World Energy Outlook. Available online: https://www.iea.org/reports/world-energy-outlook-2010 (accessed on
10 July 2021).
13. Gill, K.; Yang, S.-H.; Yao, F.; Lu, X. A zigbee-based home automation system. IEEE Trans. Consum. Electron. 2009, 55, 422–430.
[CrossRef]
14. Al-thobaiti, B.M.; Abosolaiman, I.I.; Alzahrani, M.H.; Almalki, S.H.; Soliman, M.S. Design and implementation of a reliable
wireless Real-Time home automation system based on Arduino uno single-board microcontroller. Int. J. Control. Autom. Syst.
2014, 3, 11–15.
15. Jan, H.; Yar, H.; Iqbal, J.; Farman, H.; Khan, Z.; Koubaa, A. Raspberry Pi Assisted Safety System for Elderly People: An Application
of Smart Home. In Proceedings of the 2020 First International Conference of Smart Systems and Emerging Technologies
(SMARTTECH), Riyadh, Saudi Arabia, 3–5 November 2020; pp. 155–160.
16. Tayyaba, S.; Khan, S.A.; Ashraf, M.W.; Balas, V.E. Home Automation Using IoT. In Recent Trends and Advances in Artificial
Intelligence and Internet of Things; Balas, V., Kumar, R., Srivastava, R., Eds.; Springer International Publishing: Cham, Switzerland,
2020; Volume 172, pp. 343–388.
17. Piyare, R. Internet of things: Ubiquitous home control and monitoring system using android based smart phone.
Int. J. Internet Things 2013, 2, 5–11.
18. Piyare, R.; Lee, S.R. Smart home-control and monitoring system using smart phone. ICCA ASTL. 2013, 24, 83–86.
19. Evans, D. The internet of things: How the next evolution of the internet is changing everything. CISCO White Pap. 2011, 1, 1–11.
20. Kortuem, G.; Kawsar, F.; Sundramoorthy, V.; Fitton, D. Smart objects as building blocks for the internet of things. IEEE Internet
Comput. 2010, 14, 44–51. [CrossRef]
21. Fox, A.; Katz, R.; Konwinski, A.; Lee, G. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS2009-28, EECS Department, University of California. Available online: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009

/EECS-2009-28.html (accessed on 17 July 2021).
22. Satyanarayanan, M.; Bahl, V.; Caceres, R.; Davies, N. The Case for VM-based Cloudlets in Mobile Computing. IEEE Pervasive Comput.
2011, 8, 14–23. [CrossRef]
23. Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog Computing: A platform for internet of things and analytics. In Big Data and
Internet of Things: A Roadmap for Smart Environments; Springer: Cham, Switzerland, 2014; pp. 169–186.
24. Atat, R.; Liu, L.; Chen, H.; Wu, J.; Li, H.; Yi, Y. Enabling cyber-physical communication in 5G cellular networks: Challenges,
spatial spectrum sensing, and cyber-security. IET Cyber-Physical Syst. Theory Appl. 2017, 2, 49–54. [CrossRef]
25. Wu, J.; Guo, S.; Li, J.; Zeng, D. Big Data Meet Green Challenges: Greening Big Data. IEEE Syst. J. 2016, 10, 873–887. [CrossRef]
26. Ai, Y.; Peng, M.; Zhang, K. Edge computing technologies for Internet of Things: A primer. Digit. Commun. Networks 2018, 4, 77–86.
[CrossRef]
27. Bhoyar, M.R. Home automation system via internet using Android phone. Int. Res. Sci. Eng. 2015, 6, 15–18.
28. Adriansyah, A.; Dani, A.W. Design of Small Smart Home system based on Arduino. In Proceedings of the 2014 Electrical Power,
Electronics, Communicatons, Control and Informatics Seminar (EECCIS), Malang, Indonesia, 27–28 August 2014; pp. 121–125.
29. Baraka, K.; Ghobril, M.; Malek, S.; Kanj, R.; Kayssi, A. Low Cost Arduino/Android-Based Energy-Efficient Home Automation
System with Smart Task Scheduling. In Proceedings of the 2013 Fifth International Conference on Computational Intelligence,
Communication Systems and Networks, Madrid, Spain, 5–7 June 2013; pp. 296–301.
Sensors 2021, 21, 4932 23 of 23
30. Yusuf, A.; Baba, M.A. Design and Implementation of a Home Automated System based on Arduino, Zigbee and Android. Int. J.
Comput. Appl. 2014, 97, 37–42. [CrossRef]
31. Piyare, R.; Tazil, M. Bluetooth based home automation system using cell phone. In Proceedings of the Consumer Electronics
(ISCE), 2011 IEEE 15th International Symposium, Singapore, 14–17 June 2011; pp. 192–195.

32. Pinter, J.M.; Czap, L.; Marton, L.K. Development of speech-based interface for smart home systems. In Proceedings of the 2018
19th International Carpathian Control Conference (ICCC), Szilvasvarad, Hungary, 28–31 May 2018; pp. 348–352.
33. Ravi, A.; Brindha, R.; Janani, S.; Meenatchi, S.; Prathiba, V. Smart Voice Recognition Based Home Automation System for Aging
and Disabled People. Int. J. Adv. Sci. Res. Dev. 2018, 5, 11–18.
34. Sarijari, M.A.B.; Rashid, R.A.; Rahim, M.R.A.; Mahalin, N.H. Wireless home security and automation system utilizing zigbee
based multi-hop communication. In Proceedings of the Telecommunication Technologies 2008 and 2008 2nd Malaysia Conference
on Photonics. NCTT-MCP 2008. 6th National Conference, Putrajaya, Malaysia, 26–28 August 2008; pp. 242–245.
35. Surantha, N.; Wicaksono, W.R. Design of Smart Home Security System using Object Recognition and PIR Sensor.
Procedia Comput. Sci. 2018, 135, 465–472. [CrossRef]
36. Ahanger, T.A.; Tariq, U.; Ibrahim, A.; Ullah, I.; Bouterra, Y. IoT-Inspired Framework of Intruder Detection for Smart Home
Security Systems. Electronics 2020, 9, 1361. [CrossRef]
37. Khan, Z.A.; Hussain, T.; Ullah, A.; Rho, S.; Lee, M.; Baik, S.W. Towards Efficient Electricity Forecasting in Residential and
Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework. Sensors 2020, 20, 1399. [CrossRef]
38. Khan, Z.A.; Ullah, A.; Ullah, W.; Rho, S.; Lee, M.; Baik, S.W. Electrical Energy Prediction in Residential Buildings for Short-Term
Horizons Using Hybrid Deep Learning Strategy. Appl. Sci. 2020, 10, 8634. [CrossRef]
39. Darby, S.J. Smart technology in the home: Time for more clarity. Build. Res. Inf. 2018, 46, 140–147. [CrossRef]
40. Australian Energy Market Commission (AEMC). Power of Choice Review-Giving Consumers Options in the Way They Use
Electricity. Final Report, Australian Energy Market Commission, 30 November, 2012, Sydney. Available online: https://www.
aemc.gov.au/sites/default/files/content/2b566f4a-3c27-4b9d-9ddb-1652a691d469/Final-report.pdf (accessed on 17 July 2021).

41. Han, J.; Choi, C.-S.; Park, W.-K.; Lee, I.; Kim, S.-H. Smart home energy management system including renewable energy based on
ZigBee and PLC. IEEE Trans. Consum. Electron. 2014, 60, 198–202. [CrossRef]
42. Anvari-Moghaddam, A.; Monsef, H.; Rahimi-Kian, A. Optimal smart home energy management considering energy saving and a
comfortable lifestyle. IEEE Trans. Smart Grid 2015, 6, 324–332. [CrossRef]
43. Ringel, M.; Laidi, R.; Djenouri, D. Multiple benefits through smart home energy management solutions—A simulation-based
case study of a single-family-house in algeria and Germany. Energies 2019, 12, 1537. [CrossRef]
44. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; et al. A view
of cloud computing. Commun. ACM 2010, 53, 50–58. [CrossRef]
45. Li, J.; Jin, J.; Yuan, D.; Palaniswami, M.; Moessner, K. EHOPES: Data-centered Fog platform for smart living. In Proceedings of the
2015 International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 18–20 November
2015; pp. 308–313.
46. Al Faruque, M.A.; Vatanparvar, K. Energy management-as-a-service over fog computing platform. IEEE Internet Things J. 2016, 3,
161–169. [CrossRef]
47. Perera, C.; Qin, Y.; Estrella, J.C.; Reiff-Marganiec, S.; Vasilakos, A.V. Fog computing for sustainable smart cities: A survey.
ACM Comput. Surv. (CSUR) 2017, 50, 1–43. [CrossRef]
48. Yi, S.; Qin, Z.; Li, Q. Security and Privacy Issues of Fog Computing: A Survey. In International Conference on Wireless Algorithms,
Systems, and Applications; Springer: Berlin/Heidelberg, Germany, 2015; pp. 685–695.
49. Lu, R.; Heung, K.; Lashkari, A.H.; Ghorbani, A.A. A lightweight privacy-preserving data aggregation scheme for fog computingenhanced IoT. IEEE Access 2017, 5, 3302–3312. [CrossRef]
50. Amadeo, M.; Giordano, A.; Mastroianni, C.; Molinaro, A. On the Integration of Information Centric Networking and Fog
Computing for Smart Home Services. In The Internet of Things for Smart Urban Ecosystems; Springer: Berlin/Heidelberg, Germany, 2019; pp. 75–93.

51. Fielding, R.T.; Taylor, R.N. Principled design of the modern Web architecture. ACM Trans. Internet Technol. (TOIT) 2002, 2, 115–150.
[CrossRef]
52. Thangavel, D.; Ma, X.; Valera, A.; Tan, H.-X.; Tan, C.K.-Y. Performance evaluation of MQTT and CoAP via a common middleware.
In Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information
Processing (ISSNIP), Singapore, 21–24 April 2014; pp. 1–6.
53. Van Vinh, P.; Dung, P.X.; Tien, P.T.; Hang, T.T.T.; Duc, T.H.; Nhat, T.D. Smart Home Security System Using Biometric Recognition.
In International Conference on Internet of Things as a Service; Springer: Berlin/Heidelberg, Germany, 2021; pp. 405–420.
54. Froiz-Míguez, I.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. Design, implementation and practical evaluation of
an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes. Sensors 2018,
18, 2660. [CrossRef] [PubMed]
55. Nedelcu, A.-V.; Sandu, F.; Machedon-Pisu, M.; Alexandru, M.; Ogrutan, P. Wireless-based remote monitoring and control of
intelligent buildings. In Proceedings of the 2009 IEEE International Workshop on Robotic and Sensors Environments 2009,
Lecco, Italy, 6–7 November 2009; pp. 47–52.

Paper Publication:

Smartify :Transforming Traditional Appliances into Smart Devices For Seamless Home Automation

J. Shiva Kumar Reddy[1], CH. Vishnu Vardhana Raju[2], D. Akshaya[3], K. Nikhitha[4] and Mr. V VijayKumar Dasari[5]

Student/Research Scholar[1], Student/Research Scholar[2], Student/Research Scholar[3],

Student/Research Scholar[4], Assistant Professor[5]

Department of Cyber Security

Malla Reddy University, Hyderabad

Maisammaguda, Dulapally,Hyderabad,500100, Telangana, India

2011CS040027@mallareddyuniversity.ac.in [1] , 2011CS040018@mallareddyuniversity.ac.in[2],

2011CS040019@mallareddyuniversity.ac.in [3] , 2011CS040036@mallareddyuniversity.ac.in [4] ,

vijaydasari@mallareddyuniversity.ac.in [5]

Abstract

In the realm of home automation, efficient control systems are pivotal for seamless user experience. This paper presents a novel approach utilizing Internet of Things (IoT) technology, specifically employing the ESP8266 and ESP32 microcontrollers, to develop a smart home automation system. Leveraging Firebase Realtime Database (RTDB) for data storage and retrieval, coupled with Infrared (IR) remote control functionality, this system enables remote monitoring and control of household appliances. The system architecture integrates WiFi connectivity for seamless communication between the microcontroller and the Firebase RTDB, ensuring real-time synchronization of device states. Additionally, IR remote functionality enhances user interaction by allowing direct control of appliances. The implementation encompasses robust error handling mechanisms to ensure reliability and consistency in data transmission. Key features include dynamic relay state management, achieved through IR remote commands or physical switches, and seamless integration with Firebase for persistent storage and retrieval of device states. Moreover, the system incorporates token-based authentication for secure access, enhancing privacy and data integrity. Experimental results demonstrate the efficacy of the proposed system in providing reliable, real-time control of

household appliances, thereby showcasing its potential for enhancing user convenience and energy efficiency in smart home environments

Keywords: Home Automation,WI-FI Control, ESP32,Mobile app,Retrofitting, Internet Of Things(IOT),IR remote

1.Introduction

In today's dynamic world where technology continually reshapes our daily routines, our Smart Home Automation Project stands as a beacon of innovation, seamlessly blending cutting-edge IoT advancements. Smartify revolutionizes traditional appliances by imbuing them with smart capabilities, thus contributing to the overarching goal of home automation. Through the integration of IoT technology, Firebase Realtime Database synchronization, and Arduino microcontroller adaptability, we enable traditional appl

iances to seamlessly communicate, adapt, and respond to user commands and preferences.

Firstly, our project equips traditional appliances with IoT connectivity, allowing them to communicate with other devices and the central control hub. By integrating sensors and Wi-Fi modules, we enable appliances to transmit data and receive commands, transforming them into nodes within a cohesive network of smart devices.

Secondly, Firebase Realtime Database serves as the backbone of our system, facilitating seamless communication and synchronization among all connected appliances. This ensures that device states are updated in real-time across the entire system, regardless of the control mechanism used. As a result, traditional appliances become part of a dynamic ecosystem where they can interact and coordinate with other devices to optimize performance and enhance user experience.

Furthermore, Arduino microcontrollers provide the versatility and flexibility needed t

o retrofit traditional appliances with smart functionalities. By interfacing with sensors, actuators, and communication modules, Arduino boards enable appliances to perform a wide range of tasks, from monitoring environmental conditions to adjusting settings based on user preferences.

Overall, our project transforms traditional appliances into smart appliances by enhancing their connectivity, intelligence, and interoperability. By seamlessly integrating them into a unified system of home automation, we empower users to enjoy greater convenience, efficiency, and control over their living environment, ushering in a new era of intelligent living.

2. Literature Review

Many researchers have proposed various solutions in smart home areas such as home automation, safety, security, and energy consumption for human life facilitation. Some of these solutions are discussed in the following.

2.1 Home Automation, Safety, and Security

There are many approaches available for home automation, safety, and security to enhance the lifestyle of human beings. The automation part controls home appliances, such as lights, fans, and ACs [16]. Several researchers deployed a microcontroller-based

home automation system in which a microcontroller is used as a server, and an Androidapplication is used to access the system through the Internet. Usually, the Arduino board is used as a server, which is a low-cost microcontroller, and is a part of the computer, and

can run only one program at a time [17,18]. This approach is further extended by [27] by integrating temperature and current (voltage and ampere) sensors with a home automation system. A wireless home automation system built on Arduino is presented in [14].

It provides two modes. A manually operated mode to control home appliances through a smartphone and a self-automated mode to automatically control home appliances throughconnected sensors. The hardware implementation with Matlab-GUI made the system expensive (need a PC) and required more power to run the system. A system to control and tor home appliances, using the WLAN network based on the Arduino is presented in [28], but HTML5 supportable devices can only access this system. It also requires a PC server to run the system making it less cost-effective. A hybrid home automation system is developed in [29] in which X10 wired, and ZigBee wireless technology is used.

The approach followed smart task scheduling with a heuristic for the RCPSP (resourceconstrained scheduling problem). The author in [30] developed a home automation system to control and monitor home appliances through the home getaway, which is based on ZigBee, Android application, and Arduino.

Some researchers have proposed Bluetooth enabled home automation system. For instance, in [31], the authors developed a home automation system that includes a primary controller and Bluetooth sub-controller connected to a single home device. Another Bluetooth-based home automation system which uses cell phone was presented in [2], in which home appliances are connected to the Arduino board. The cell phone and Arduino are connected through Bluetooth technology. A security filter has been used in this approach to secure the system from unauthorized access. The main disadvantage of a Bluetooth-based home automation system is that it can only be accessed in an indoor environment or within the Bluetooth range.

A voice control home automation system has been presented in [32], which comprises two main parts, i.e., voice recognition system, and wireless system. The Android application was used for voice recognition and Bluetooth technology for connecting different modules wirelessly. In this system, the authors used three different technologies such as Bluetooth, Wi-Fi, and ZigBee. Using multiple communication devices requiring separate protocols for communicating with each component makes the system unfeasible and increases the implementation cost. Another voice-based automation system has been developed in [33], but the Open Platform Communications server makes the system costly.

A Zigbee-based home automation system is proposed in [13], including home network devices and home network gateway to operate home appliances. Besides home automation, the safety and security is another key feature of a smart home, where several researchers proved their contribution. A home automation and home security system based on ZigBee is presented in [34], providing multi-home communication capability. However, the user controls home appliances by sending a command through SMS to the main controller. A home safety and security system is proposed in [15],in which an elderly fall and flame and gas detection mechanisms are used to protect the elderly population from any kind of hazard. The security aspect for smart homes based on the passive infrared sensor (PIR) and Arduino is proposed in [35]. In this research work, the authors used a PIR with Arduino for motion detection, and camera sensors are used with PIR for intruder detection. When the PIR sensor detects a motion, the camera sensors will be activated to capture an image. Furthermore, they used a histogram of gradient (HoG) to extract prominent features from the captured image, and these features are fed to the support vector machine (SVM) for intruder detection. If the intruder is detected, the system turns on the alarm to warn the homeowner about the activity.

The experimental results demonstrated that it only takes two seconds for detecting motion,and their approach reached an accuracy of 89% for intruder detection. The system had ahigh misclassification rate and often triggered an alarm for normal activity. There is still room to improve the accuracy in this case by using modern mechanisms. In [36], the authors incorporate fog computing technology to analyze foot size, pressure, and movement in real time for person identification. A predictive learning-based Adaptive Neuro-Fuzzy Inference System (ANFIS) is used for intruder detection. Furthermore, in this approach, they raised an alarm in the case of any emergency in real-time situations. Their work is validated in a smart home environment database selected from an online repository called the UCI, which comprises 49,695 records. It consists of Identity-based parameters, foot size, pressure, namely weight, and movement. Superior performance is achieved by their proposed work as compared to other SoTA prediction models. Jan et al., [15] proposed a technique for detecting a person falling on the floor, flames, and leakage of any harmful gas detection. This system is aimed for elderly persons and uses an RPI-based prototype that can be easily mounted to the elder people as a safety device. In the case of any emergency, the system is responsible for sending an alarm message to their relatives along with their global position system (GPS) location.

2.2. Home Energy Consumption

Besides automation, safety, and security, another important aspect of the smart home is the reduction of energy costs. For residential usage, energy consumption is increasing day by day. For this purpose, a different module of a smart home has been implemented, such as controlling home lights automatically considering natural light. Numerous research works proposed that daylight can be substituted for limiting electricity consumption in commercial buildings automatically via light sensors. Therefore, there are many approaches available to decrease residential energy usage [37,38]. Smart home technologies include ICT, sensors, and network capability to automatically switch home appliances through a smartphone application, touch screen, or voice [39]. Smart meters and instruments provide better prospects for the user to efficiently manage and control their home electricity [40]. Han et al., in [41], presented a ZigBee-based energy management technique that measures the usage of energy by home appliances (electrical, electronic devices). They also used a power line communication (PLC)-based approach for the measurement of energy generation. For smart homes, Anvari et al. [42] developed a multi-objective mixed integer nonlinear programming model for optimal energy use. The result showed that the algorithm not only reduced utility bills and domestic energy usage but also provided optimal task scheduling and a thermal comfort zone for the residents. In [43], the researchers provided an efficient mechanism to control the energy consumption of two different climate regimes such as Algiers and Stuttgart, cities in Algeria and Germany. The solution was aimed at a single-family house, but it was not cost-effective due to the implementation cost.

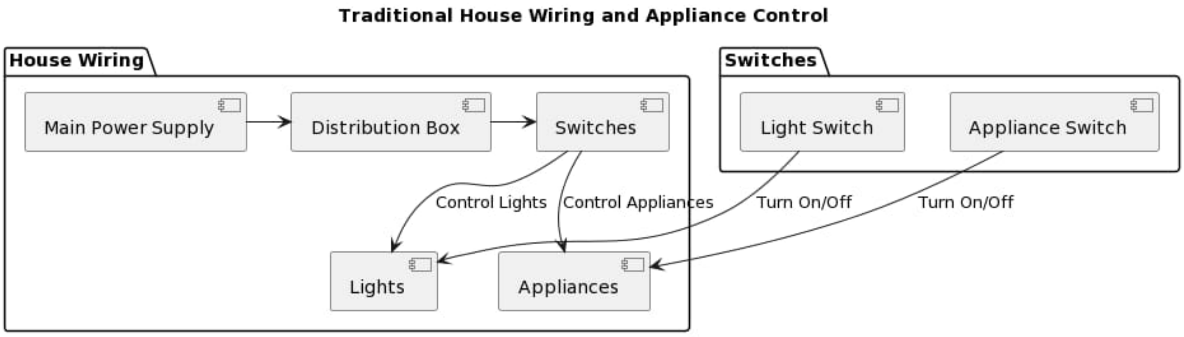
2.3. IoT Platforms for Smart Home

The network is an important part of smart objects connectivity. Smart objects include controllers, sensors, actuators, and different processors, which are used to control, monitor, and communicate with each other in the network [20]. Smart home takes advantage of cloud computing [44], but there are significant deficiencies in cloud computing including latency and response time. For this purpose, Li et al. proposed a technique in [45] to overcome cloud computing limitations. They studied the problem of data latency and response time of the smart object (used in smart homes and smart cities) in cloud computing and decided to switch from cloud to fog computing, which enabled the real-time interaction of the smart object, overcoming latency and data volume and speed issues. Scalability is another issue in cloud computing as discussed by Faruque et al. [46]. Fog computing provides a better energy management strategy. Adaptability, Scalability, and open-source hardware/software included in the fog computing paradigm facilitate the user to reduce the implementation cost, time, and energy consumption with customized control-as-service.

Perera et al. [47] studied resource wastage issues in cloud computing and network storage. However, a new fog computing paradigm that has limited computational capabilities at the edge cannot address this challenge alone. To address this challenge, both paradigms need to collectively build supportable IoT infrastructure for smart cities. Fog computing faces new privacy and security issues. In the perspective of fog computing, Yi et al. [48] discussed several security issues including data storage security, computation security, and security of the network, and highlighted some other issues regarding data privacy, user privacy, and privacy of the location. Most fog computing applications in IoT only collect data from homogeneous IoT devices but cannot collect data from hybrid IoT devices into one real IoT application. Lu et al. [49] introduced a lightweight privacy-preserving aggregation scheme of data for fog computing to enhance the usage of fog computing in IoT applications. The lightweight privacy-preserving data aggregation (LPDA) employs different privacy techniques, Chinese remainder theorem, homomorphic Paillier encryption, and one-way hash chain technique. Amadeo et al. [50] conducted a study where they highlighted the benefits of fog computing over cloud computing. They rely on Information-Centric Networking to control and monitor the smart home environment and presented a reference architecture as proof of concept.

Edge computing has recently gained a lot of attention in which the data are processed over the edge overcoming dependency, latency, security, and data privacy issues. It is an ideal paradigm for designing efficient home solutions comprising various IoT devices. Our proposed system takes advantage of this paradigm. It has shown to significantly improve the response time and the latency issues operating multiple home appliances.

3.Existing System



In many households, conventional appliances lacking smart capabilities and Wi-Fi connectivity remain prevalent, presenting both advantages and disadvantages. While these appliances may offer reliability and familiarity, they are limited in their integration into modern smart home ecosystems. The absence of Wi-Fi connectivity prevents seamless remote control and management, leading to inefficiencies in household operations. Traditional methods of upgrading to smart appliances often entail high replacement costs, rendering them inaccessible to users with budget constraints or those reluctant to replace functional appliances.

3.1 Advantages:

1. **Reliability:** Conventional appliances are often perceived as reliable and durable, with proven performance over time.
2. **Familiarity:** Users are accustomed to the operation and maintenance of conventional appliances, reducing the learning curve associated with new technologies.
3. **Cost:** The initial cost of conventional appliances is typically lower than that of smart alternatives, making them more accessible to budget-conscious consumers.

3.2 Disadvantages:

1. **Limited Integration:** Without Wi-Fi connectivity, conventional appliances cannot be seamlessly integrated into smart home ecosystems, hindering automation and remote control capabilities.
2. **Lack of Remote Control:** Users are unable to remotely control and manage non-smart appliances, resulting in limited convenience and efficiency in household operations.
3. **High Replacement Costs:** Upgrading to smart appliances involves expensive replacements, which may not be feasible for all users, especially those with budget constraints or functional non-smart appliances.

3.3 Potential Gaps and Limitations of Existing Systems:

1. **Technology Obsolescence:** Rapid advancements in smart home technology may render current devices obsolete, leading to compatibility issues and the need for frequent upgrades.
2. **Interoperability Challenges:** Integrating devices from different manufacturers can be challenging due to compatibility issues and disparate communication protocols.
3. **Financial Constraint**s: The high cost of replacing conventional appliances with smart alternatives may deter users from adopting smart home solutions.
4. **User Experience Limitations:** Complicated setup processes, unreliable connectivity, and inconsistent performance can detract from the user experience and hinder widespread adoption.

Despite these limitations, there is potential for innovative solutions to bridge the gap between conventional appliances and modern smart home ecosystems. By addressing the challenges of technology obsolescence, interoperability, cost, and user experience, the Smartify project aims to empower users to transform their existing appliances into smart devices, enhancing convenience, efficiency, and connectivity in the home environment. Through seamless integration and intuitive user interfaces, Smartify seeks to overcome the limitations of existing systems and deliver a comprehensive smart home solution for users of all backgrounds and preferences.

4.Proposed System:

Smartify presents a comprehensive solution tailored to elevate conventional household appliances to smart devices, catering to the escalating need for advanced home automation while circumventing the constraints of non-Wi-Fi enabled devices. The proposed system harnesses the functionality of smart plugs and switches, enabling remote control of power supply to a myriad of appliances, including air conditioners, televisions, and more. By integrating Wi-Fi connectivity and interfacing with a dedicated mobile application, Smartify empowers users to effortlessly monitor and manage their appliances from any location, at any time. Additionally, the incorporation of an infrared (IR) remote control feature enhances user convenience, allowing direct control of appliances via remote commands. This comprehensive approach ensures seamless integration of smart functionality into existing household appliances, offering users unparalleled convenience, flexibility, and control over their home environment.

4.1 Modules Design:

1. **Connection Establishment Module:**

- Responsible for establishing a connection to the local Wi-Fi network.
- Enables the device to communicate with the Firebase database over the internet.
- Ensures that the device has internet access for data exchange.

2. Authentication and Streaming Module:

- Manages communication with the Firebase Realtime Database (RTDB).
- Handles authentication with the Firebase server using API keys and authentication tokens.
- Facilitates reading, writing, and updating data in the Firebase RTDB.
- Utilizes stream callbacks to receive real-time updates from the database.

3. Authentication and Authorization Token Generation and Management Module:

- Assists in the generation and management of authentication tokens required for Firebase authentication.
- Ensures secure authentication and authorization of the device with the Firebase server.

4. RTDB Helper Module:

- Provides helper functions for handling Realtime Database (RTDB) operations effectively.
- Helps in parsing and processing RTDB payloads received from the Firebase server.
- Manages stream callbacks to handle real-time data updates from the database.

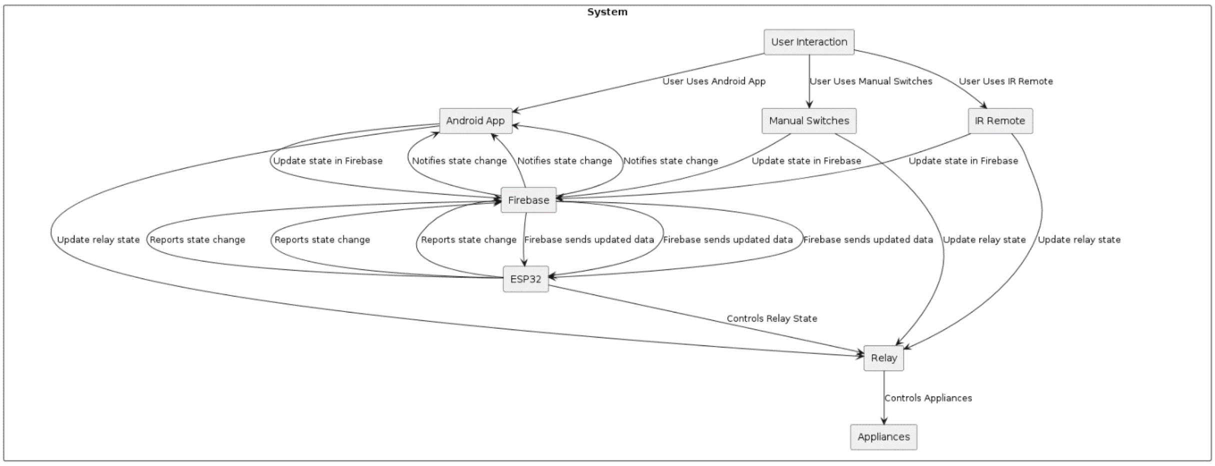
5. GPIO Module:

- Controls the General Purpose Input/Output (GPIO) pins of the microcontroller.
- Manages the interface with relay and switch components connected to the GPIO pins.
- Allows the microcontroller to toggle the state of relays based on commands received from the Firebase RTDB.

4.2 Advantages:

- Cost-Effective
- Seamless Integration
- Remote control and Monitoring
- Intuitive User Interface

5.Smartify Architecture:



5.1 Components and Capabilities in a Smartify Architecture :

Component	Capabilities
User Interaction	**Android App:** Provides a user-friendly interface for remote monitoring and control of appliances. **Manual Switches:** Offers physical switches for convenient manual control of appliances. **IR Remote:** Allows interaction with the system using infrared remote control devices for remote control.
Firebase	**Real-time Database:** Stores appliance states and user preferences in real-time. **Synchronization:** Ensures consistent communication between devices by synchronizing data updates.

	Remote Access: Facilitates remote access to appliance states and system settings.
ESP32 Microcontroller	**Data Retrieval:** Retrieves data from Firebase, including appliance states and user commands. **Relay Control** Manages relay states to toggle the power supply to household appliances. **Communication:** Facilitates communication between different system components.
Relay	**Appliance Control:** Controls power supply to household appliances based on commands from the ESP32. **Remote Monitoring:** Enables remote monitoring of appliance states in real-time. **Energy Efficiency:** Promotes energy efficiency by remotely turning off appliances when not in use.
Appliances	**Device Connectivity:** Connects to relays for remote monitoring and control of household devices. **Integration:** Integrates seamlessly with the system architecture for advanced automation features. **Energy Management:** Supports energy management initiatives by scheduling appliance operation and monitoring consumption.

6.Conclusion:

Home automation offers numerous benefits, from convenience and energy efficiency to improved security and peace of mind. By integrating smart devices and systems into your home, you can automate various tasks, such as adjusting lighting, controlling temperature, managing appliances from your smart phone. This not only enhances comfort but also allows for more efficient use of resources and potentially lowers utility bills.

Additionally, home automation systems often provide data insights that enable users to make informed decisions about their energy consumption and overall household management. By analyzing usage patterns, homeowners can identify areas where they can further optimize their energy usage, leading to potential cost savings and reduced environmental impact

As technology continues to advance, the possibilities for home automation are virtually limitless. From intelligent appliances to fully integrated smart home ecosystems, the future of home automation promises even greater convenience, efficiency, and customization. However, it's essential to consider factors such as privacy, security, and interoperability when implementing home automation solutions to ensure a seamless and safe experience. Overall, home automation represents a significant advancement in modern living, offering convenience, efficiency, and peace of mind to homeowners worldwide.

References:

1. Ashton, K. That ‘internet of things’ thing. RFID J. 2009, 22, 97–114.

2. Evans, D. The internet of everything: How more relevant and valuable connections will change the world. Cisco IBSG 2012,

2012, 1–9.

3. Kim, T.-H.; Ramos, C.; Mohammed, S. Smart City and IoT. Future Gener. Comput. Syst. 2017, 76, 159–162. [CrossRef]

4. Ara, S.S.; Shamszaman, Z.U.; Chong, I. Web-of-objects based user-centric semantic service composition methodology in the

internet of things. Int. J. Distrib. Sens. Netw. 2014, 10, 482873. [CrossRef]

5. Fielding, R.T.; Taylor, R.N. Architectural Styles and the Design of Network-Based Software Architectures, Irvine.

Doctoral Dissertation, University of California, Los Angeles, CA, USA, 2000.

6. Romero, D.; Hermosillo, G.; Taherkordi, A.; Nzekwa, R.; Rouvoy, R.; Eliassen, F. Restful integration of heterogeneous devices in

pervasive environments. In IFIP International Conference on Distributed Applications and Interoperable Systems; Springer: Berlin,

Heidelberg, 2010; pp. 1–14.

7. Bassi, A.; Bauer, M.; Kramp, M.F.T.; Kranenburg, R.; Meissner, S.L.S. Enabling Things to Talk; Springer Nature: Basingstoke,

UK, 2013.

8. Tseloni, A.; Farrell, G.; Thompson, R.; Evans, E.; Tilley, N. Domestic burglary drop and the security hypothesis. Crime Sci. 2017, 6,

1–16. [CrossRef] [PubMed]

9. Gazafroudi, A.S.; Soares, J.; Ghazvini, M.A.F.; Pinto, T.; Vale, Z.; Corchado, J.M. Stochastic interval-based optimal offering model

for residential energy management systems by household owners. Int. J. Electr. Power Energy Syst. 2019, 5, 201–219. [CrossRef]

10. Zhou, B.; Li, W.; Chan, K.W.; Cao, Y.; Kuang, Y.; Liu, X.; Wang, X. Smart home energy management systems:

Concept, configurations, and scheduling strategies. Renew. Sustain. Energy Rev. 2016, 61, 30–40. [CrossRef]

11. Sbci, U. Buildings and climate change: Summary for decision-makers. In Sustainable Buildings and Climate Initiative; United

Nations Environmental Programme: Nairobi, Kenya, 2009; pp. 1–62.

12. Birol, F. World Energy Outlook. Available online: https://www.iea.org/reports/world-energy-outlook-2010 (accessed on

10 July 2021).

13. Gill, K.; Yang, S.-H.; Yao, F.; Lu, X. A zigbee-based home automation system. IEEE Trans. Consum. Electron. 2009, 55, 422–430.

[CrossRef]

14. Al-thobaiti, B.M.; Abosolaiman, I.I.; Alzahrani, M.H.; Almalki, S.H.; Soliman, M.S. Design and implementation of a reliable

wireless Real-Time home automation system based on Arduino uno single-board microcontroller. Int. J. Control. Autom. Syst.

2014, 3, 11–15.

15. Jan, H.; Yar, H.; Iqbal, J.; Farman, H.; Khan, Z.; Koubaa, A. Raspberry Pi Assisted Safety System for Elderly People: An Application

of Smart Home. In Proceedings of the 2020 First International Conference of Smart Systems and Emerging Technologies

(SMARTTECH), Riyadh, Saudi Arabia, 3–5 November 2020; pp. 155–160.

16. Tayyaba, S.; Khan, S.A.; Ashraf, M.W.; Balas, V.E. Home Automation Using IoT. In Recent Trends and Advances in Artificial

Intelligence and Internet of Things; Balas, V., Kumar, R., Srivastava, R., Eds.; Springer International Publishing: Cham, Switzerland,

2020; Volume 172, pp. 343–388.

17. Piyare, R. Internet of things: Ubiquitous home control and monitoring system using android based smart phone.

Int. J. Internet Things 2013, 2, 5–11.

18. Piyare, R.; Lee, S.R. Smart home-control and monitoring system using smart phone. ICCA ASTL. 2013, 24, 83–86.

19. Evans, D. The internet of things: How the next evolution of the internet is changing everything. CISCO White Pap. 2011, 1, 1–11.

20. Kortuem, G.; Kawsar, F.; Sundramoorthy, V.; Fitton, D. Smart objects as building blocks for the internet of things. IEEE Internet

Comput. 2010, 14, 44–51. [CrossRef]

21. Fox, A.; Katz, R.; Konwinski, A.; Lee, G. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS2009-28, EECS Department, University of California. Available online: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009

/EECS-2009-28.html (accessed on 17 July 2021).

22. Satyanarayanan, M.; Bahl, V.; Caceres, R.; Davies, N. The Case for VM-based Cloudlets in Mobile Computing. IEEE Pervasive Comput.

2011, 8, 14–23. [CrossRef]

23. Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog Computing: A platform for internet of things and analytics. In Big Data and

Internet of Things: A Roadmap for Smart Environments; Springer: Cham, Switzerland, 2014; pp. 169–186.

24. Atat, R.; Liu, L.; Chen, H.; Wu, J.; Li, H.; Yi, Y. Enabling cyber-physical communication in 5G cellular networks: Challenges,

spatial spectrum sensing, and cyber-security. IET Cyber-Physical Syst. Theory Appl. 2017, 2, 49–54. [CrossRef]

25. Wu, J.; Guo, S.; Li, J.; Zeng, D. Big Data Meet Green Challenges: Greening Big Data. IEEE Syst. J. 2016, 10, 873–887. [CrossRef]

26. Ai, Y.; Peng, M.; Zhang, K. Edge computing technologies for Internet of Things: A primer. Digit. Commun. Networks 2018, 4, 77–86.

[CrossRef]

27. Bhoyar, M.R. Home automation system via internet using Android phone. Int. Res. Sci. Eng. 2015, 6, 15–18.

28. Adriansyah, A.; Dani, A.W. Design of Small Smart Home system based on Arduino. In Proceedings of the 2014 Electrical Power,

Electronics, Communicatons, Control and Informatics Seminar (EECCIS), Malang, Indonesia, 27–28 August 2014; pp. 121–125.

29. Baraka, K.; Ghobril, M.; Malek, S.; Kanj, R.; Kayssi, A. Low Cost Arduino/Android-Based Energy-Efficient Home Automation

System with Smart Task Scheduling. In Proceedings of the 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, 5–7 June 2013; pp. 296–301.

30. Yusuf, A.; Baba, M.A. Design and Implementation of a Home Automated System based on Arduino, Zigbee and Android. Int. J.

Comput. Appl. 2014, 97, 37–42. [CrossRef]

31. Piyare, R.; Tazil, M. Bluetooth based home automation system using cell phone. In Proceedings of the Consumer Electronics

(ISCE), 2011 IEEE 15th International Symposium, Singapore, 14–17 June 2011; pp. 192–195.

32. Pinter, J.M.; Czap, L.; Marton, L.K. Development of speech-based interface for smart home systems. In Proceedings of the 2018

19th International Carpathian Control Conference (ICCC), Szilvasvarad, Hungary, 28–31 May 2018; pp. 348–352.

33. Ravi, A.; Brindha, R.; Janani, S.; Meenatchi, S.; Prathiba, V. Smart Voice Recognition Based Home Automation System for Aging

and Disabled People. Int. J. Adv. Sci. Res. Dev. 2018, 5, 11–18.

34. Sarijari, M.A.B.; Rashid, R.A.; Rahim, M.R.A.; Mahalin, N.H. Wireless home security and automation system utilizing zigbee

based multi-hop communication. In Proceedings of the Telecommunication Technologies 2008 and 2008 2nd Malaysia Conference

on Photonics. NCTT-MCP 2008. 6th National Conference, Putrajaya, Malaysia, 26–28 August 2008; pp. 242–245.

35. Surantha, N.; Wicaksono, W.R. Design of Smart Home Security System using Object Recognition and PIR Sensor.

Procedia Comput. Sci. 2018, 135, 465–472. [CrossRef]

36. Ahanger, T.A.; Tariq, U.; Ibrahim, A.; Ullah, I.; Bouterra, Y. IoT-Inspired Framework of Intruder Detection for Smart Home

Security Systems. Electronics 2020, 9, 1361. [CrossRef]

37. Khan, Z.A.; Hussain, T.; Ullah, A.; Rho, S.; Lee, M.; Baik, S.W. Towards Efficient Electricity Forecasting in Residential and

Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework. Sensors 2020, 20, 1399. [CrossRef]

38. Khan, Z.A.; Ullah, A.; Ullah, W.; Rho, S.; Lee, M.; Baik, S.W. Electrical Energy Prediction in Residential Buildings for Short-Term

Horizons Using Hybrid Deep Learning Strategy. Appl. Sci. 2020, 10, 8634. [CrossRef]

39. Darby, S.J. Smart technology in the home: Time for more clarity. Build. Res. Inf. 2018, 46, 140–147. [CrossRef]

40. Australian Energy Market Commission (AEMC). Power of Choice Review-Giving Consumers Options in the Way They Use

Electricity. Final Report, Australian Energy Market Commission, 30 November, 2012, Sydney. Available online: https://www.

aemc.gov.au/sites/default/files/content/2b566f4a-3c27-4b9d-9ddb-1652a691d469/Final-report.pdf (accessed on 17 July 2021).

41. Han, J.; Choi, C.-S.; Park, W.-K.; Lee, I.; Kim, S.-H. Smart home energy management system including renewable energy based on

ZigBee and PLC. IEEE Trans. Consum. Electron. 2014, 60, 198–202. [CrossRef]

42. Anvari-Moghaddam, A.; Monsef, H.; Rahimi-Kian, A. Optimal smart home energy management considering energy saving and a

comfortable lifestyle. IEEE Trans. Smart Grid 2015, 6, 324–332. [CrossRef]

43. Ringel, M.; Laidi, R.; Djenouri, D. Multiple benefits through smart home energy management solutions—A simulation-based

case study of a single-family-house in algeria and Germany. Energies 2019, 12, 1537. [CrossRef]

44. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; et al. A view

of cloud computing. Commun. ACM 2010, 53, 50–58. [CrossRef]

45. Li, J.; Jin, J.; Yuan, D.; Palaniswami, M.; Moessner, K. EHOPES: Data-centered Fog platform for smart living. In Proceedings of the

2015 International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 18–20 November

2015; pp. 308–313.

46. Al Faruque, M.A.; Vatanparvar, K. Energy management-as-a-service over fog computing platform. IEEE Internet Things J. 2016, 3,

161–169. [CrossRef]

47. Perera, C.; Qin, Y.; Estrella, J.C.; Reiff-Marganiec, S.; Vasilakos, A.V. Fog computing for sustainable smart cities: A survey.

ACM Comput. Surv. (CSUR) 2017, 50, 1–43. [CrossRef]

48. Yi, S.; Qin, Z.; Li, Q. Security and Privacy Issues of Fog Computing: A Survey. In International Conference on Wireless Algorithms,

Systems, and Applications; Springer: Berlin/Heidelberg, Germany, 2015; pp. 685–695.

49. Lu, R.; Heung, K.; Lashkari, A.H.; Ghorbani, A.A. A lightweight privacy-preserving data aggregation scheme for fog computingenhanced IoT. IEEE Access 2017, 5, 3302–3312. [CrossRef]

50. Amadeo, M.; Giordano, A.; Mastroianni, C.; Molinaro, A. On the Integration of Information Centric Networking and Fog

Computing for Smart Home Services. In The Internet of Things for Smart Urban Ecosystems; Springer: Berlin/Heidelberg, Germany, 2019; pp. 75–93.

51. Fielding, R.T.; Taylor, R.N. Principled design of the modern Web architecture. ACM Trans. Internet Technol. (TOIT) 2002, 2, 115–150.

[CrossRef]

52. Thangavel, D.; Ma, X.; Valera, A.; Tan, H.-X.; Tan, C.K.-Y. Performance evaluation of MQTT and CoAP via a common middleware.

In Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information

Processing (ISSNIP), Singapore, 21–24 April 2014; pp. 1–6.

53. Van Vinh, P.; Dung, P.X.; Tien, P.T.; Hang, T.T.T.; Duc, T.H.; Nhat, T.D. Smart Home Security System Using Biometric Recognition.

In International Conference on Internet of Things as a Service; Springer: Berlin/Heidelberg, Germany, 2021; pp. 405–420.

54. Froiz-Míguez, I.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. Design, implementation and practical evaluation of

an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes. Sensors 2018,

18, 2660. [CrossRef] [PubMed]

55. Nedelcu, A.-V.; Sandu, F.; Machedon-Pisu, M.; Alexandru, M.; Ogrutan, P. Wireless-based remote monitoring and control of

intelligent buildings. In Proceedings of the 2009 IEEE International Workshop on Robotic and Sensors Environments 2009,

Lecco, Italy, 6–7 November 2009; pp. 47–52

Submission 1	
Title	Smartify: Transforming Traditional Appliances into Smart Devices For Seamless Home Automation
Paper:	(Apr 14, 21:09 GMT)
Author keywords	Smart Home Automation WI-FI Control ESP32 Firebase Realtime Database Mobile app Retrofitting Internet Of Things(IOT) Infrared (IR) Remote
Abstract	In the realm of home automation, efficient control systems are pivotal for a seamless user experience. This paper presents a novel approach utilizing Internet of Things (IoT) technology, specifically employing the ESP8266 and ESP32 microcontrollers, to develop a smart home automation system. Leveraging Firebase Realtime Database (RTDB) for data storage and retrieval, coupled with Infrared (IR) remote control functionality, this system enables remote monitoring and control of household appliances. The system architecture integrates WiFi connectivity for seamless communication between the microcontroller and the Firebase RTDB, ensuring real-time synchronization of device states. Additionally, IR remote functionality enhances user interaction by allowing direct control of appliances. The implementation encompasses robust error-handling mechanisms to ensure reliability and consistency in data transmission. Key features include dynamic relay state management, achieved through IR remote commands or physical switches, and seamless integration with Firebase for persistent storage and retrieval of device states. Moreover, the system incorporates token-based authentication for secure access, enhancing privacy and data integrity. Experimental results demonstrate the efficacy of the proposed system in providing reliable, real-time control of household appliances, thereby showcasing its potential for enhancing user convenience and energy efficiency in smart home environments
Submitted	Apr 14, 21:09 GMT
Last update	

Authors						
first name	last name	email	country	affiliation	Web page	corresponding?
Shiva Kumar Reddy	Jakka	jakkashivakumarreddy8@gmail.com	India	Malla Reddy University		✓
Vishnu Vardhana Raju	Ch	2011CS040018@mallareddyuniversity.ac.in	India	Malla Reddy University		✓
Akshaya	Diddi	2011CS040019@mallareddyuniversity.ac.in	India	Malla Reddy University		✓

Publication Link :

https://easychair.org/conferences/submission?a=32645978&submission=6875324

Web Link of the project:

https://github.com/iamlearner1/Batch7MajorProject