Smart Home Automation System

System Requirements Document

Author: Garrett Kellogg

Table of Contents

1. Customer Problem Statement and System Requirements

Customer Problem Statement

Modern households often require integrated control over various devices, but managing multiple smart devices independently can become cumbersome. Homeowners want a unified platform to control, schedule, and monitor their devices efficiently. This system is designed to provide a solution that is easy to operate, customizable, and ensures efficient energy usage and security for the household.

System Requirements

Non-functional Requirements:

• Usability: The system must be intuitive and user-friendly.
• Performance: The system should operate smoothly with minimal delays.
• Reliability: Ensure stable operation and accurate status reporting.
• Security: Access should be controlled based on user roles, with secure data handling.

- Functional Requirements:
  • Device Control
  • Role Management
  • Diagnostics
  • Scheduling
  • Notifications and Alerts

2. Functional Requirement Specification

1. Device Control

Description: Allows users to add, remove, and control devices.
Actors: Admin, Homeowner.
Preconditions: User must be logged in with appropriate access.

Flow:

1. 1. User selects a device from the list.
   2. User can toggle the device on or off.

2. Role Management

Description: Admin can assign roles with specific access levels.
Actors: Admin.
Preconditions: User must have Admin privileges.

Flow:

2. 1. Admin navigates to user roles.
   2. Admin selects a user and assigns or modifies the role.

3. Scheduling

Description: Users can set device actions to occur at specific times.
Actors: Homeowner.
Preconditions: Device must be available and online.

Flow:

3. 1. User selects a device and chooses an action.
   2. User schedules the action for a specified time.

4. Diagnostics

Description: Technicians can run diagnostic tests on devices.
Actors: Technician.
Preconditions: Device must be operational.

Flow:

4. 1. Technician selects a device to test.
   2. System runs diagnostic checks and returns a status report.

5. Alerts and Notifications

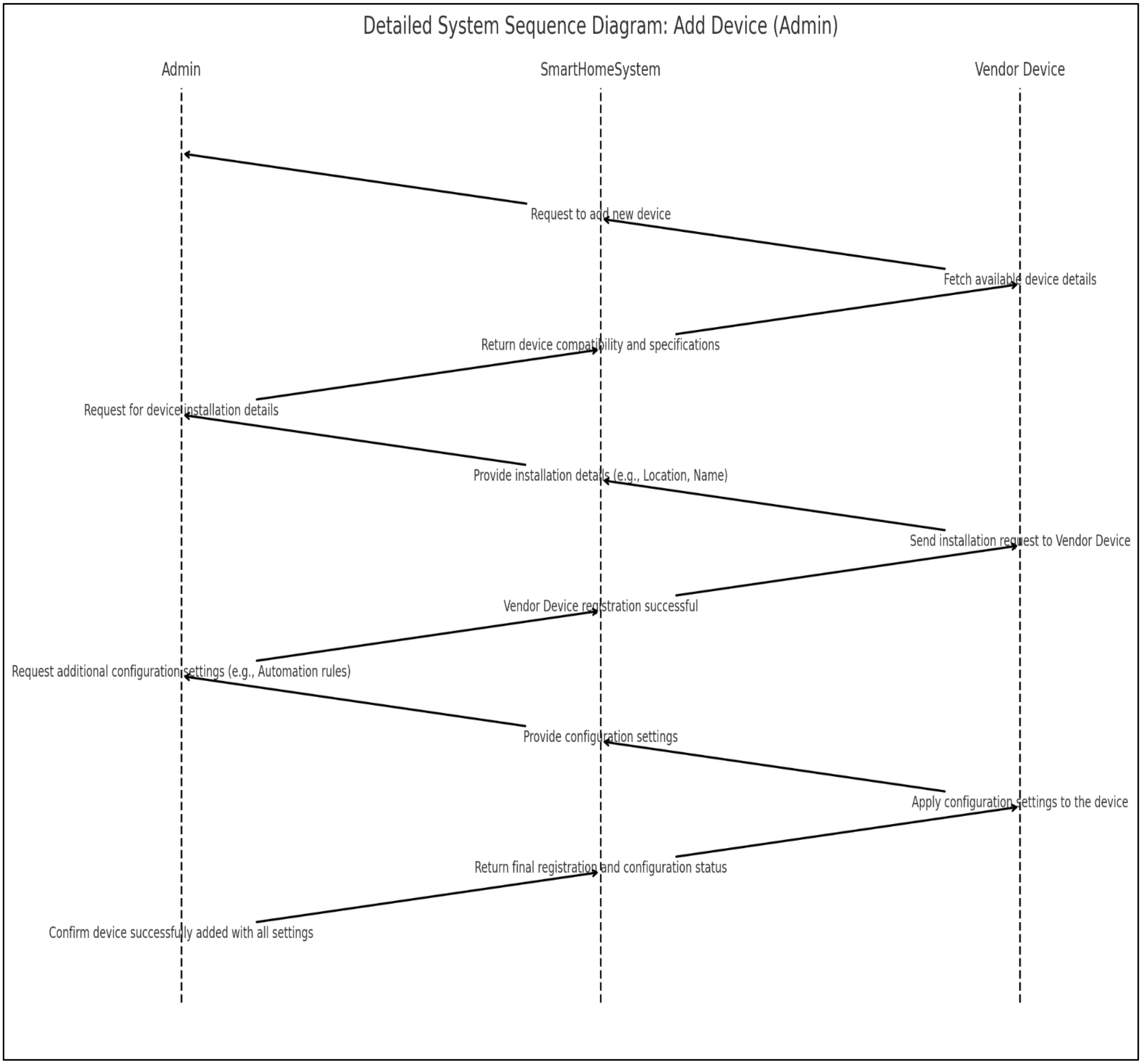
Description: System sends notifications for device status or scheduled actions.
Actors: All users.
Preconditions: Relevant events or schedules are in place.

Flow:

5. 1. User sets notification preferences.
   2. System sends alerts based on settings.

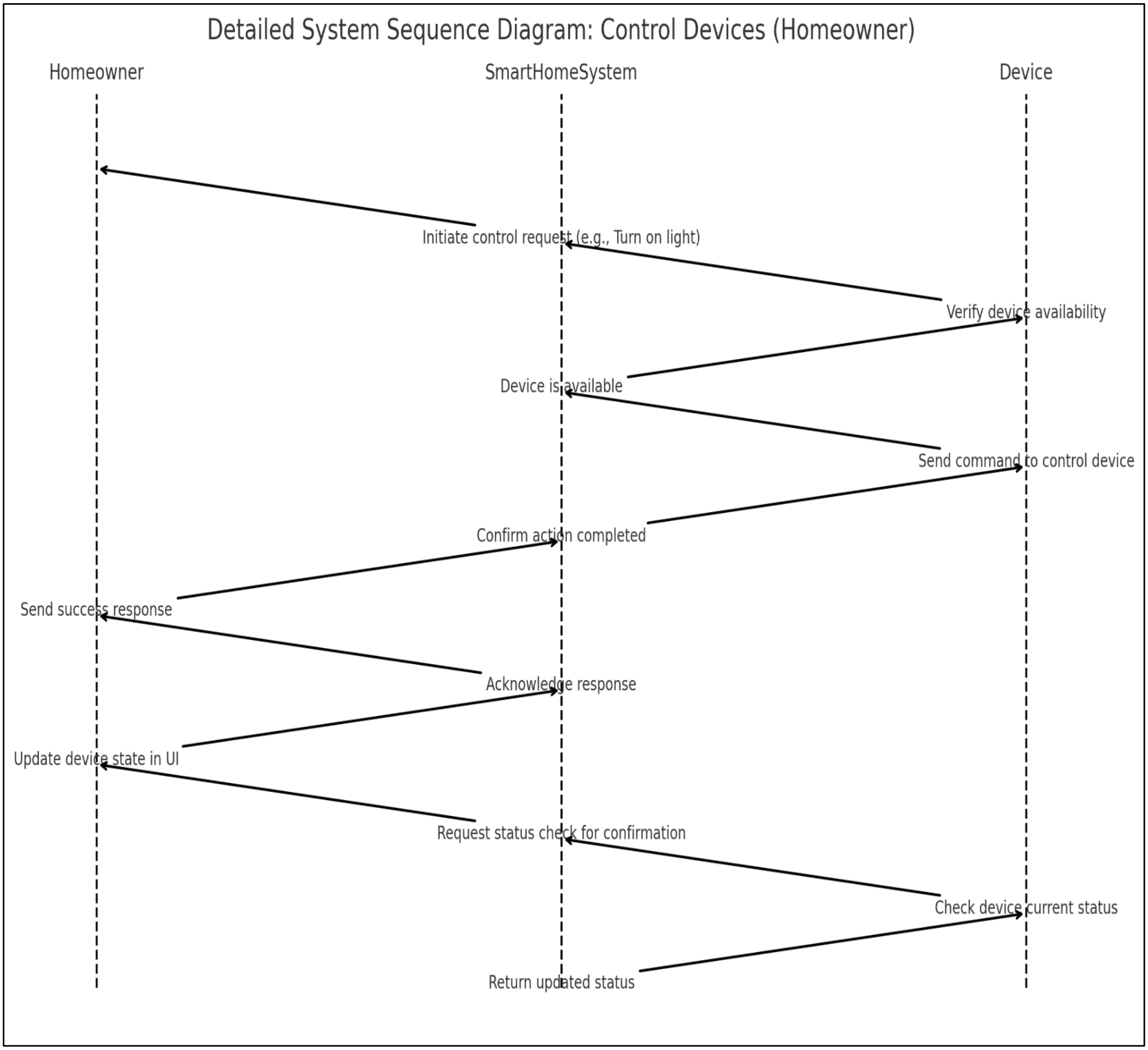
3. System Sequence Diagram

Sequence Diagram 1



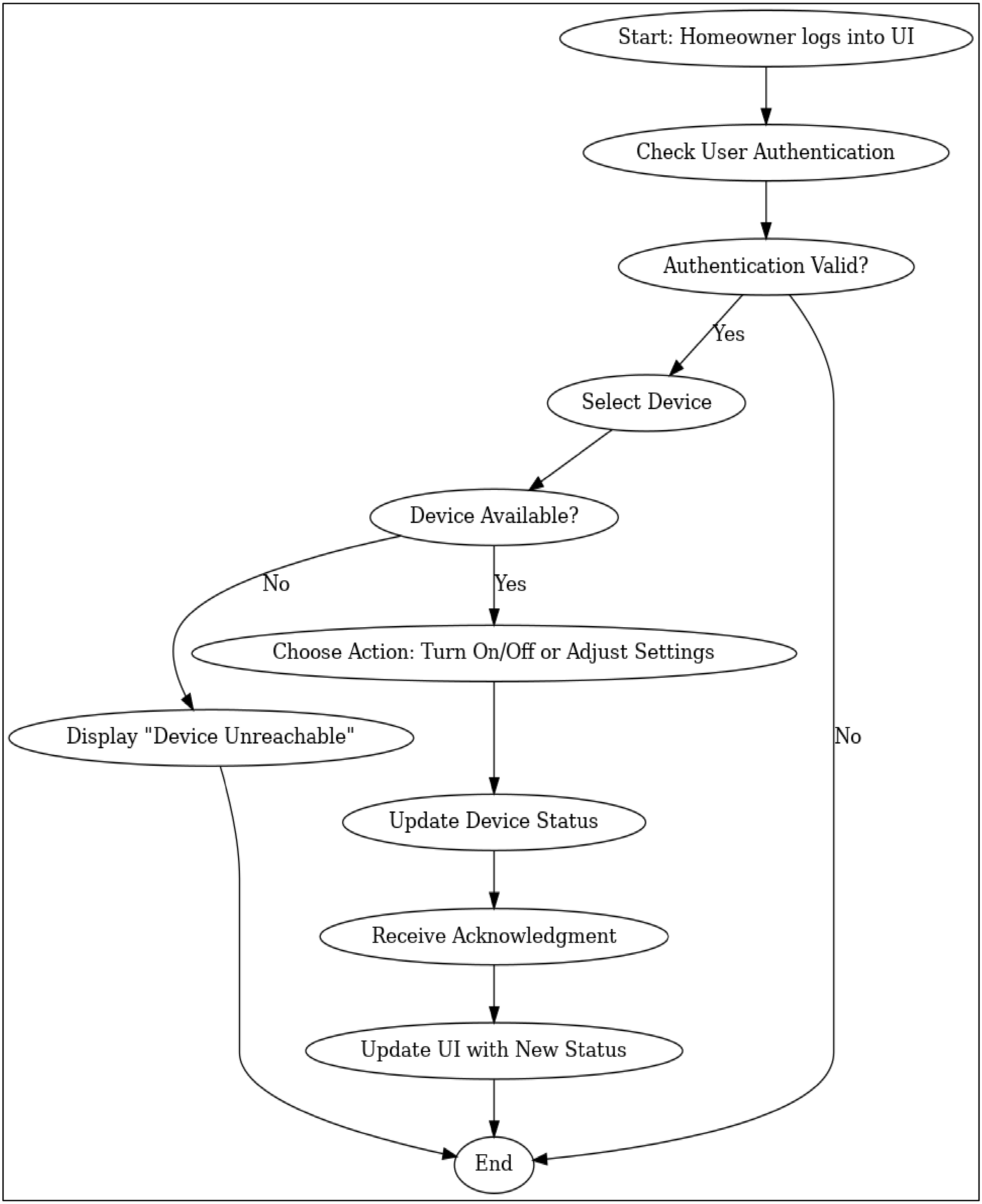
Detailed System Sequence Diagram: Add Device (Admin)
Admin
SmartHomeSystem
Vendor Device
Request to add new device
Fetch available device details
Return device compatibility and specifications
Request for device installation details
Provide installation details (e.g., Location, Name)
Send installation request to Vendor Device
Vendor Device registration successful
Request additional configuration settings (e.g., Automation rules)
Provide configuration settings
Apply configuration settings to the device
Return final registration and configuration status
Confirm device successfully added with all settings

Sequence Diagram 2

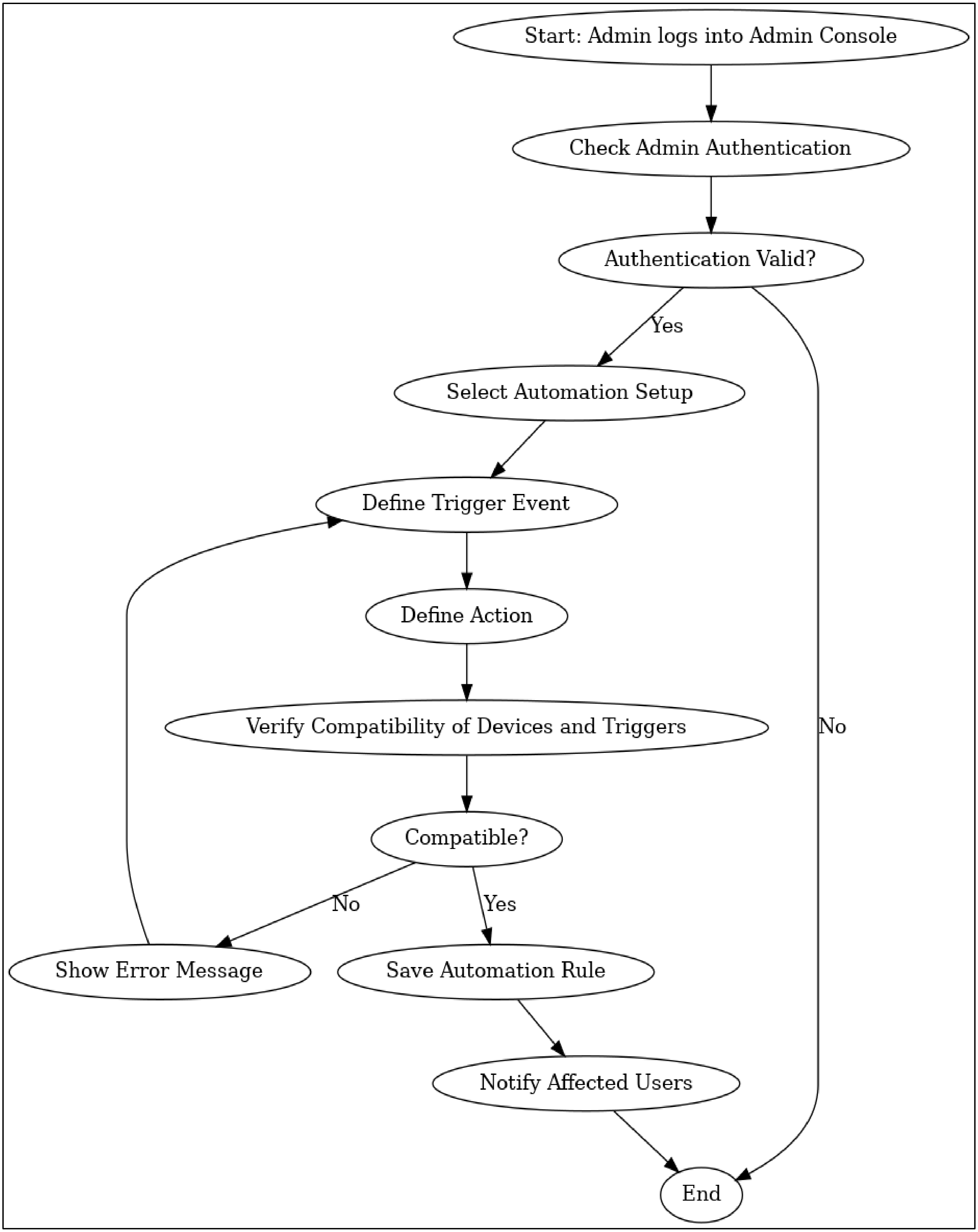

Detailed System Sequence Diagram: Control Devices (Homeowner)
Homeowner
SmartHomeSystem
Device
Initiate control request (e.g., Turn on light)
Verify device availability
Device is available
Send command to control device
Confirm action completed
Send success response
Acknowledge response
Update device state in UI
Request status check for confirmation
Check device current status
Return updated status

4. Activity Diagram

Activity Diagram 1:


Start: Homeowner logs into UI
Check User Authentication
Authentication Valid?
Yes
Select Device
Device Available?
No
Yes
Choose Action: Turn On/Off or Adjust Settings
Display "Device Unreachable"
No
Update Device Status
Receive Acknowledgment
Update UI with New Status
End

Activity Diagram 2:



5. User Interface Specification

Main Dashboard

The central hub for all users, providing quick access to the system's primary functionalities. The dashboard includes a tiled layout where each tile represents a key feature (Manage Devices, User Roles, Scheduling, Diagnostics, Alerts). Users can click on each tile to access specific features. The dashboard is customized based on user roles, displaying relevant features according to the user's permissions.

Device Management Screen

This screen allows users to manage and control connected devices. It displays a list of devices with their names, types, and current statuses. Users with appropriate permissions can add or remove devices from the system, toggle device statuses, and view additional details by selecting a device. Each device in the list includes an on/off toggle switch and a settings option for advanced configurations.

Role Management Screen

Designed for Admin users to assign roles and manage user access within the system. The screen displays a list of all registered users along with their assigned roles (Admin, Homeowner, Technician). Admin users can update roles by selecting a user and choosing a new role from a dropdown menu. Changes in role assignments are reflected across the system instantly, impacting the user's access to features.

Scheduling Screen

Provides an interface for scheduling actions for specific devices. Users can select a device, choose an action (e.g., Turn On, Turn Off), and set a time or recurrence for the action. The scheduling interface includes an easy-to-use date and time picker, along with options for recurring schedules. Scheduled actions are listed on the screen with details like device name, action, and scheduled time.

Diagnostics Screen

The diagnostics screen is accessible to Technicians for troubleshooting and running diagnostic tests on devices. The screen displays a list of devices that can be selected for diagnostics. Users can run tests by selecting a device, which initiates a diagnostic process that checks device connectivity, health, and performance. Test results are displayed on the screen, providing Technicians with data to assess device functionality.

Alerts and Notifications Screen

This screen shows all alerts and notifications for the user, including device status changes, scheduled action completions, and system alerts. Each alert displays a brief description, timestamp, and read/unread status. Users can acknowledge alerts by marking them as read, helping them keep track of system activity. Notifications are organized chronologically to ensure that the most recent alerts are easily accessible.

6. Traceability Matrix

System Requirements

No.	Priority Weight (1-5)	Description
REQ1	5	Enable user login/logout functionality
REQ2	2	Provide real-time alerts and notifications
REQ3	5	Admin and Agents can log in/out from the system
REQ4	4	Allow device scheduling functionality
REQ5	2	Enable user role management
REQ6	1	Provide diagnostics for devices
REQ7	2	Allow device status monitoring
REQ8	1	View activity history of devices
REQ9	1	Display system usage analytics

Use Cases

No.	Description
UC1	User can log in and log out of the system
UC2	System sends alerts and notifications based on events
UC3	Admin and Agents have access control over login/logout
UC4	Users can schedule actions for specific devices
UC5	Admin manages user roles and permissions
UC6	Technicians can run diagnostics on devices
UC7	User can monitor the status of each device

No.	**Description**
UC8	Users can view activity history of devices

Traceability Matrix

Requirement	**UC1**	**UC2**	**UC3**	**UC4**	**UC5**	**UC6**	**UC7**	**UC8**	**Max PW**	**Total PW**
REQ1	1	0	1	0	0	0	0	0	5	10
REQ2	0	1	0	0	0	0	0	0	2	2
REQ3	1	0	1	0	0	0	0	0	5	10
REQ4	0	0	0	1	0	0	0	0	4	4
REQ5	0	0	0	0	1	0	0	0	2	2
REQ6	0	0	0	0	0	1	0	0	1	1
REQ7	0	0	0	0	0	0	1	0	2	2
REQ8	0	0	0	0	0	0	0	1	1	1
REQ9	0	0	0	0	0	0	0	0	1	0

7. System Architecture and System Design

1. Architectural Styles

The Smart Home Automation System adopts the following architectural styles:

- **Client-Server Architecture**: Enables separation of the user interface and backend processing for device control and user management.

- **Layered Architecture**: Divides the system into layers for UI, business logic, and data storage, ensuring scalability and maintainability.

- **Event-Driven Architecture**: Handles real-time notifications and triggers based on device activity or user input.

2. Identifying Subsystems

The system is composed of several subsystems, each representing a specific functional area. The key subsystems are:

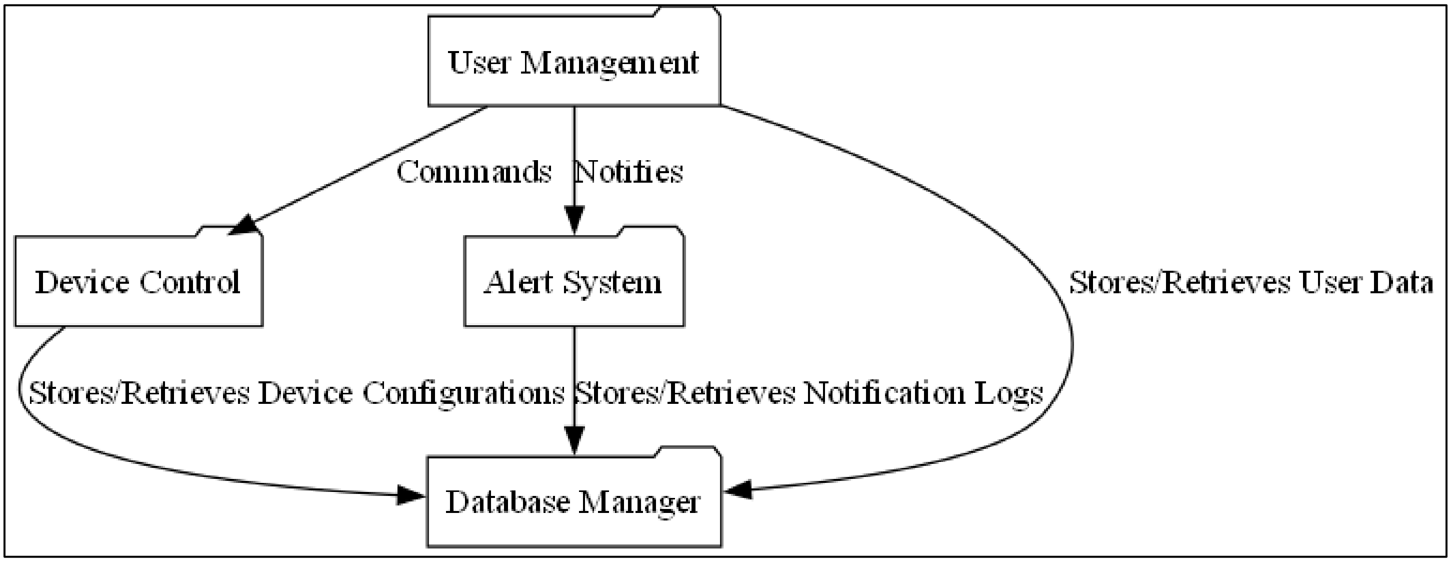
- **User Management**: Handles login/logout functionality, role assignment, and user permissions.

- **Device Control**: Allows users to monitor and control smart devices.

- **Alert System**: Sends real-time notifications and device diagnostics.

- **Database Manage**: Manages persistent storage of user data, device configurations, and logs.

A UML package diagram of these subsystems is included to visually represent their structure.


User Management
Commands
Notifies
Device Control
Alert System
Stores/Retrieves User Data
Stores/Retrieves Device Configurations
Stores/Retrieves Notification Logs
Database Manager

3. Mapping Subsystems to Hardware

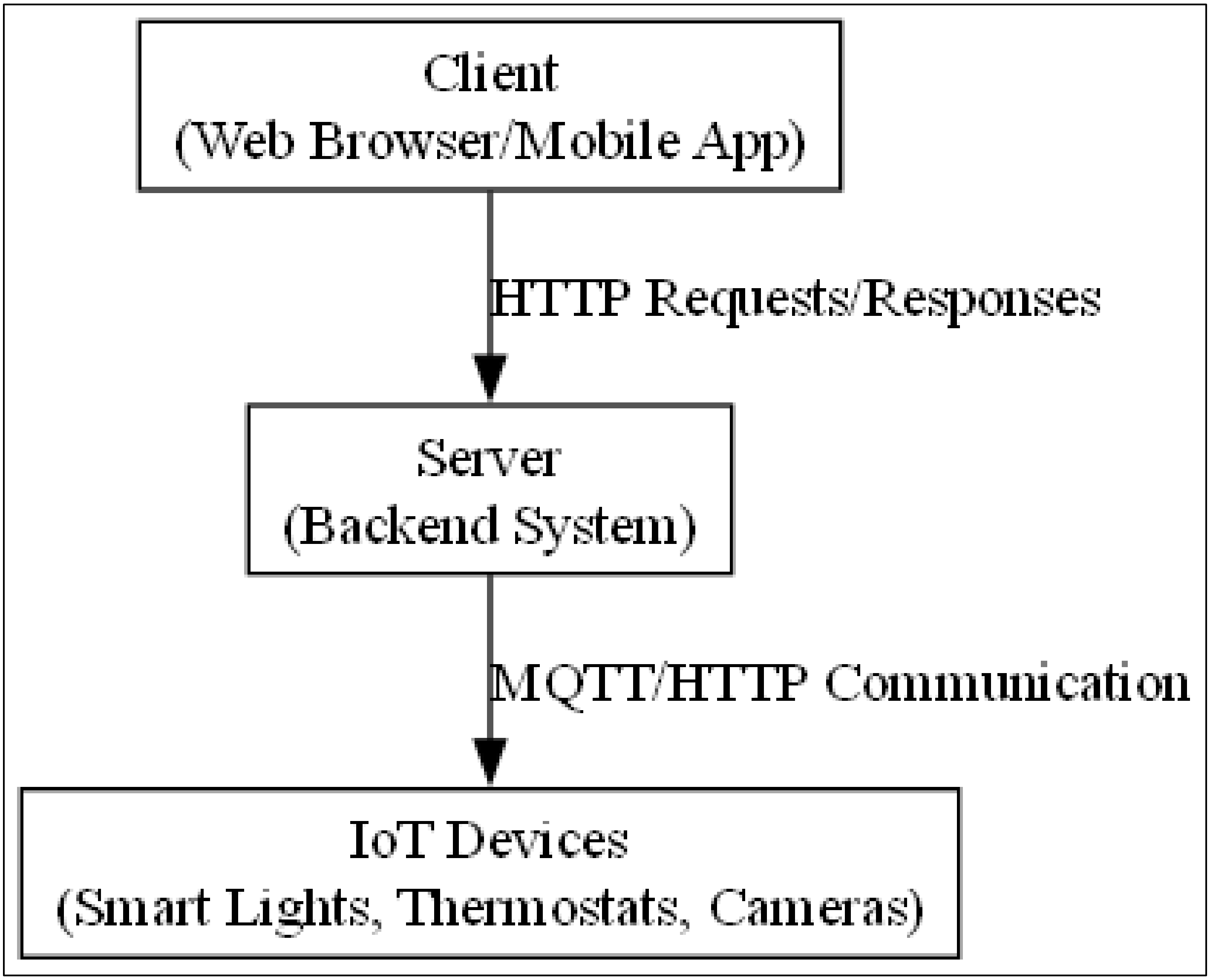
The Smart Home Automation System runs on a distributed architecture. Key subsystems and their corresponding hardware mappings are:

- **Client**: Web browser or mobile app for user interaction.

- **Server**: Backend server for processing requests and managing persistent data.

- **IoT Devices**: Smart devices such as lights, thermostats, and cameras that interact with the system.

A hardware mapping diagram is provided to illustrate these relationships.



4. Persistent Data Storage

The system requires persistent storage for user data, device configurations, activity logs, and scheduled actions. A relational database such as MySQL is used to store structured data, while flat files may be used for logging.

5. Network Protocol

The system communicates over a network using the HTTP protocol for client-server communication. For IoT devices, MQTT is employed due to its lightweight nature and efficiency in handling real-time updates.

6. Global Control Flow

- **Execution Order:** The system is event-driven, responding to user commands or device triggers.

- **Time Dependency**: Periodic diagnostics occur every 24 hours, while alerts are handled in real-time.

- **Concurrency**: Multithreading is employed to handle simultaneous user requests and device interactions.

Synchronization mechanisms such as locks ensure data consistency.

7. Hardware Requirements

The system requires the following hardware specifications:

- **Display**: A minimum resolution of 800 × 600 pixels.

- **Storage**: At least 5 GB for database and log files.

- **Network**: A minimum bandwidth of 128 Kbps for real-time updates.

- **IoT Devices**: Compatible devices with proper APIs for integration.