ITPR7.508: Business Application Programming (BAP)

2024

PROJECT:
Emergency Device Management System *(EDMS)*

Project Client	E.I.T. Jonathan Bixley (EIT Facilities Coordinator & Maintenance)
Team Members	+ Liam Palmers + Alex Scott + Qiao Yin + Aidan Willis + James Sadler
Observer/Mentor	Daniel Dang
Document	Technical Specifications and Design Document Version: 2.0

Contents

INTRODUCTION

Problems and the application/software aims to address

EIT currently relies on an Excel spreadsheet to track fire extinguishers, which is insufficient for their needs. Jonathan Bixley, the Facilities Operations Coordinator, is tasked with ensuring that each fire extinguisher is inspected four times annually, following a detailed checklist.
The current system makes it difficult to track these inspections and maintain up-to-date records, potentially leading to safety risks. Fire extinguishers have a lifespan of five years, so keeping track of their replacement is crucial. The limitations of the spreadsheet highlight the need for a more robust digital solution to ensure compliance and safety across the campus.

Project and objectives

Project Overview:
The project aims to replace EIT's Excel-based fire extinguisher tracking system with a scalable digital solution that improves safety compliance, data management, and efficiency.

Objectives:

- Track and Manage Devices:
  Enable tracking of fire extinguishers and safety devices with details like type, location, inspection dates, and expiration dates.

- Automated Notifications:
  Automatically notify users of upcoming inspections and replacements to ensure compliance.

- Admin and User Accounts:
  Implement user accounts with admins managing devices and users viewing records.

- Interactive Mapping:
  Provide an interactive map to locate devices visually across campus.

- Scalability:
  Ensure the system supports future devices and multiple locations.

Application: key features and functionalities

This application will include a user-friendly interface to display device information, as well as features to add, delete, and update device records. Users will be able to search for specific devices, and the system will differentiate between user and admin accounts to control access to certain functions. Automated notifications will be sent to alert users of upcoming device expirations and required maintenance. Additionally, the application will include an interactive map to visually track device locations, enhancing the overall management and safety of the campus.

USE CASES / USER STORIES / SCENARIOS

Use case 1: ....

+ Role (actor) — “As a User”
+ Feature — “I want to log in to the system using my credentials”
+ Benefit — “So that I can securely access my account and its associated features.”

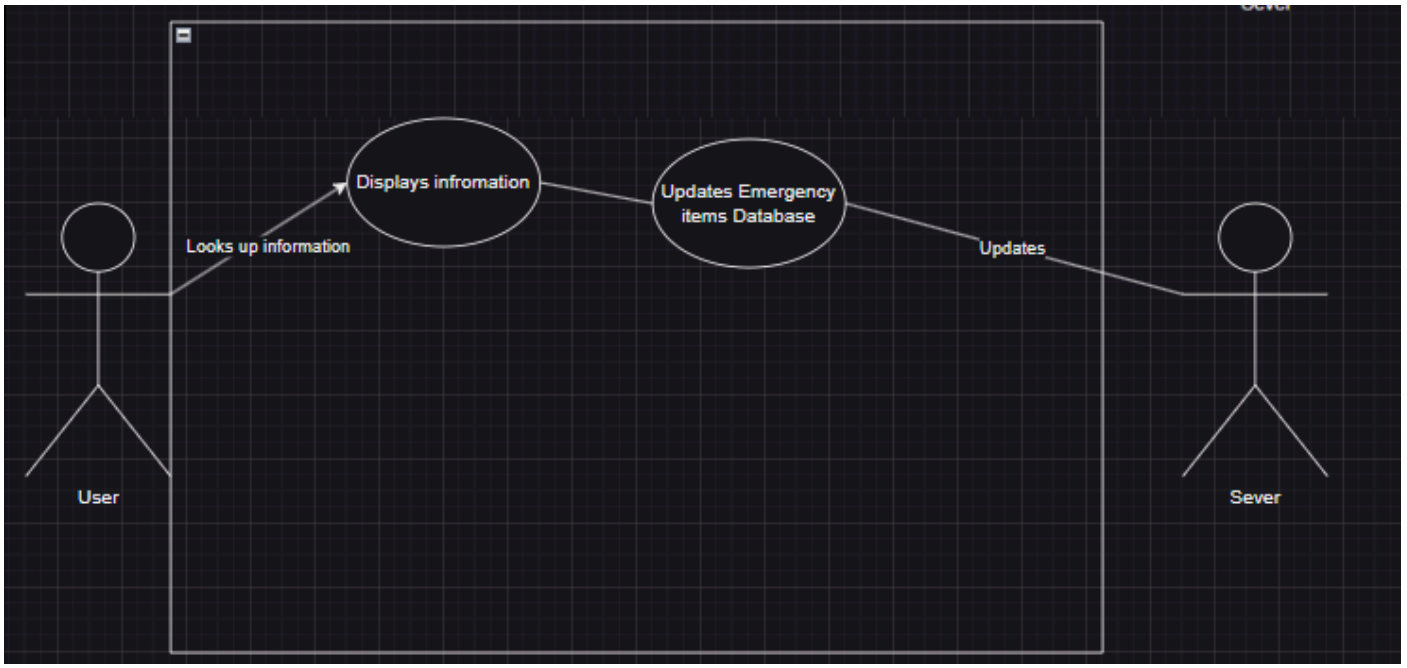


Figure 1.1: Use Case 1

Use case 2: ....

+ Role (actor) — “As a New User”
+ Feature — “I want to create a new account”
+ Benefit — “So that I can access the emergency device management system.”

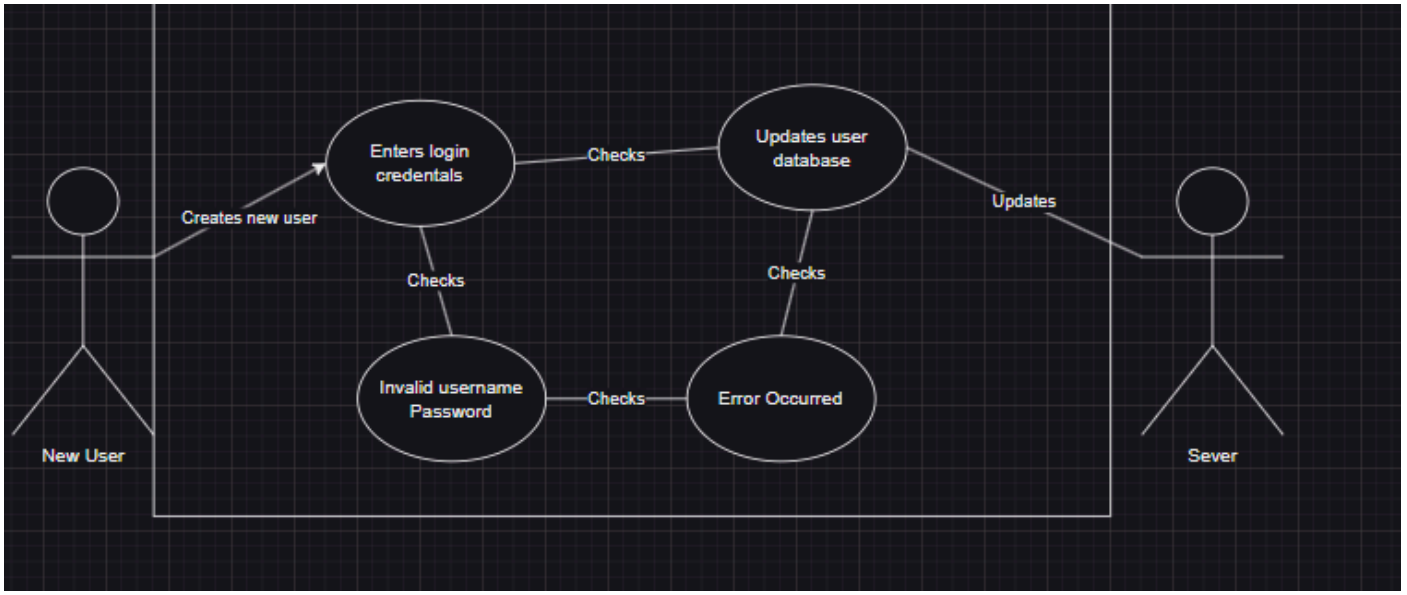


Figure 1.2: Use Case 2

Use case 3: ....

+ Role (actor) — “As an Admin”
+ Feature — “I want to create, update, or delete emergency devices in the system”
+ Benefit — “So that I can maintain an accurate inventory and manage device lifecycles efficiently.”

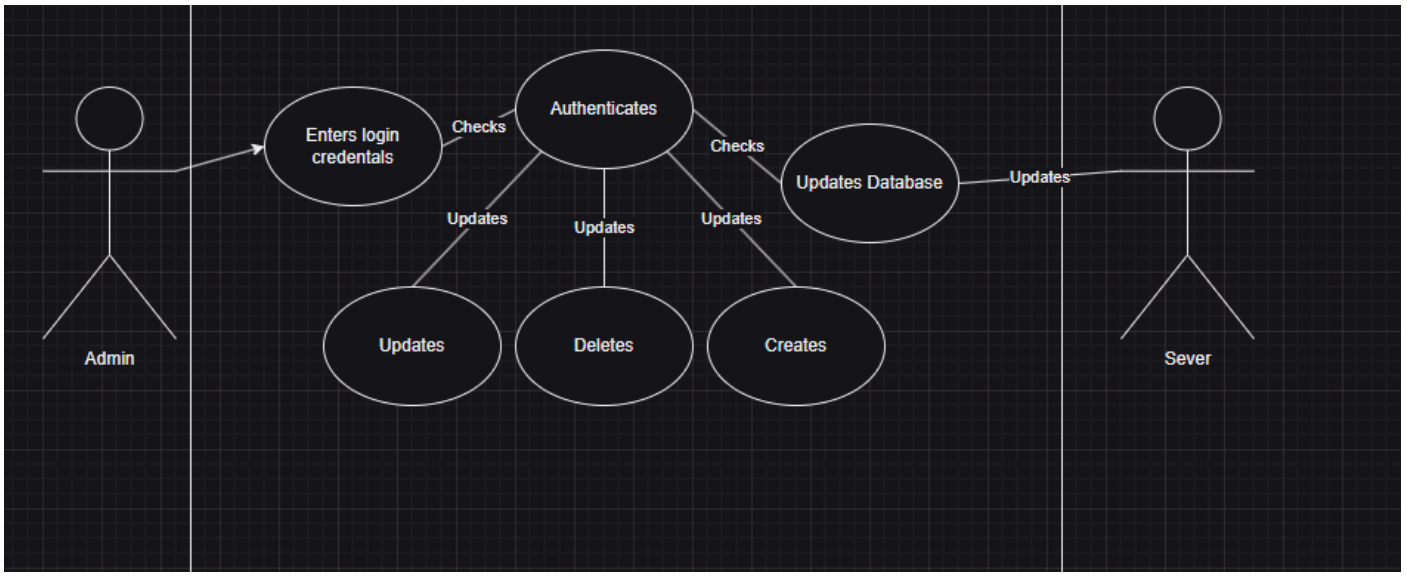


Figure 1.3: Use Case 3

Use case 4: ....

+ Role (actor) — “As a User”
+ Feature — “I want to view and filter emergency devices by location, type, or status using the interactive map, dropdowns, and search functions”
+ Benefit — “So that I can easily find specific devices based on various criteria”

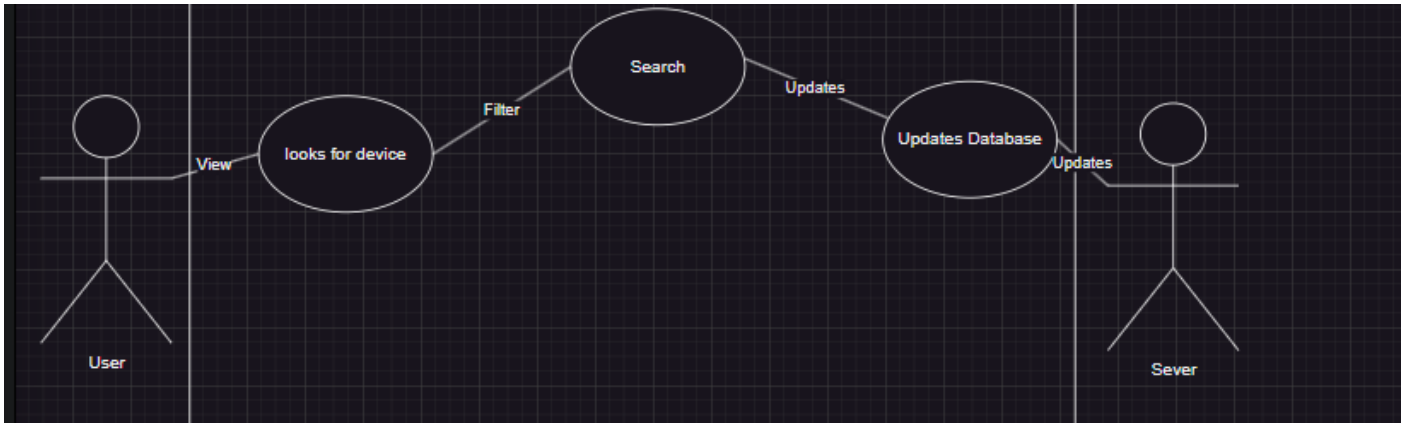


Figure 1.4: Use Case 4

Use case 5: ....

+ Role (actor) — “As an Admin”
+ Feature — “I want to create, update, or delete inspection records for fire extinguishers”
+ Benefit — “So that I can ensure compliance with inspection schedules and keep records up-to-date”

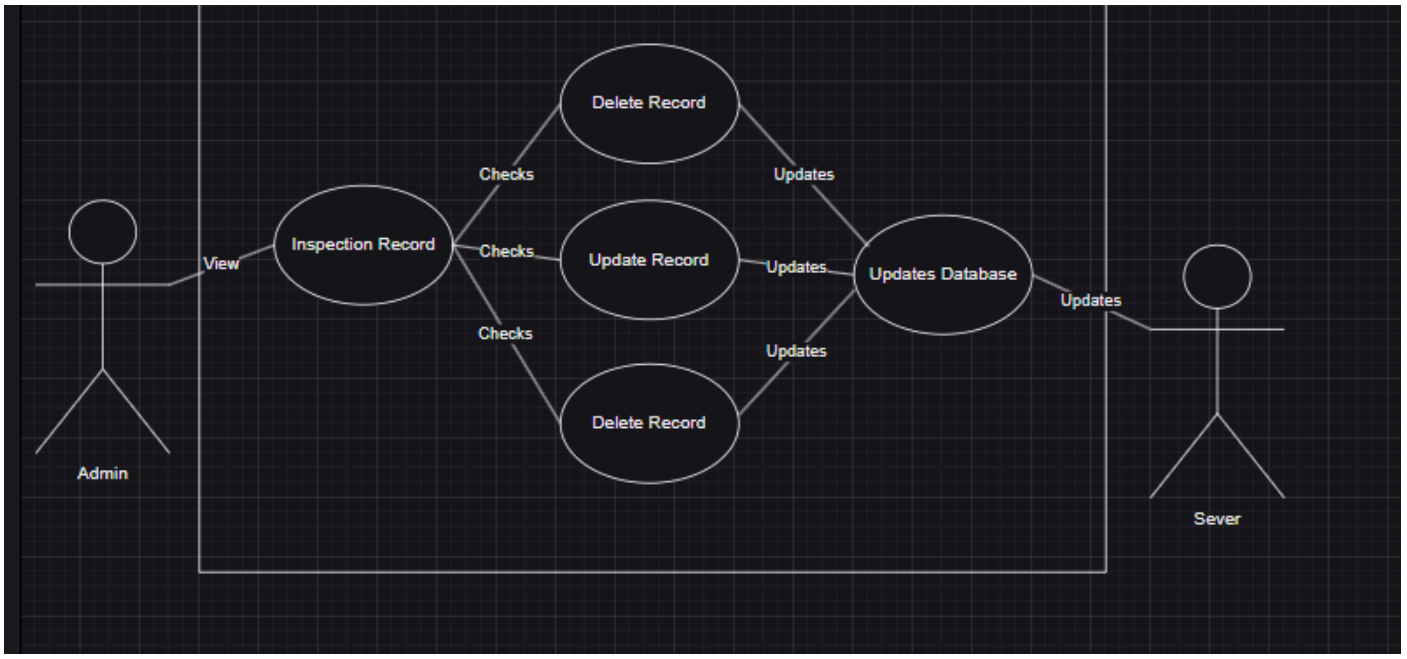


Figure 1.5: Use Case 5

Use case 6: ....

+ Role (actor) — “As a User”
+ Feature — “I want to receive notifications for upcoming device inspections and expirations”
+ Benefit — “So that I can stay informed about required actions and maintain safety compliance”

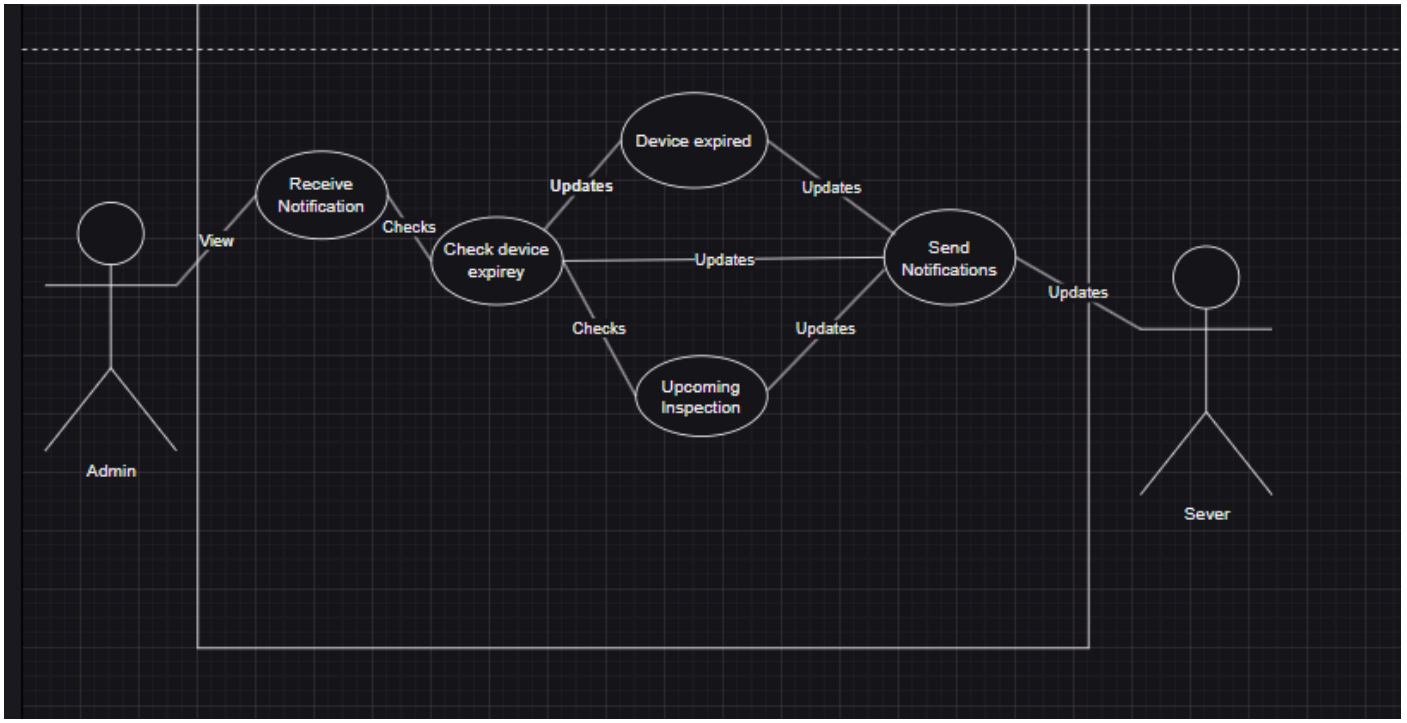


Figure 1.6: Use Case 6

Use case 7: ….

+ Role (actor) — “As an Admin”
+ Feature — “I want to create, update, or delete room, building, and site data”
+ Benefit — “So that I can accurately track where devices are located across the campus”

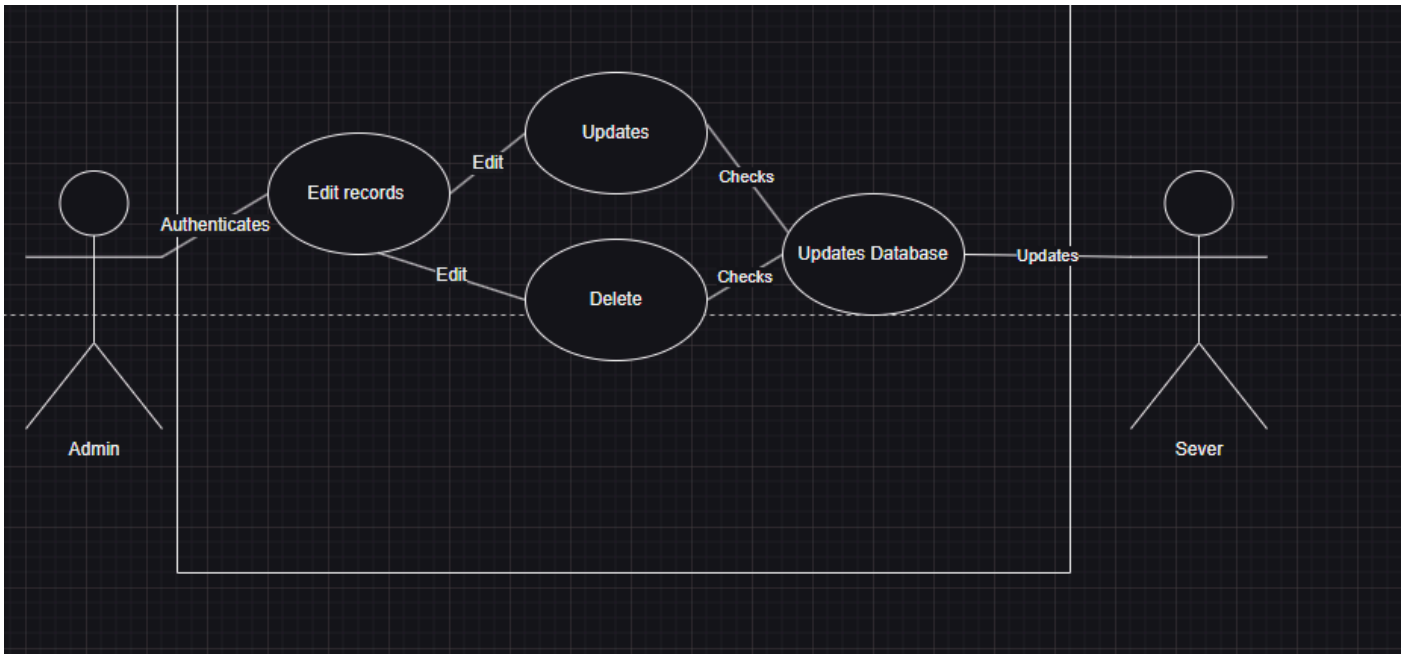


Figure 1.7: Use Case 7

Use case 8: ….

+ Role (actor) — “As an Admin”
+ Feature — “I want to create, update, or delete user accounts and device types”
+ Benefit — “So that I can manage system access and maintain accurate records of different emergency devices”

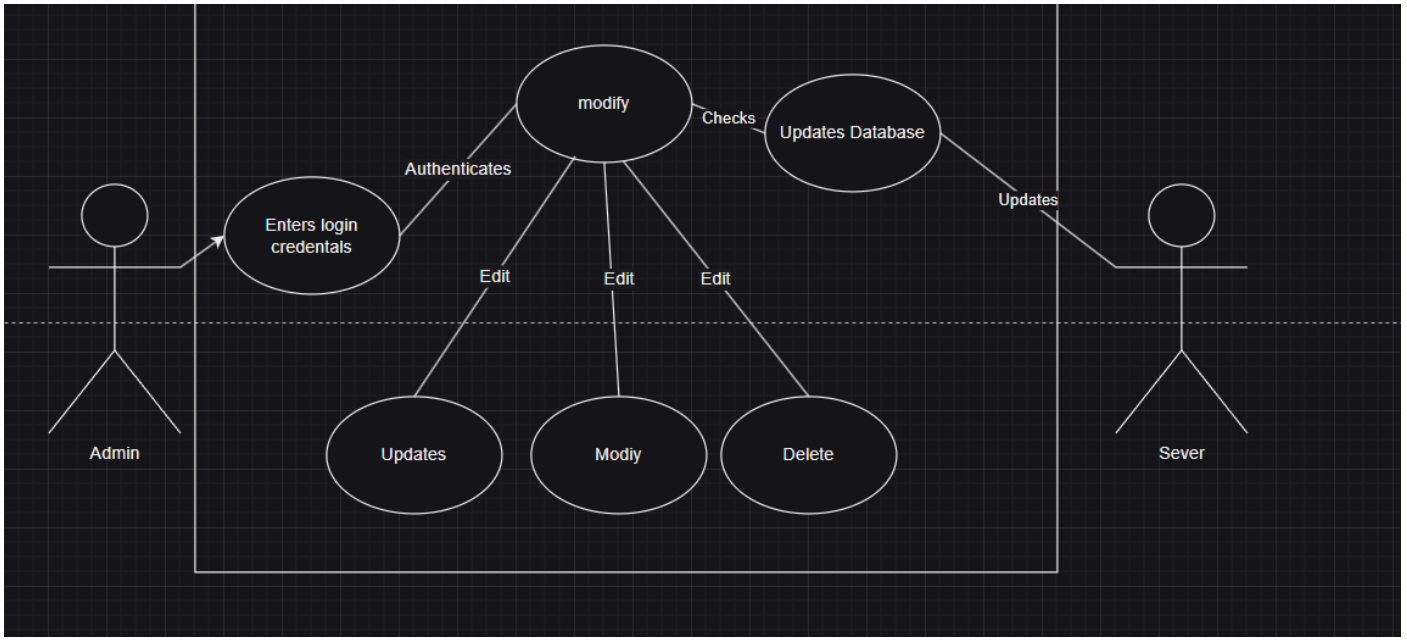


Figure 1.8: Use Case 8

Use case 9: ….

+ Role (actor) — “As an Admin”
+ Feature — “I want to login and see a notifications”
+ Benefit — “to make sure all the devices are kept up to date”

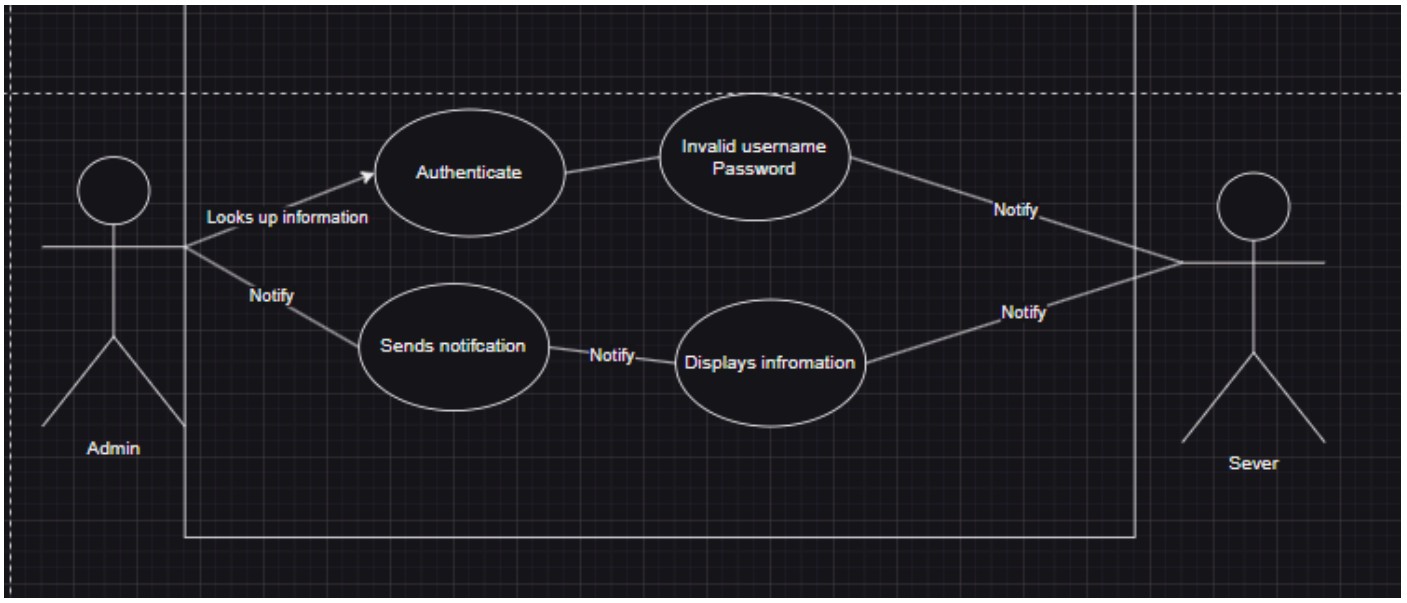


Figure 1.9: Use Case 9

SYSTEM / COMPONENT DIAGRAMS

+ System Diagram 1 based on three-tier client-server architecture:

The three major components in the client-server model: presentation, server (application/business logic), and data storage.

- "Client" – "Server" (business logic) – "database":

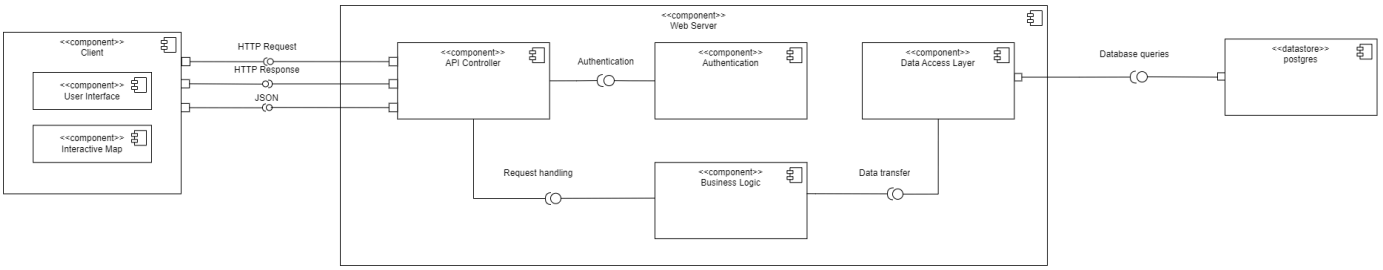


Figure 2.1: System Diagram 1 of the proposed web application

+ Component Diagram 2 based on on application functions:

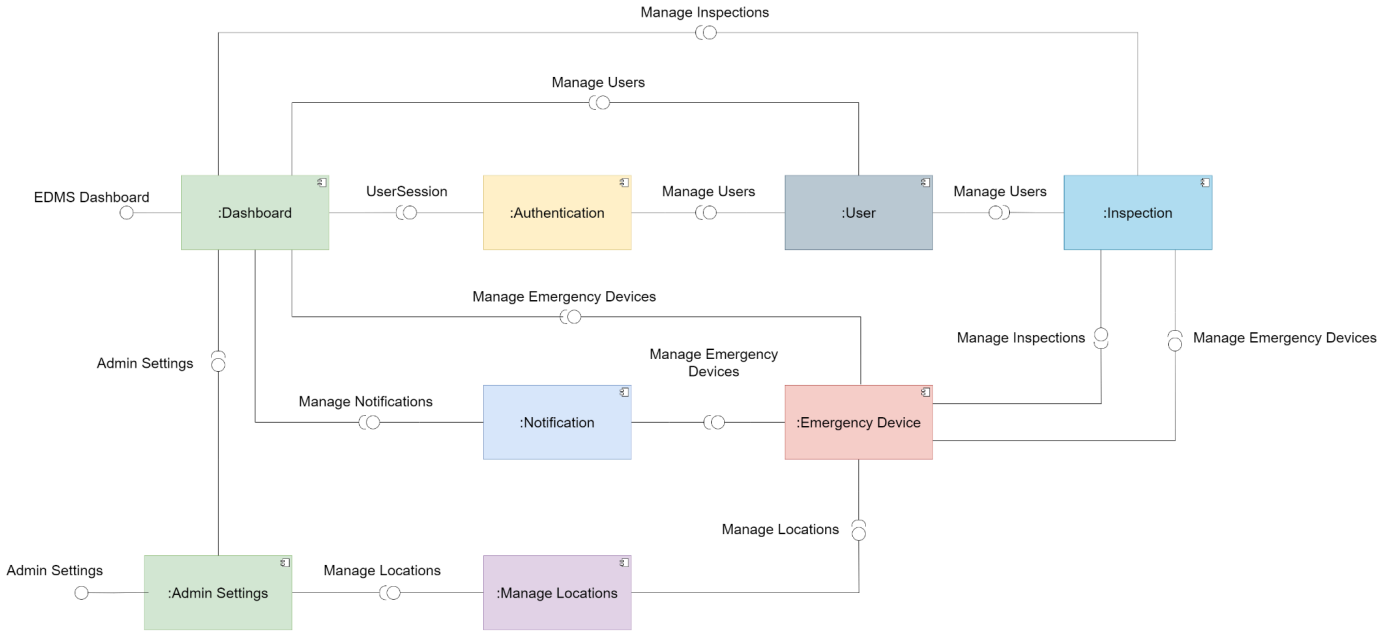


Figure 2.2: Component Diagram 2 of the proposed web application

ACTIVITY DIAGRAMS:

COMPONENT: "AUTHENTICATION"

- **Purpose:**
  - The "Authentication" component is responsible for verifying user credentials to ensure that only authorized individuals can access the system. It manages the login, registration, and password recovery processes.
- **Inputs:**
  - Login: Username/email and password.
  - Registration: New user details including username, email, and password.
  - Forgot Password: Email address for password reset requests
- **Operation:**
  - Login: Validates user credentials against stored data. If correct, access to the system is granted.
  - Registration: Saves new user details in the database, creates a new user account, and displays a success message.
  - Forgot Password: Sends a password reset email to the provided address and displays a success message.
- **Outputs:**
  - Successful Login: Grants access to the system.
  - Successful Registration: Creates a new user account and displays a success message.
  - Successful Password Recovery: Sends a password reset email and displays a success message.
  - Errors: Messages for failed login attempts, registration issues, or password recovery problems.

Authentication Process

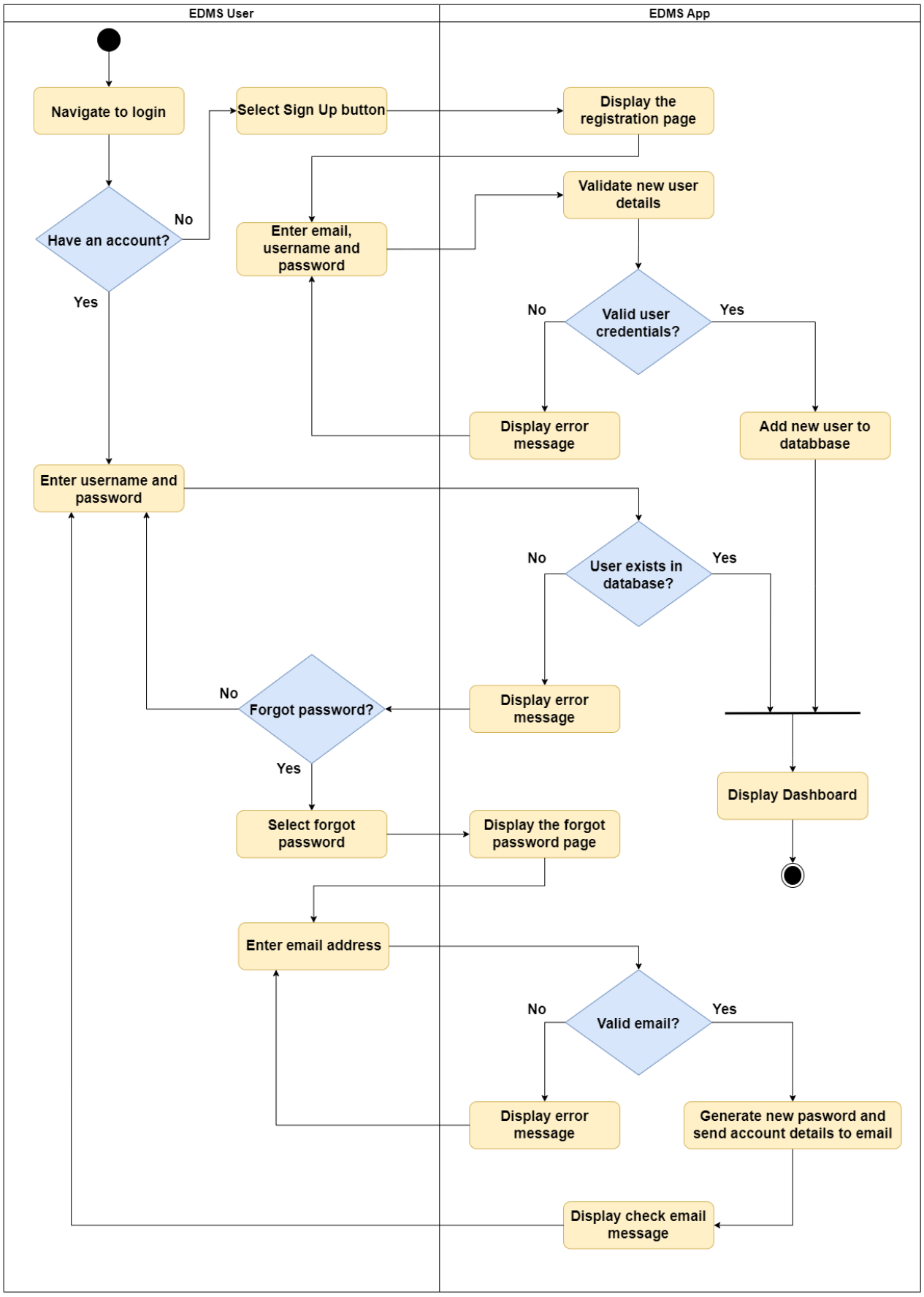


Figure 3.1: Activity Diagram for the authentication process of the proposed web application

COMPONENT: "ADMIN"

- **Purpose:**
  - The "Admin" component is used to manage location and user information in the system, like locations (Sites, Buildings, Rooms), users, and device types. It allows admins to add, view, update, and delete records for these categories.
- **Inputs:**
  - Admin actions:
    - Add, view, update, and delete Locations (Sites, Buildings, Rooms)
    - View, update, and delete Users and their roles
    - Add, view, update, and delete Device Types
  - Location Data: Names of sites, addresses, building codes, room codes
  - User Data: Username, email, roles
  - Device Type Data: Device Type names
- **Operation:**
  - Add new sites, buildings, or rooms
  - View details of existing locations
  - Update information for sites, buildings, or rooms
  - Delete sites, buildings, or rooms
  - View user accounts
  - Update user information (e.g. , role)
  - Delete user accounts
  - Add new device types
  - View existing device types
  - Update device type information
  - Delete device types
- **Outputs:**
  - Records: New and updated Site, Building, Room, User and Device Type records stored in the database.
  - Messages: Success and error notifications for creation, updates, or deletions.

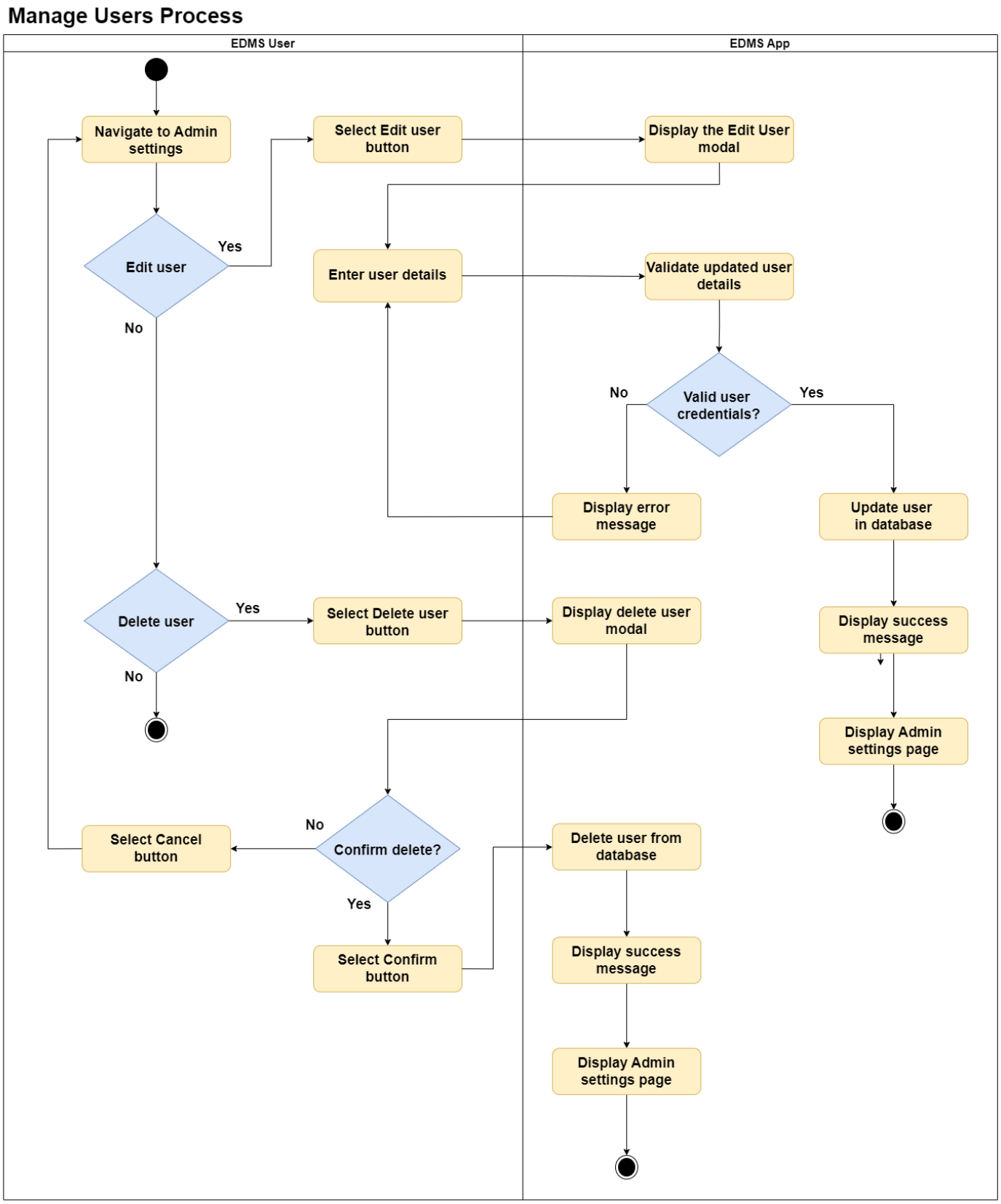


Figure 3.2: Activity Diagram for the manage users process of the proposed web application

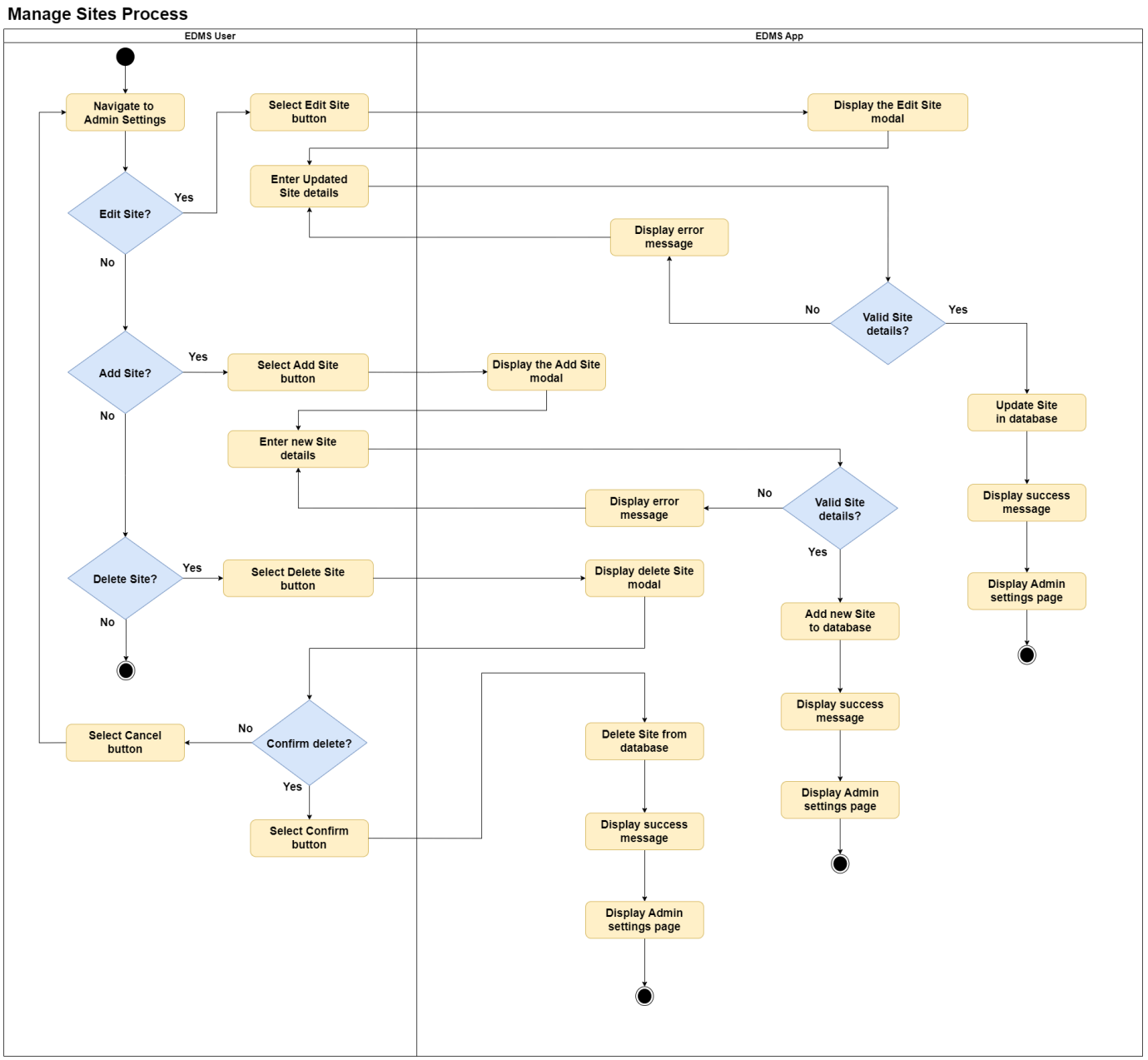


Figure 3.3: Activity Diagram for the manage sites process of the proposed web application

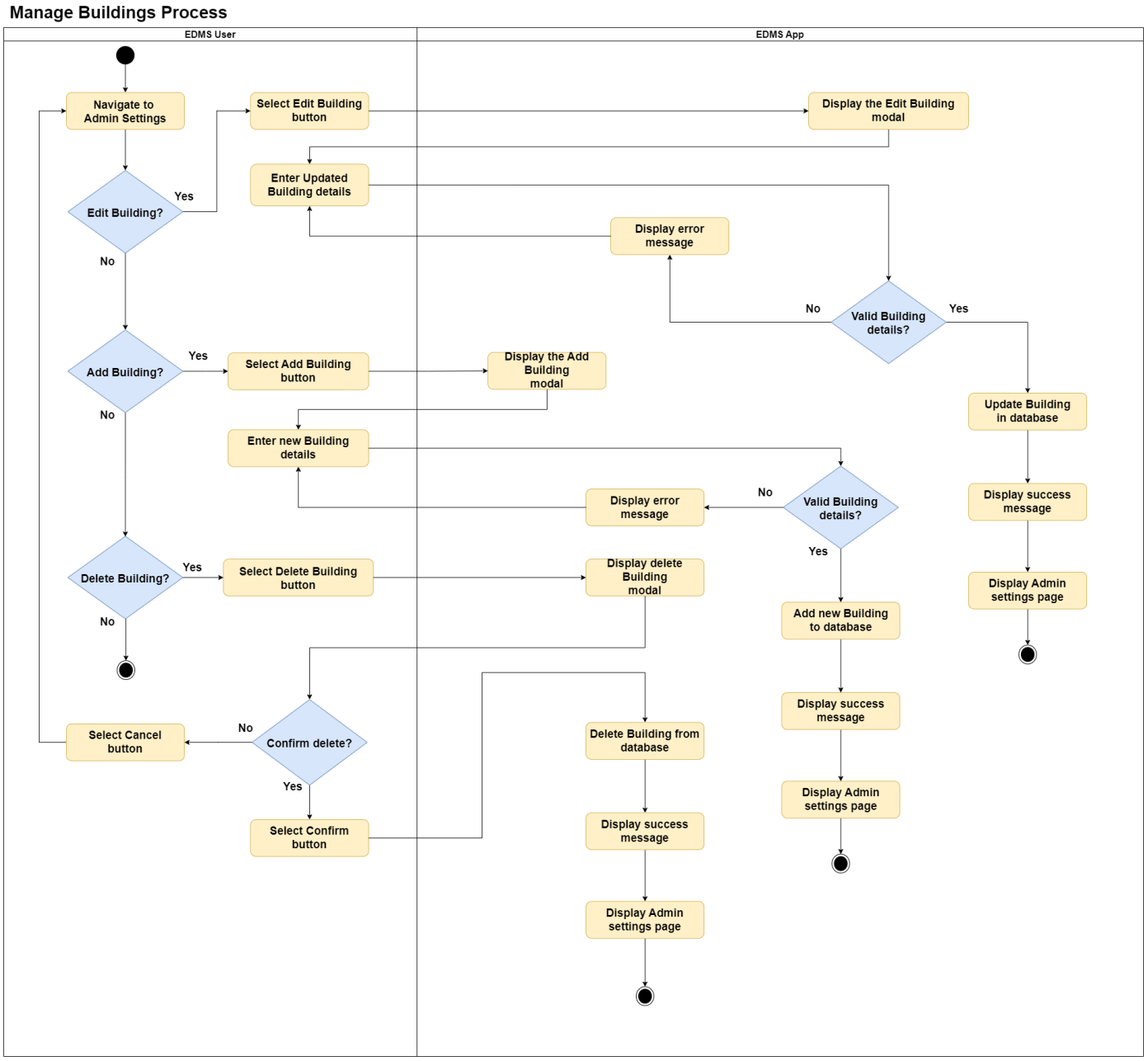


Figure 3.4: Activity Diagram for the manage buildings process of the proposed web application

Manage Rooms Process

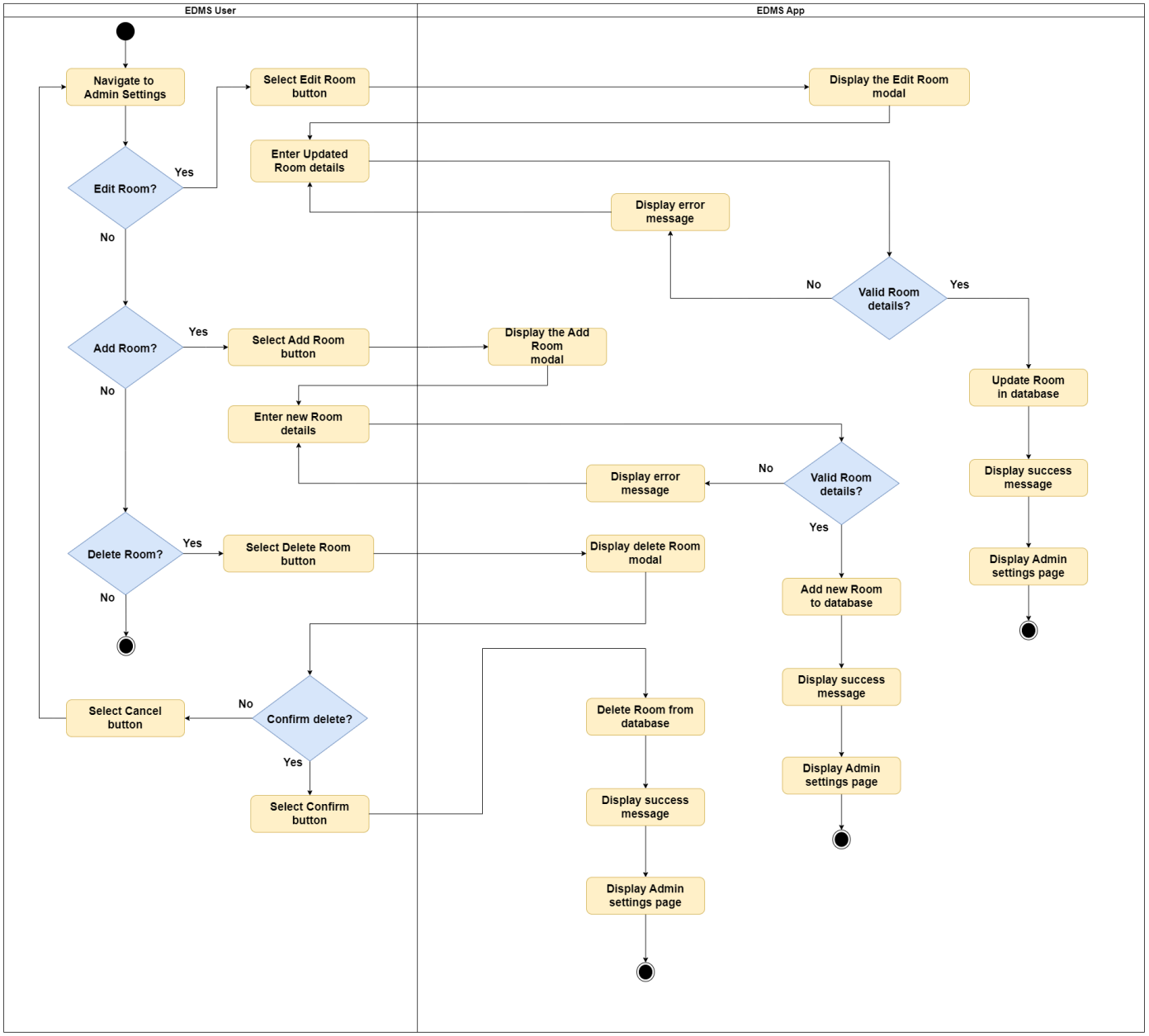


Figure 3.5: Activity Diagram for the manage rooms process of the proposed web application

Manage Device Types Process

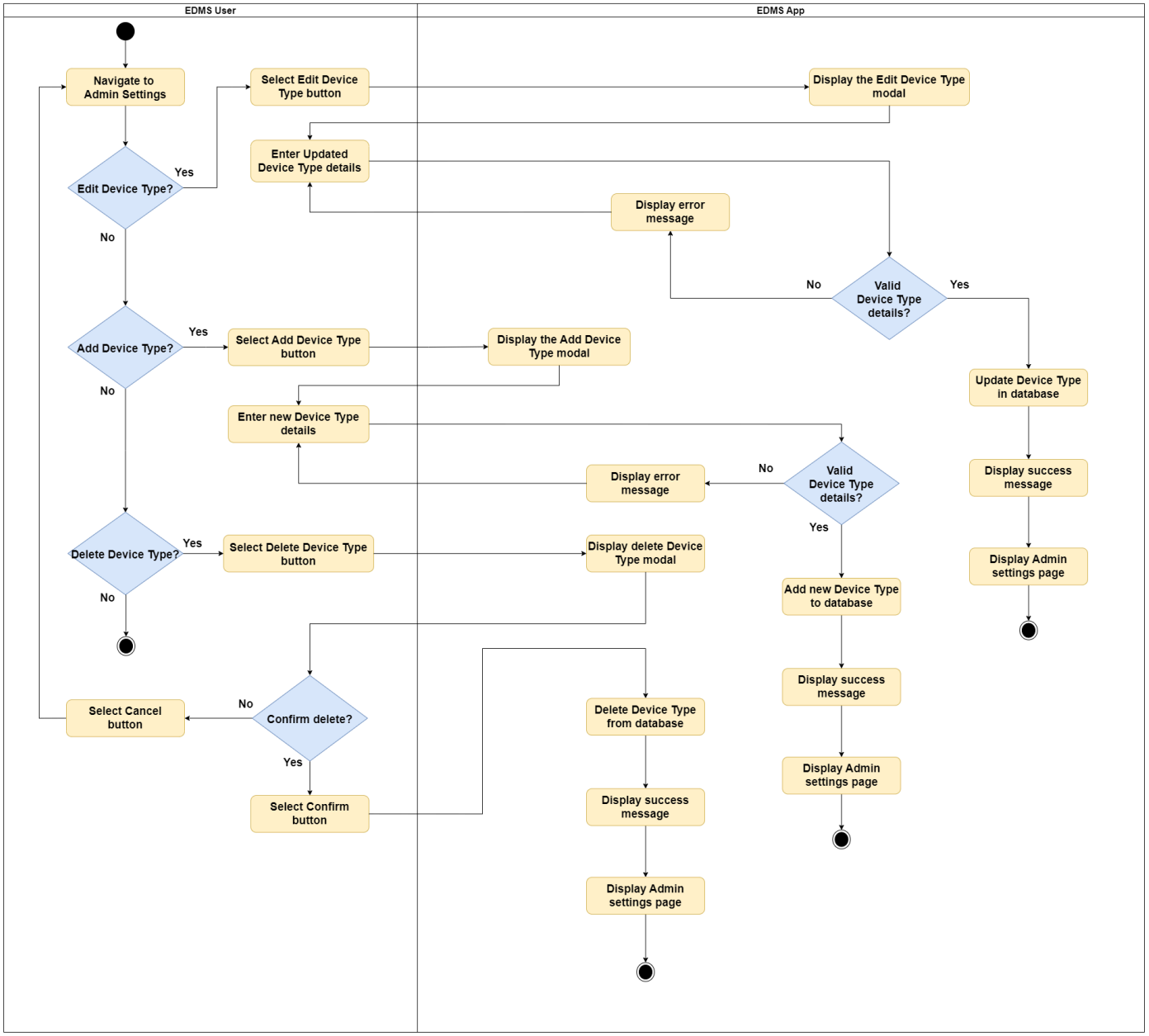


Figure 3.6: Activity Diagram for the manage device types process of the proposed web application

COMPONENT: "INSPECTION"

- **Purpose:**
  - The "Inspection" component is responsible for creating and viewing inspections for fire extinguishers. It enables viewing past inspection details for a fire extinguisher and creating new inspections for fire extinguishers by Admin users.
- **Inputs:**
  - User Actions: Create and read operations on inspection data.
  - Inspection Data: Inspection date, inspector's name, and device condition.
- **Operation:**
  - Admins can create new inspection records for fire extinguishers.
  - Admins can view existing inspection records.
- **Outputs:**
  - Records: Inspection records stored in the database.
  - Messages: Success and error notifications for inspection creation.

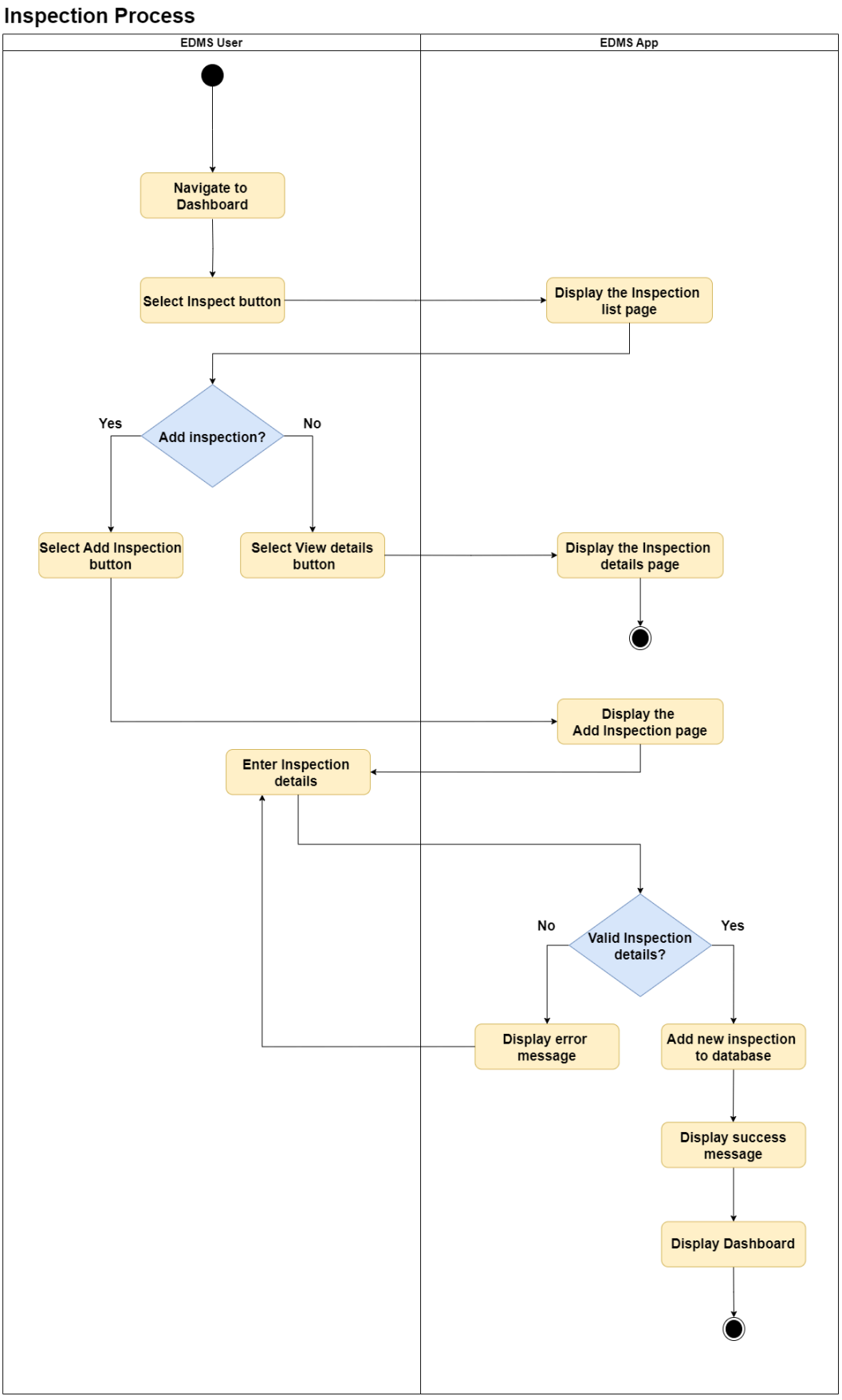


Figure 3.7: Activity Diagram for the inspection process of the proposed web application

COMPONENT: "DASHBOARD"

- **Purpose:**
  - The "Dashboard" component is responsible for the management of emergency devices across locations. Admins can create, read, update, and delete device records (regular users can only read), Devices can be filtered by building using an interactive map, while additional filtering and searching are available through a search box and dropdown menus.
- **Inputs:**
  - User Actions: Create, read, update, and delete operations for Admins; read operations for regular users.
  - Device Data: Includes Device ID, Device Type, Room, Serial Number, Status, Manufacture Date, Description, Size, and Last Inspection Date.
  - Search and Filter Inputs: Search box input and dropdown selections for filtering by device type, status, or location.
- **Operation:**
  - Admins can add, view, and modify device records.
  - Regular users can view device records.
  - Devices can be filtered by building using an interactive map.
  - Devices can be searched with a search box.
  - Devices can be filtered using dropdown menus.
- **Outputs:**
  - Records: New and updated device records stored in the database.
  - Display: Filtered and searched devices shown based on the interactive map, search box, and dropdown selections.
  - Messages: Success and error notifications for device creation, updates and deletions.

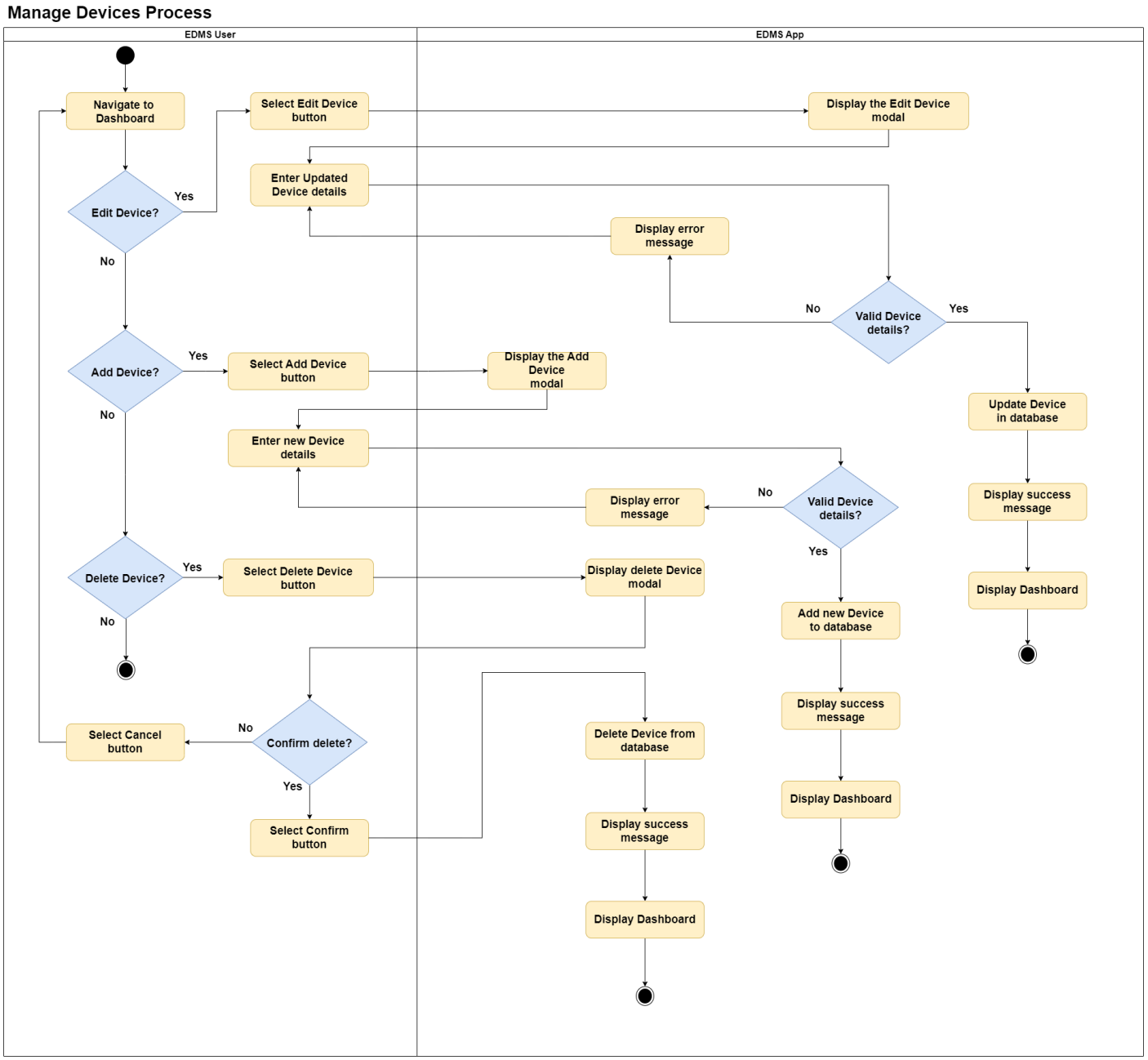


Figure 3.8: Activity Diagram for the manage devices process of the proposed web application

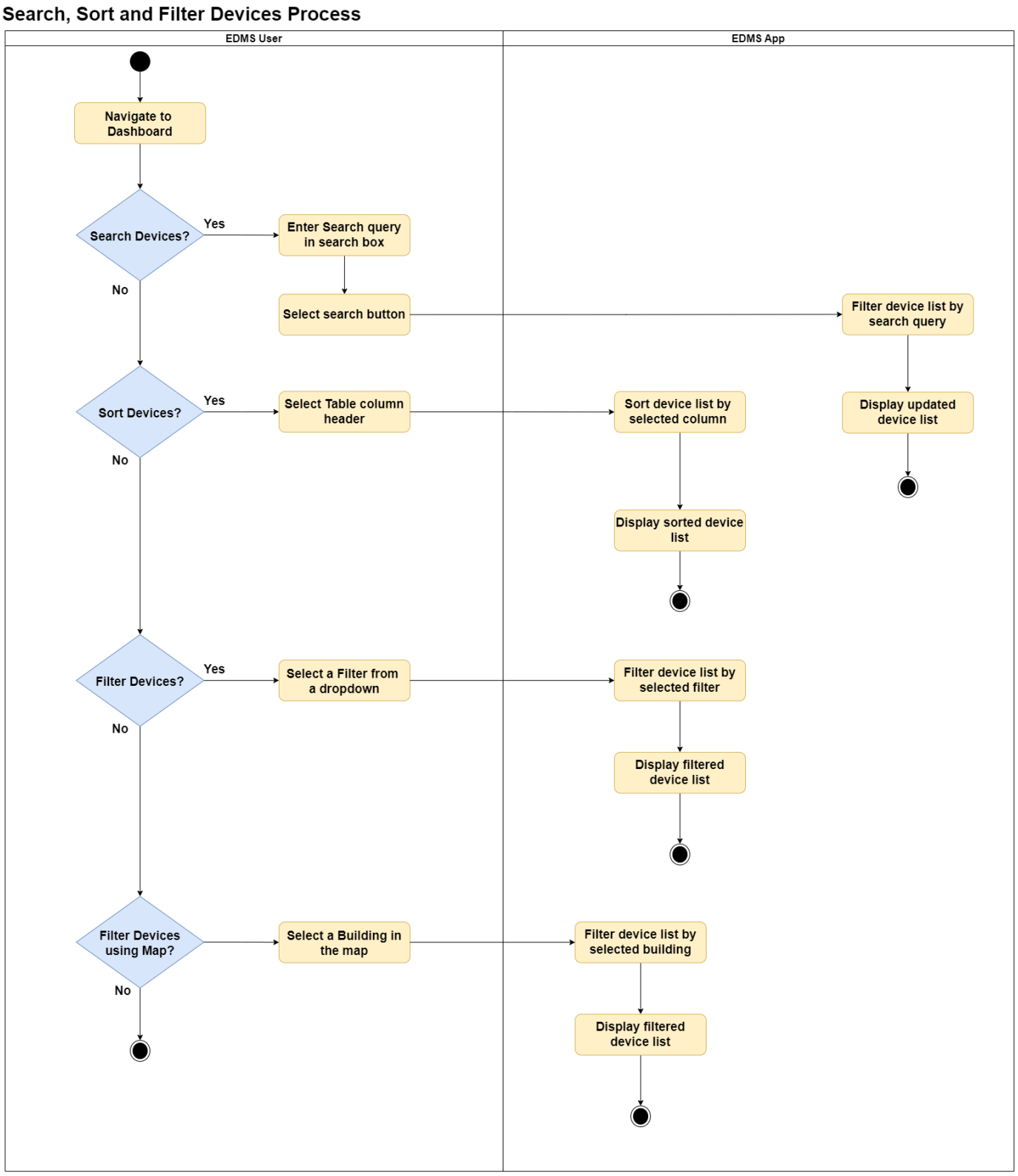


Figure 3.9: Activity Diagram for search, sort and filter devices process of the proposed web application

COMPONENT: "NOTIFICATION"

- **Purpose:**
  - The "Notification" component is responsible for notifying users about devices that are nearing their expiry or have an upcoming inspection due date (within the next 30 days). Notifications are displayed in a bell icon dropdown on the dashboard after the user logs in.
- **Inputs:**
  - Device information: Expiry dates and inspection due dates for devices.
- **Operation:**
  - After a successful login, the system checks device records for expiry and inspection dates.
  - Generates notifications for devices with upcoming expiry or inspections due within the next 30 days.
  - Displays these notifications in the dropdown menu associated with the bell icon on the dashboard.
- **Outputs:**
  - Notifications**:** Listed in the dropdown menu, showing devices with upcoming expiry or inspection dates.

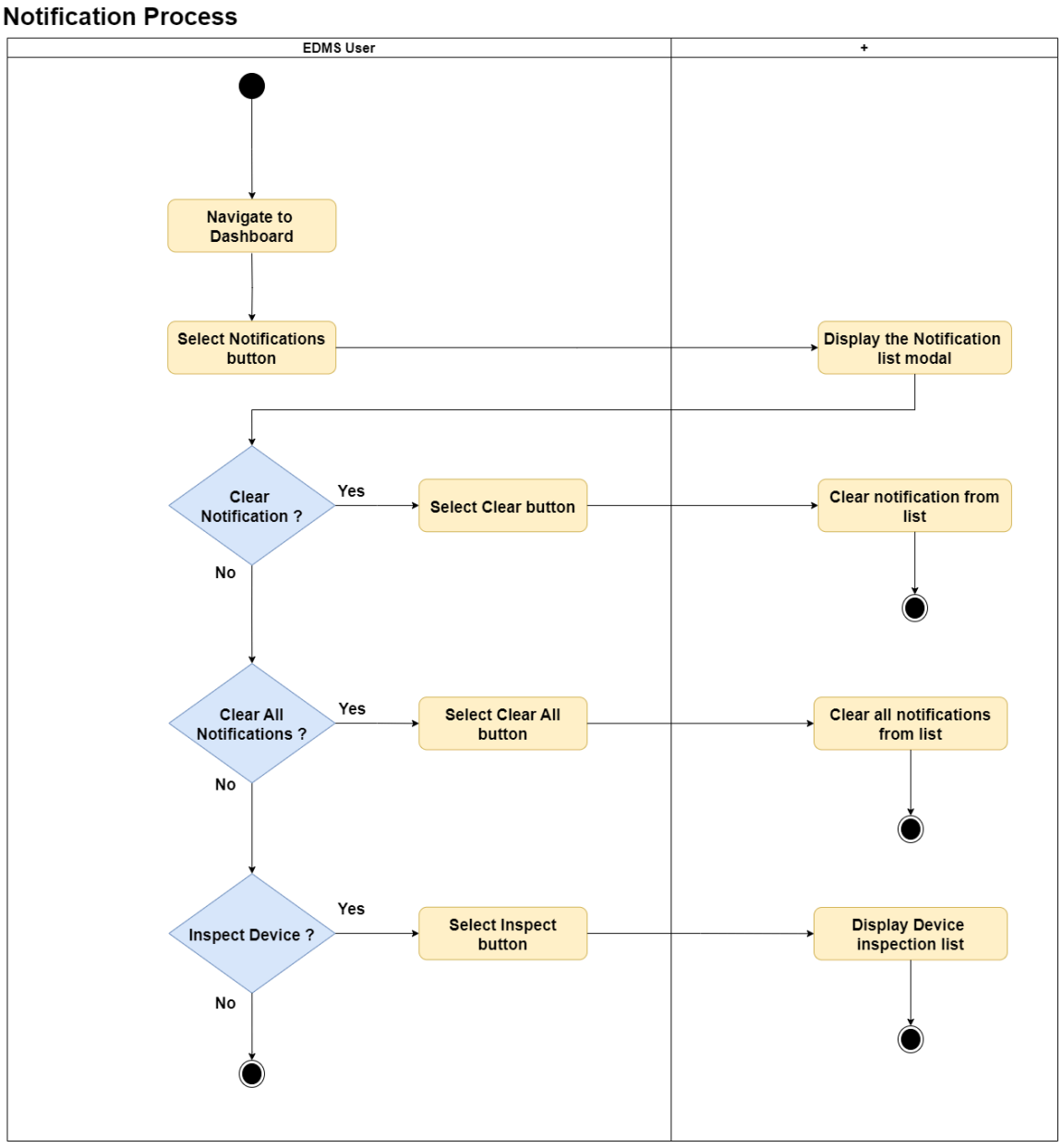


Figure 3.10: Activity Diagram for the notification process of the proposed web application

DATABASE DESIGN: ERD & DATA FLOW

ENTITY RELATION DIAGRAM

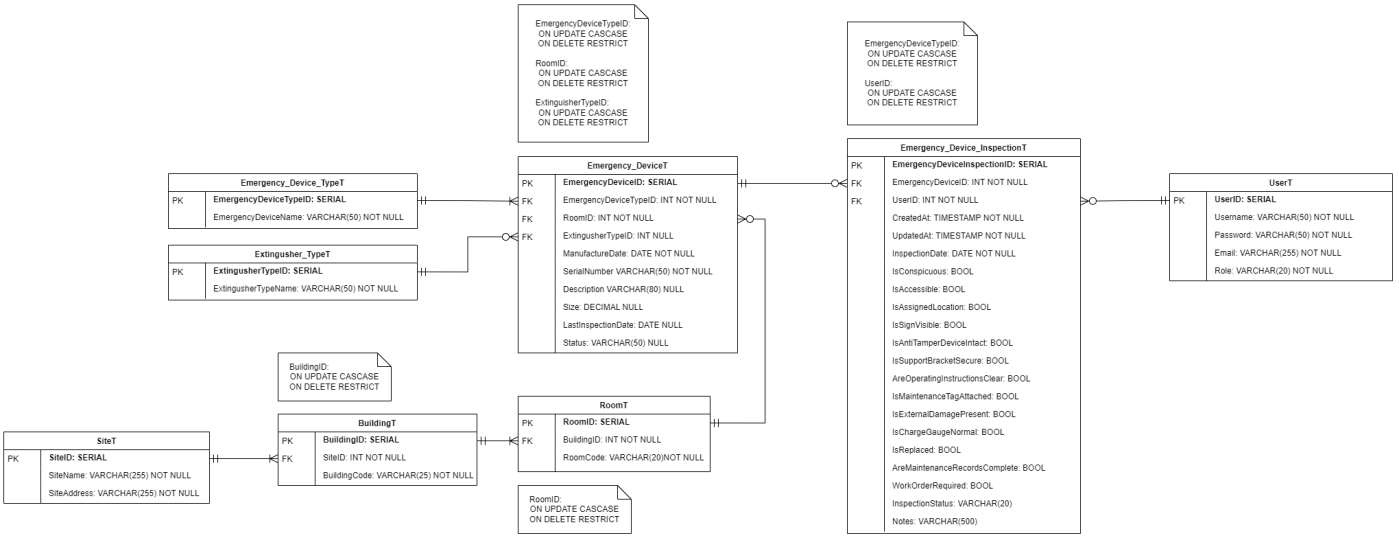


Figure 4.1: Entity Relation Diagram of the proposed web application

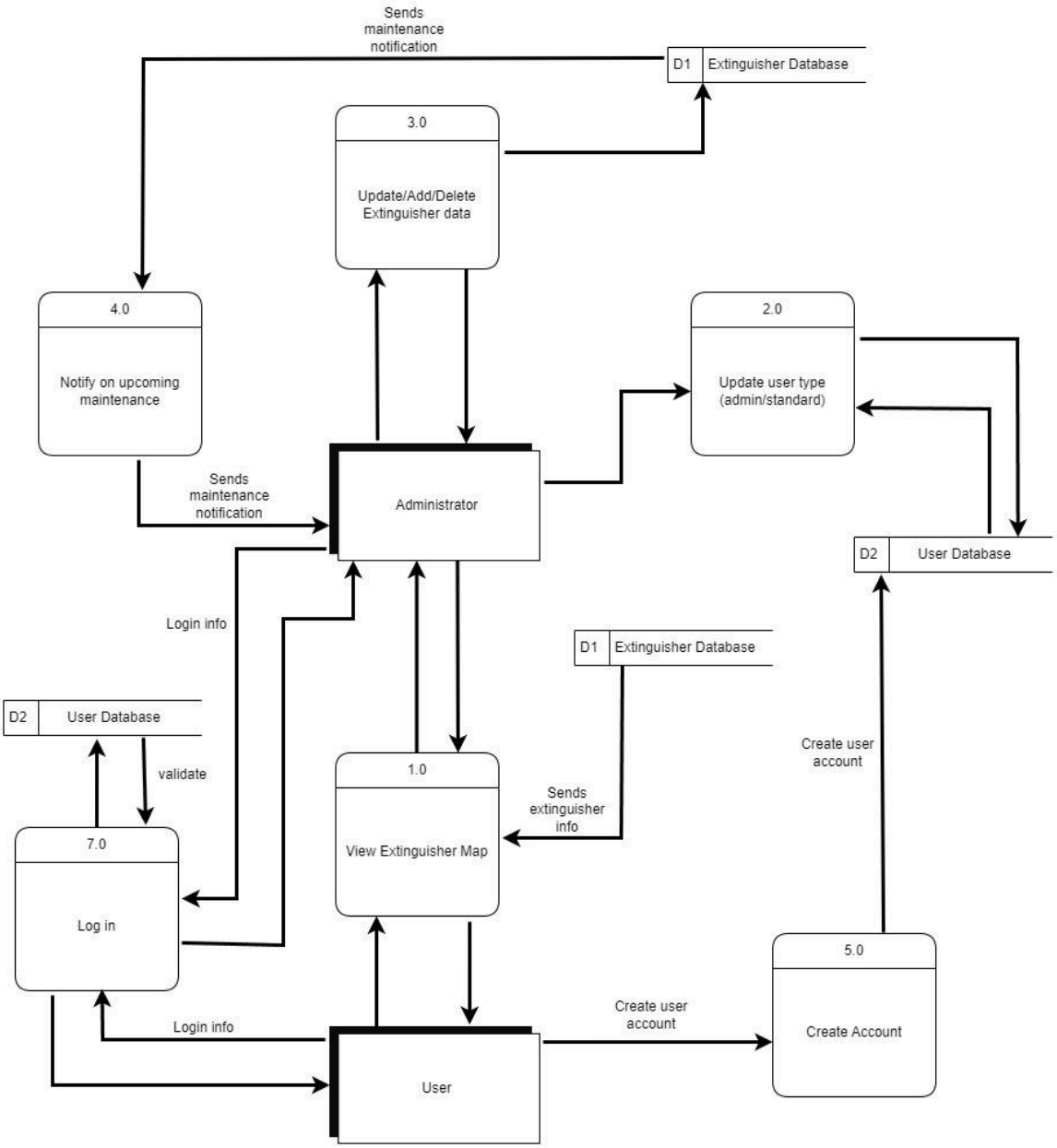


Figure 4.2: Data Flow Diagram of the proposed web application

UI / UX: WIREFRAMES

PAGE 1: HOMEPAGE

Figure 4.1: Wireframe designs of the home (landing) page on smartphones

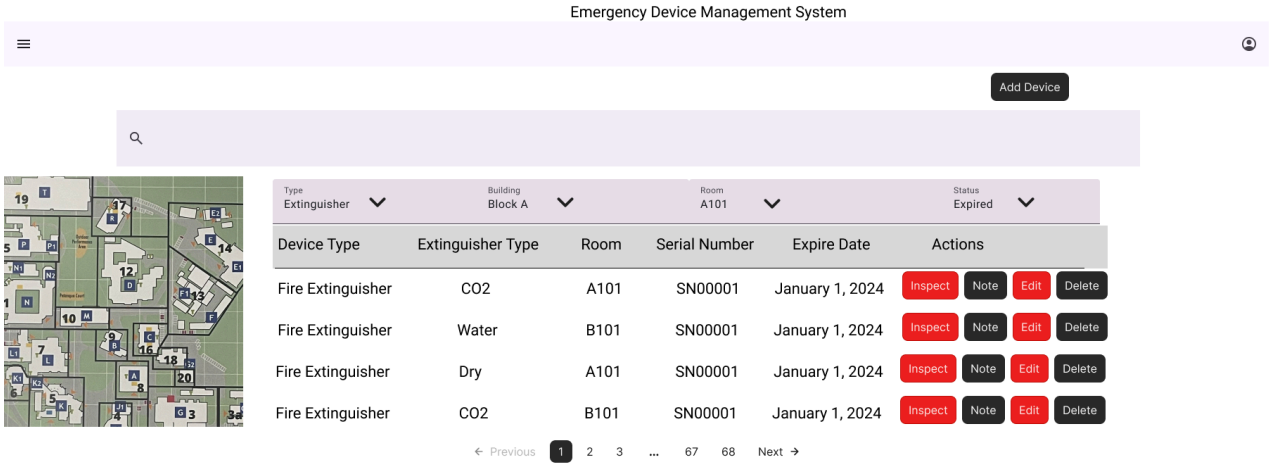
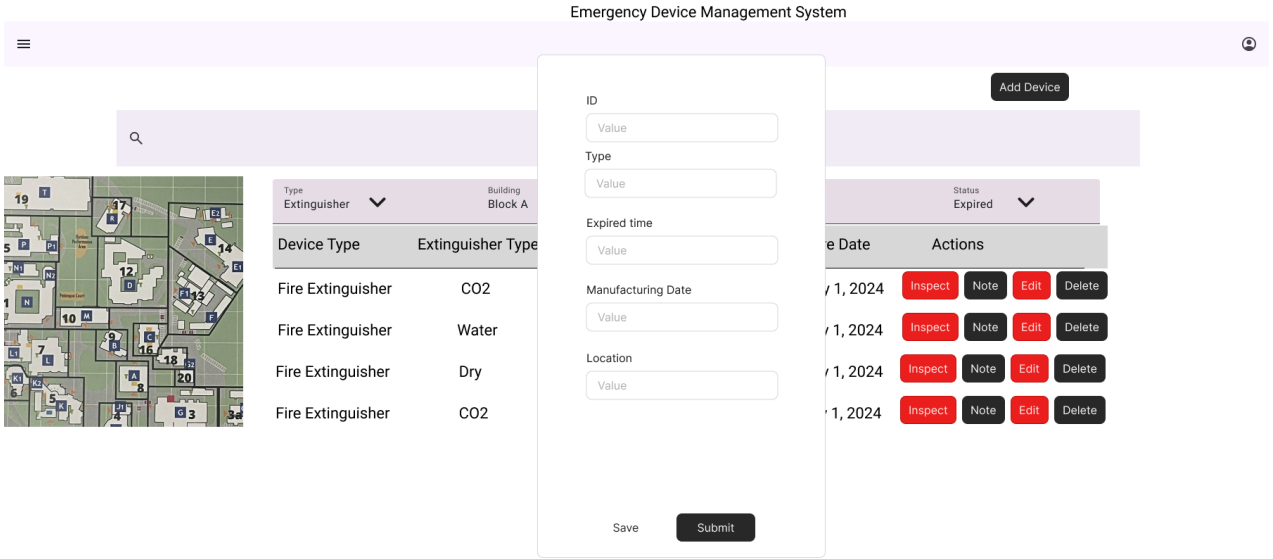
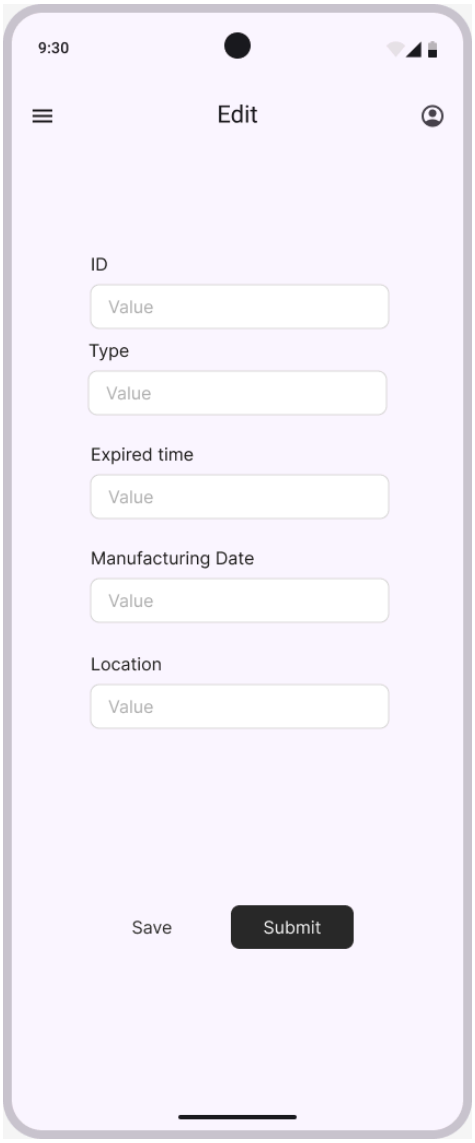


Figure 4.2: Wireframe designs of the home (landing) page on desktops / laptops

COMPONENT UI:
"DASHBOARD" UI (ADD/EDIT DEVICE):



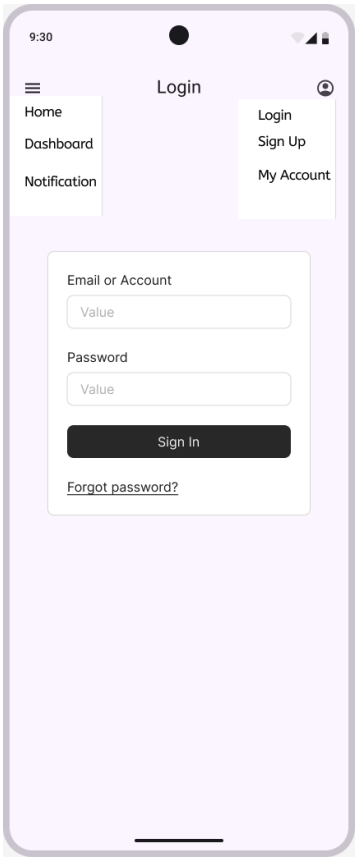
DESKTOP ADD/EDIT DEVICE



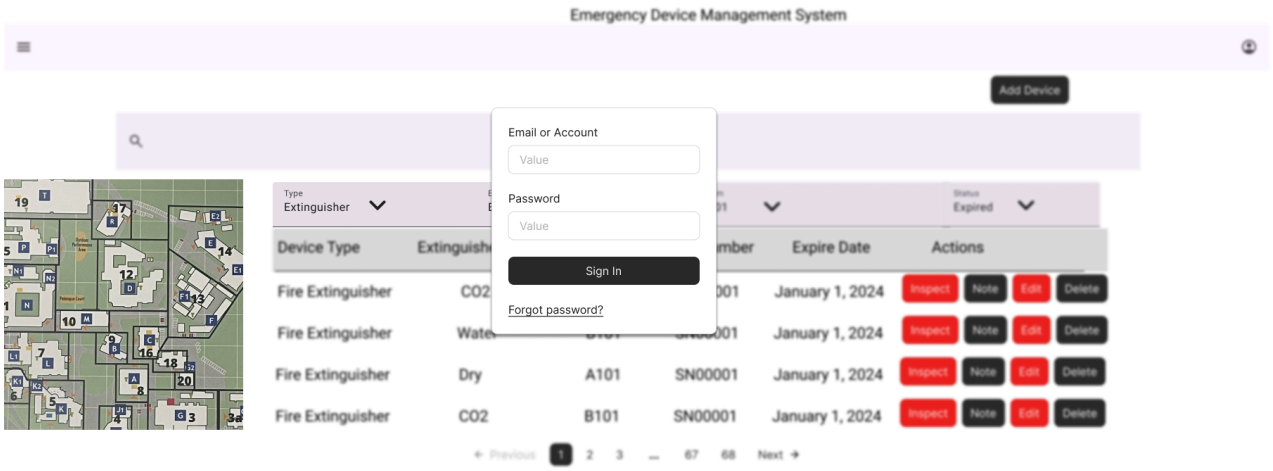
SMARTPHONE ADD/EDIT DEVICE

Figure 4.6: Check / Update UI

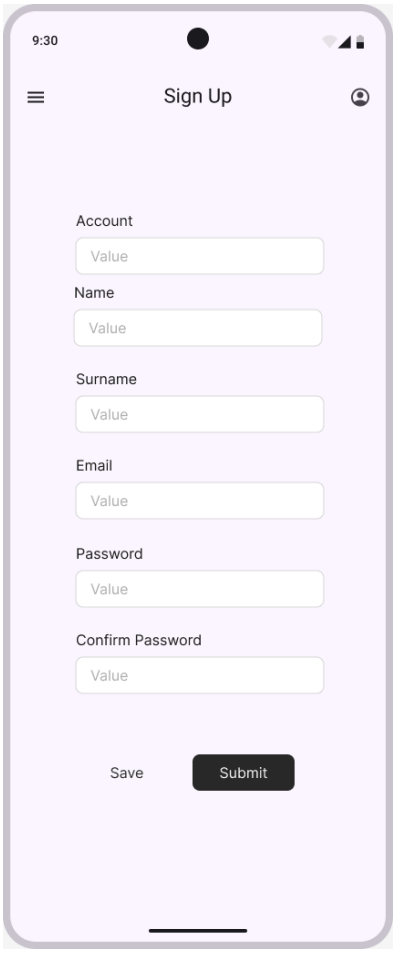
“AUTHENTICATION” UI (SIGN-UP / LOGIN):



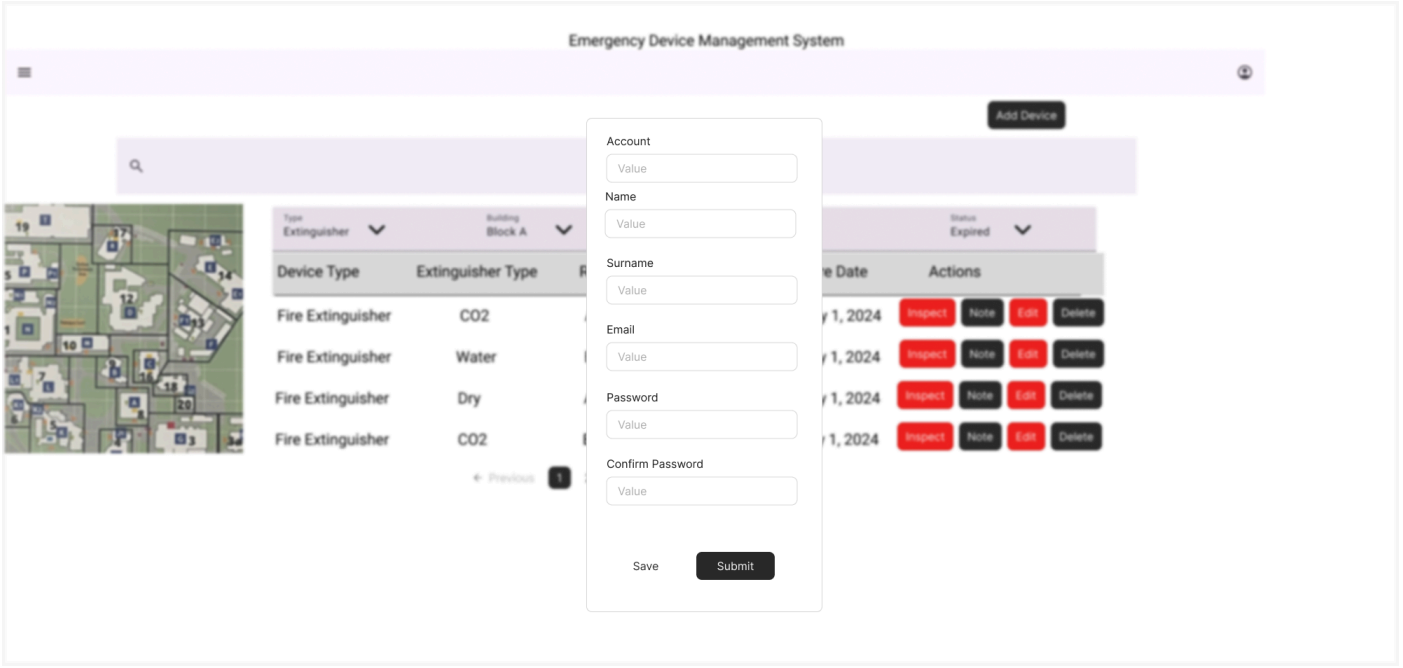
SMARTPHONE LOGIN



DESKTOP LOGIN



MOBILE SIGN UP



DESKTOP SIGN UP

Figure 4.4: Authentication UI

“ADMIN” UI:

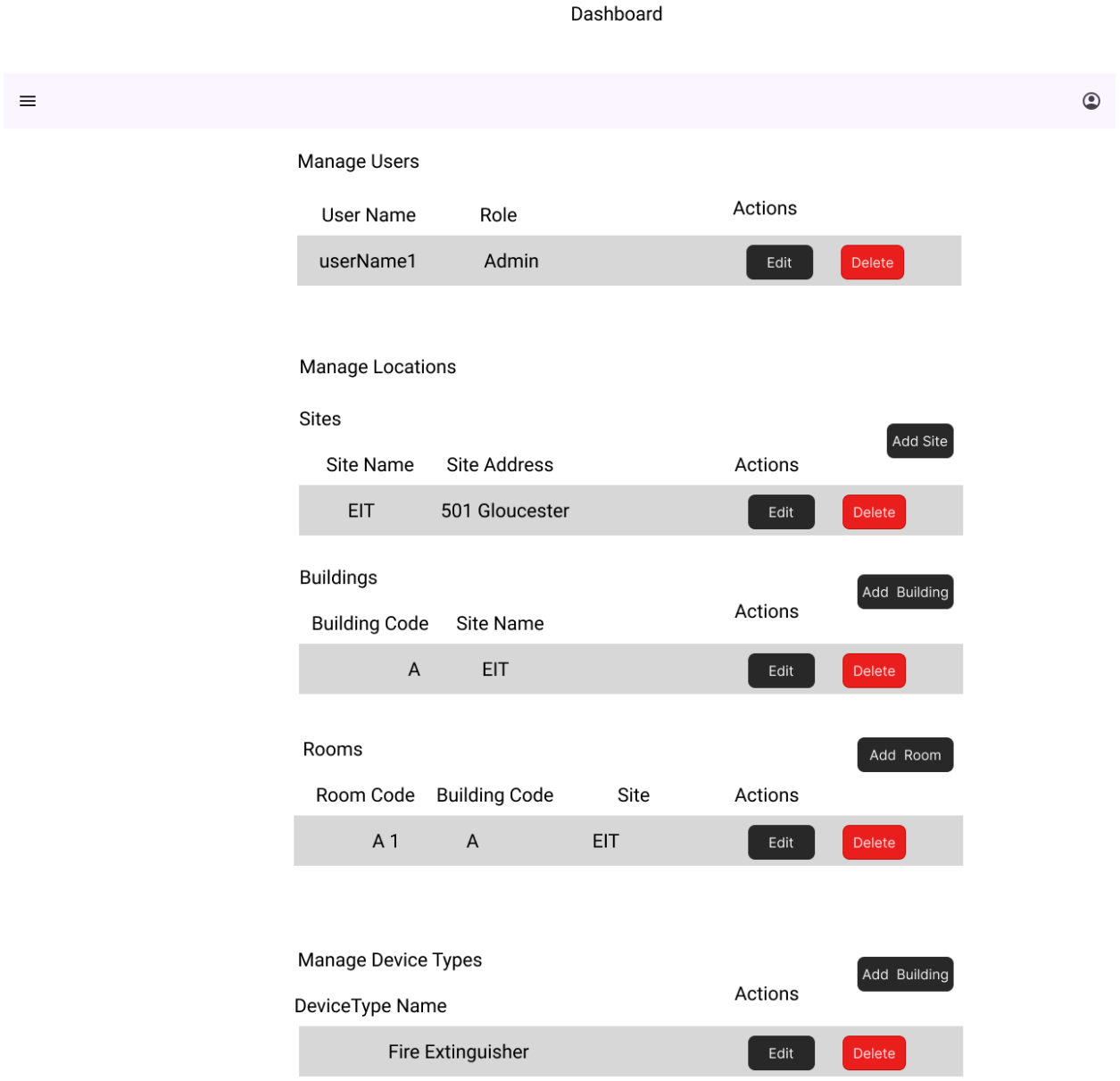
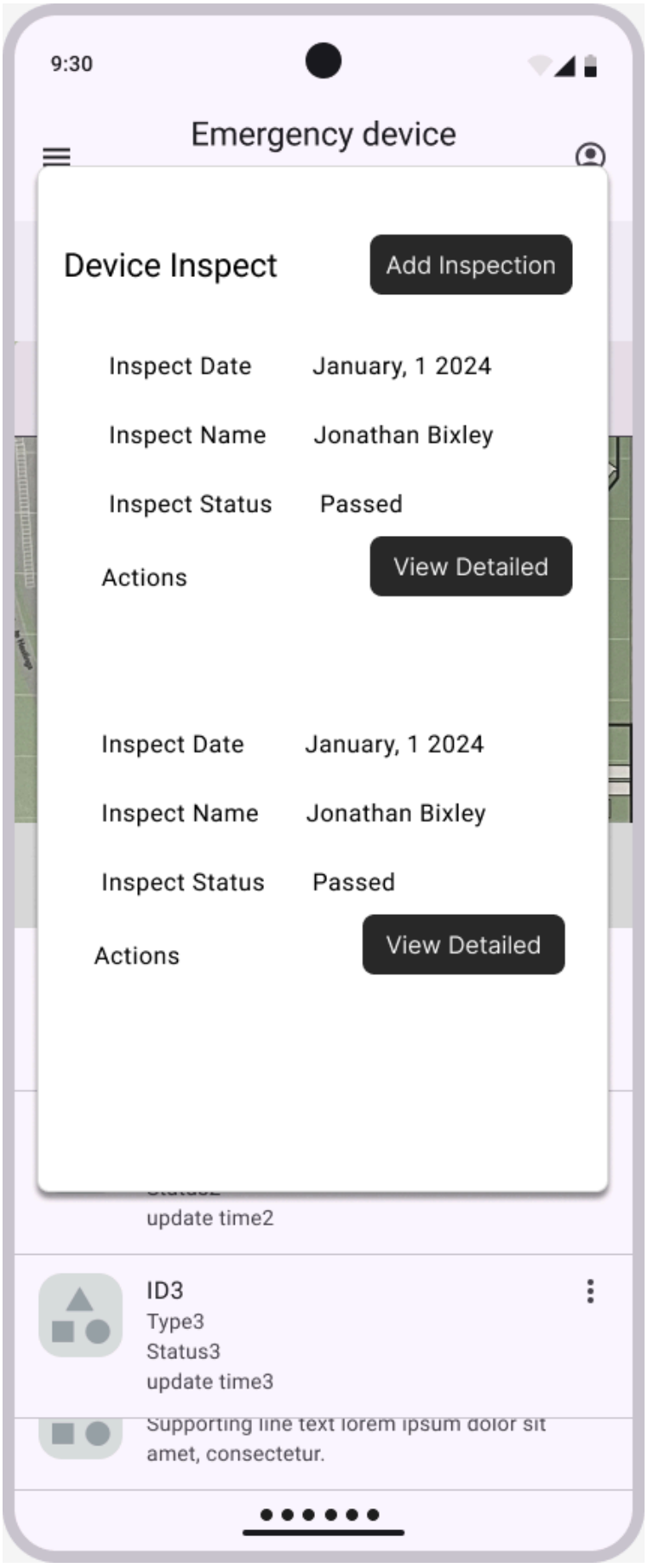


Figure 4.5: Admin UI

“INSPECTION” UI:



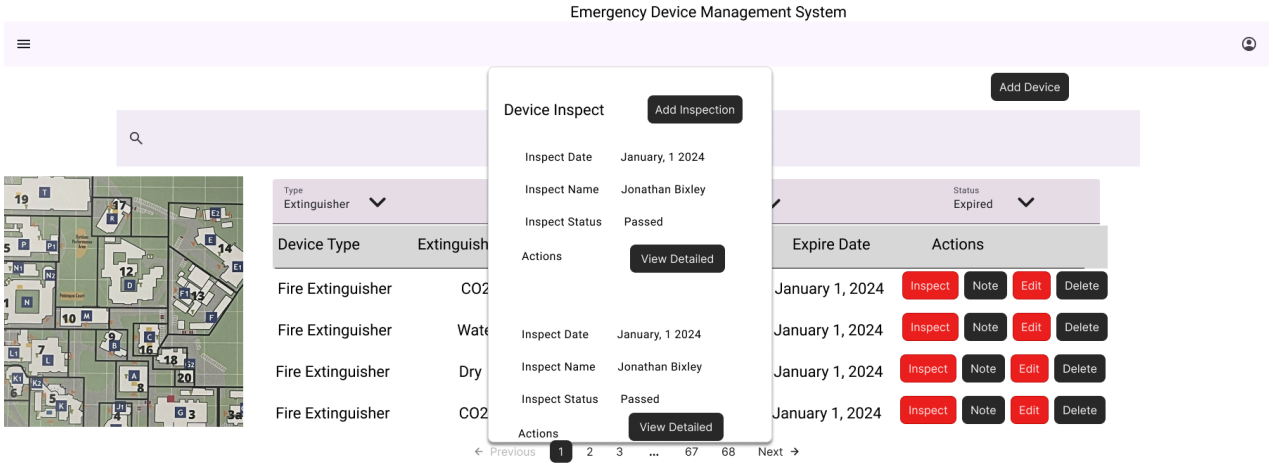
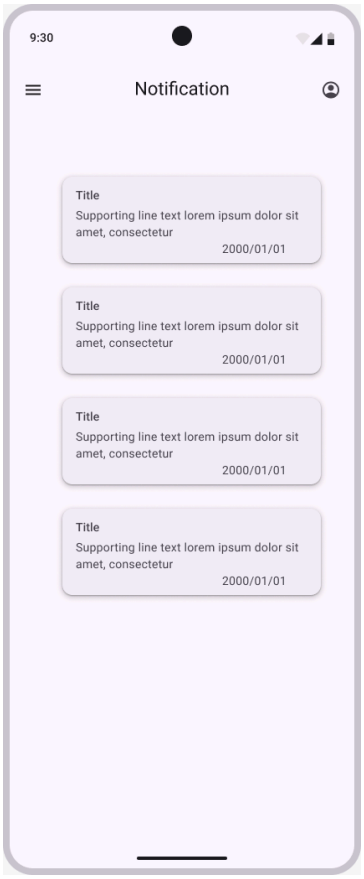
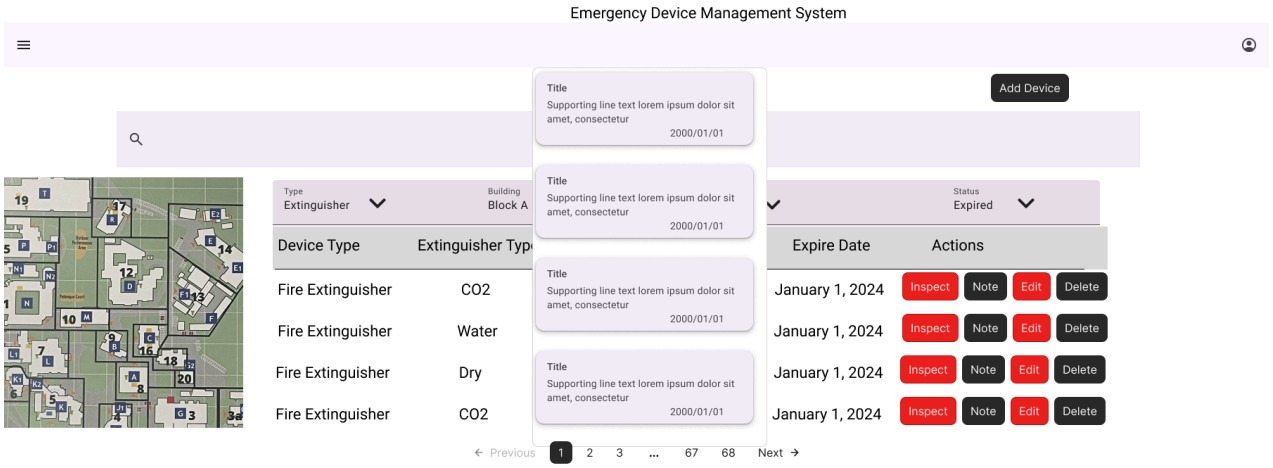


Figure 4.6: Check / Update UI

“NOTIFICATION” UI:





CLASS/OBJECT DIAGRAM

Figure 5.1: Class Diagram of the proposed web application

Test STRATEGIES/PLAN

Unit Testing

Test 1: Login

Component: Authentication

Prerequisites: Standard login

Steps

- User opens app
- User inputs required information in fields (ie Email in email field, password in password field)
- User clicks login

Test Results

Input	Expected Result	Actual Result
Username: "user1" Password: "Password1!"	User successfully logs in and can see a map of EIT's Campus	UNTESTED
Username: Password: "Password1!"	Login Fails due to missing input/s	UNTESTED
Username: "user1" Password:	Login Fails due to missing input/s	UNTESTED
Username: Password:	Login Fails due to missing input/s	UNTESTED

Test 2: Add Emergency Device

Component: Dashboard

Prerequisites: Admin account

Steps

- User clicks on add device
- User inputs relevant information pertaining to the fire extinguisher
- User clicks add

Test Results

Input	Expected Result	Actual Result
Type: Fire Extinguisher Extinguisher Type: CO2 Building: A Room: A1 Serial Number: SN00001 Description: By charging station Manufactured: 6/7/24 Last inspected: 8/9/24 Status: Active	User successfully adds a fire extinguisher type device into the system	UNTESTED
Type: Fire Blanket Building: B Room: B1 Description: On wall outside classroom Manufactured: 6/8/20	User successfully adds a fire blanket type device into the system	UNTESTED
Type: Med-Kit Building: C Room: C1 Serial Number: MD00001 Description: In a cupboard	User successfully adds a med kit into the system	UNTESTED
Any blank information	Adding device fails due to missing information	UNTESTED

Test 3: User account to Admin account

Component: Administrator

Prerequisites: Admin account (isAdmin=True)

Steps

- User opens user list
- User finds other user to change into admin (or searches for)
- User clicks on them, then on a drop down field saying “Set Admin”
- User can repeat steps as many times as required

Test data: None

Expected Results: User can update a user into an Admin

Actual Results: UNTESTED

Test Status: UNTESTED

Test 4: View Emergency Device

Component: Dashboard

Prerequisites: User account

Steps

- User finds an extinguisher to open (map)
- User clicks on fire extinguisher

Test data: None

Expected Results: User can view a fire extinguishers info

Actual Results: UNTESTED

Test Status: UNTESTED

Test 5: Sign up

Component: Authentication

Prerequisites: None

Steps

- User opens app and clicks on "sign up"
- User fills in fields with relevant information
- User clicks sign up

Test Results

Input	Expected Result	Actual Result
Username: "user1" Password: "Password1!" Email:"user1email@gmail.com"	User successfully signs up and gets redirected to log in page	UNTESTED
Username: "user1" Password: "Password1!" Email:	Sign up Fails due to missing input/s	UNTESTED
Username: "user1" Password: Email:"user1email@gmail.com"	Sign up fails due to missing input/s	UNTESTED
Username: Password: "Password1!" Email:"user1email@gmail.com"	Sign up fails due to missing input/s	UNTESTED

Test 6: Deleting account

Component: Dashboard

Prerequisites: User account

Steps

- User logs into account
- User clicks on account settings in app (From home: Settings -> Account Settings)
- User clicks delete account

Test data: None

Expected Results: User account is successfully deleted and logs user out

Actual Results: UNTESTED

Test Status: UNTESTED

Test 7: Device Inspection

Component: Dashboard

Prerequisites: User account

Steps

- User logs into account
- User clicks on account settings in app (From home: Settings -> Account Settings)
- User clicks delete account

Test data: None

Input	Expected Result	Actual Result

Inspection Date: 8/9/24 Notes: Small dent Is conspicuous: true Operation instructions clear: true Is accessible: true Is in correct location: true Has maintenance tag: true Is sign visible: true Is external damage present: true Is anti-tamper device intact: true is charge gauge normal: true is support bracket secure: true Is replaced: false Is Work order required: false Maintenance records complete: true	Device passes inspection	UNTESTED
Inspection Date: 8/9/24 Notes: Gone Is conspicuous: false Operation instructions clear: true Is accessible: false Is in correct location: false Has maintenance tag: false Is sign visible: true Is external damage present: true Is anti-tamper device intact: false is charge gauge normal: false is support bracket secure: true Is replaced: false Is Work order required: true Maintenance records complete: false	Device fails inspection	UNTESTED

Test 8: View Notifications

Component: Notification

Prerequisites: User account

Steps

- User logs into account
- User clicks on notifications to check if they have anything that needs to be updated

Test data: None

Expected Results: User can see notifications if anything new has occurred

Actual Results: UNTESTED

Test Status: UNTESTED

DATABASE TESTING

Database Testing Plan

Objective:
Ensure the accuracy, consistency, reliability, and performance of the database system.

Scope:
Test all CRUD (Create, Read, Update, Delete) operations, data validation, integrity constraints, and performance under different data volumes.

Test Cases:

- Data Validation

  Test Case: Insert, update, delete data and check the results in the database.

  Steps:

    Insert a new device record.

    Verify that the record exists in the database.

    Update a field in the record (e.g., last inspection date).

    Verify that the updated record is correct.

    Delete the record and ensure it no longer exists.

  Expected Outcome: The record is inserted, updated, and deleted successfully, and the database maintains consistency.

- Data Integrity

  Test Case: Validate relationships between tables (e.g., foreign key constraints).

  Steps:

    Try to insert a record with an invalid foreign key reference (e.g., non-existent device type).

    Verify that the operation is blocked by the database.

    Ensure cascading updates are correctly implemented (e.g., deleting a user should cascade and remove related entries).

  Expected Outcome: Foreign key constraints prevent invalid operations, and cascading rules work correctly.

- Performance Testing

  Test Case: Measure query performance under different loads.

  Steps:

Insert a large dataset of devices (e.g., 10,000 records).

Run queries that filter, group, and join data from multiple tables.

Measure response times.

Expected Outcome: Queries should complete within acceptable response times.

Integration Testing

Integration Testing Plan

Objective:
Verify that different modules in the application work together seamlessly and that data flows correctly between the front-end, back-end, and database.

Scope:
Focus on interactions between modules like Authentication, Dashboard, and Database.

Test Cases:

- Login and Dashboard Integration

  Test Case: Ensure that upon login, the user is directed to the dashboard with the correct data loaded.

  Steps:

  Login using valid credentials.

  Redirect to the dashboard.

  Verify that user-specific data is correctly loaded on the dashboard.

  Expected Outcome: The user successfully logs in, and the correct data is loaded.

- Database Interaction

  Test Case: Ensure that adding or updating a device in the front-end reflects correctly in the database.

  Steps:

  Add a new fire extinguisher using the admin panel.

  Verify that the new entry exists in the database.

  Update the extinguisher's status from the dashboard.

  Verify that the updated status is correctly reflected in the database.

  Expected Outcome: Changes made in the UI are reflected in the database without errors.

System Testing

System Testing Plan

Objective:
Evaluate the application's overall security, performance, and compatibility.

Scope:
Test the application across multiple environments, including different browsers, operating systems, and devices, along with performance and security aspects.

Test Cases:

- Compatibility Testing

  Test Case: Ensure the application works on multiple browsers (Chrome, Firefox, Safari) and devices (mobile, tablet, desktop).

  Steps:

  Open the application in different browsers and devices.

  Verify that all elements (UI, functionality, etc.) load correctly.

  Test various features (e.g., map interactions, data filtering).

  Expected Outcome: The application performs consistently across all tested platforms.

- Performance Testing

  Test Case: Measure load times and responsiveness under different user loads.

  Steps:

  Simulate multiple users (e.g., 100, 500, 1000) accessing the application simultaneously.

  Measure load times and response delays.

  Expected Outcome: The application remains responsive and loads within acceptable times even under heavy load.

- Security Testing

  Test Case: Ensure that user data is secure and that unauthorized users cannot access restricted areas.

  Steps:

  Try logging in with invalid credentials.

  Check that no sensitive data is displayed in error messages.

  Attempt to access admin features as a regular user.

  Verify encryption of sensitive data like passwords.

  Expected Outcome: The system blocks unauthorized access and secures sensitive data.

USER ACCEPTANCE TESTING (UAT)

Objective:
Validate the system's functionality from an end-user perspective and ensure that it meets the project requirements.

Scope:
Test real user interactions for key features such as registration, device management, and notifications.

Test Cases:

- User Registration

  Test Case: Ensure new users can register and create accounts.

  Steps:

  Open the sign-up page.

  Fill in valid registration details.

  Complete the registration process.

  Expected Outcome: The user account is created, and the user can log in successfully.

- Device Management

  Test Case: Ensure that admins can manage (add, update, delete) emergency devices.

  Steps:

  Log in as an admin.

  Add a new emergency device.

  Update its details.

  Delete the device.

  Expected Outcome: All device management operations succeed, and changes are reflected correctly in the system.

- Notifications

  Test Case: Ensure users receive notifications for upcoming device inspections.

  Steps:

  Log in to the system.

  Verify if notifications are displayed for upcoming inspections.

  Expected Outcome: Notifications are correctly shown based on the system's data for inspections and expiration.

Conclusion

In conclusion, this document outlines the key business and design decisions made to address EIT's current fire extinguisher tracking challenges. The proposed solution replaces the outdated spreadsheet system with a scalable digital web application, ensuring improved compliance, data management, and campus safety. By implementing features like automated notifications, role-based access, and an interactive map, the application improves efficiency and supports future growth, providing a comprehensive tool for managing safety devices across multiple locations.