RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

CB23332
SOFTWARE ENGINEERING LAB

Laboratory Record Note Book

Name : . . . . . .

Year / Branch / Section : . . . . . .

Register No. : . . . . . .

Semester : . . . . . .

Academic Year : . . . . . .

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

RAJALAKSHMI NAGAR, THANDALAM – 602-105

BONAFIDE CERTIFICATE

NAME:_____**REGISTER NO.:** _____

ACADEMIC YEAR: 2024-25 **SEMESTER:** III **BRANCH:** _____B.E/B.Tech

This Certification is the bonafide record of work done by the above student in the

CB23332-SOFTWARE ENGINEERING - Laboratory during the year 2024 – 2025.

Signature of Faculty -in – Charge

Submitted for the Practical Examination held on _____

Internal Examiner External Examiner

INDEX

S. No.	Name of the Experiment	Expt. Date	Faculty Sign
1.	Preparing Problem Statement		
2.	Software Requirement Specification (SRS)		
3.	Entity-Relational Diagram		
4.	Data Flow Diagram		
5.	Use Case Diagram		
6.	Activity Diagram		
7.	State Chart Diagram		
8.	Sequence Diagram		
9.	Collaboration Diagramt		
10.	Class Diagram		

EX NO:1	WRITE THE COMPLETE PROBLEM STATEMENT
DATE	

AIM:

To prepare a PROBLEM STATEMENT for a Home automation system.

ALGORITHM:

- The problem statement is the initial starting point for a project.
- A problem statement describes what needs to be done without describing how.
- It is generally a one-to-three-page document that all project stakeholders agree upon, describing the goals of the project at a high level.
- The problem statement is intended for a broad audience and should be written in non-technical terms.
- It helps both technical and non-technical personnel communicate effectively by providing a clear description of the problem.
- The problem statement does not describe the solution to the problem.

INPUT:

- The input to requirement engineering is the problem statement prepared by the customer.
- It may include an overview of the existing system and the broad expectations from the new system.
- The first phase of requirements engineering begins with requirements elicitation, i.e., gathering information about the requirements.

  Here, requirements are identified with the help of the customer and existing system processes.

Problem:

Traditional home management systems often rely on manual control of appliances, leading to inefficiencies, increased energy consumption, and inconvenience. Users must physically interact with devices for tasks such as turning off lights, adjusting thermostats, or securing doors. This approach lacks flexibility and does not optimize resources, especially for individuals with busy schedules or mobility limitations.

Background:

The concept of home automation emerged to address the inefficiencies and limitations of traditional systems. With the advancement of IoT technologies, smart sensors, and actuators, home automation systems now offer enhanced control, energy efficiency, and security. By integrating devices into a centralized system, users can remotely operate appliances, receive alerts, and automate tasks, creating a seamless living experience.

Relevance:

Home automation systems are increasingly relevant as the demand for convenience, energy efficiency, and security grows. They align with the modern trend of digital transformation, enabling smart and sustainable living. These systems empower users to monitor and manage their homes effectively, reducing energy waste and improving quality of life.

Objectives:

The objective of the Home Automation System (HAS) is to provide users with a convenient, efficient, and secure way to manage their homes through centralized control and automation. The system aims to improve energy efficiency, enhance security, and simplify daily tasks.

1. **Simplify Home Management**: Offer a centralized platform to control appliances, lighting, and HVAC systems, either locally or remotely.
2. **Enhance Energy Efficiency**: Provide real-time energy monitoring and analytics to help users reduce consumption and optimize usage.
3. **Improve Security**: Enable integration with surveillance cameras, motion detectors, and smart locks for comprehensive home security.
4. **Automation**: Allow users to automate routine tasks, such as scheduling lighting or temperature adjustments, for increased convenience.
5. **Provide Alerts and Notifications**: Notify users of critical events, such as security breaches or system malfunctions, to ensure timely action.
6. **Support Scalability and Integration**: Ensure compatibility with a wide range of IoT devices and future upgrades to meet evolving user needs.
7. **Boost Accessibility**: Design intuitive interfaces that are user-friendly and accessible for all, including individuals with mobility or vision impairments.

Result:

EX NO:2	WRITE THE SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT
DATE	

AIM:

To do requirement analysis and develop Software Requirement Specification Sheet(SRS) for E-voting system.

ALGORITHM:

SRS shall address are the following:

a) **Functionality.** What is the software supposed to do?

b) **External interfaces.** How does the software interact with people, the system‘s hardware, other hardware, and other software?

c) **Performance.** What is the speed, availability, response time, recovery time of various software functions, etc.?

d) **Attributes.** What is the portability, correctness, maintainability, security, etc. considerations?

e) **Design constraints imposed on an implementation.** Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) defines the functional and non-functional requirements for a Home Automation System (HAS). The HAS aims to provide users with centralized control over various smart devices within their homes, enhancing convenience, energy efficiency, and security. The intended audience for this document includes system developers, testers, and project managers.

1.2 Scope

The HAS will encompass the following major functionalities:

- Device Control: Remotely control various smart devices such as lights, thermostats, locks, and appliances.
- Scheduling: Create automated schedules for device actions based on time, events, or environmental conditions.
- Energy Monitoring: Track energy consumption of connected devices to optimize usage and reduce costs.
- System Alerts: Receive notifications for system events, such as security breaches, device failures, or energy usage thresholds.

1.3 Definitions, Acronyms, and Abbreviations

- HAS: Home Automation System
- IoT: Internet of Things

1.4 References

- IEEE Std 830-1998: Software Requirements Specification

2. Overall Description

2.1 Product Perspective

The HAS will integrate with a variety of smart devices through a central hub or directly via Wi-Fi or other IoT protocols. Users will interact with the system through a mobile app or voice commands.

2.2 Product Features

- Device Control: Remotely turn devices on/off, adjust settings (e.g., temperature, brightness), and create custom scenes.
- Scheduling: Set up automated routines for daily tasks, such as waking up, going to sleep, or leaving home.
- Energy Monitoring: Track real-time and historical energy usage, identify areas for optimization, and receive energy-saving tips.
- System Alerts: Receive notifications for security breaches, device failures, or unexpected energy spikes.
- Voice Control: Interact with the system using voice commands to control devices and access information.

2.3 User Classes and Characteristics

- Homeowners: Primary users who control the system and manage devices.
- Guests: Temporary users with limited access to specific devices or functions.

2.4 Operating Environment

- Hardware: Smart devices compatible with the HAS, central hub (if applicable), mobile devices (iOS and Android).
- Software: HAS mobile app, cloud-based backend services.

2.5 Design and Implementation Constraints

- Security: Robust security measures to protect user data and prevent unauthorized access.
- Interoperability: Compatibility with a wide range of smart devices and IoT protocols.
- User Experience: Intuitive and user-friendly interface for seamless control.
- Performance: Real-time response to user commands and efficient data processing.

2.6 Assumptions and Dependencies

- Reliable internet connectivity for cloud-based features.
- Consistent power supply for the central hub and smart devices.
- Timely updates and support from device manufacturers.

3. System Features

3.1 Device Control

- Functional Requirements:
  - Remotely turn devices on/off.
  - Adjust device settings (e.g., temperature, brightness, volume).
  - Create and manage custom scenes (e.g., "Good Morning," "Movie Night").
  - Control multiple devices simultaneously.
- Non-Functional Requirements:
  - Real-time response to user commands.
  - Secure communication between the HAS and devices.

3.2 Scheduling

- Functional Requirements:
  - Create time-based schedules for device actions (e.g., turn on lights at sunset).
  - Set up event-based schedules (e.g., activate security mode when leaving home).
  - Use environmental conditions (e.g., temperature, humidity) to trigger actions.
  - Manage multiple schedules and adjust them as needed.
- Non-Functional Requirements:
  - Reliable execution of scheduled tasks.
  - Flexibility to modify schedules.

3.3 Energy Monitoring

- Functional Requirements:
  - Track real-time energy consumption of individual devices and the entire home.
  - Generate historical energy usage reports.
  - Identify energy-saving opportunities and provide recommendations.
  - Set energy usage alerts and notifications.
- Non-Functional Requirements:
  - Accurate energy consumption data.
  - Clear and concise energy usage reports.

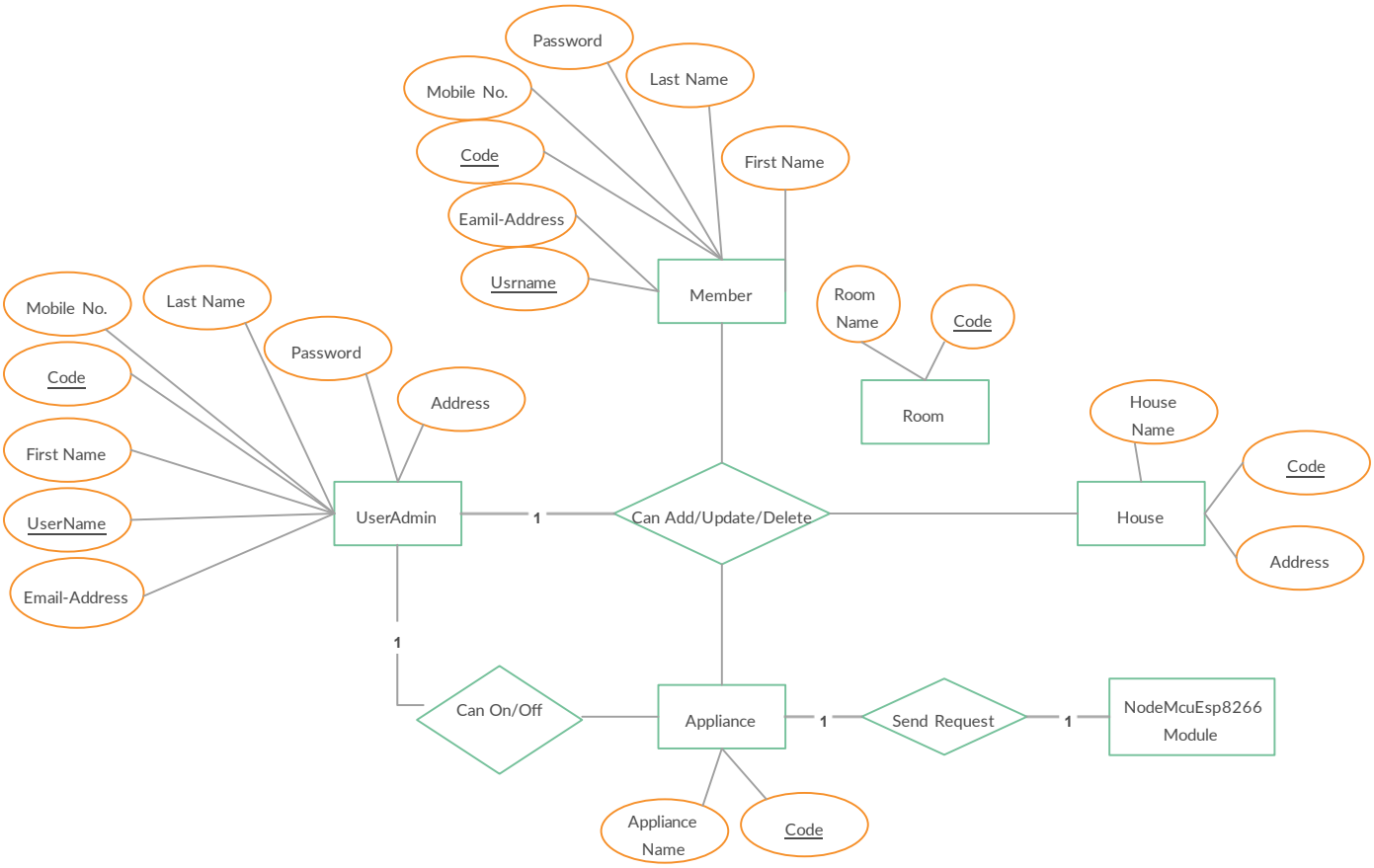
3.4 System Alerts

- Functional Requirements:
  - Receive notifications for security breaches (e.g., unauthorized access, sensor triggers).
  - Get alerts for device failures or malfunctions.
  - Receive energy usage alerts (e.g., high energy consumption).
  - Customize alert preferences and severity levels.
- Non-Functional Requirements: *

Result:

SAMPLE OUTPUT:

ER DIAGRAM:



EX NO:3	DRAW THE ENTITY RELATIONSHIP DIAGRAM
DATE	

AIM:

To Draw the Entity Relationship Diagram for Home automation system.

ALGORITHM:

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multivalued attributes.

INPUT:

Entities

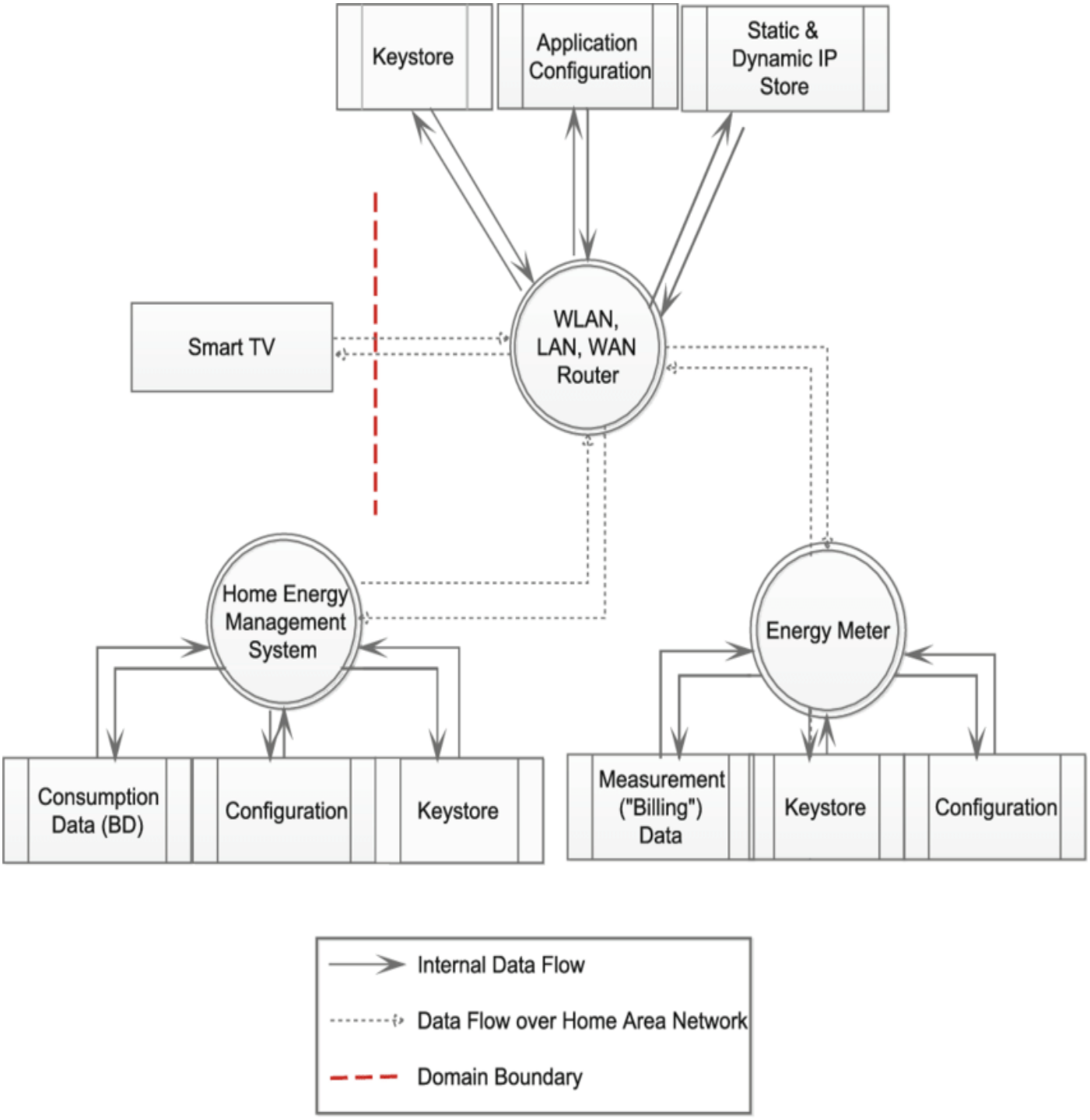
Entity Relationship Matrix

Primary Keys

Attributes

Mapping of Attributes with Entities

Result:


Keystore
Application Configuration
Static & Dynamic IP Store
Smart TV
WLAN, LAN, WAN Router
Home Energy Management System
Energy Meter
Consumption Data (BD)
Configuration
Keystore
Measurement ("Billing") Data
Keystore
Configuration
Internal Data Flow
Data Flow over Home Area Network
Domain Boundary

EX NO:4	DRAW THE DATA FLOW DIAGRAMS AT LEVEL 0 AND LEVEL 1
DATE	

AIM:

To Draw the Data Flow Diagram for Home automation system and List the Modules in the Application.

ALGORITHM:

1. Open the Visual Paradigm to draw DFD (Ex.Lucidchart)
2. Select a data flow diagram template
3. Name the data flow diagram
4. Add an external entity that starts the process
5. Add a Process to the DFD
6. Add a data store to the diagram
7. Continue to add items to the DFD
8. Add data flow to the DFD
9. Name the data flow
10. Customize the DFD with colours and fonts
11. Add a title and share your data flow diagram

INPUT:

Processes

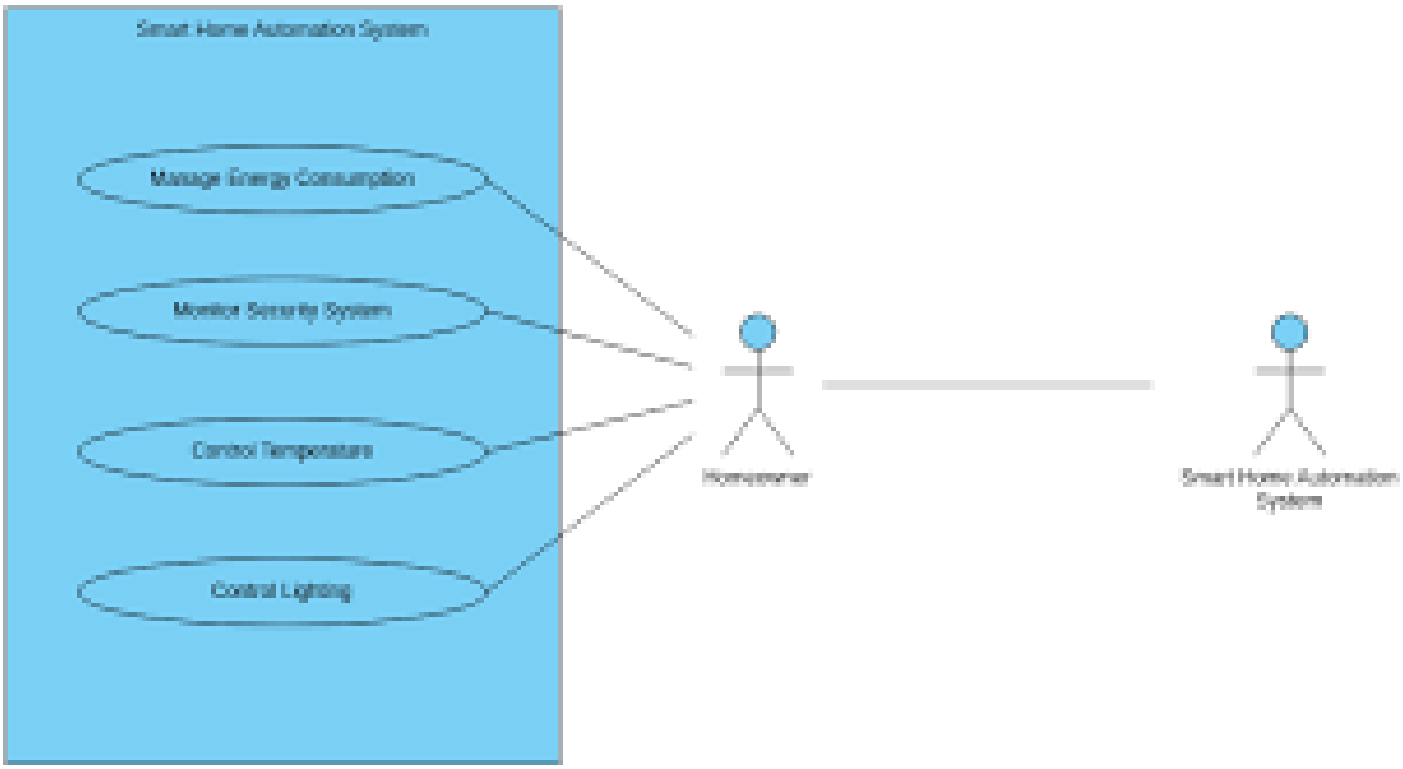
Datastores

External Entities

Result:

SAMPLE OUTPUT:

USE CASE DIAGRAM:



EX NO:5	DRAW USE CASE DIAGRAM
DATE	

AIM:

To Draw the Use Case Diagram for Home automation system**.**

ALGORITHM:

Step 1: Identify Actors

Step 2: Identify Use Cases

Step 3: Connect Actors and Use Cases

Step 4: Add System Boundary

Step 5: Define Relationships

Step 6: Review and Refine

Step 7: Validate

INPUTS:

Actors

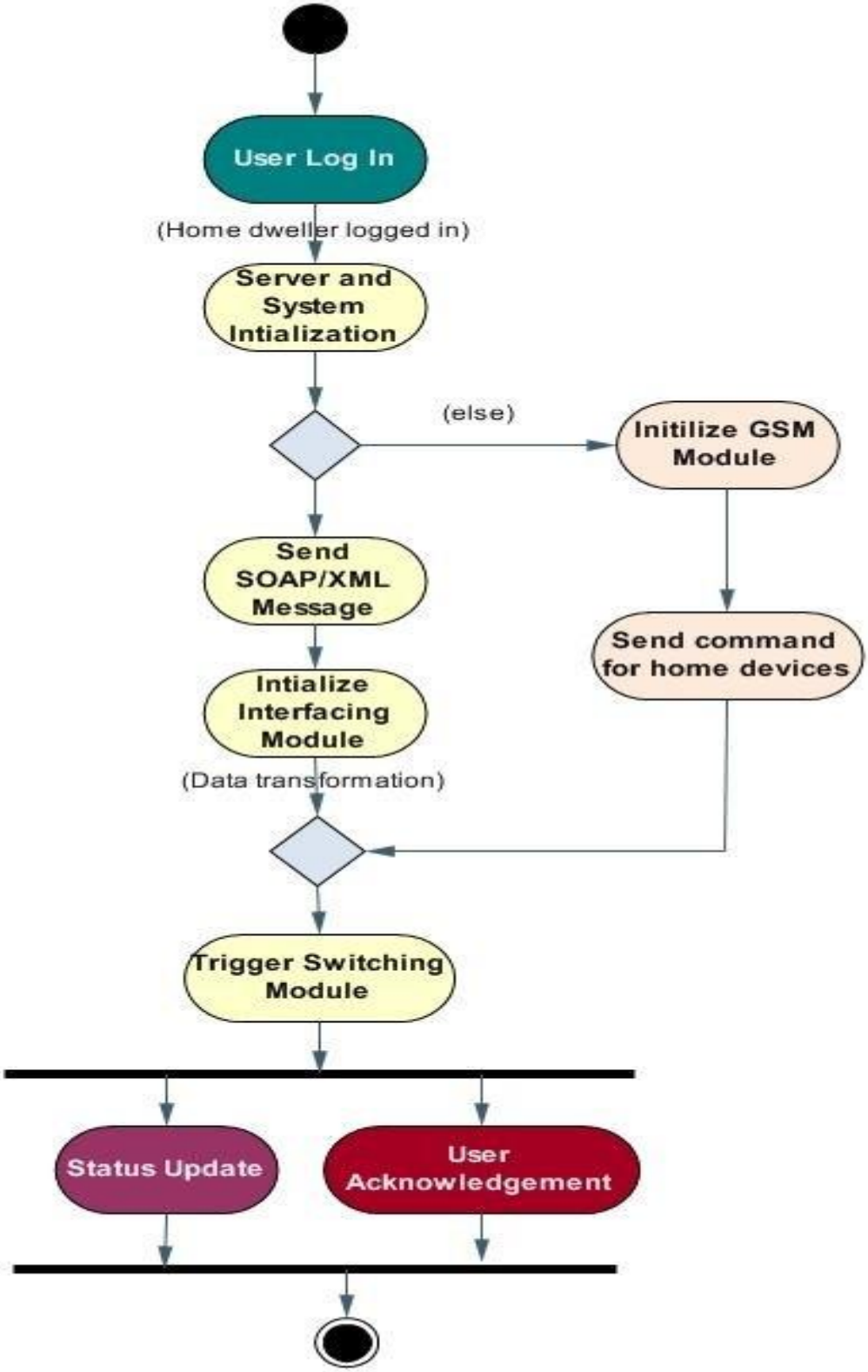
Use Cases

Relations

Result:

SAMPLE OUTPUT:

ACTIVITY DIAGRAM:



EX NO:6	**DRAW ACTIVITY DIAGRAM OF ALL USE CASES.**
DATE	

AIM:

To Draw the activity Diagram for home automation system.

ALGORITHM:

Step 1: Identify the Initial State and Final States

Step 2: Identify the Intermediate Activities Needed

Step 3: Identify the Conditions or Constraints

Step 4: Draw the Diagram with Appropriate Notations

INPUTS:

Activities

Decision Points

Guards

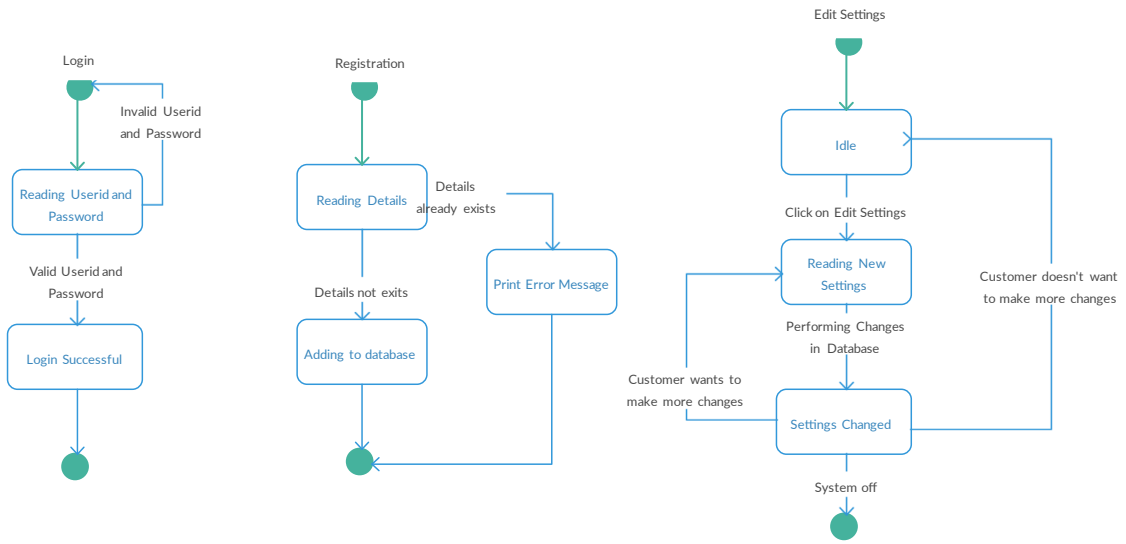
Parallel Activities

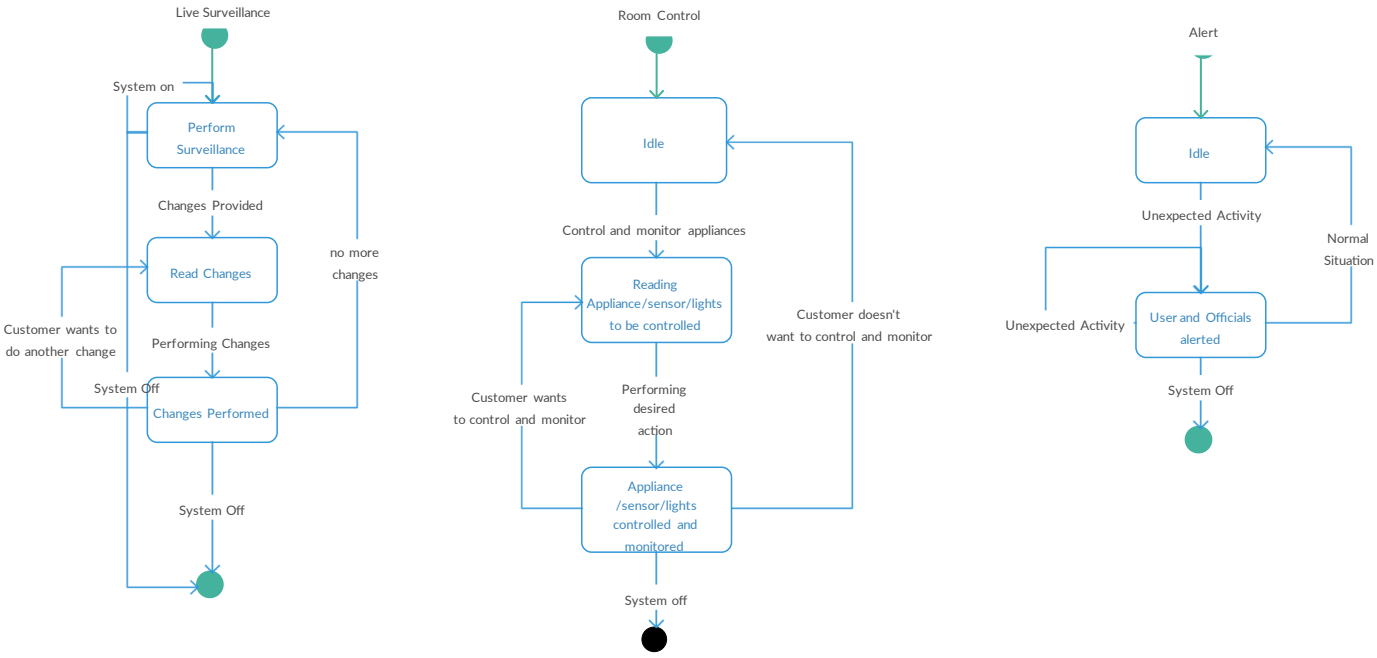
Conditions

Result:

SAMPLE OUTPUT:

STATE CHART DIAGRAM:





EX NO:7	DRAW STATE CHART DIAGRAM OF ALL USE CASES.
DATE	

AIM:

To Draw the State Chart Diagram for Home automation system.

ALGORITHM:

STEP-1: Identify the important objects to be analysed.

STEP-2: Identify the states.

STEP-3: Identify the events.

INPUTS:

Objects

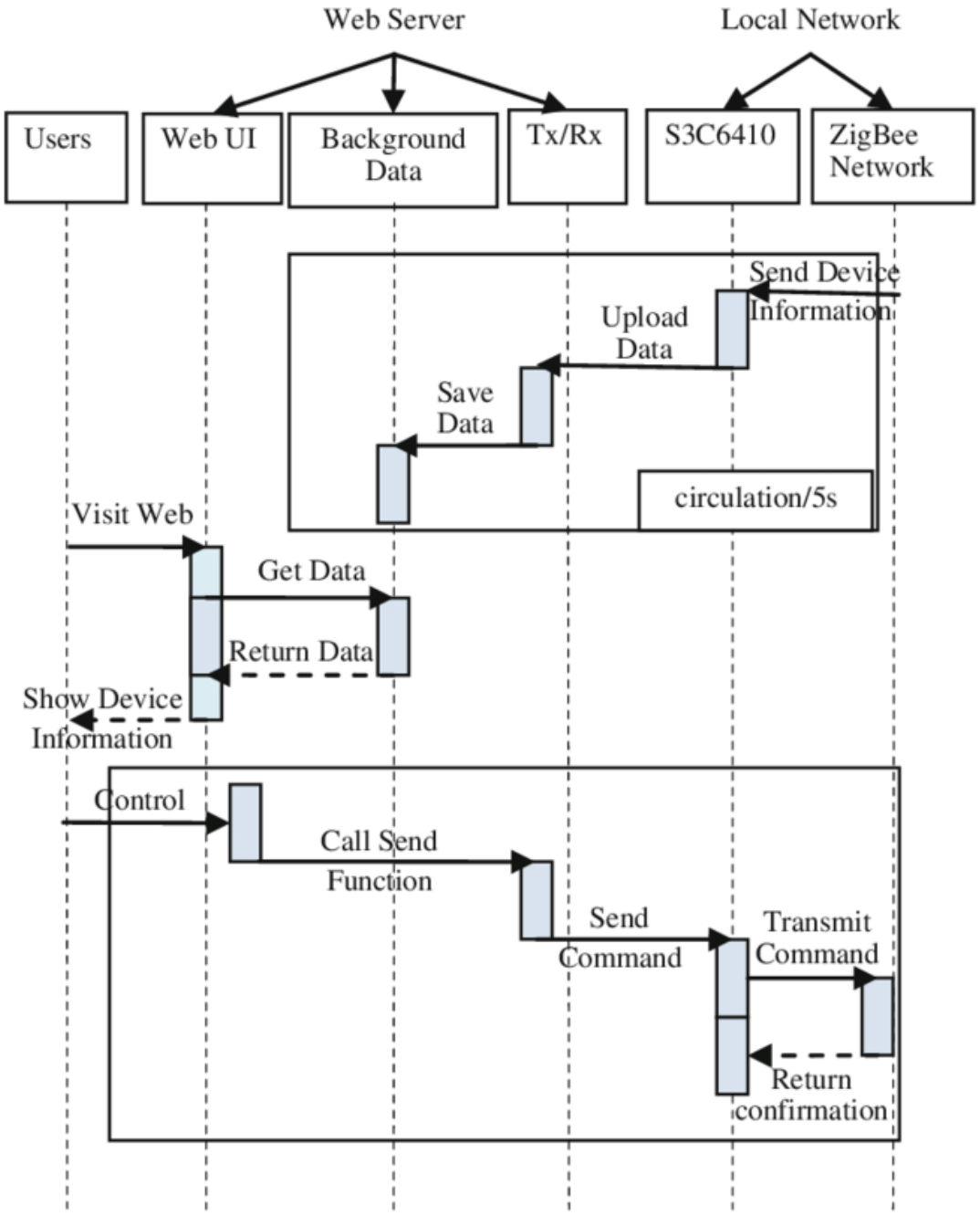
States

Events

Result:

SAMPLE OUTPUT:

SEQUENCE DIAGRAM:



EX NO:8	DRAW SEQUENCE DIAGRAM OF ALL USE CASES.
DATE	

AIM:

To Draw the Sequence Diagram for Home automation system.

ALGORITHM:

1. Identify the Scenario
2. List the Participants
3. Define Lifelines
4. Arrange Lifelines
5. Add Activation Bars
6. Draw Messages
7. Include Return Messages
8. Indicate Timing and Order
9. Include Conditions and Loops
10. Consider Parallel Execution
11. Review and Refine
12. Add Annotations and Comments
13. Document Assumptions and Constraints
14. Use a Tool to create a neat sequence diagram

INPUTS:

Objects taking part in the interaction.

Message flows among the objects.

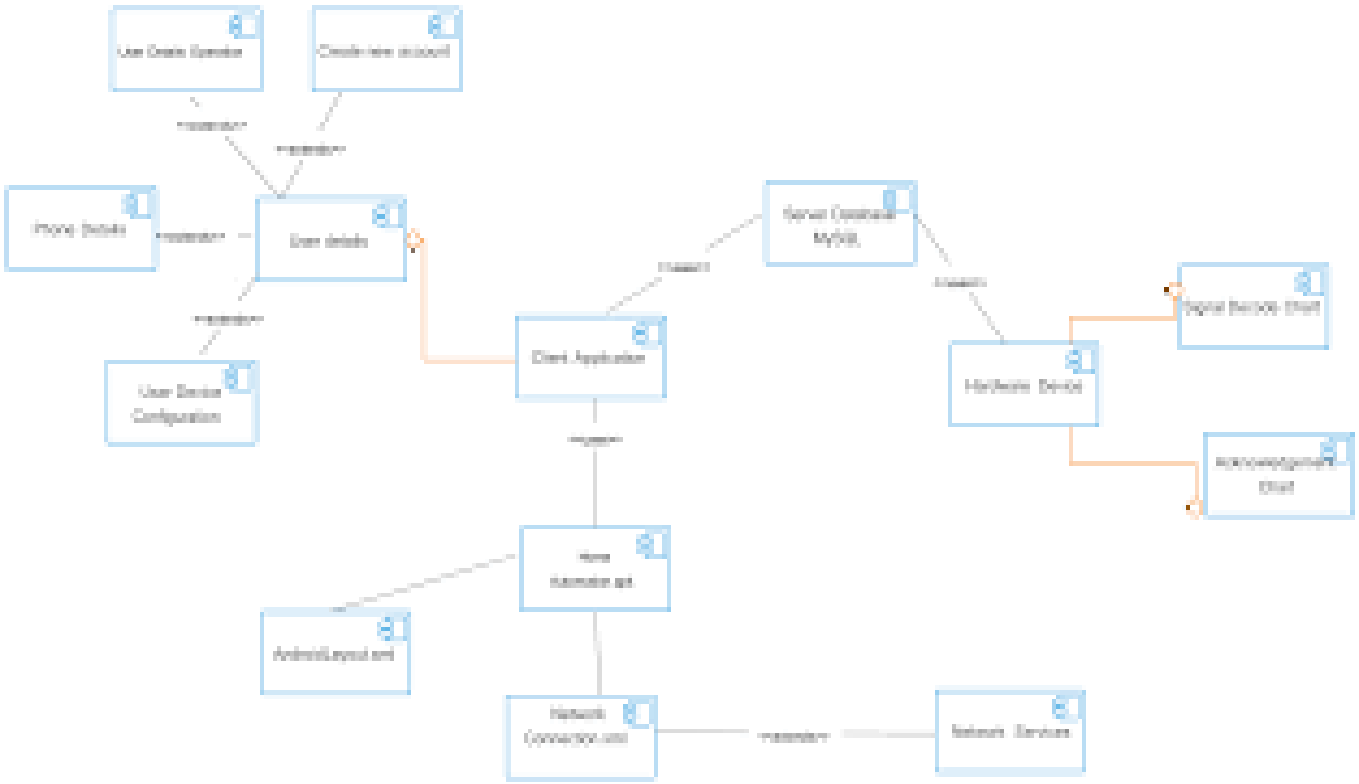
The sequence in which the messages are flowing.

Object organization.

Result:

SAMPLE OUTPUT:

COLLABORATION DIGRAM:



EX NO:9	DRAW COLLABORATION DIAGRAM OF ALL USE CASES
DATE	

AIM:

To Draw the Collaboration Diagram for Home automation system.

ALGORITHM:

Step 1: Identify Objects/Participants

Step 2: Define Interactions

Step 3: Add Messages

Step 4: Consider Relationships

Step 5: Document the collaboration diagram along with any relevant explanations or annotations.

INPUTS:

Objects taking part in the interaction.

Message flows among the objects.

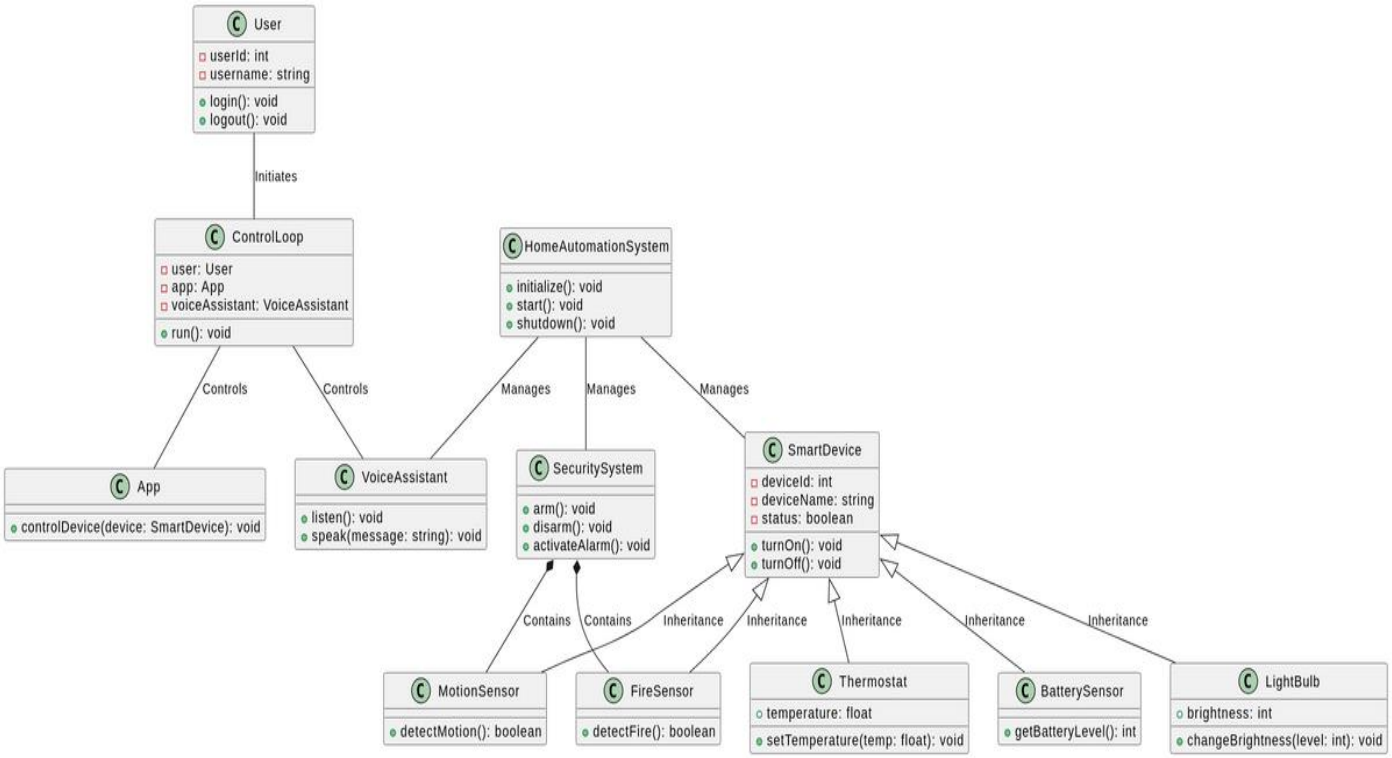
The sequence in which the messages are flowing.

Object organization.

Result:

SAMPLE OUTPUT:

CLASS DIAGRAM:



EX NO:10	ASSIGN OBJECTS IN SEQUENCE DIAGRAM TO CLASSES AND MAKE CLASS DIAGRAM.
DATE	

AIM:

To Draw the Class Diagram for Home automation system.

ALGORITHM:

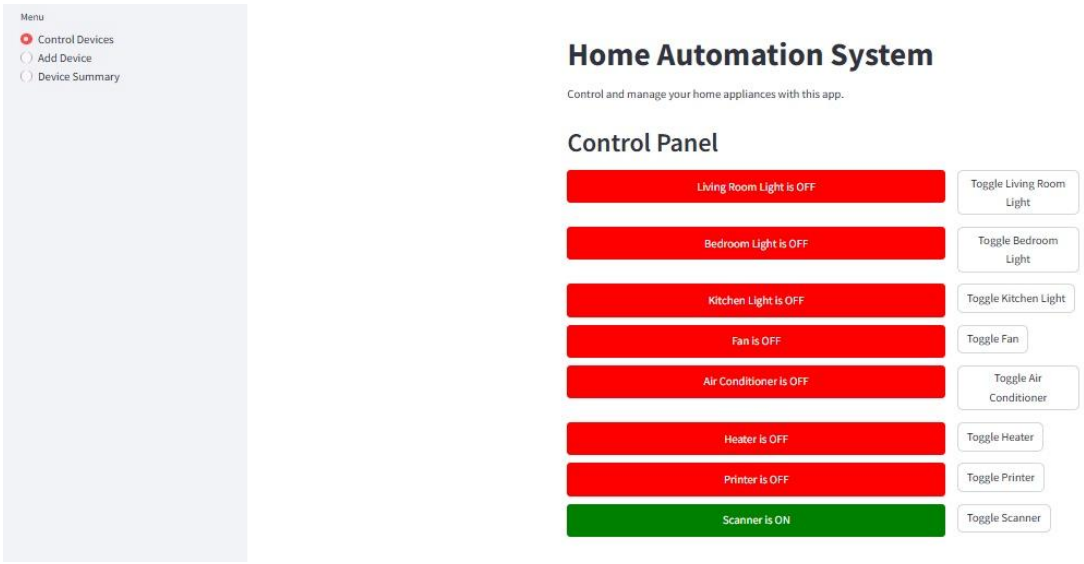
1. Identify Classes
2. List Attributes and Methods
3. Identify Relationships
4. Create Class Boxes
5. Add Attributes and Methods
6. Draw Relationships
7. Label Relationships
8. Review and Refine
9. Use Tools for Digital Drawing

INPUTS:

1. Class Name
2. Attributes
3. Methods
4. Visibility Notation

RESULT:

OUTPUT:


Menu
Control Devices
Add Device
Device Summary
Home Automation System
Control and manage your home appliances with this app.
Control Panel
Living Room Light is OFF
Toggle Living Room Light
Bedroom Light is OFF
Toggle Bedroom Light
Kitchen Light is OFF
Toggle Kitchen Light
Fan is OFF
Toggle Fan
Air Conditioner is OFF
Toggle Air Conditioner
Heater is OFF
Toggle Heater
Printer is OFF
Toggle Printer
Scanner is ON
Toggle Scanner

EX NO:11	MINI PROJECT- HOME AUTOMATION
DATE	

AIM:

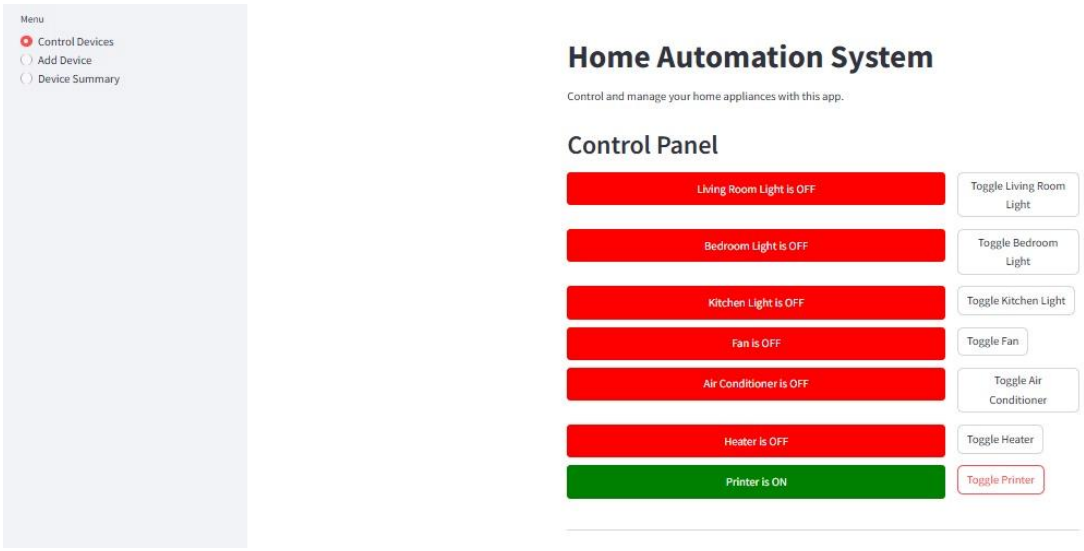
The primary aim of this mini-project is to develop a secure and user-friendly Home Automation system. By utilizing MySQL for robust data storage and Streamlit for a seamless user interface, we aim to streamline home control processes, ensuring convenience, energy efficiency, and enhanced user experience.

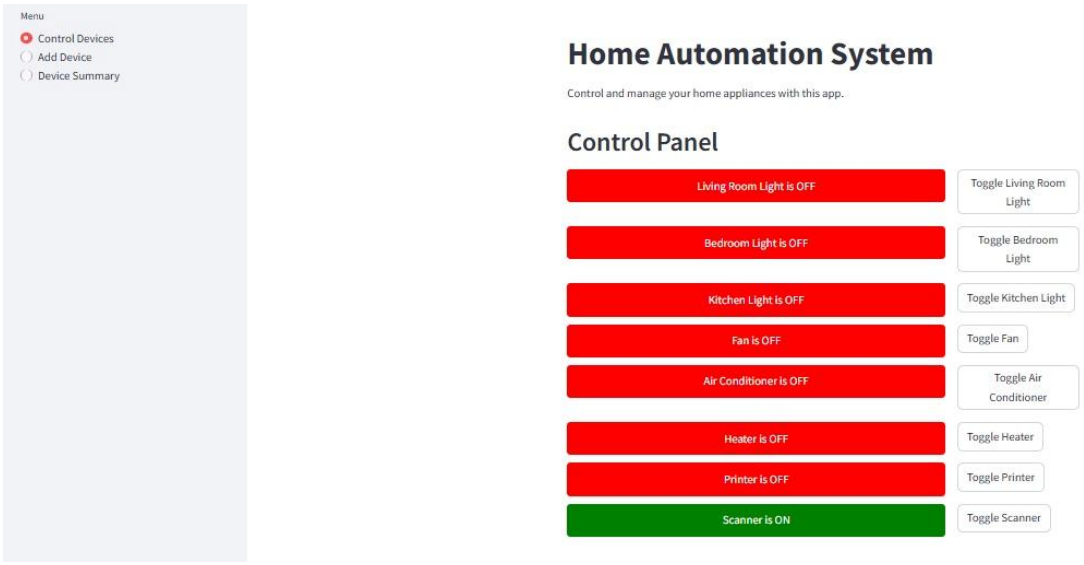
ALGORITHM:

1. User registers with valid credentials.
2. User logs in using their credentials.
3. System displays a list of connected smart devices.
4. User selects a device to control or monitor.
5. User's commands are securely encrypted and transmitted to the device.
6. System verifies the successful execution of the command.
7. Device state or feedback is securely logged in the database.
8. Comprehensive usage reports are published transparently for user review.

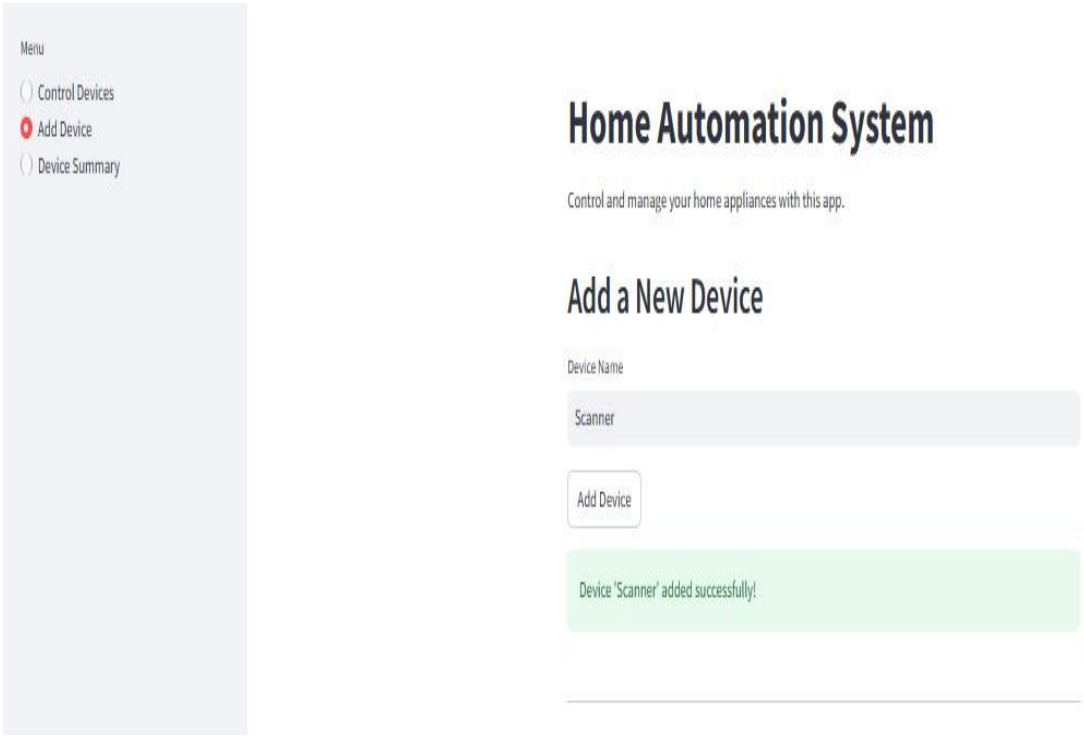
PROGRAM:

```
import streamlit as st


# Initialize session state for device management
if 'devices' not in st.session_state:
    st.session_state.devices = {
        'Living Room Light': False,
        'Bedroom Light': False,
        'Kitchen Light': False,
        'Fan': False,
        'Air Conditioner': False,
```

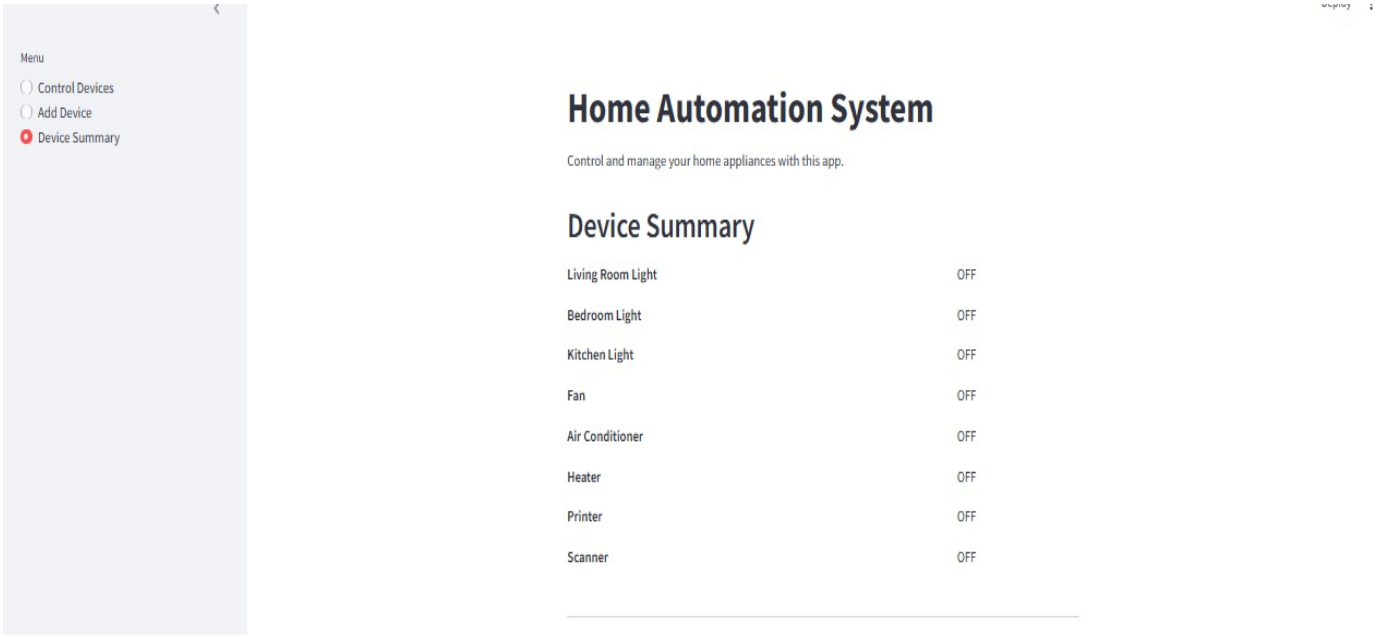

Menu
Control Devices
Add Device
Device Summary
Home Automation System
Control and manage your home appliances with this app.
Control Panel
Living Room Light is OFF
Toggle Living Room Light
Bedroom Light is OFF
Toggle Bedroom Light
Kitchen Light is OFF
Toggle Kitchen Light
Fan is OFF
Toggle Fan
Air Conditioner is OFF
Toggle Air Conditioner
Heater is OFF
Toggle Heater
Printer is ON
Toggle Printer


Menu
Control Devices
Add Device
Device Summary
Home Automation System
Control and manage your home appliances with this app.
Control Panel
Living Room Light is OFF
Toggle Living Room Light
Bedroom Light is OFF
Toggle Bedroom Light
Kitchen Light is OFF
Toggle Kitchen Light
Fan is OFF
Toggle Fan
Air Conditioner is OFF
Toggle Air Conditioner
Heater is OFF
Toggle Heater
Printer is OFF
Toggle Printer
Scanner is ON
Toggle Scanner

```
        'Heater': False,
    }

# Function to set box color based on state
def get_box_style(state):
    color = "green" if state else "red"
    return f"background-color: {color}; color: white; padding: 10px; border-radius: 5px; text-align: center;"

# Application title
st.title("Home Automation System")
st.write("Control and manage your home appliances with this app.")

# Sidebar Menu
menu = st.sidebar.radio("Menu", ["Control Devices", "Add Device", "Device Summary"])

# Control Devices Section
if menu == "Control Devices":
    st.header("Control Panel")
    for device, state in st.session_state.devices.items():
        col1, col2 = st.columns([3, 1])
        with col1:
            # Display device with dynamic color
            st.markdown(
                f"<div style='{get_box_style(state)}'>{device} is {'ON' if state else 'OFF'}</div>",
                unsafe_allow_html=True,
```


Menu
Control Devices
Add Device
Device Summary
Home Automation System
Control and manage your home appliances with this app.
Add a New Device
Device Name
Scanner
Add Device
Device 'Scanner' added successfully!

```
            )
        with col2:
            # Toggle button
            if st.button(f"Toggle {device}", key=f"toggle_{device}"):
                st.session_state.devices[device] = not state


# Add Device Section
elif menu == "Add Device":
    st.header("Add a New Device")
    new_device = st.text_input("Device Name")
    if st.button("Add Device"):
        if new_device and new_device not in st.session_state.devices:
            st.session_state.devices[new_device] = False
            st.success(f"Device '{new_device}' added successfully!")
        elif new_device in st.session_state.devices:
            st.warning(f"Device '{new_device}' already exists.")
        else:
            st.error("Please enter a valid device name.")


# Device Summary Section
elif menu == "Device Summary":
    st.header("Device Summary")
    if st.session_state.devices:
        for device, state in st.session_state.devices.items():
            col1, col2 = st.columns([3, 1])
            with col1:
```


Menu
Control Devices
Add Device
Device Summary
Home Automation System
Control and manage your home appliances with this app.
Device Summary
Living Room Light OFF
Bedroom Light OFF
Kitchen Light OFF
Fan OFF
Air Conditioner OFF
Heater OFF
Printer OFF
Scanner OFF

```
            st.write(f"{device}")
        with col2:
            st.write("ON" if state else "OFF")
    else:
        st.write("No devices found. Add some devices first!")


# Footer
st.markdown("---")
st.markdown("Powered by [Streamlit](https://streamlit.io)")
```

Conclusion:

The Home Automation system offers a secure, efficient, and convenient solution to traditional home management methods. By leveraging technology, it enhances user comfort, optimizes energy usage, and provides seamless control over smart devices. This project demonstrates the potential of technology to transform daily life, promoting a smarter and more sustainable living environment.