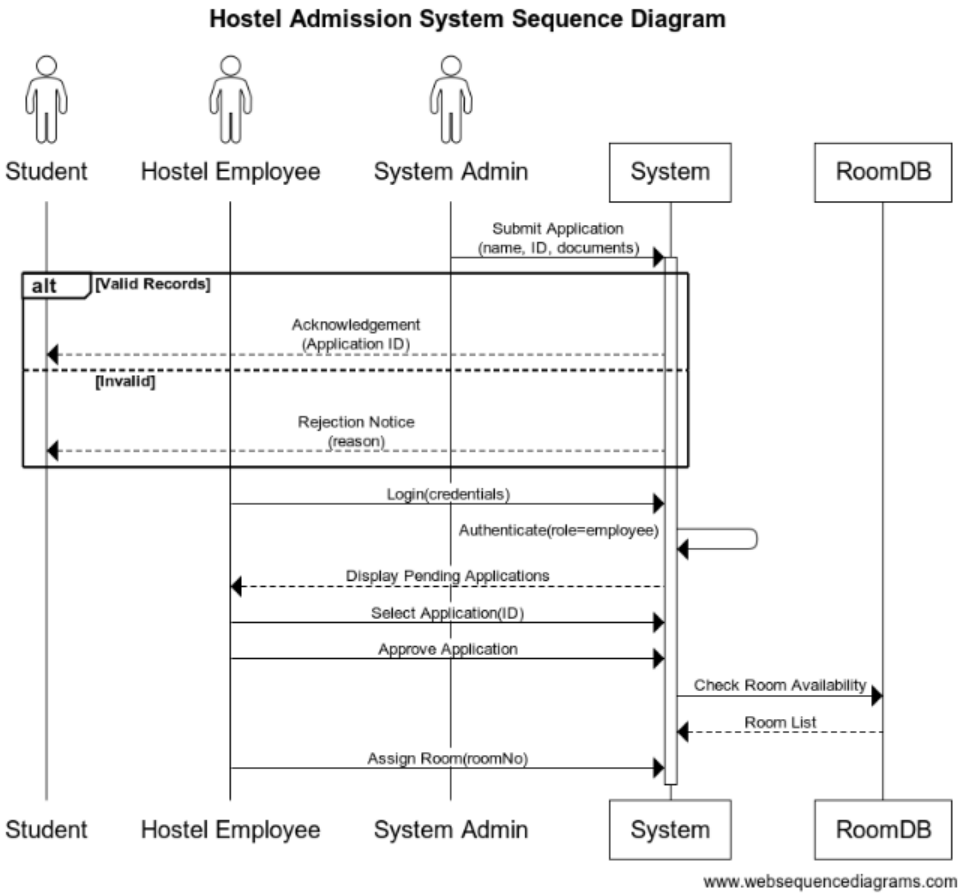
COMSATS
ISLAMABAD UNIVERSITY

Assignment No : 01

Name : ***Uzair Arif***

Registration No : ***Sp23-BSE-168***

Use Case Element	Details
Use Case Name	New Student Admission
Primary Actor	Hostel Warden
Secondary Actors	Admin
Stakeholders	• Student: Wants smooth room allocation • Warden: Needs accurate record-keeping • Admin: Ensures policy compliance
Preconditions	1. Student is institutionally admitted 2. Required documents submitted 3. System operational with valid warden access 4. Rooms available (or waiting list active)
Postconditions	1. Student registered in hostel system 2. Room assigned and marked occupied 3. Attendance record initialized 4. Welcome notification sent
Main Success Scenario	1. Warden logs in → Student Management 2. Selects "New Admission" 3. Enters student details + documents 4. System verifies institutional admission 5. Auto-assigns room (or manual selection) 6. Generates hostel ID card 7. Sys sends confirmation to all parties
Alternative Flows	A1. No rooms available → Waitlist A2. Document missing → System flags for completion A3. Duplicate entry → System alerts warden
Exception Flows	E1. System crash during entry → Data recovery protocol E2. Unauthorized access attempt → Logs incident + alerts admin
Special Requirements	• Biometric integration for check-in • Automated bed/room inventory updates • GDPR-compliant data storage



CODE

```
import java.util.*;

class Student {
    String id, name, email;
    boolean admitted, docs;
    int room;
```

```
    public Student(String name, String email, boolean admitted, boolean
docs, int room) {
        this.id = UUID.randomUUID().toString();
        this.name = name;
        this.email = email;
        this.admitted = admitted;
        this.docs = docs;
        this.room = room;
    }
}

public class NewAdmissionShort {

    static List<Student> students = new ArrayList<>();
    static boolean[] rooms = new boolean[5]; // 5 rooms, false = available

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Student Name: ");
        String name = sc.nextLine();
```

```
        System.out.print("Enter Student Email: ");
        String email = sc.nextLine();

        System.out.print("Is the student institutionally admitted? (true/false): ");
        boolean admitted = sc.nextBoolean();

        System.out.print("Are documents submitted? (true/false): ");
        boolean docs = sc.nextBoolean();

        if (!admitted || !docs) {
            System.out.println("❌ Admission or documents missing.");
            return;
        }

        int assignedRoom = assignRoom();
        if (assignedRoom == -1) {
            System.out.println("⚠ No rooms available. Student added to waitlist.");
            return;
        }
```

```
        Student s = new Student(name, email, admitted, docs,
assignedRoom);
        students.add(s);


        System.out.println("✓ " + s.name + " admitted and assigned Room
" + s.room);
    }


    static int assignRoom() {
        for (int i = 0; i < rooms.length; i++) {
            if (!rooms[i]) {
                rooms[i] = true;
                return i + 1; // Room numbers start from 1
            }
        }
        return -1; // No rooms available
    }
}
```