SMART HOME AUTOMATION USING RASPBERRY PI

Submitted by

AYUSHI GUPTA [Reg No:RA2111003010662]

JOITA GHOSH [Reg No:RA2111003010664]

VISHAL KHUMAR P D [Reg No:RA2111003010667]

Under the Guidance of

MR. M. ARUL PRAKASH

Assistant Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
of
FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603203

MAY 2023

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled **"SMART HOME AUTOMATION USING RASPBERRY PI"** is the bonafide work done by **AYUSHI GUPTA [RA2111003010662], JOITA GHOSH [RA2111003010664] and VISHAL KHUMAR P D [RA2111003010667]** who carried out under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

M. Pushpalatha

SIGNATURE

Faculty In-Charge
Mr.M.Arul Prakash
Assistant Professor
Department of Computing Technologies
SRM Institute of Science and Technology

SIGNATURE

HEAD OF THE DEPARTMENT
Dr. M. Pushpalatha
Professor and Head,
Department of Computing Technologies
SRM Institute of Science and Technology

ABSTRACT

This report focuses on the implementation of smart home automation using Raspberry Pi, a low-cost, single-board computer. The project begins with the identification of the software project and creating a business case for it. A problem statement is formulated based on the identified process methodology and stakeholder description. The project's system, functional, and non-functional requirements are defined in detail. A project plan is prepared based on the scope, and project effort is calculated based on resources and job roles and responsibilities. A work breakdown structure and timeline chart are also prepared, and risks are identified and documented in a risk identification table. The system architecture is designed, and use case and class diagrams are created. An entity relationship diagram is designed to capture the relationships between entities in the system. A data flow diagram up to level 1 is developed to show how data flows within the system. Sequence and collaboration diagrams are also created to show the interaction between system components. A testing framework and user interface are developed, and test cases are written and executed to ensure the system meets the requirements. Manual test case reporting is done to document the results of the testing. Finally, the details of the architecture design, framework, and implementation are provided, including the technologies used, programming languages, and libraries. Overall, the project successfully implements smart home automation using Raspberry Pi, demonstrating the potential for low-cost, effective home automation solutions.

LIST OF EXPERIMENTS

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	1
Title of Experiment	To identify the Software Project, Create Business Case, Arrive at a Problem Statement
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	27-01-2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

AIM:

To Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the **SMART HOME AUTOMATION USING RASPBERRY PI** project.

TEAM MEMBERS:

S. No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	Lead/Rep
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

PROJECT TITLE:

SMART HOME AUTOMATION USING RASPBERRY PI

PROJECT DESCRIPTION:

Smart home automation using Raspberry Pi is a technology that allows users to control various devices in their homes using a central hub. The Raspberry Pi is programmed to receive and interpret commands from various input sources, such as smartphones, voice assistants, or web interfaces. It can then perform actions on connected devices, such as turning on lights, adjusting the temperature, or controlling the security system. To connect the devices in a home to the Raspberry Pi, various wired or wireless technologies may be used, such as Wi-Fi, Bluetooth, or Zigbee. The communication between the Raspberry Pi and the connected devices is managed through the use of software programs and APIs. These programs allow the Raspberry Pi to receive commands from the input sources and to send commands to the connected devices.

Smart home automation using Raspberry Pi offers many benefits, including increased convenience, improved security, and reduced energy consumption. Additionally, the Raspberry Pi can be programmed to detect motion in the home and to trigger actions, such as turning on the lights or sounding an alarm. Overall, smart home automation using Raspberry Pi provides an easy and cost-effective way to make any home a smart home.

BUSINESS CASE

DATE	27-01-2023
SUBMITTED BY	VISHAL KHUMAR P D
TITLE / ROLE	LEADER

THE PROJECT

The principle of smart home automation using Raspberry Pi is based on using the Raspberry Pi as a central hub to control and automate various devices in a home. The Raspberry Pi is programmed to receive and interpret commands from various input sources (such as smartphones, voice assistants, etc.) and to perform actions on connected devices (such as lights, fans, etc.) based on those commands.

The devices are connected to the Raspberry Pi through wired or wireless means, such as Wi-Fi or Bluetooth, and communication between the devices and the Raspberry Pi is managed through the use of software programs and APIs. The result is a fully automated and controllable smart home environment.

THE HISTORY

The current situation of smart home automation using Raspberry Pi is promising. Raspberry Pi, being a low-cost, compact, and powerful device, has made it possible for DIY enthusiasts and hobbyists to easily create custom smart home automation systems. Raspberry Pi is compatible with various sensors, actuators, and other devices that can be used to control lighting, heating, cooling, and other household appliances. The open-source nature of Raspberry Pi has also led to the development of many software packages and libraries that make it easier to build and manage smart home systems. Additionally, the large community of Raspberry Pi users and developers has made it easier to find solutions and tutorials for building smart home systems.

ESTIMATED COST

- TP-Link Smart Wi-Fi LED Bulb: 3,000 rupees
- Raspberry Pi Official USB Bluetooth Adapter: 980 rupees
- Amazon Echo Dot: 5,000 rupees
- PIR Motion Sensor: 1,300 rupees
- DHT11 Temperature and Humidity Sensor: 950 rupees
- SONOFF Basic R3 Smart Switch: 2,000 rupees
- 840-Point Breadboard: 300 rupees
- SanDisk Ultra microSDXC UHS-I: 1,300 rupees
- TP-Link Smart Plug Mini: 2,000 rupees
- TP-Link Archer C1200 Dual Band Gigabit Router: 5,000 rupees
- 0.96 Inch 128 x 64 Pixel OLED Display: 2,000 rupees
- 36kHz IR Infrared Remote Control Transmitter Module: 1,000 rupees
- Raspberry Pi 4 Case: 1,200 rupees

TOTAL COST: 26,030 rupees(approx.)

APPROACH

1. **Raspberry Pi:** Raspberry Pi 4 Model B with 4GB RAM
2. **Power supply:** Raspberry Pi Official Power Supply 5.1V 3A USB-C
3. **Connected devices:** TP-Link Smart Wi-Fi LED Bulb
4. **Wired or wireless connections:** Raspberry Pi Official USB Bluetooth Adapter
5. **Software and libraries:** Python, Node-RED
6. **Input devices:** Amazon Echo Dot
7. **Sensors:** PIR Motion Sensor, or DHT11 Temperature and Humidity Sensor
8. **Relays or controllers:** SONOFF Basic R3 Smart Switch
9. **Breadboard or circuit board:** 840-Point Breadboard
10. **SD card:** SanDisk Ultra microSDXC UHS-I
11. **Smart plug:** TP-Link Smart Plug Mini
12. **Wi-Fi Router:** TP-Link Archer C1200 Dual Band Gigabit Router
13. **OLED Display:** 0.96 Inch 128 x 64 Pixel OLED Display
14. **IR Blaster:** 36kHz IR Infrared Remote Control Transmitter Module
15. **Cases:** Raspberry Pi 4 Case

BENEFITS

- This system is designed to be low cost, low energy and expandable allowing a variety of devices to be controlled.
- Saves money and energy and all in one user friendly system
- Remote control of home functions.
- Increased energy efficiency.
- Improved appliance functionality. Smart homes can also help you run your appliances better.

LIMITATIONS

- High Cost
- Not user friendly
- The basic requirement for the smart home system is the internet. So there will high dependency on internet.
- All the smart home products must be protected with strong and unbreakable passwords, else it can be hacked.

RESULT:

Thus, the project team formed, the project is described, the business case was prepared and the problem statement was arrived.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	2
Title of Experiment	Identification of Process Methodology and Stakeholder Description
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	03-02-2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Team Members:

Sl No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	**Rep/Member**
2	RA2111003010662	AYUSHI GUPTA	**Member**
3	RA2111003010664	JOITA GHOSH	**Member**

Project Title:

SMART HOME AUTOMATION USING RASPBERRY PI

Selection of Methodology: The Waterfall Model

The Waterfall model is a linear sequential software development life cycle (SDLC) model, which means that each phase must be completed before the next phase can begin and there is no overlapping in the phases. It is named so because the phases flow downwards like a waterfall.

The phases of the Waterfall model are:

- Requirements Gathering and Analysis: This phase involves gathering and documenting the requirements for the software.
- Design: In this phase, the software design is created based on the requirements gathered in the previous phase. This includes creating system architecture, interface design, and database design.
- Implementation or Coding: In this phase, the actual coding of the software takes place. The design created in the previous phase is used as a reference while coding.
- Testing: In this phase, the software is tested to make sure it meets the requirements and is free of bugs.
- Deployment: In this phase, the software is deployed for use in a live environment.
- Maintenance: This phase involves providing support and fixing any bugs that may be discovered after deployment.
- The Waterfall model is best suited for projects with well-defined and fixed requirements and a limited scope. It is not suitable for projects where requirements are rapidly changing, or the scope is unclear.

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
Internet service provider	provides mode of connectivity	Low	High	14
Electricians	provides electrical support, accountable for inter-connections between the various devices in the system	Low	High	13
IT support companies	ensures proper connectivity between devices and the network and setting up of the automation in homes	Low	High	12
Homeowner	achieve targets, increase sales margin	High	High	1
Home security companies	provides safety and security methods	Low	High	11
Home automation retailers	provides the core software	High	High	10
Hardware distributors	provides the necessary hardware elements	Low	High	8
Smart devices manufacturers	provides the necessary components to make the smart devices including Raspberry Pi	Low	High	7
Software developers	accountable for maintaining the entire framework of the automaton	Low	Low	9
Data privacy and security companies	accountable for keeping consumer confidentiality and ensuring safe working practices	High	High	15
Ownership	Exceeds goal, improves sales margin	Low	High	6
Sponsors	Negotiate funding for projects that exceed targets and	Low	High	3

	improve sales margins. Investigate changes in the project environment. Team members should receive rewards. Renew and maintain enthusiasm for new products.			
Project Manager	Provides leadership for the entire group. Take responsibility for the project, the team, and the overall scope for its success or failure.	Low	High	5
Investors	Promoters of investments. provide the necessary financial resources.	Low	High	2
Resource manager	Resource Planning and Allocation. Ensure appropriate resource allocation according to project needs and budget.	Low	Low	4
Suppliers	Make sure everything is possible and realistic. Manage cost variances from budget.	Low	High	16
End User	Provide Recommendations	High	Low	17

Result

Thus, the Project Methodology was identified and the stakeholders were described.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	3
Title of Experiment	System, Functional and Non-Functional Requirements of the Project
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	27-01-2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To identify the system, functional and non-functional requirements for the project.

Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	**Rep/Member**
2	RA2111003010662	AYUSHI GUPTA	**Member**
3	RA2111003010664	JOITA GHOSH	**Member**

Project Title:

SMART HOME AUTOMATION USING RASPBERRY PI

System Requirements:

- **Raspberry Pi:** Raspberry Pi computer with sufficient processing power is needed to handle the smart home automation tasks. Raspberry Pi 3 or later is recommended.
- **Operating System:** An operating system on the Raspberry Pi is required. Raspbian is the recommended operating system for Raspberry Pi.
- **Power Supply:** It is recommended to use a power supply with at least 2.5A output.
- **MicroSD Card:** microSD card is needed to store the operating system and data.
- **Sensors:** Temperature, humidity, motion, and light sensors.
- **Actuators:** Actuators such as relays, motors, and servos to control devices such as lights, fans, and doors.
- **Home Automation Software:** Home automation software such as Home Assistant, OpenHAB, or Node-RED to control the sensors and actuators.
- **Wi-Fi Adapter:** A Wi-Fi adapter to connect the Raspberry Pi to the internet.

Functional Requirements:

- **Control of smart devices:** Smart home automation using Raspberry Pi should allow users to control smart devices such as lights, thermostats, locks, cameras, and other devices.
- **Voice Control:** Integrating voice assistants such as Amazon Alexa, Google Assistant or Siri can give users the ability to control their smart home devices with voice commands.

- **Sensor Integration:** Raspberry Pi can be used to integrate various sensors like motion sensors, temperature sensors, humidity sensors, and so on, which can help automate certain functions.
- **Customization:** Users should have the ability to customize their automation system based on their preferences, including creating routines or rules to automate tasks, setting schedules, and creating custom notifications.
- **Security:** The automation system should provide security features such as password protection, data encryption, and secure communication protocols to protect user data and prevent unauthorized access.
- **Remote Access**: Users should be able to control their smart home automation system remotely, whether via a smartphone app, a web browser, or another device.
- **Expandability:** The system should be scalable and easy to expand as the user adds new smart devices or features.
- **Energy efficiency:** Smart home automation using Raspberry Pi can help users save energy by controlling devices automatically, setting schedules, or monitoring energy usage.
- **Integration with other systems:** The system should be able to integrate with other home automation systems and protocols, such as Zigbee or Z-Wave, to ensure interoperability.
- **User-friendly Interface:** A user-friendly interface for configuring the automation system and monitoring device activity is a must for smart home automation.

Non-Functional Requirements:

- **Performance and scalability:**. Smart home automation has high quality and fast performance provided the devices connected to the software works perfectly and frequent servicing is conducted. when a command is given to switch off a particular light bulb in a particular room, it should be executed within 2 secs.
- **Portability and compatibility**: It runs on the principle of IoT(internet of things) and raspberry pi software
- **Reliability, maintainability, availability:** The system can face a complete shutdown if the internet connection is not provided and can work very slowly if the connection isn't strong enough. internet service providers should come for maintenance check in every 2 months while other appliances such as the fans and lights should be serviced by the electrician once in every 1 month. If such conditions are followed, the automation system is very reliable.

- **Security:** a smart home technology is based completely on its software and codes it is written upon. Just like any code,it is susceptible to hackers and viruses. It is the responsibility of the developers and the software company to provide in-built antivirus and to develop strong and efficient protection against possible malware.
- **Usability:** The technology is extremely easy and convenient to use as it is connected to an application software in our mobile device and it can simply be operated like any other app through the touch of our fingers.

Result:

Thus the requirements were identified and accordingly described.

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	4
Title of Experiment	Prepare Project Plan based on scope, Calculate Project effort based on resources and Job roles and responsibilities for Smart Home Automation using Raspberry Pi
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	20/02/2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim:

To Prepare Project Plan based on scope, Calculate Project effort based on resources, Find Job roles and responsibilities for Smart Home Automation using Raspberry Pi

Team Members:

Sl No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	Lead
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

Requirements:

1. Project Management Plan

Describe the key issues driving the project. **[Min 3 Focus Areas]**

Focus Area	Details
Integration Management	Identify the components that need to be integrated, such as sensors, controllers, actuators, and user interfaces. Choose hardware and software components that are compatible with each other and can work together seamlessly. Develop a detailed project plan that includes tasks, timelines, resources, and deliverables, as well as dependencies between various components.
Scope Management	Scope management for smart home automation using Raspberry Pi involves defining and controlling the project's boundaries to ensure that it stays within the defined scope. Effective scope management ensures that the project is completed on time, within budget, and to the satisfaction of all stakeholders. It also helps to minimize risks and ensure that the project meets its goals and objectives.
Schedule Management	Schedule management for smart home automation using Raspberry Pi involves creating a project schedule that outlines the tasks, timelines, and resources required for each phase of the project. It is also important to monitor the schedule regularly and adjust it as necessary to ensure that the project stays on track. Using tools such as Gantt charts can help to visualize the project schedule and identify critical paths and potential bottlenecks.

Cost Management	Cost management for smart home automation using Raspberry Pi involves keeping track of expenses related to hardware, software, and installation. To manage costs effectively, it is essential to create a budget and stick to it. This can be achieved by comparing prices, looking for discounts or sales, and selecting cost-effective components that meet project requirements.
Quality Management	Quality assurance for smart home automation using Raspberry Pi involves ensuring that the system functions as intended, meets the project requirements, and is reliable and secure. This includes testing the system thoroughly during development and before deployment, identifying and addressing any defects or issues, and verifying that the system meets the expected performance and functional requirements.
Resource Management	Smart home automation using Raspberry Pi requires a team of experts including a software developer, a hardware engineer, and an IoT specialist. The physical requirements include a Raspberry Pi board, sensors, and other components depending on the desired functionality. The finance required depends on the complexity of the project and the number of sensors used. Resource management involves identifying the required resources, estimating the cost, and scheduling the activities.
Communication Management	Smart home automation using Raspberry Pi requires effective communication management to ensure smooth operation of devices and systems. Communication between the Raspberry Pi and other devices can be achieved through various protocols such as Wi-Fi, Bluetooth, and Zigbee.
Risk Management	Smart home automation using Raspberry Pi can present security risks such as unauthorized access, data breaches, and system malfunctions. To manage these risks, it is essential to implement security measures such as strong passwords, regular software updates, and secure network configuration. It is also important to limit the use of third-party applications, enable encryption for sensitive data, and monitor the system for any suspicious activity.
Procurement Management	Procurement management of smart home automation using Raspberry Pi involves identifying the necessary hardware and software components, selecting reliable suppliers, and ensuring timely delivery of the required items. The procurement process also includes negotiating prices, evaluating quality, and managing inventory.

2. Estimation

2.1. Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the user screen	E1R1A1T1 (Effort-Requirement-Activity-Task)	Confirm the user requirements (acceptance criteria)	10	5000
	E1R1A1T2 (Design and Architecture)	Design System Architecture and Hardware Layout	20	10000
	E1R1A1T3 (Software Development)	Develop software for Raspberry Pi and sensors	50	25000
Identify Data Source for displaying units of Energy Consumption	E1R1A1T4 (Integration and Testing)	Integrate hardware and software and test system	30	15000
	E1R1A1T5 (Deployment and Maintenance)	Deploy system in user's home and provide maintenance support	20	10000

Effort (hr)	Cost (INR)
1	500

2.2. Infrastructure/Resource Cost [CapEx]

< OneTime Infra requirements >

Infrastructure Requirement	Qty	Cost per qty	Cost per item
IR1 Raspberry Pi Board	1	3000	3000
IR2 Sensors	10	500	5000
IR3 Micro SD card	1	1000	1000
IR4 Power Supply	1	500	500
IR5 Cables and Connectors	10	50	500

2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin Developer , Support Consultant	3	4,000,000	8,000,000
License	Operating System Database Middleware IDE	10	20000	300,000
Infrastructures	Server, Storage and Network	20	50000	900,000

3. Project Team Formation

3.1. Identification Team members

Name	Role	Responsibilities
Vishal Khumar	Key Business User (Product Owner)	Provide clear business and user requirements
Joita Ghosh	Project Manager	Manage the project
Ayushi Gupta	Business Analyst	Discuss and Document Requirements
Kingsley	Technical Lead	Design the end-to-end architecture
Rahul	UX Designer	Design the user experience
Ayushi Gupta	Frontend Developer	Develop user interface
Joita Ghosh	Backend Developer	Design, Develop and Unit Test Services/API/DB
Sharan	Cloud Architect	Design the cost effective, highly available and scalable architecture
Ajay	Cloud Operations	Provision required Services
Ganesh	Tester	Define Test Cases and Perform Testing

3.2. Responsibility AssignmentMatrix

RACI Matrix	Team Members			
Activity	**Name (BA)**	**Name (Developer)**	**Name (Project Manager)**	**Key Business User**
User Requirement Documentation	A	C/I	I	R
Design and Architecture	A	R	I	C
Hardware Selection	C	R	A	I
Software Development	C	A	R	I
Integration and Testing	C	R	A	I
Deployment and Maintenance	C	R	A	I

A	Accountable
R	Responsible
C	Consult
I	Inform

Reference

1. https://www.pmi.org/
2. https://www.projectmanagement.com/
3. https://www.tpsgc-pwgsc.gc.ca/biens-property/sngp-npms/ti-it/ervcpgpm-dsfvpmpt-eng.html

Result:

Thus, the Project Plan was documented successfully.

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	5
Title of Experiment	Prepare Work breakdown structure, Timeline chart, Risk identification table
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	27-02-2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

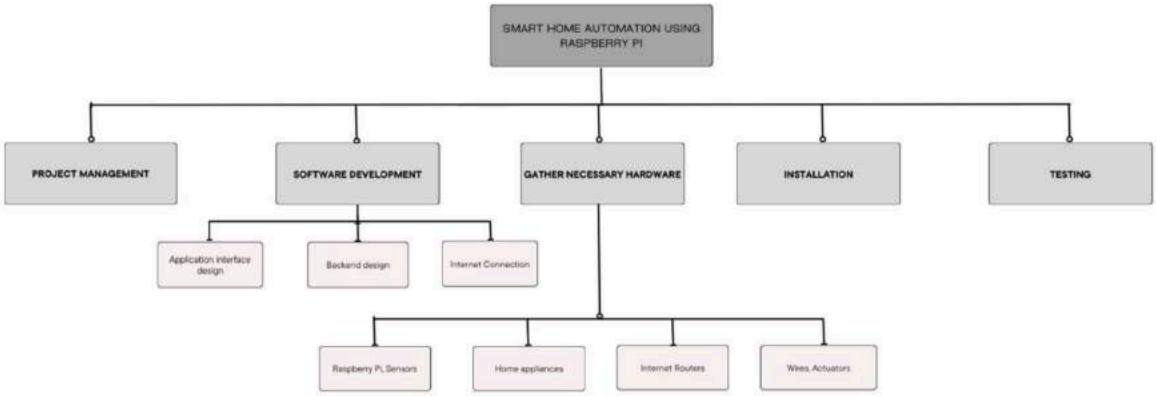
Aim:

To Prepare Work breakdown structure, Timeline chart and Risk identification table for Smart Home Automation using Raspberry Pi

Team Members:

Sl No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	Rep
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

Work Breakdown Structure:



- Smart Home Automation using Raspberry Pi
- Project Management
- 2.0 Software Development
  - 2.1 Application Interface Design
  - 2.2 Backend Design
  - 2.3 Internet Connection
- 3.0 Gather Necessary Hardware
  - 3.1 Raspberry Pi, Sensors
  - 3.2 Home Appliances
  - 3.3 Internet Routers
  - 3.4 Wires, Actuators
- 4.0 Installation
- 5.0 Testing

Timeline Chart:

Task	Start Date	End Date	Duration	Description
Research and Planning	07-03-2023	21-03-2023	2 weeks	Involves researching Smart Home Automation systems and components, determining the project's scope, identifying required components, planning the system layout and design, creating a budget and timeline, and setting project milestones.
Raspberry Pi Set-Up	22-03-2023	28-03-2023	1 week	Involves purchasing and assembling the Raspberry Pi components, installing and configuring the operating system and necessary software, connecting the Raspberry Pi to the internet, and testing its functionality.
Sensor and Actuator Integration	29-03-2023	11-04-2023	2 weeks	Involves purchasing and integrating the necessary sensors and actuators, developing code to interface the sensors and actuators with the Raspberry Pi, testing and debugging the sensor and actuator system, and setting up communication protocols.
Smart Home Automation System Development	12-04-2023	02-05-2023	3 weeks	Involves developing the software to control and monitor the Smart Home Automation system, creating user interfaces for controlling and monitoring the system, integrating the sensor and actuator system with the Smart Home Automation software, testing and debugging the Smart Home Automation system, and creating documentation.
Deployment	03-05-2023	09-05-2023	1 week	Involves installing the Smart Home Automation system in the home, configuring the system for the specific home layout and needs, testing the system, and making any necessary adjustments.
Maintenance	10-05-2023	31-07-2023	Ongoing	Involves monitoring the Smart Home Automation system for errors and malfunctions, performing regular maintenance and updates, expanding or modifying the system as needed, and providing technical support to users.

Risk Identification table:

SWOT Analysis:

Strengths:

- Cost-effective: Using Raspberry Pi as the brain of the Smart Home Automation system can significantly reduce costs compared to other automation systems.
- Customizable: The open-source nature of Raspberry Pi allows for easy customization and modification of the Smart Home Automation system.
- User-friendly: The system can be controlled and monitored via a user-friendly interface, which makes it accessible to a wide range of users.
- Energy-efficient: The Smart Home Automation system can be programmed to optimize energy usage and reduce energy costs.

Weaknesses:

- Technical expertise required: Setting up and configuring the Raspberry Pi and sensors and actuators may require technical expertise, which may be a barrier to entry for some users.
- Limited processing power: Raspberry Pi has limited processing power compared to more powerful computers, which may limit the functionality and performance of the Smart Home Automation system.
- Security concerns: The Smart Home Automation system may be vulnerable to cyber attacks, and security measures must be taken to protect the system and user data.

Opportunities:

- Growing market: The demand for Smart Home Automation systems is growing, and there is an opportunity for businesses to enter the market with cost-effective solutions.
- Integration with other systems: The Smart Home Automation system can be integrated with other home automation systems and devices, such as voice assistants and smart thermostats, to create a more comprehensive and efficient system.
- Collaboration with other industries: The Smart Home Automation system can be integrated with other industries, such as renewable energy, to create a more sustainable and eco-friendly home.

Threats

- Competition: There is competition in the Smart Home Automation market, and other companies may offer more advanced and sophisticated systems.
- Technological advancements: Technological advancements may quickly make the Raspberry Pi obsolete, and the Smart Home Automation system may need to be updated or replaced.
- Regulatory changes: Regulatory changes related to data privacy and cybersecurity may require additional measures and investments to ensure compliance.

Risk Mitigation, Monitoring, and Management (RMMM):

Response	Strategy	Examples
AVOID	Risk avoidance means trying to avoid compromising events as a way to eliminate liability exposures.	• Unauthorized access: Unauthorized access to the smart home automation system can result in theft, misuse of personal data, or even physical harm to individuals in the home. Please make sure that access to the system is limited to authorized users only and use strong passwords to prevent unauthorized access. • Data breaches: The smart home automation system may collect and store sensitive personal data, such as names, addresses, and financial information. Protect data by encrypting it both in transit and at rest, and use reputable vendors for all components. • Malware and viruses: Malware and viruses can be introduced into the system through various sources, including email attachments or downloads. Install security software and regularly update software to prevent malware and virus infections.
ASSESS	A systematic process of evaluating the potential risks that may be involved in a projected activity or undertaking	• Assess the likelihood and potential impact of each identified risk. Determine the level of risk and the potential consequences of the risk occurring. • Develop a risk management plan to address identified risks. This may involve implementing safeguards or controls to reduce the likelihood of risks occurring or mitigating the potential impact of risks if they do occur.
MONITOR AND REVIEW	Risk monitoring is the process that tracks and evaluates the levels of risk in an organization.	Continuously monitor the system and review the risk management plan to ensure that it remains effective and relevant over time. Be prepared to make adjustments to the plan as necessary to address new or emerging risks.
MITIGATE	Risk mitigation is a strategy to prepare for and lessen the effects of threats faced by a business.	• Use secure passwords for all devices and accounts. • Update software regularly to ensure that security patches are applied. • Enable encryption to secure data in transit and at rest. • Implement firewalls and other security measures to protect the network. • Use reputable vendors and suppliers for all components. • Limit access to the system and data to authorized users only.

		▯ Regularly backup system data to prevent loss of important information.

Result:

Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	6
Title of Experiment	Design a System Architecture, Use Case and Class Diagram
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	13-03-2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

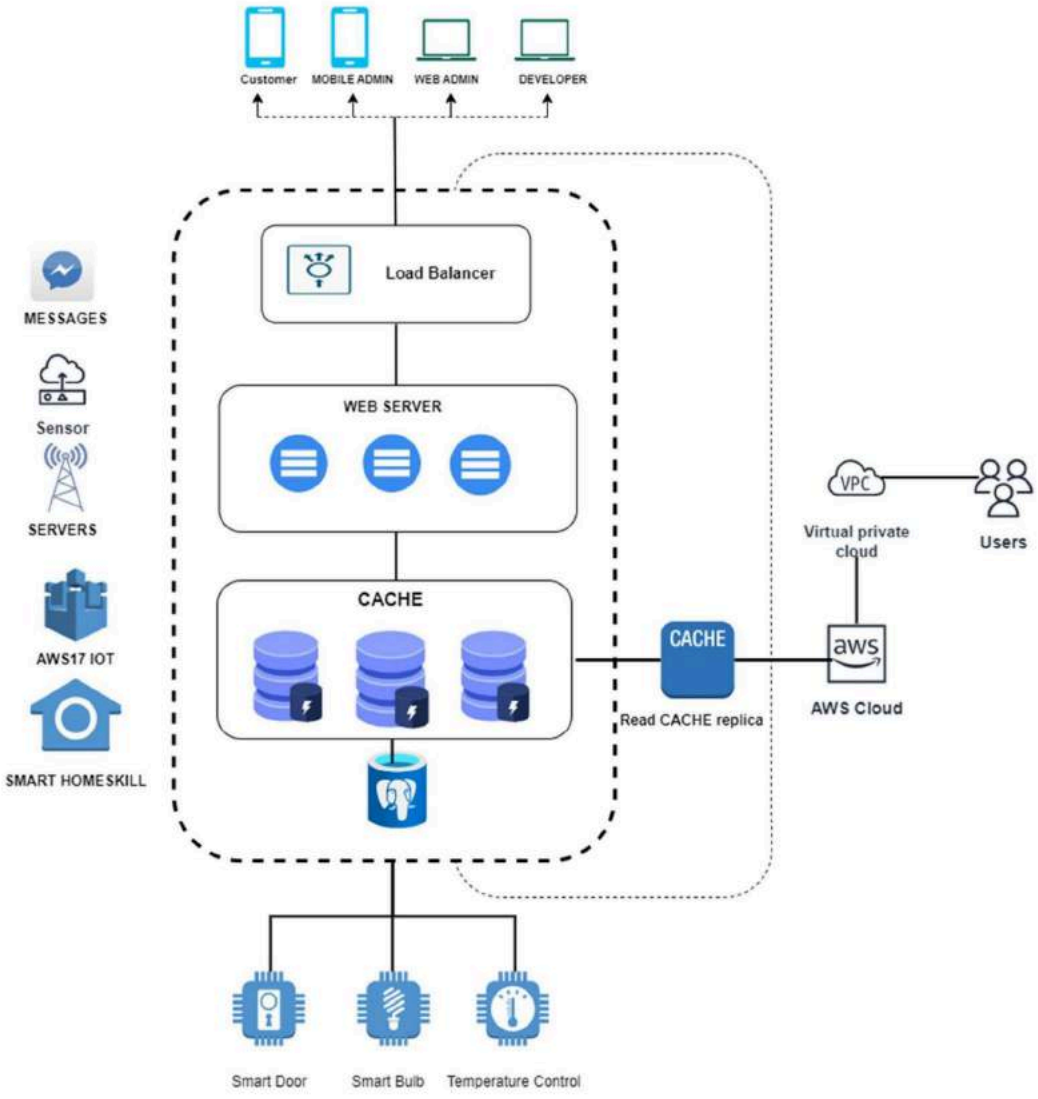
Aim

To Design a System Architecture, Use case and Class Diagram

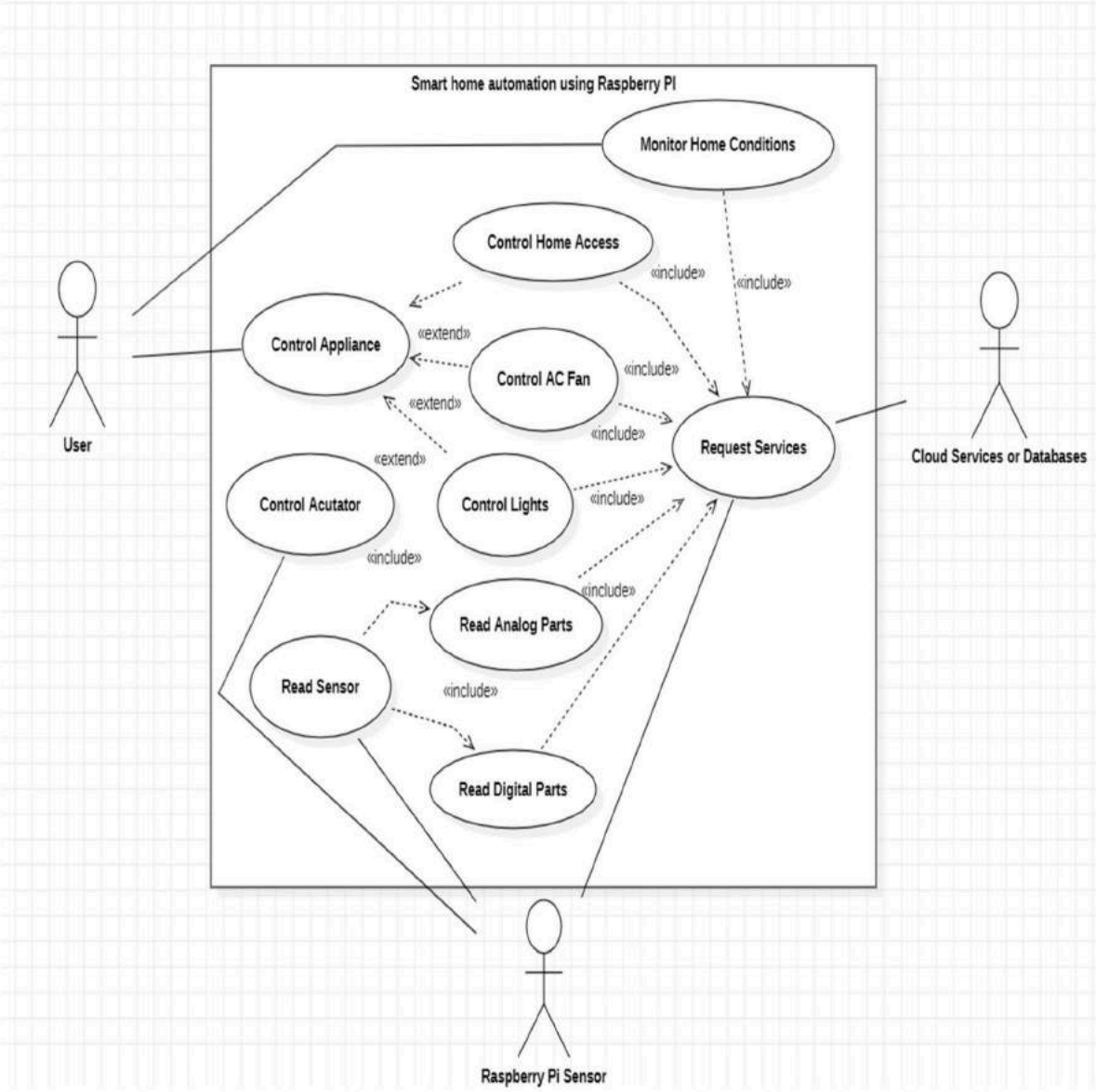
Team Members:

Sl No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	Rep
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

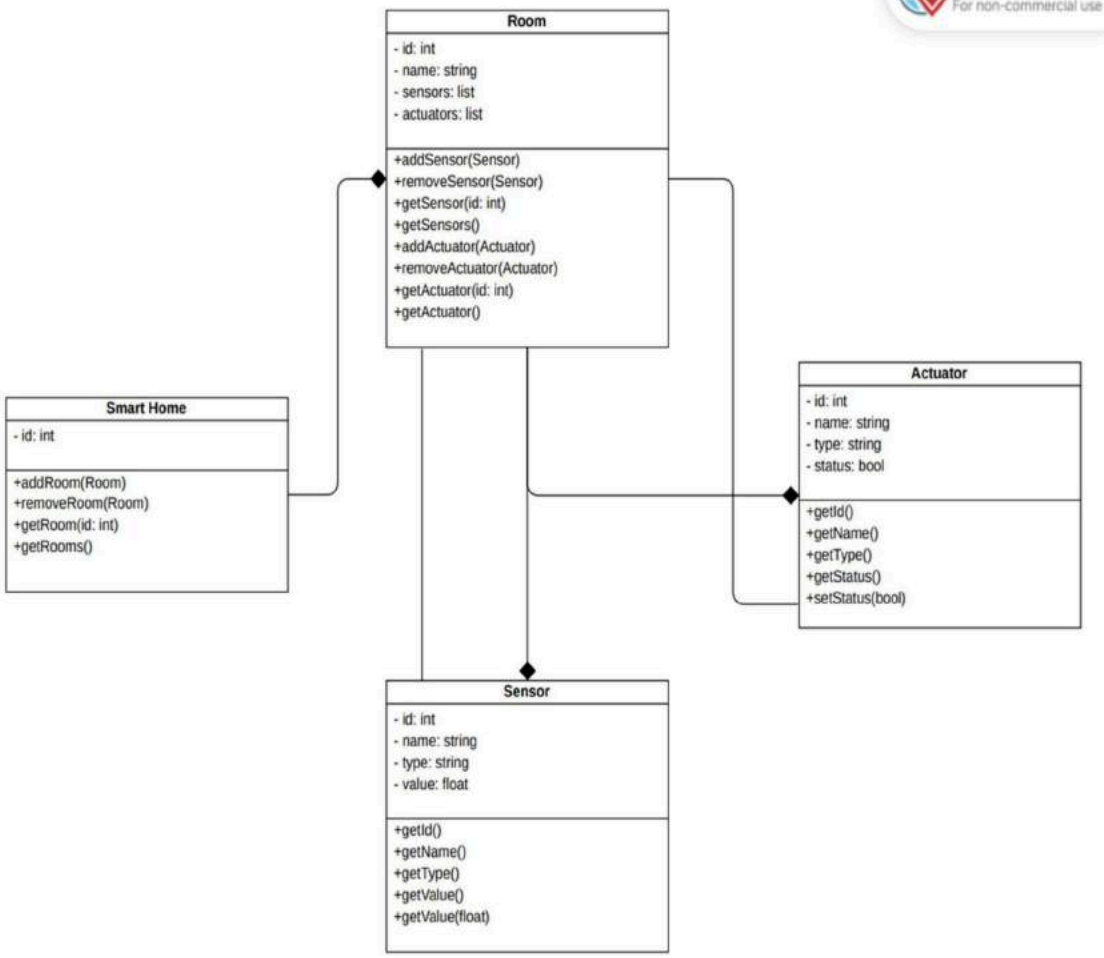
SYSTEM ARCHITECTURE



USE CASE DIAGRAM



CLASS DIAGRAM



Result:

Thus, the system architecture, use case and class diagram created successfully.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	7
Title of Experiment	DESIGN AN ENTITY RELATIONSHIP DIAGRAM
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	13/03/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

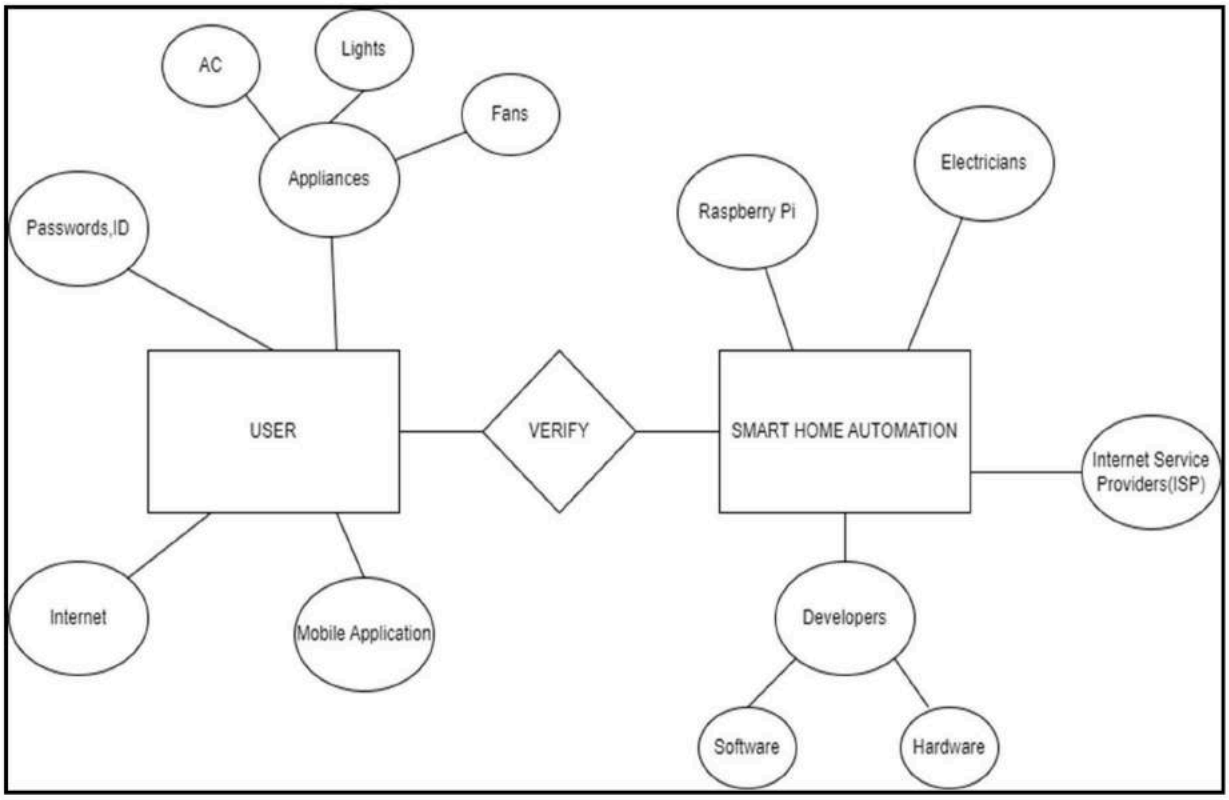
Aim:

To create the Entity Relationship Diagram

Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR PD	Rep
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

ER DIAGRAM :



An Entity-Relationship (ER) diagram is a visual representation of the relationships between entities in a system. In the context of a smart home automation system using Raspberry Pi, the components that can be represented in an ER diagram include:

1. **User:** This is the person who interacts with the smart home automation system. The user can use a mobile application to control the appliances in their home.

2. **Mobile application:** The mobile application is the interface through which the user can control the smart home automation system. The mobile application can be developed by software developers.

3. **Passwords:** To ensure the security of the smart home automation system, passwords are required to authenticate the user before they can access the mobile application.

4. **ID:** Each user is assigned a unique ID in the smart home automation system. This ID is used to identify the user and authenticate them when they log in to the mobile application.

5. **Appliances:** These are the electronic devices that are connected to the smart home automation system. Examples of appliances include lights, fans, air conditioners, and security cameras.

6. **Raspberry Pi:** This is a small computer that is used as the central hub of the smart home automation system. The Raspberry Pi communicates with the appliances and the mobile application to facilitate control of the appliances.

7. **Software developers:** These are the professionals who develop the software that powers the smart home automation system. The software developers are responsible for creating the mobile application and ensuring that it functions properly.

8. **Hardware developers:** These are the professionals who develop the hardware components of the smart home automation system, such as the Raspberry Pi and the appliances.

9. **Electricians:** These are the professionals who install the smart home automation system in the user's home. They are responsible for wiring the appliances and connecting them to the Raspberry Pi.

10. **Internet service provider (ISP):** The ISP is the company that provides the internet connection to the smart home automation system. The quality of the internet connection can impact the performance of the smart home automation system, so it is important to choose a reliable ISP.

Result:

Thus, Entity Relationship diagram was created successfully.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	8
Title of Experiment	DEVELOP A DATA FLOW DIAGRAM (PROCESS UP-TO LEVEL 1)
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	20/03/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

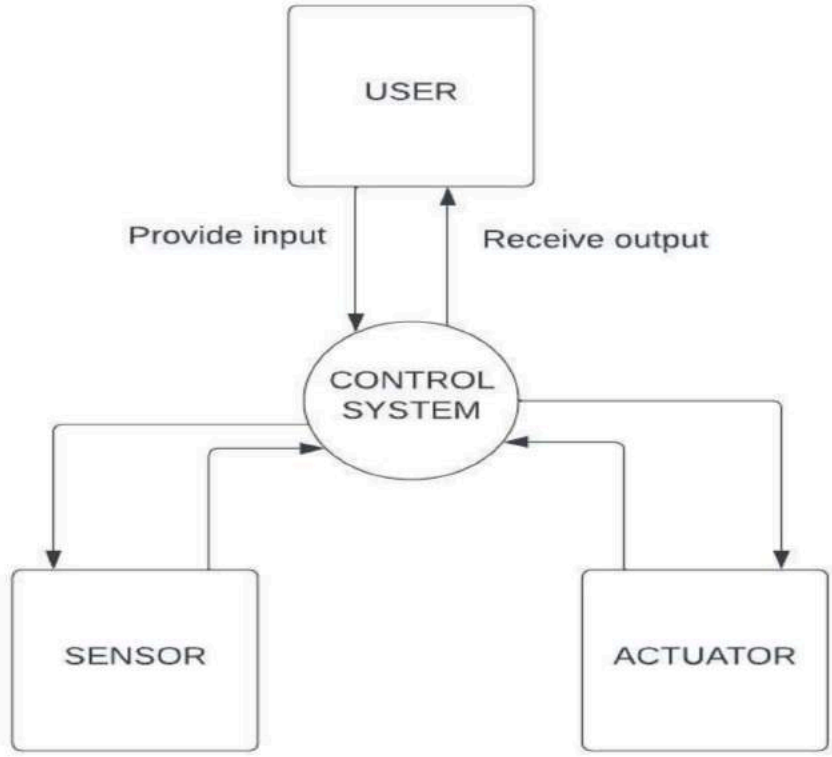
Aim:

To develop the data flow diagram up to level 1 for the Smart Home Automation using Raspberry PI.

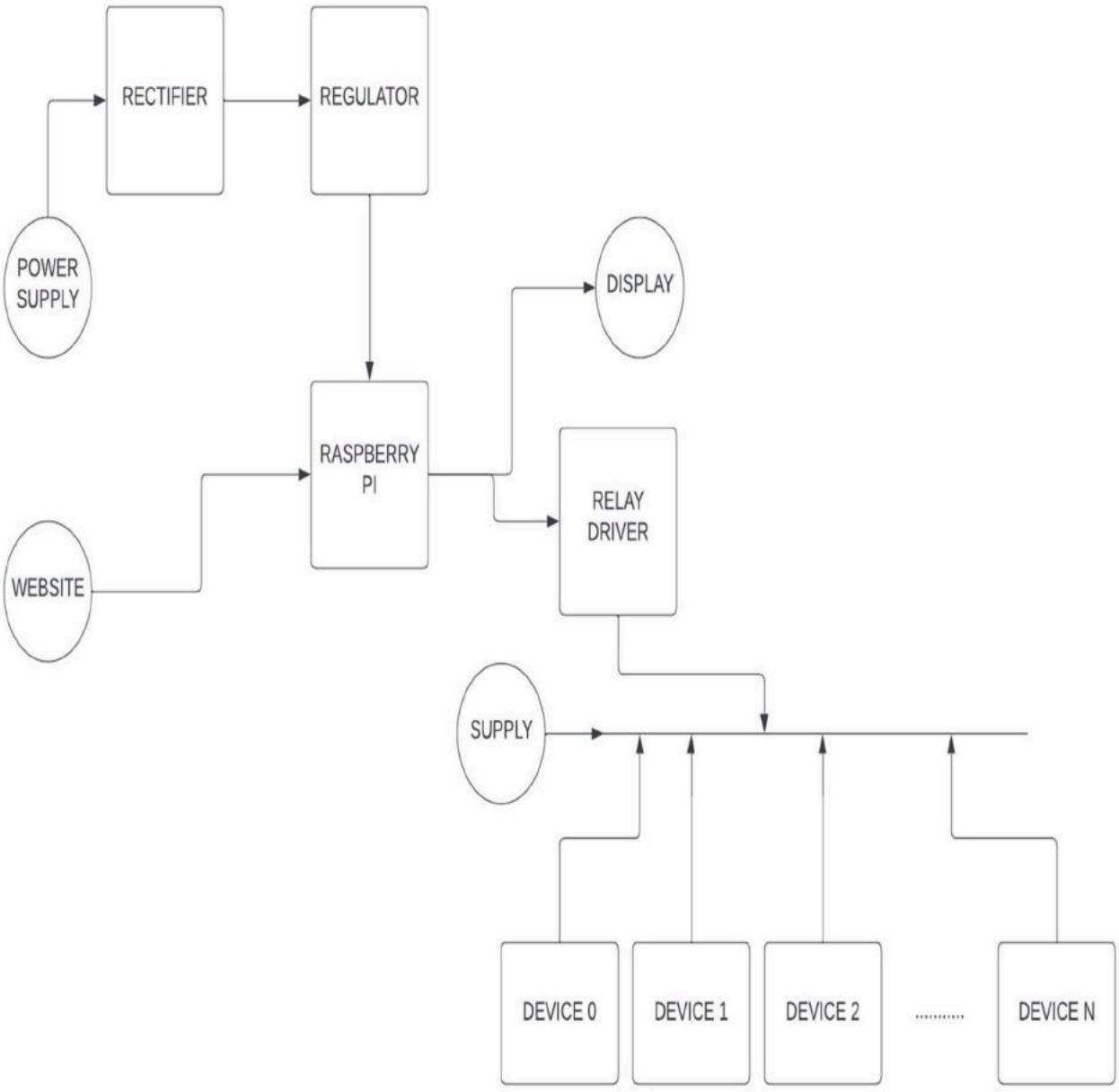
Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	**Rep**
2	RA2111003010662	AYUSHI GUPTA	**Member**
3	RA2111003010664	JOITA GHOSH	**Member**

Level 0 Data Flow Diagram:



Level 1 Data Flow Diagram:



Result:

Thus, the data flow diagrams have been created for the Smart Home Automation using Raspberry PI.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	9
Title of Experiment	DESIGN A SEQUENCE AND COLLABORATION DIAGRAM
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA,JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	26/03/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

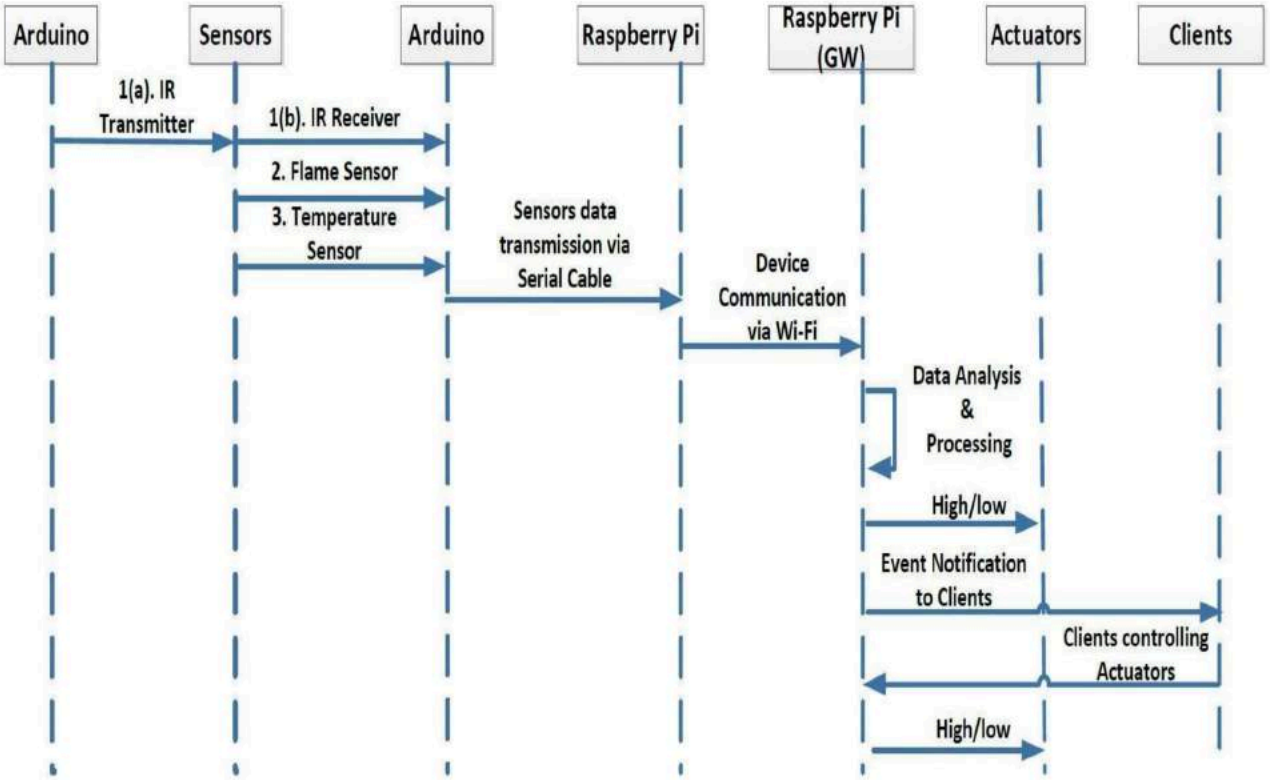
Aim:

To create the sequence and collaboration diagram for the Smart Home Automation using Raspberry Pi.

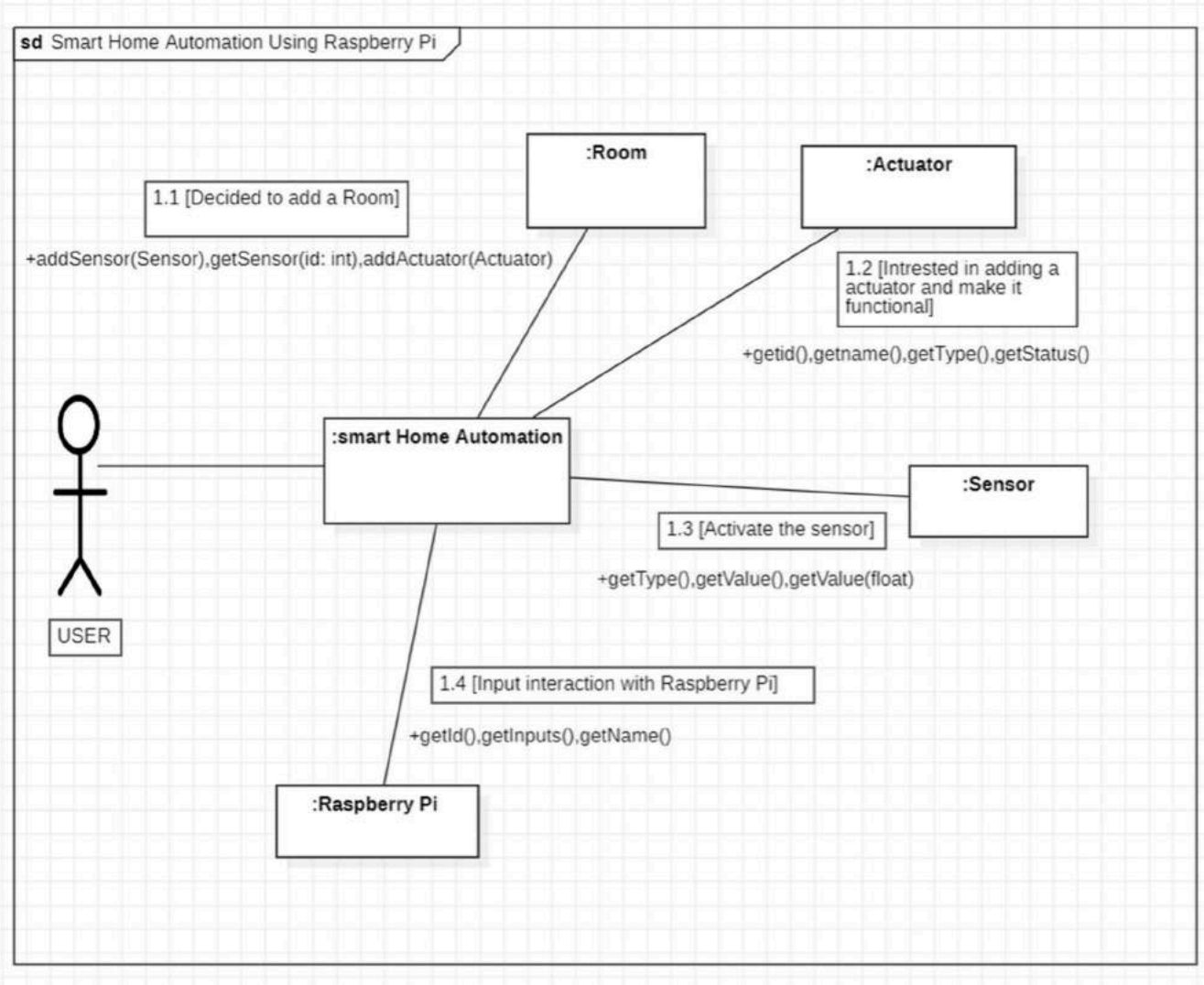
Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	Rep/Member
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

Sequence Diagram:



Collaboration Diagram:



Result:

Thus, the sequence and collaboration diagrams were created for the Smart Home Automation using Raspberry Pi.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	10
Title of Experiment	DEVELOP A TESTING FRAMEWORK/USER INTERFACE
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	30/04/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

To develop the testing framework and/or user interface framework for the smart home automation using Raspberry Pi.

Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	Rep/Member
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

User Interface:

1. Dashboard: The dashboard should provide users with an overview of all connected devices and their status. Users should be able to turn devices quickly and easily on or off, adjust settings, and view device details.

2. Device Control: Users should be able to control devices individually or in groups, using buttons, sliders, or other controls. The controls should be intuitive and easy to use, with clear feedback on device status.

3. Automation: The app should allow users to set up automation rules based on time, location, or other triggers. Users should be able to create rules that turn on or off devices automatically, adjust settings based on temperature or other environmental factors, or send notifications to the user's phone.

4. Personalization: The app should allow users to customize their experience, with options for changing the layout, colors, or other visual elements. Users should be able to set up user profiles, with different settings and access levels for each user.

Testing:

1. Functional Testing: This type of testing ensures that all features of the smart home automation app are working as intended. This includes testing device control, automation rules, and user profiles.

2. Compatibility Testing: This ensures that the smart home automation app is working correctly across different devices and platforms, including different operating systems and screen sizes.

3. Usability Testing: This involves testing the user interface of the smart home automation app and ensuring that it is easy to use and understand. This type of testing is crucial for a smart home automation app, as it should be accessible to users who may not be as familiar with technology or software.

4. Security Testing: This ensures that the smart home automation app is secure and protected against unauthorized access or tampering. This includes testing authentication and authorization mechanisms, encryption of data in transit and at rest, and vulnerability scanning.

Conclusion:

- A well-designed user interface is crucial for a smart home automation app, with easy-to-use controls and personalization options.
- The app should be thoroughly tested for functionality, compatibility, usability, and security, to ensure a high level of user satisfaction and protection of sensitive data.
- Overall, a well-designed smart home automation app using Raspberry Pi can provide users with a convenient and efficient way to control their home devices and improve their quality of life.

Testing Type	Methodology	Tools Required
Functional Testing	**Test all functions of the smart home automation system such as turning on/off lights, adjusting temperature, controlling appliances, etc.**	**Test automation frameworks such as Selenium, Appium, and Robot Framework.**
Security Testing	**Test the security of the smart home automation system to ensure that it is protected against potential cyber-attacks.**	**Vulnerability scanners such as Nessus and OpenVAS, penetration testing tools such as Metasploit and Nmap.**
Compatibility Testing	**Test the compatibility of the smart home automation system with different devices and platforms, including different operating systems, browsers, and screen sizes.**	**Emulators and simulators such as Android Studio and Xcode, real devices, and cross-browser testing tools such as Browser Stack and Sauce Labs.**
Usability Testing	**Test the user interface of the smart home automation system and ensure that it is easy to use and understand for a non-technical user.**	**Usability testing tools such as User Zoom and Usability Hub, survey tools such as SurveyMonkey and Google Forms.**
Performance Testing	**Test the performance of the smart home automation system, including the speed of the system, the response time, and the overall user experience.**	**Load testing tools such as Apache JMeter and LoadRunner, monitoring tools such as New Relic and AppDynamics.**

Result:

Thus, the testing framework/user interface framework has been created for the Smart Home Automation using Raspberry Pi.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	11
Title of Experiment	TEST CASES
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Number	RA2111003010667
Date of Experiment	04/04/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim:

To develop the test cases manual for the smart home automation using Raspberry Pi.

Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	Rep
2	RA2111003010662	AYUSHI GUPTA	Member
3	RA2111003010664	JOITA GHOSH	Member

Functional Test Cases:

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
F001	Control Lights	Turn on/off lights	1. Open the mobile app 2. Select the lights to be controlled 3. Turn on/off the light	1.The app should display the list of available devices. 2. The app should display the controls for the selected light. 3.The light should on/off accordingly.	1.The app displayed the list of available devices. 2. The app displayed the controls for the selected light. 3. The light turned on/off accordingly.	Pass Pass Pass	NIL
F002	Control Temperature	Adjust Temperature	1. Open the mobile app 2. Select the temperature controller 3. Adjust the temperature	1. The app should display the list of available devices. 2. The app should display the current temperature and controls to adjust it. 3.The temperature should change accordingly.	1. The app displayed the current temperature and controls to adjust it. 2. The app displayed the current temperature. 3.The temperature changed accordingly.	Pass Pass Pass	NIL

F003	Set Schedule.	Set device schedules.	1.Open the mobile app. 2. Select the device to schedule. 3. Set the schedule.	1.The app should display list of available devices. 2. The app should display the controls to set a schedule. 3. The device should turn on/off according to the schedule.	1.The list of available devices are displayed. 2. The controls to set a schedule, are displayed. 3. The device turned on/off accordingly.	Pass Pass Pass	NIL

Non-Functional Test Cases:

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
NF001	Performance	Response Time	Send a command and monitor response time	The device should respond and the response time should be within expected range.	The device responded and the response time was within the expected range	Pass	NIL
NFOO2	Security	Authentication	Open the app, enter incorrect authentication details, and then enter the correct authentication details.	The app should require the authentication before allowing access and the app shouldn't give access for incorrect authentication details	The app did not allow access when we provided wrong credentials and it gave access while providing correct credentials	Pass	NIL
NF003	Compatibility	Integration with other devices	Connect new device to raspberry Pi and test the device control.	The device should be recognised and added to the list of available devices and the device should respond for the command controls.	The device got added to list of available devices and responded for the control commands	Pass	NIL

Result:

Thus, the test case manual created for the Smart Home Automation using Raspberry Pi.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	12
Title of Experiment	Manual Test Case Reporting
Name of the candidate	VISHAL KHUMAR PD
Team Members	JOITA GHOSH, AYUSHI GUPTA
Register Number	RA2111003010667
Date of Experiment	12/04/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim:

To prepare the manual test case report for the Smart Home Automation using Raspberry Pi.

Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR PD	Rep/Member
2	RA2111003010664	JOITA GHOSH	Member
3	RA2111003010662	AYUSHI GUPTA	Member

Category	ProgressAgainstPlan	Status
Functional Testing	Green	Completed
Non-Functional Testing	Green	Completed

Functional Manual Test Case Report for Smart Home Automation using Raspberry Pi

Functional	Test Case Coverage (%)	Status
Install the application on the Android Operating System	100%	Completed
Application Starts successfully on clicking app icon	100%	Completed
Home Screen shows the appliances which can be controlled through the app	100%	Completed
Click on Light and select the room whose light needs to turn on	100%	Completed

Click on Fan and select the room whose fan needs to turn on	100%	Completed
Click on AC and select the room whose AC needs to turn on	100%	Completed
Click on Light and select the room whose light needs to turn off	100%	Completed
Click on Fan and select the room whose fan needs to turn off	100%	Completed
Click on AC and select the room whose AC needs to turn off	100%	Completed

Non-Functional Manual Test Case Report for Smart Home Automation using Raspberry Pi

Non-Functional	Test Case Coverage (%)	Status
To validate that the mandatory fields are displayed in the screen in a distinctive way.	100%	Completed
To validate that the application successfully connects to the home internet service	100%	Completed
To validate that the application successfully connects to the required home appliances	100%	Completed
To validate that the installed application enables other applications to perform satisfactorily, and it does not eat into the memory of the other applications.	100%	Completed

To validate whether the installation of the automation can be done smoothly provided the user has the necessary resources and it does not lead to any significant errors.	100%	Completed
To validate whether the smart home technology does not consume more electricity than what is feasible.	100%	Completed
To validate whether the Raspberry Pi is successfully incorporated into the automation system.	100%	Completed
To validate whether the application provides an available user guide for those who are not familiar to the app	100%	Completed

Result:

Thus, the test case report has been created for the Smart Home Automation using Raspberry Pi.

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	13
Title of Experiment	Provide the details of Architecture Design/Framework/Implementation
Name of the candidate	VISHAL KHUMAR P D
Team Members	AYUSHI GUPTA, JOITA GHOSH
Register Numbers	RA2111003010667
Date of Experiment	20-04-23

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	
2	Viva	5	
	Total	**10**	

Staff Signature with date

Aim

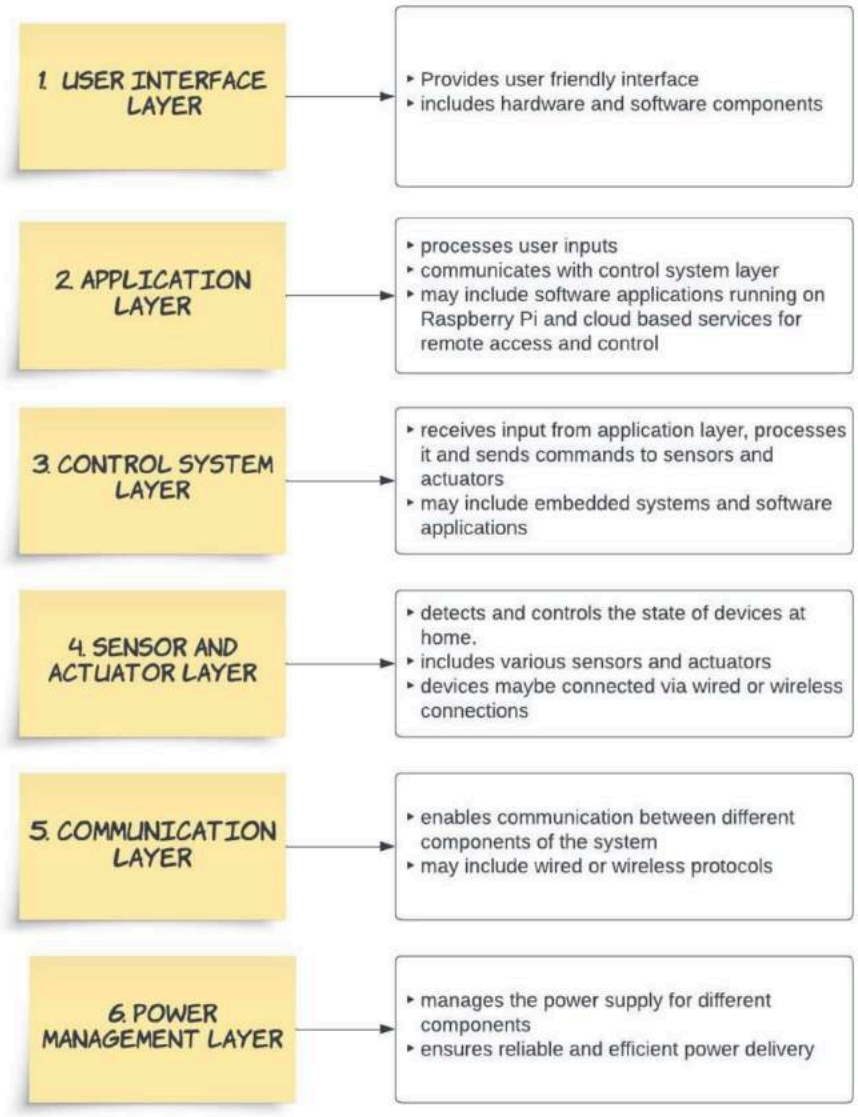
To provide the details of architectural design/framework/implementation for the Smart Home Automation using Raspberry PI.

Team Members:

S No	Register No	Name	Role
1	RA2111003010667	VISHAL KHUMAR P D	**Rep/Member**
2	RA2111003010662	AYUSHI GUPTA	**Member**
3	RA2111003010664	JOITA GHOSH	**Member**

Architectural Design:

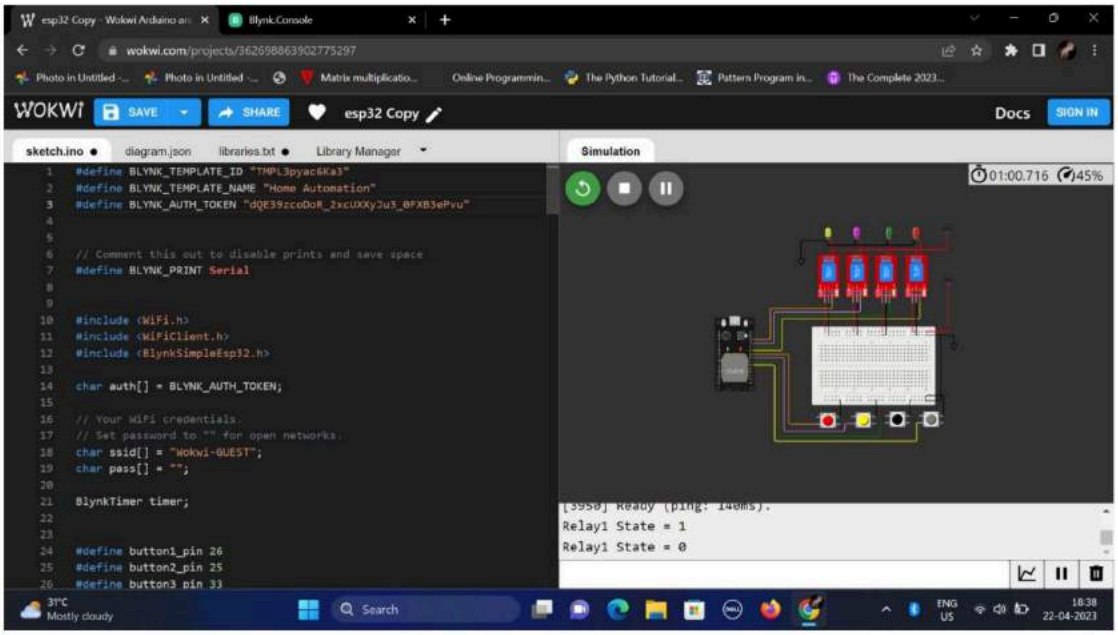
The Smart Home Automation using Raspberry PI project has a layered architectural style. It involves a complex and interconnected system of hardware and software components, each of which plays a critical role in enabling the system to function effectively and reliably. By carefully designing and integrating each of these layers, it is possible to create a system that provides convenient and efficient control over the devices in a home, while minimizing energy consumption and maximizing user satisfaction.

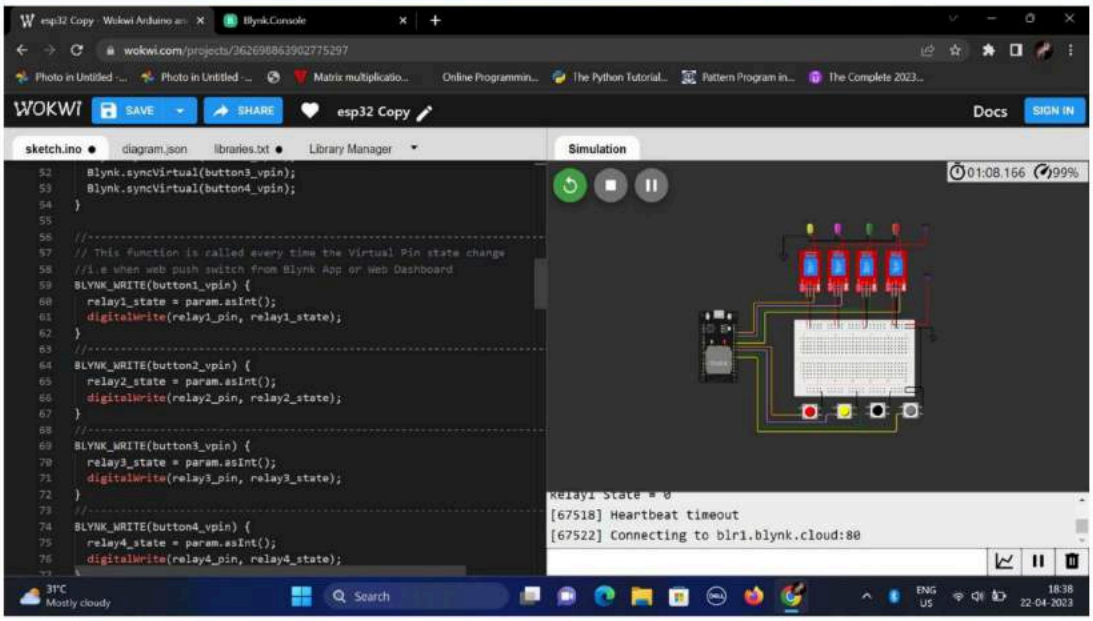


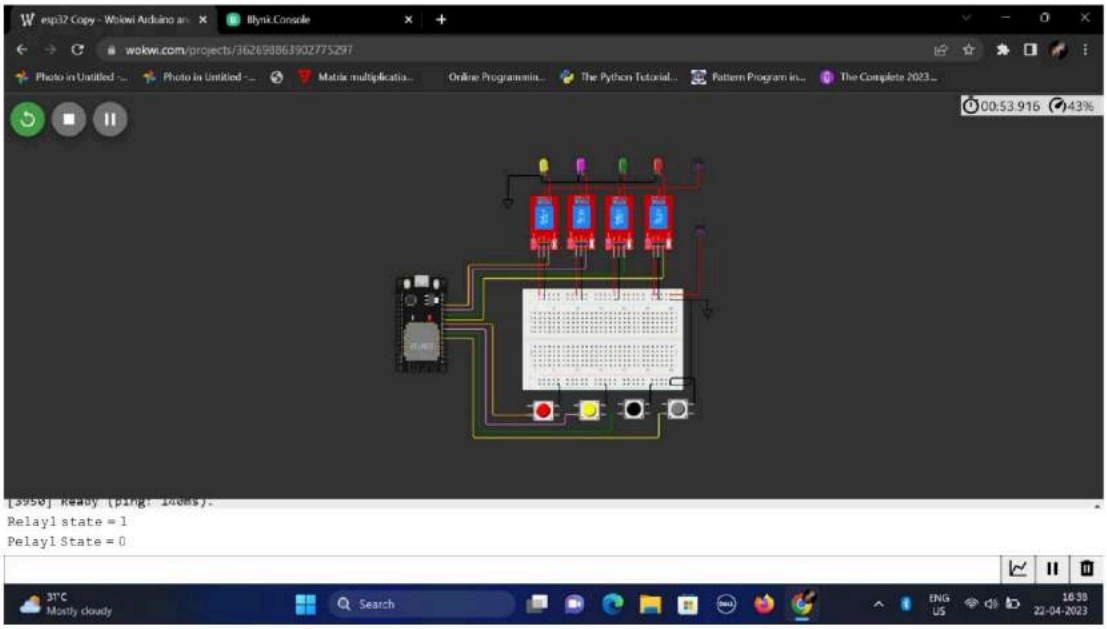
Implementation Details:

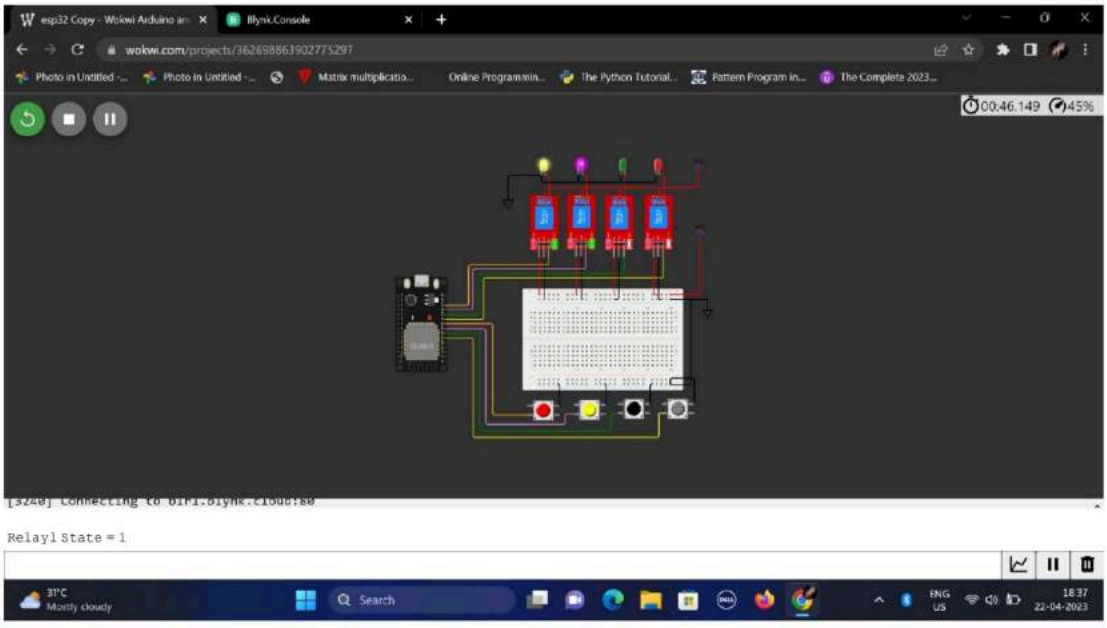
The working prototype is created using the Wokwi online simulator and the libraries included for the project are uploaded from the blynk platform.

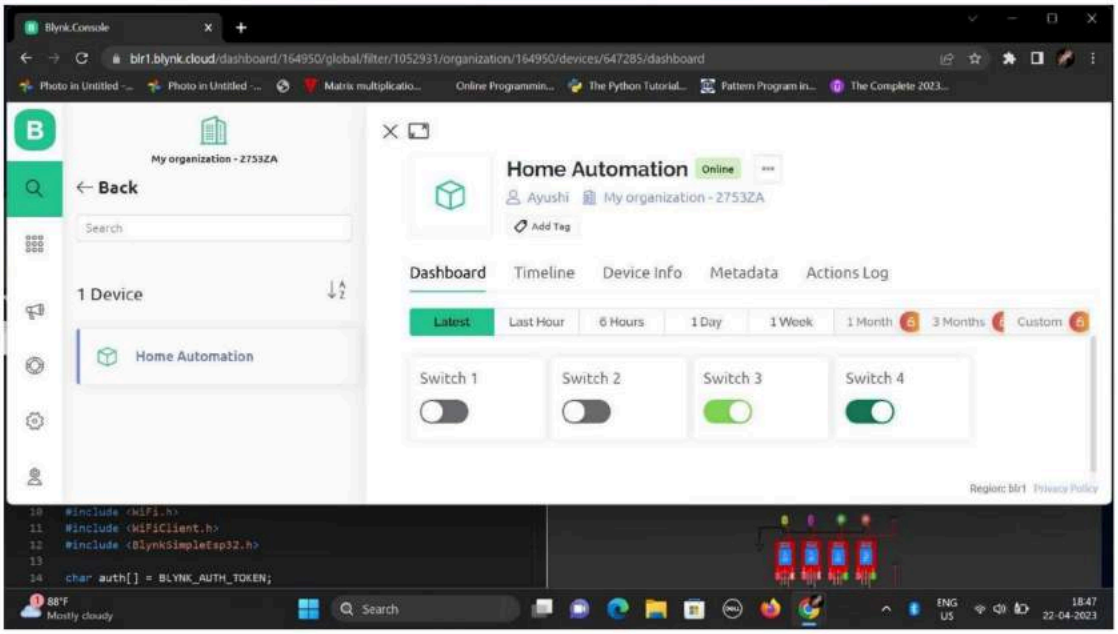
The components like the relay module, pushbutton, LEDs, raspberry Pi, breadboard, and connections were arranged to give a proper working module. The code was formulated to make it a working prototype. Blynk was then used to connect the wokwi simulator online using Blynk libraries.

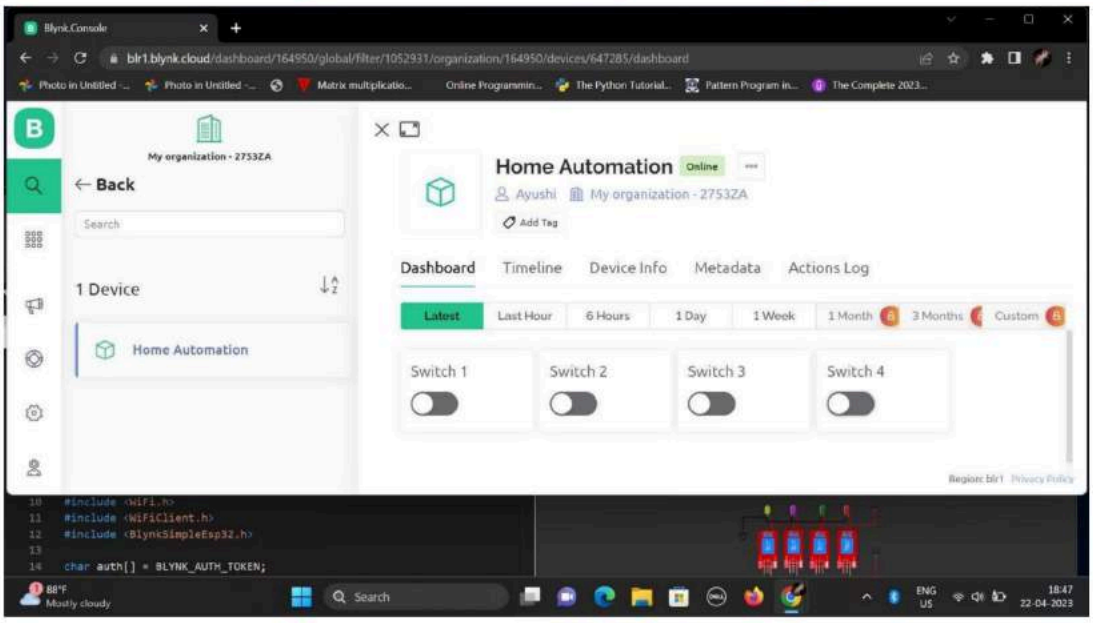





esp32 Copy - Wokwi Arduino an
Blynk.Console
wokwi.com/projects/362698863902775297
00:53.916
43%
Relay1 state = 1
Pelay1 State = 0
31°C
Mostly cloudy
Search
ENG
US
18:38
22-04-2023


esp32 Copy - Wokwi Arduino an
Blynk.Console
wokwi.com/projects/362698863902775297
00:46.149
45%
Relay1 State = 1
31°C
Mostly cloudy
Search
ENG
US
18:37
22-04-2023


Blynk.Console
blr1.blynk.cloud/dashboard/164950/global/filter/1052931/organization/164950/devices/647285/dashboard
My organization - 2753ZA
Back
Search
1 Device
Home Automation
Home Automation
Online
Ayushi
My organization - 2753ZA
Add Tag
Dashboard
Timeline
Device Info
Metadata
Actions Log
Latest
Last Hour
6 Hours
1 Day
1 Week
1 Month
3 Months
Custom
Switch 1
Switch 2
Switch 3
Switch 4
Region: blr1
Privacy Policy
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
char auth[] = BLYNK_AUTH_TOKEN;
88°F
Mostly cloudy
Search
ENG
US
18:47
22-04-2023


Blynk.Console
blr1.blynk.cloud/dashboard/164950/global/filter/1052931/organization/164950/devices/647285/dashboard
My organization - 2753ZA
Back
Search
1 Device
Home Automation
Home Automation
Online
Ayushi
My organization - 2753ZA
Add Tag
Dashboard
Timeline
Device Info
Metadata
Actions Log
Latest
Last Hour
6 Hours
1 Day
1 Week
1 Month
3 Months
Custom
Switch 1
Switch 2
Switch 3
Switch 4
Region: blr1
Privacy Policy
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
char auth[] = BLYNK_AUTH_TOKEN;
88°F
Mostly cloudy
Search
ENG
US
18:47
22-04-2023

Code section:

```
#define BLYNK_TEMPLATE_ID "TMPL3pyac6Ka3"
#define BLYNK_TEMPLATE_NAME "Home Automation"
#define BLYNK_AUTH_TOKEN "dQE39zcoDoR_2xcUXXyJu3_0FXB3ePvu"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

BlynkTimer timer;

#define button1_pin 26
#define button2_pin 25
#define button3_pin 33
#define button4_pin 32

#define relay1_pin 13
#define relay2_pin 12
#define relay3_pin 14
#define relay4_pin 27

int relay1_state = 0;
int relay2_state = 0;
int relay3_state = 0;
int relay4_state = 0;

//Change the virtual pins according the rooms
#define button1_vpin    V1
#define button2_vpin    V2
#define button3_vpin    V3
#define button4_vpin    V4

//------------------------------------------------------------------------------
// This function is called every time the device is connected to the Blynk.Cloud
// Request the latest state from the server
BLYNK_CONNECTED() {
  Blynk.syncVirtual(button1_vpin);
  Blynk.syncVirtual(button2_vpin);
  Blynk.syncVirtual(button3_vpin);
  Blynk.syncVirtual(button4_vpin);
```

```
}

//--------------------------------------------------------------------
// This function is called every time the Virtual Pin state change
//i.e when web push switch from Blynk App or Web Dashboard
BLYNK_WRITE(button1_vpin) {
  relay1_state = param.asInt();
  digitalWrite(relay1_pin, relay1_state);
}
//--------------------------------------------------------------------
BLYNK_WRITE(button2_vpin) {
  relay2_state = param.asInt();
  digitalWrite(relay2_pin, relay2_state);
}
//--------------------------------------------------------------------
BLYNK_WRITE(button3_vpin) {
  relay3_state = param.asInt();
  digitalWrite(relay3_pin, relay3_state);
}
//--------------------------------------------------------------------
BLYNK_WRITE(button4_vpin) {
  relay4_state = param.asInt();
  digitalWrite(relay4_pin, relay4_state);
}
//--------------------------------------------------------------------

void setup()
{
  // Debug console
  Serial.begin(115200);
  //------------------------------------------------------------------
  pinMode(button1_pin, INPUT_PULLUP);
  pinMode(button2_pin, INPUT_PULLUP);
  pinMode(button3_pin, INPUT_PULLUP);
  pinMode(button4_pin, INPUT_PULLUP);
  //------------------------------------------------------------------
  pinMode(relay1_pin, OUTPUT);
  pinMode(relay2_pin, OUTPUT);
  pinMode(relay3_pin, OUTPUT);
  pinMode(relay4_pin, OUTPUT);
  //------------------------------------------------------------------
  //During Starting all Relays should TURN OFF
  digitalWrite(relay1_pin, HIGH);
  digitalWrite(relay2_pin, HIGH);
  digitalWrite(relay3_pin, HIGH);
  digitalWrite(relay4_pin, HIGH);
  //------------------------------------------------------------------
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
```

```
  //------------------------------------------------------------
  //Blynk.virtualWrite(button1_vpin, relay1_state);
  //Blynk.virtualWrite(button2_vpin, relay2_state);
  //Blynk.virtualWrite(button3_vpin, relay3_state);
  //Blynk.virtualWrite(button4_vpin, relay4_state);
  //------------------------------------------------------------
}

void loop()
{
  Blynk.run();
  timer.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!

  listen_push_buttons();
}

void listen_push_buttons(){
    //-------------------------------------------------------------------
    if(digitalRead(button1_pin) == LOW){
      delay(200);
      control_relay(1);
      Blynk.virtualWrite(button1_vpin, relay1_state); //update button state
    }
    //-------------------------------------------------------------------
    else if (digitalRead(button2_pin) == LOW){
      delay(200);
      control_relay(2);
      Blynk.virtualWrite(button2_vpin, relay2_state); //update button state
    }
    //-------------------------------------------------------------------
    else if (digitalRead(button3_pin) == LOW){
      delay(200);
      control_relay(3);
      Blynk.virtualWrite(button3_vpin, relay3_state); //update button state
    }
    //-------------------------------------------------------------------
    else if (digitalRead(button4_pin) == LOW){
      delay(200);
      control_relay(4);
      Blynk.virtualWrite(button4_vpin, relay4_state); //update button state
    }
    //-------------------------------------------------------------------
}



void control_relay(int relay){
  //-----------------------------------------------
```

```
  if(relay == 1){
    relay1_state = !relay1_state;
    digitalWrite(relay1_pin, relay1_state);
    Serial.print("Relay1 State = ");
    Serial.println(relay1_state);
    delay(50);
  }
  //-----------------------------------------------
  else if(relay == 2){
    relay2_state = !relay2_state;
    digitalWrite(relay2_pin, relay2_state);
    delay(50);
  }
  //-----------------------------------------------
  else if(relay == 3){
    relay3_state = !relay3_state;
    digitalWrite(relay3_pin, relay3_state);
    delay(50);
  }
  //-----------------------------------------------
  else if(relay == 4){
    relay4_state = !relay4_state;
    digitalWrite(relay4_pin, relay4_state);
    delay(50);
  }
  //-----------------------------------------------
}
```

Result:

Thus, the details of architectural design/framework/implementation along with the screenshots were provided.