Supervised by Dr. Elaf Islam

Smart Campus Management System Report

by: **Amal Ahmad Al-turkistani**

Led a collaborative team of 8 additional developers
(Names withheld for privacy)

Table of Contents

Idea Summary

Our system (Smart University Campus Management) is an intelligent system that manages the classrooms in the university campus. When entering the program, three paths are displayed for system users:

(Data Entry) Path

This path is for the administrator, where classroom data is entered (room type, room ID, capacity, attachments, and tools).

Book a Room Path:

This path is for the user, and it provides 7 different sub-paths:

1. VIEW ROOM REPORTS

displays (available rooms, booked rooms, smart rooms, and rooms under maintenance).

2. VIEW AVAILABLE ROOM

allows viewing only the rooms available for booking.

3. VIEW BOOKED ROOMS

allows viewing only the rooms that are currently booked.

4.BOOK ROOM

allows the user to book rooms and displays the instructions that must be followed in case of booking.

5.CANCEL A BOOKING

allows the user to cancel room bookings.

6.VIEW ALL ROOM DETAILS

displays (room type, room ID, capacity, whether it is available or not, maintenance status).

7.RETURN TO MAIN MENU

after finishing, it returns you to the main menu

Exit program

where you should select the third path to close the program.

How does the project contribute to Vision 2030 ?

The Smart Campus Management project contributes to supporting Vision 2030 by providing a developmental environment based on modern technology.
It facilitates the smart management of halls and reduces wasted time and effort, aligning with the vision's goal of improving the quality of government and educational services.
This project also supports the vision's objectives by building an advanced knowledge society, as it provides an electronic reservation system and connects colleges to their facilities. This type of system contributes to raising the quality of education and improving the experience for faculty members and students.

Screenshots of the program outputs

```
=== Classroom Management System ===
Welcome! Choose your role:
1. Data Entry (Administrator)
2. Book a Room (User)
3. Exit Program
Enter choice (1, 2, or 3): 1

Enter College Name: College of Computing and Information Technology.
How many rooms do you want to add? 4

Room #1
Choose room type:
1. Lecture Room
2. Lab
3. Smart Classroom
Enter choice: 1
Enter Lecture Room ID: 6102
Enter capacity: 80
Has projector? (true/false): true
Has whiteboard? (true/false): true

Room #2
Choose room type:
1. Lecture Room
2. Lab
3. Smart Classroom
Enter choice: 2
Enter Lab Room ID: 12102
Enter capacity: 40
Needs maintenance? (true/false): false
Computers count: 24
```

```
Room #3
Choose room type:
1. Lecture Room
2. Lab
3. Smart Classroom
Enter choice: 2
Enter Lab Room ID: 12103
Enter capacity: 20
Needs maintenance? (true/false): true
Computers count: 20

Room #4
Choose room type:
1. Lecture Room
2. Lab
3. Smart Classroom
Enter choice: 3
Enter Smart Room ID: 13102
Enter capacity: 45
Has smart screen? (true/false): true
Has projector? (true/false): true

*** Data saved successfully! ***
Total rooms added: 4

Redirecting to Booking Menu...
Invalid choice! Please try again.
```

Screenshots of the program outputs

```
=== Classroom Management System ===
Welcome! Choose your role:
1. Data Entry (Administrator)
2. Book a Room (User)
3. Exit Program
Enter choice (1, 2, or 3): 2

=== Room Booking System ===
Using your entered room data.

--- Booking Menu ---
1. View Room Reports
2. View Available Rooms
3. View Booked Rooms
4. Book a Room
5. Cancel a Booking
6. View All Room Details
7. Return to Main Menu
Enter choice: 1
=== Available Rooms ===
6102 - Lecture Room
12102 - Laboratory
13102 - Smart Classroom

=== Occupied Rooms ===

=== Smart Rooms ===
13102 - Smart Classroom

=== Rooms Under Maintenance ===
12103 - Laboratory
```

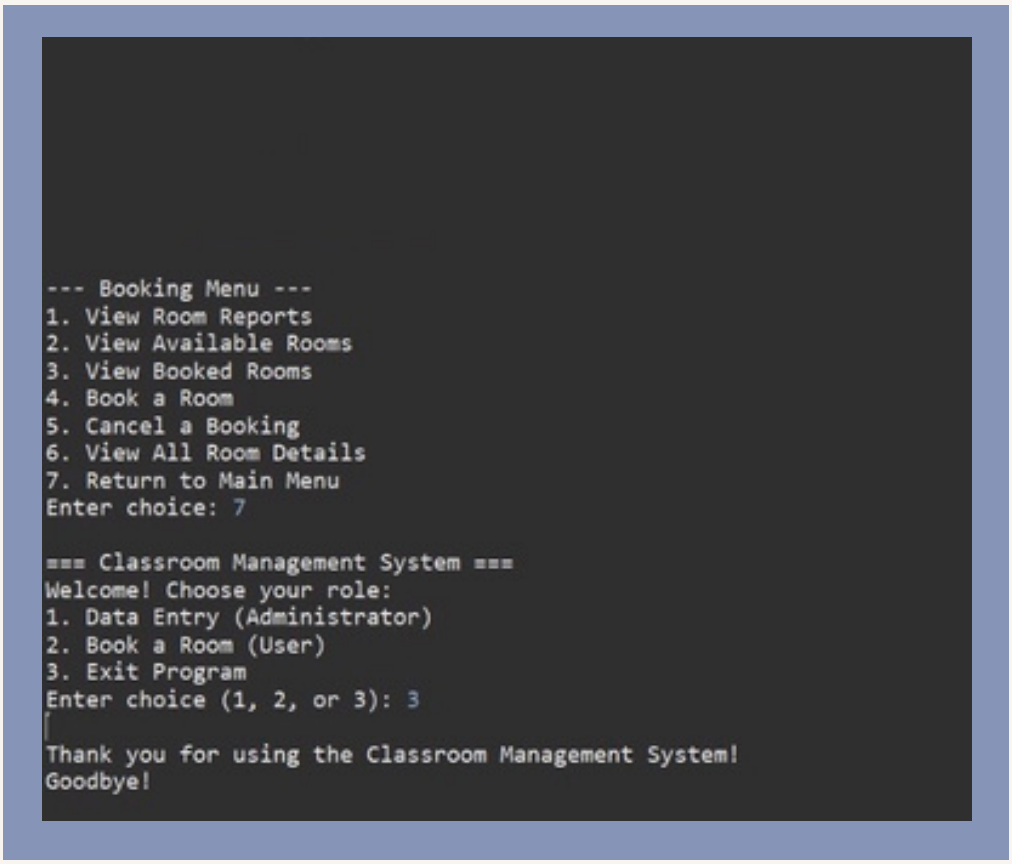
```
--- Booking Menu ---
1. View Room Reports
2. View Available Rooms
3. View Booked Rooms
4. Book a Room
5. Cancel a Booking
6. View All Room Details
7. Return to Main Menu
Enter choice: 2

=== Available Rooms for Booking ===
6102 - Lecture Room (Capacity: 80)
12102 - Laboratory (Capacity: 40)
13102 - Smart Classroom (Capacity: 45)

--- Booking Menu ---
1. View Room Reports
2. View Available Rooms
3. View Booked Rooms
4. Book a Room
5. Cancel a Booking
6. View All Room Details
7. Return to Main Menu
Enter choice: 3

=== Booked Rooms ===
No rooms are currently booked.
```

Screenshots of the program outputs

```
--- Booking Menu ---
1. View Room Reports
2. View Available Rooms
3. View Booked Rooms
4. Book a Room
5. Cancel a Booking
6. View All Room Details
7. Return to Main Menu
Enter choice: 4

Enter room number to book: 6102
Enter number of students: 65

*** Room 6102 successfully booked! ***
IMPORTANT REMINDERS:
- Maximum booking duration: 5 hours
- Please cancel your booking when you finish
- This allows other users to book the room
- Thank you for your cooperation!


--- Booking Menu ---
1. View Room Reports
2. View Available Rooms
3. View Booked Rooms
4. Book a Room
5. Cancel a Booking
6. View All Room Details
7. Return to Main Menu
Enter choice: 5

Enter room number to cancel booking: 6102
Booking for room 6102 has been cancelled.
```

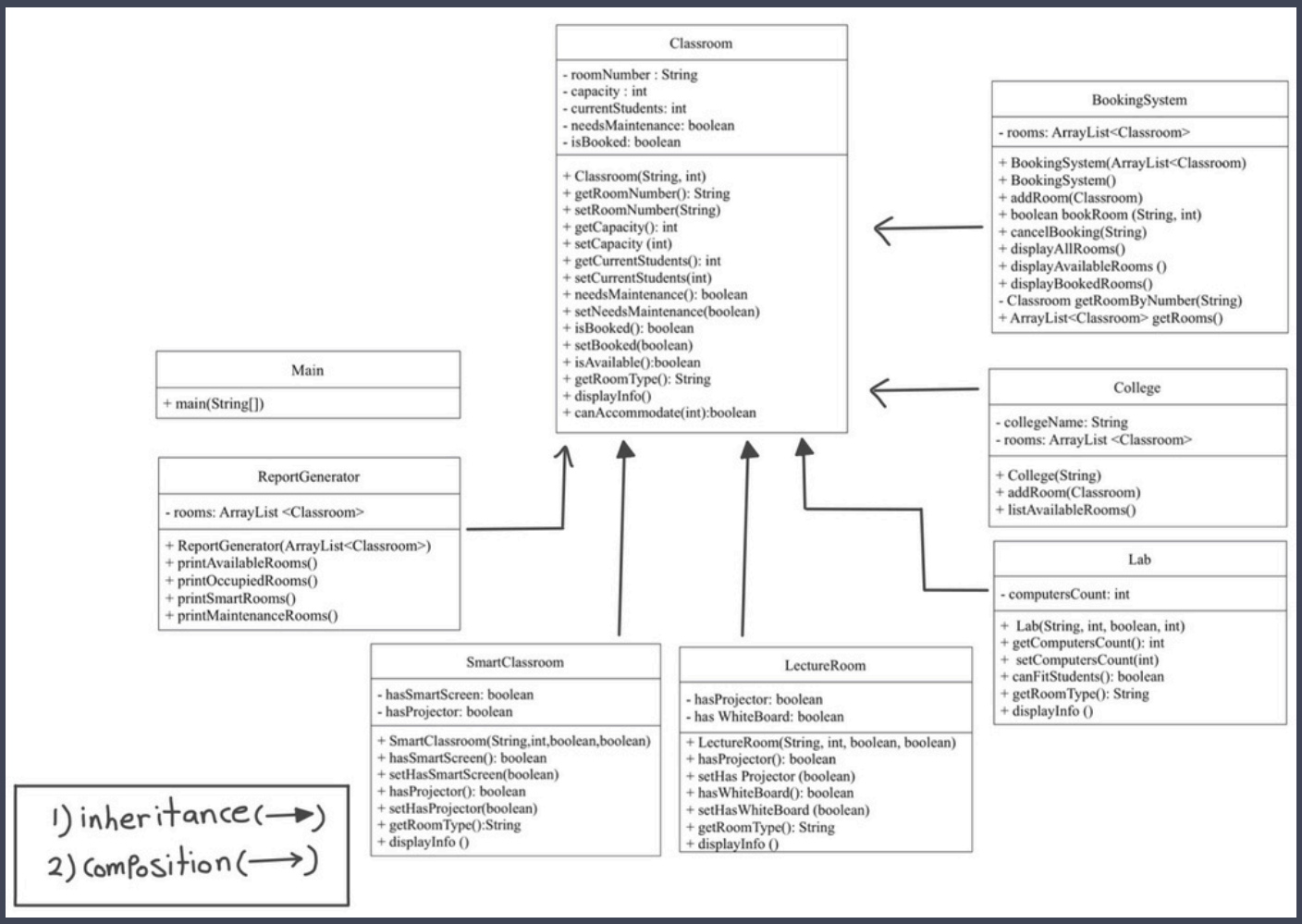
```
--- Booking Menu ---
1. View Room Reports
2. View Available Rooms
3. View Booked Rooms
4. Book a Room
5. Cancel a Booking
6. View All Room Details
7. Return to Main Menu
Enter choice: 6

=== All Rooms ===
Room Number: 6102
Type: Lecture Room
Capacity: 80 students
Status: Available
----------------------------
Room Number: 12102
Type: Laboratory
Capacity: 40 students
Status: Available
----------------------------
Room Number: 12103
Type: Laboratory
Capacity: 20 students
Status: Not Available
Under Maintenance
----------------------------
Room Number: 13102
Type: Smart Classroom
Capacity: 45 students
Status: Available
----------------------------
```

Screenshots of the program outputs



UML diagram

Please rotate your device and enlarge the page to view the diagram more clearly.



Snapshots from the source code

abstract

```
public abstract class Classroom {
    // Instance variables
    private String roomNumber;
    private int capacity;
    private int currentStudents;
    private boolean needsMaintenance;
    private boolean isBooked;
```

composition (has - a)

```
// Method to add a room
    public void addRoom(Classroom room) {
        rooms.add(room);
    }
```

Snapshots from the source code

constructor

```
public class BookingSystem {

    private ArrayList<Classroom> rooms;

    // Constructor receives a list of rooms
    public BookingSystem(ArrayList<Classroom>
rooms) {
        this.rooms = rooms;
    }
```

Inhertance

```
public class Lab extends Classroom {

    private int computersCount;

    public Lab(String roomNumber, int capacity,
boolean needsMaintenance, int computersCount) {
        super(roomNumber, capacity);
        setCurrentStudents(0);
// Start with 0 students
        setNeedsMaintenance(needsMaintenance);
        this.computersCount = computersCount;
    }
```

Snapshots from the source code

Override

```
@Override
    public String getRoomType() {
        return "Laboratory";
    }
    @Override
    public void displayInfo() {
        super.displayInfo();
// Call parent method first
        System.out.println("Computers Count: " +
computersCount);
        System.out.println(
"Can Fit Current Students: " + (canFitStudents()
? "Yes" : "No"));
    }
```