SOFTWARE DESIGN SPECIFICATION

Temperature Management System

Group Members

1. Harsh Meena	*IIT2019005*
2. Prasanth Kota	*IIT2019062*
3. Gopal Pedada	*IIT2019065*
4. Janaki Ram	*IIT2019084*

The Software Design Specification

1. Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams,class diagram,activity diagram.

1.1 Purpose of this document

This document will define the design of the temperature management system. It contains specific information about the expected input, output, classes, and functions. The interaction between the classes to meet the desired requirements are outlined in detailed figures at the end of the document.

1.2 Scope of the development of the Project

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

a. Application for managing temperature in IIIT Allahabad campus.
b. users can retrieve the temperature details of their particular rooms.
c. Control Temperature remotely using this application.

Out of Scope:

a. direct messages between users
b. Access from personal networks

1.3 Definitions, acronyms, and abbreviations

IEEE: Institute of Electrical and Electronics Engineers.
SDS: Software Design Specification.
SRS: Software Requirement Specifications.

1.4 References

1.4.1 R. S. Pressman, Software Engineering: A Practitioner's Approach, 5th Ed, McGraw-Hill, 2001.
1.4.2 IEEE SDS template

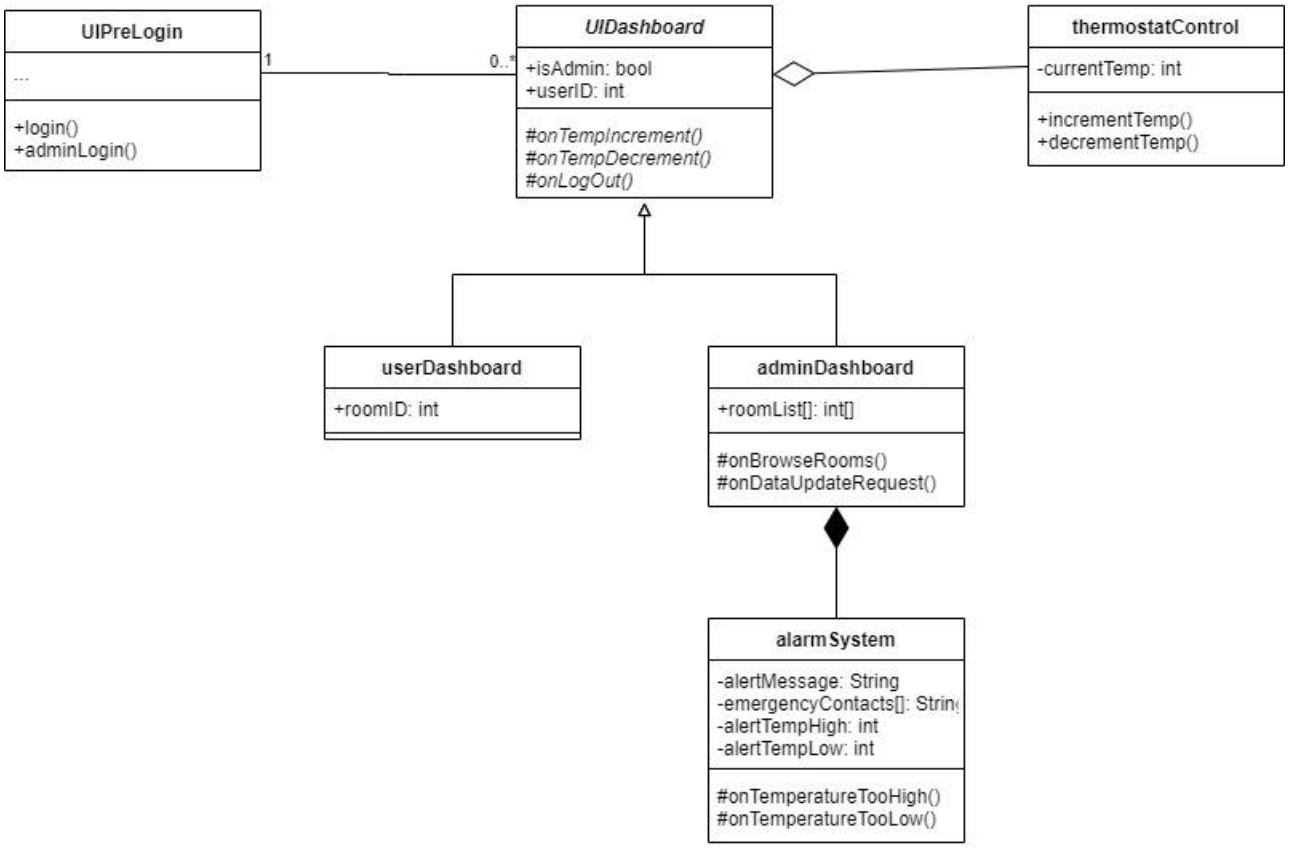
1.5 Overview of document

This SDS is divided into seven sections with various sub-sections. The sections of the Software Design Document are:

1. **Introduction**: describes about the document, purpose, scope of development project definitions and abbreviations used in the document.
2. **Logical Architecture:** describes Logical Architecture Description and Components.
3. **Execution Architecture:** defines the runtime environment, processes, deployment view.
4. **Design Decisions and Trade-offs:** describes the decisions taken along with the reason as to why they were chosen over other alternatives.
5. **Pseudocode for components:** describes pseudocode, as the name indicates.
6. **Appendices:** describes subsidiary matter

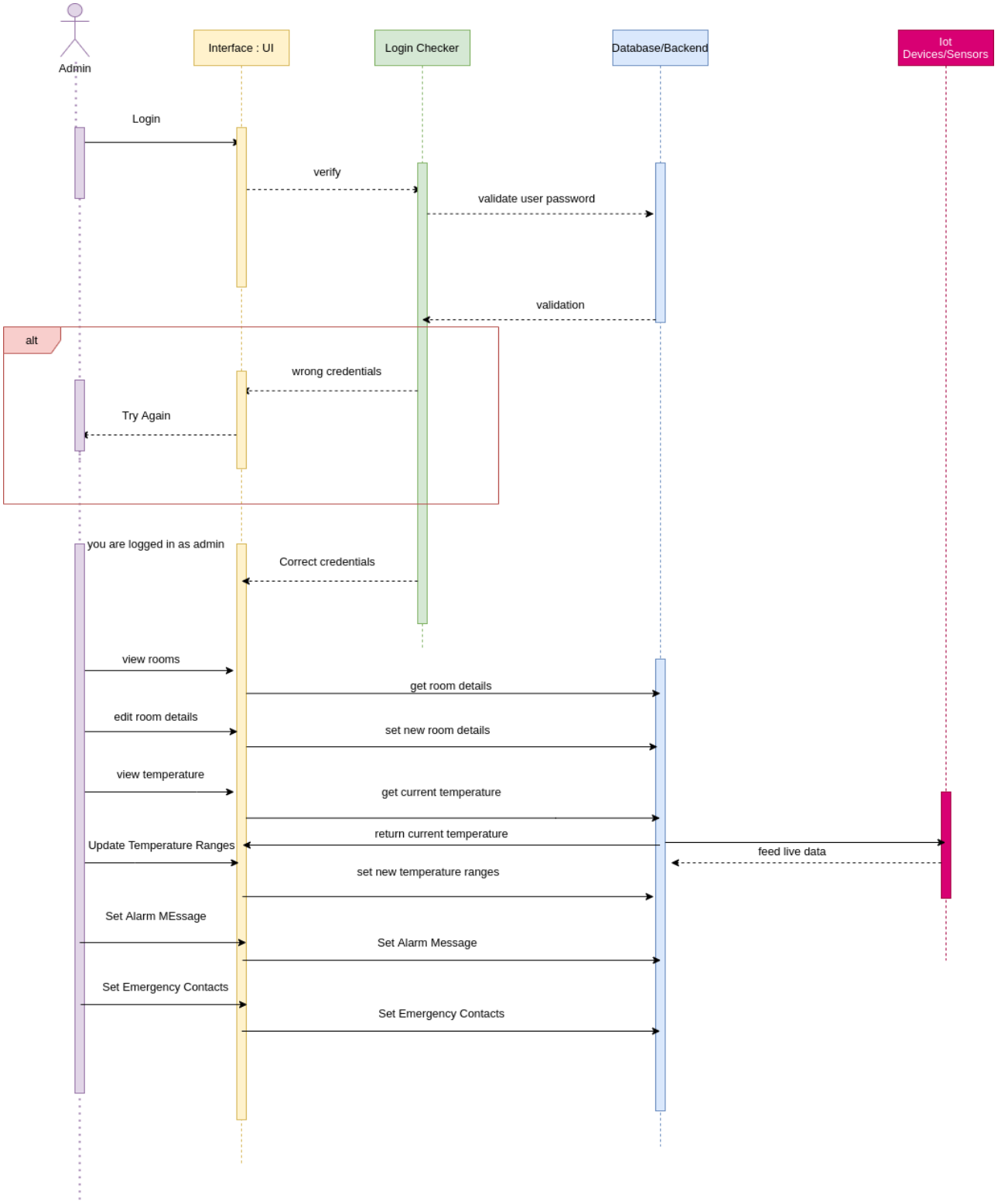
2. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

2.1. Class Diagram

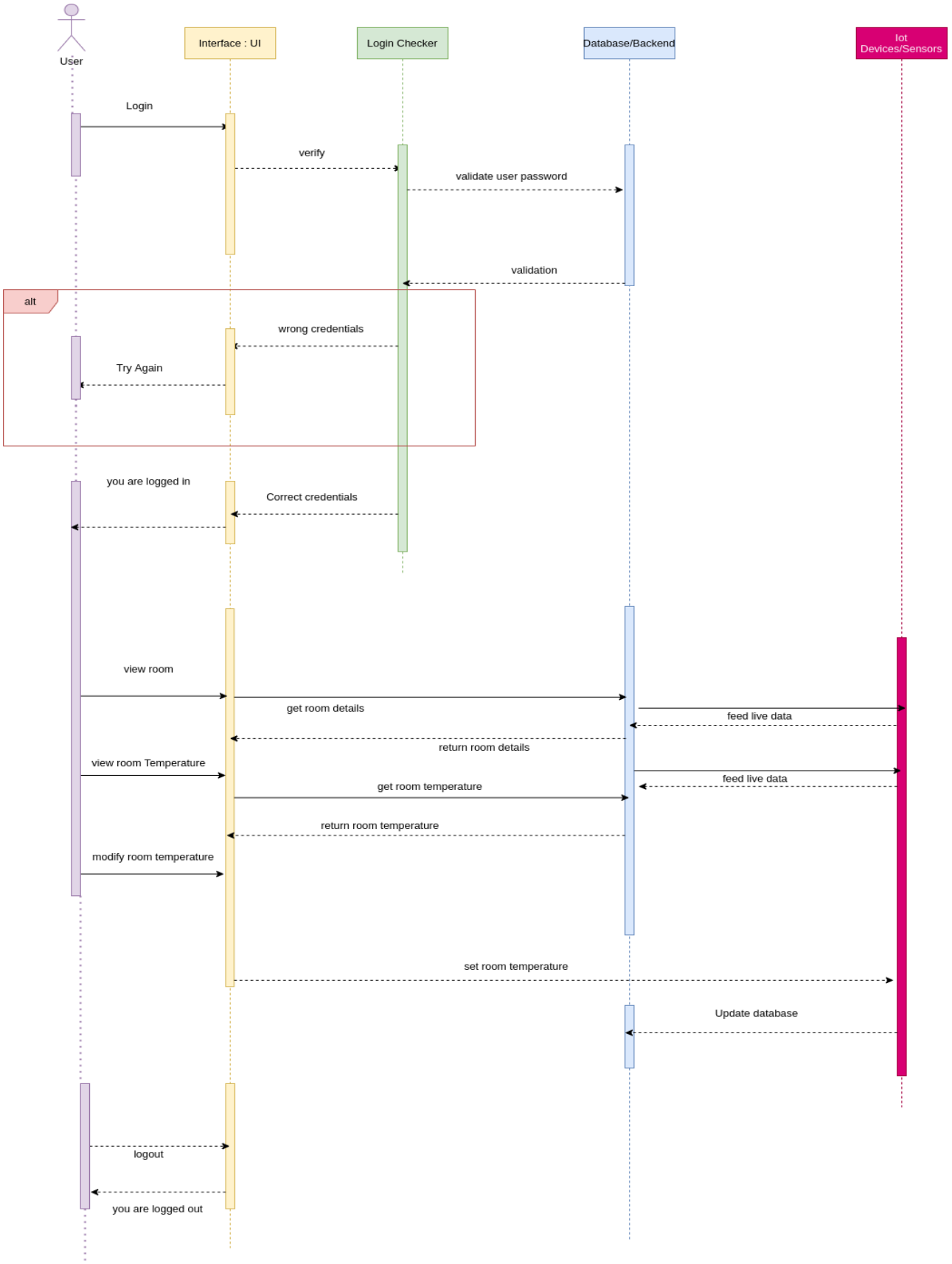


2.2 Sequence Diagrams:

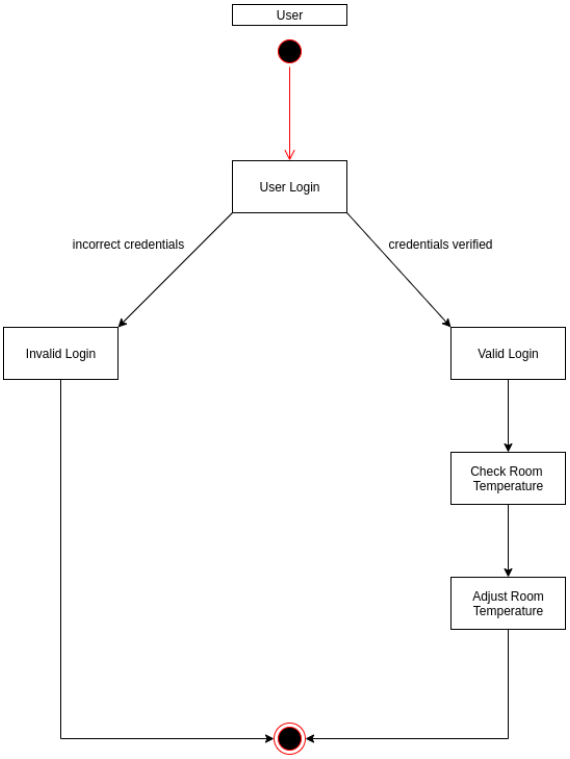
Sequence Diagram: Admin

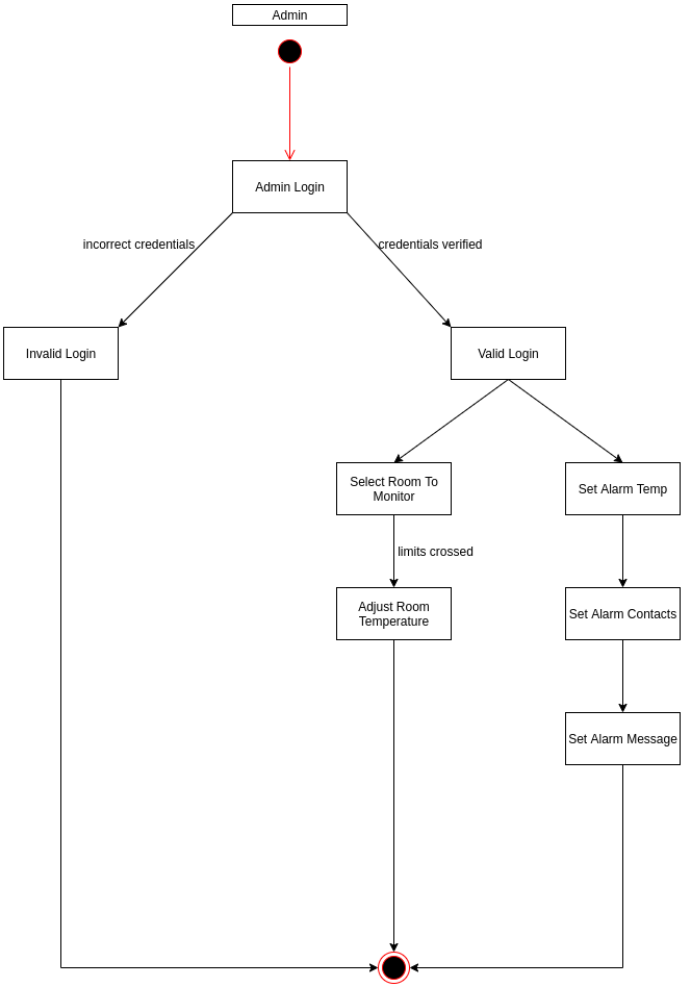


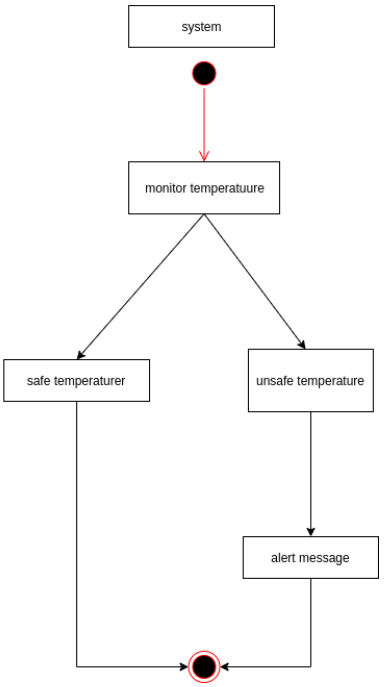
Sequence Diagram: User



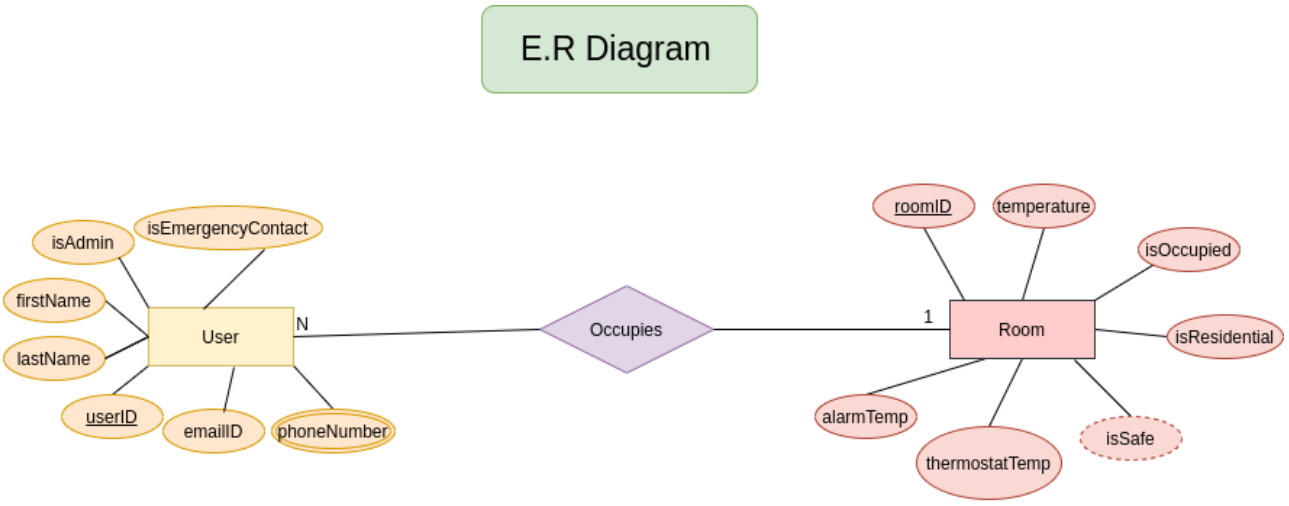
2.3 State Diagram:



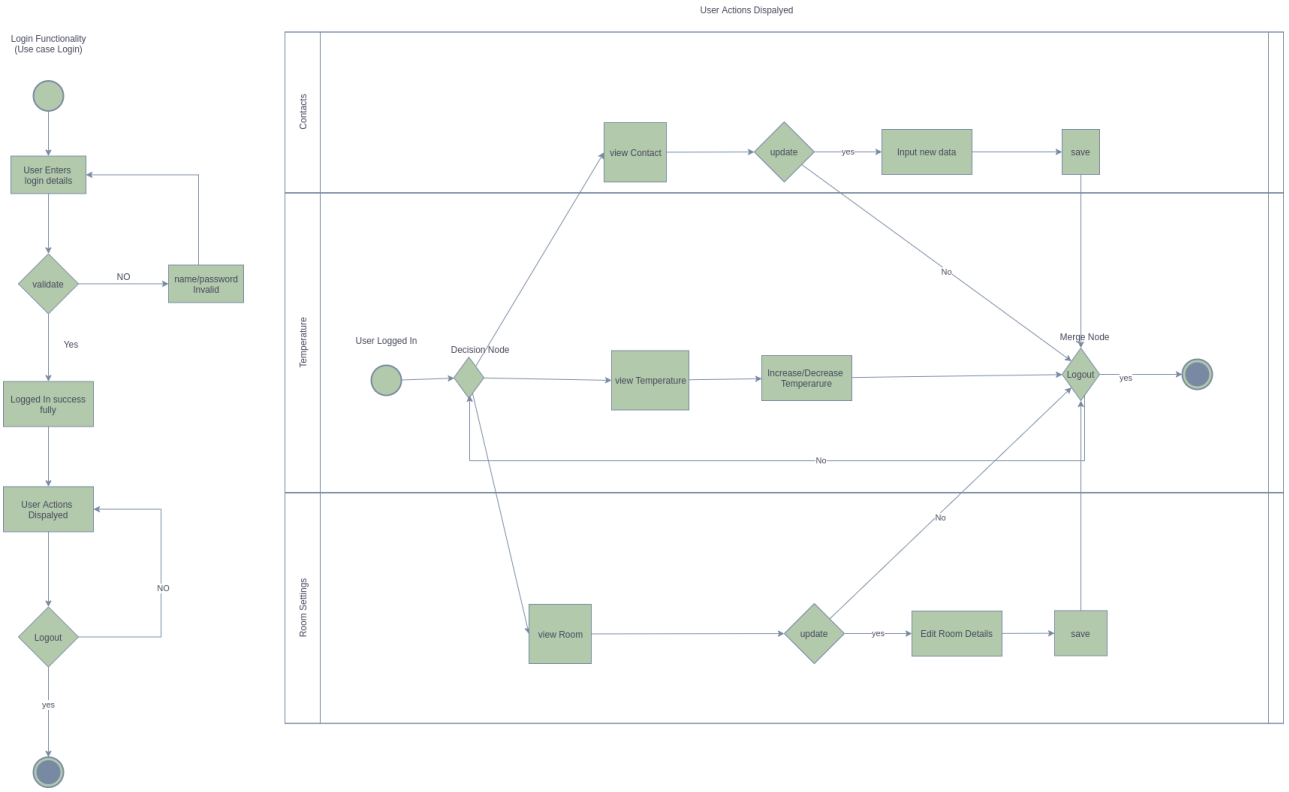




2.4 E.R Diagram :



2.5 Activity Diagram:



2.1 Logical Architecture Description

2.1.1 Class Diagram Description:

Classes userDashboard and adminDashboard both extend UIDashboard which is an abstract class. Class alarmSystem is a composition of class adminDashboard, since it can only exist if there is an admin to set the emergency contacts, alert messages etc.

Class UIPrelogin is associated with multiple UIDashboard objects, since each user will have a dashboard of their own. Class thermostatControl is an aggregation of UIDashboard class.

2.1.2 Sequence Diagram:

Arrow line signifies there is a send message taken place. Response is being shown by dotted arrows.

Admin: Admin logs in through the interface,password checker validates the username and password through login if they are invalid details then returns to home interface else admin can view and change rooms and get room details like room temperature,alarm temperature,whether room is occupied or not,modify the temperature of rooms wherever required then update the database. gets the room details,also sets the safe temperature ranges which can be used by users to maintain their own room temperature.if the temperature is high the alert message is sent to all the emergency contacts which is set by the admin.

users: user logs in through the interface then login checkers verifies the username and password.it checks through the already saved database on validation if they are wrong credentials then it returns to the home interfaces and displays try again otherwise displays you are logged in.now after logging in user can view his room temperature through the iot sensors and feed live data.he can also modify temperature according to the safe temperature needed and it is set by the iot sensors then update database.now he can logout then displays you are logged out.

3.1 Class name: UIPreLogin

Description: This class allows the user to enter the system by authenticating the entered credentials.

3.1.1. Method 1: login()

Input: username, password
Output: Launch User Dashboard
Method Description:
Whenever a user tries to login using the normal logging on function, this method is called. If the user enters the correct login credentials, he is taken to his personal dashboard. If the wrong credentials are entered then an error message is displayed.

3.1.2. Method 2: adminLogin()

Input: username, password
Output: Launch Admin Dashboard
Method Description:
Whenever a user tries to login using the admin logging on function, this method is called. If the user enters the correct login credentials, he is taken to his personal admin dashboard. If the wrong credentials are entered, or if the user is not an admin then an error message is displayed.

3.2. Class name: UIDashboard

Description: This abstract class contains functionalities to increase or decrease the thermostat temperature in a particular room, and also allows the user to log out.

3.2.1. Method 1: onTempIncrement()

Input: roomID
Output: Calls the incrementTemp() function of class thermostatControl
Method Description:
This function is an event listener which attaches itself to the temperature control buttons and whenever the increment button is clicked on, this function calls the function that implements the temperature increase functionality

3.2.2. Method 2: onTempDecrement()

Input: roomID
Output: Calls the decrementTemp() function of class thermostatControl
Method Description:
This function is an event listener which attaches itself to the temperature control buttons and whenever the decrement button is clicked on, this function calls the function that implements the temperature decrease functionality

3.2.3. Method 3: onLogOut()

Input: none
Output: Logs the user out of the dashboard and back to the login page
Method Description:
This function is an event listener which attaches itself to the logout button, and whenever the user clicks on the logout button, they are logged out of the system and taken back to the login page

3.3 Class name : adminDashboard

This class stores the list of rooms and provides functionality to view rooms and update their details. Implements attributes and methods from UIDashboard.

3.3.1. Method 1 : onBrowseRooms()

Input: roomList[]
Output: displays info of the newly chosen room
Method Description :
This method provides us with the ability to view details of each room and also provides functionality to change the room we are viewing.

3.3.2. Method 2 : onUpdateRequest()

Input: none
Output: updates required data in the database
Method Description :
This function updates necessary information in the database upon request from the administrator on clicking the button.

3.4 Class name : userDashboard

This class contains the attributes required to view the details of rooms.This child class Implements attributes and methods from UIDashboard.

3.5 Class name : thermostatControl

This abstract class provides functionality to modify current temperature.

3.5.1 Method 1 : incrementTemp()

Input: roomID
Output: increases the thermostat temperature in the database by 1 degree

Method Description :
This function is an event listener which attaches itself to the temperature control buttons and whenever the decrement button is clicked on, this function calls the function that implements the temperature decrease functionality.

3.5.2 method 2 : decrementTemp()

Input: roomID
Output: decreases the thermostat temperature in the database by 1 degree

3.6 Class name : alarmSystem

This class allows us to set alert messages and temperature constraints on each of the rooms.

3.6.1 Method 1 : onTemperatureTooHigh()

Input : alertMessage, emergencyContacts[], alertTempHigh
Output : sends an emergency message to all emergency contacts if the temperature is too high

Method Description:
This method monitors the temperature and sends alerts to the registered users if the temperature is exceeding the higher limit.

3.6.2 Method 2: onTemperatureTooLow()

Input: alertMessage, emergencyContacts[], alertTempLow
Output: sends an emergency message to all emergency contacts if the temperature is low
Method Description :
This method monitors the temperature and sends alerts to the registered users if the temperature is preceding the lower limit.

6.1 Pseudocode for components

6.1.1 Class Name: UIPreLogin

Method 1: login()
Pseudo-code:
Input: username, password
Output: Launch User Dashboard

if username exists in database and password is same as password for corresponding username in database
    grant user access
    show user dashboard
else if username does not exist in database
    show invalid username message
else if username exists in database but passwords do not match
    show invalid password message

Method 2: adminLogin()
Pseudo-code:
Input: username, password
Output: Launch Admin Dashboard

if username exists in database and password is same as password for corresponding username in database
    if user has admin access
        grant admin access
        show admin control panel
    else
        show no admin access message
else if username does not exist in database
    show invalid username message
else if username exists in database but passwords do not match
    show invalid password message

6.1.2 Class Name: UIDashboard

Method 1: onTempIncrement()
Pseudo-code:
Input: roomID
Output: Calls the incrementTemp() function of class thermostatControl
if increment button exists
    attach self to increment button
when pressed
    call incrementTemp() function and pass roomID

Method 2: onTempDecrement()
Pseudo-code:
Input: roomID
Output: Calls the decrementTemp() function of class thermostatControl
if decrement button exists
    attach self to decrement button
when pressed
    call decrementTemp() function and pass roomID

Method 3: onLogOut()
Pseudo-code:
Input: none
Output: Logs the user out of the dashboard and back to the login page
if logout button exists
    attach self to logout button
when pressed
    change instance to login screen, remove access to dashboard

6.1.3 Class Name: userDashboard

Child class inherits functions from parent class UIDashboard

6.1.4Class Name: adminDashboard

Child class inherits functions from parent class UIDashboard

Method 1: onBrowseRooms()
Pseudo-code:
Input: roomList[]
Output: displays info of the newly chosen room
```
if roomBrowse button exists
        attach self to roomBrowse button
when pressed
        store ID entered in text field in roomID
        change display to information of new roomID
```

Method 2: onDataUpdateRequest()
Pseudo-code:
Input: none
Output: updates required data in the database
```
if user has admin access
        if user wants to set new emergency contact
                initialise email to email ID entered by user
                if email exists in database
                        set it's user as emergency contact
                else
                        show user does not exist message
        if user wants to change emergency message
                set emergency message as message entered by the user
        if user wants to change temperature range
                set temperature range as range entered by the user
else
        show no admin access message
```

6.1.5 Class Name: thermostatControl

Method 1: incrementTemp()
Pseudo-code:
Input: roomID
Output: increases the thermostat temperature in the database by 1

```
if temp is 40
        show error "Maximum temperature supported by thermostat is 40 Celsius"
else
        increase temp by 1
        update data in database
```

Method 2: decrementTemp()
Pseudo-code:

```
if temp is 18
        show error "Minimum temperature supported by thermostat is 18 Celsius"
else
        decrease temp by 1
        update data in database
```

6.1.6 Class Name: alarmSystem

Method 1: onTemperatureTooHigh()
Pseudo-code:
Input: alertMessage, emergencyContacts[], alertTempHigh
Output: sends an emergency message to all emergency contacts if the temperature is too high

```
if temp is greater than alertTempHigh
        for all emails in emergencyContacts
                send message alertMessage
        if room is occupied
                send message alertMessage to room owner
```

Method 2: onTemperatureTooLow()
Pseudo-code:
Input: alertMessage, emergencyContacts[], alertTempLow
Output: sends an emergency message to all emergency contacts if the temperature is too low

```
if temp is less than alertTempLow
        for all emails in emergencyContacts
                send message alertMessage
        if room is occupied
                send message alertMessage to room owner
```

7.Traceability Matrix

requirement_id	Use case	High level design element	Low level design element-1	Low level design element-2
1	U-9,U-3	package1	6.1.4C	browseRooms
2	*U-6*,U-1,U-2	package1	6.1.1C,6.1.5C	6.1.1M1,6.1.1M2,, 6.1.5M1,6.1.5M2
3	*U-4*	package1	6.1.6C	6.1.6M1, 6.1.6,M2
4	U-7	package1	6.1.4C	6.1.4M1, 6.1.4M2