COMSATS ISLAMABAD UNIVERSITY

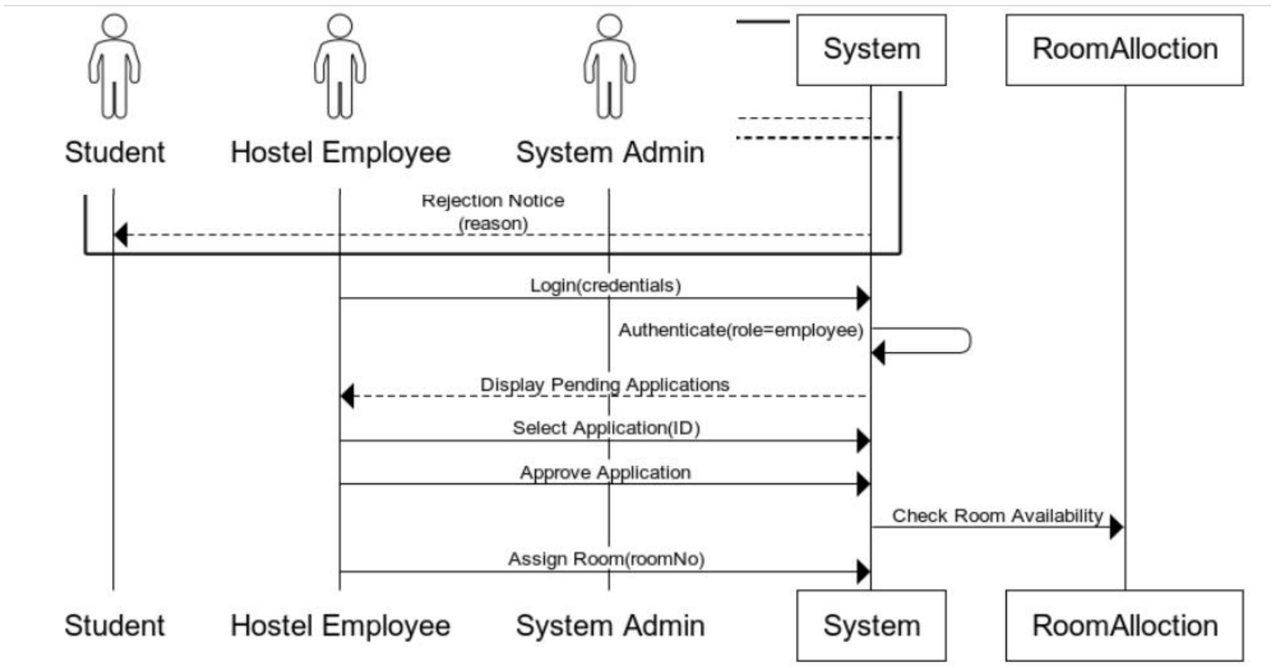
Assignment No : 01

Name : ***Uzair Arif***

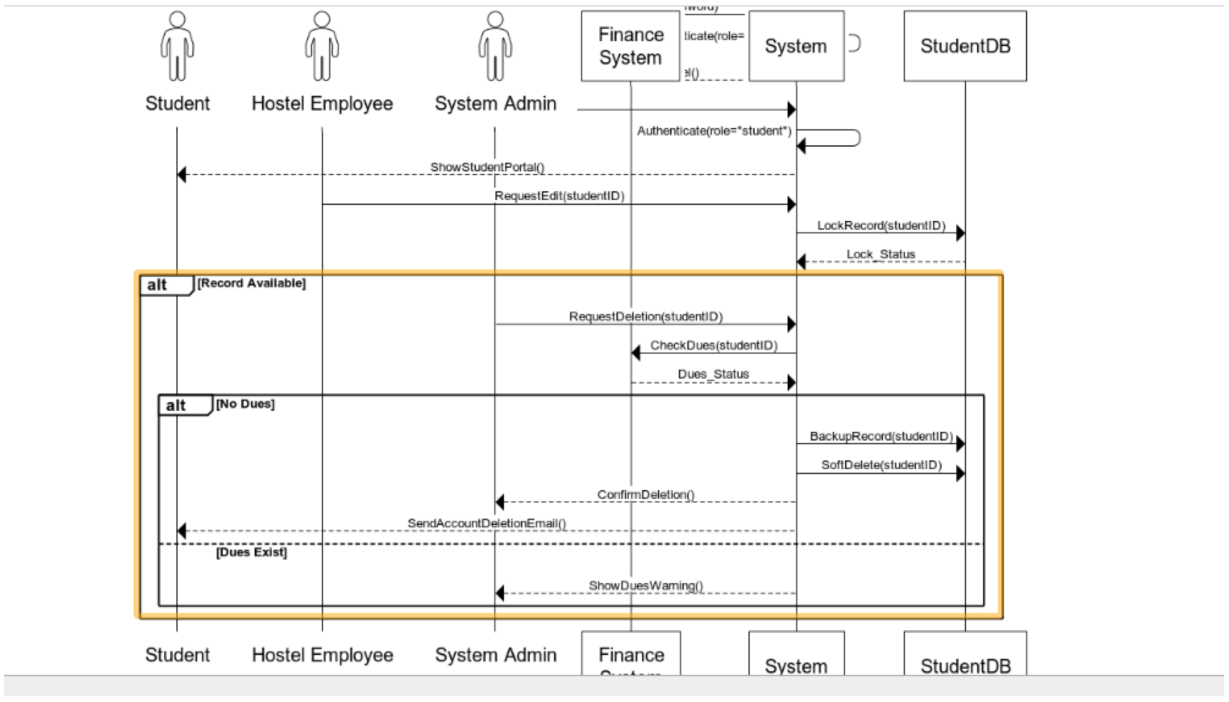
Registration No : ***Sp23-BSE-168***

Use Case Element	Update Student Details	Delete Student Record
Use Case Name	Student Information Update	Student Record Removal
Primary Actor	Hostel Warden	Hostel Warden
Secondary Actors	Admin	Admin
Stakeholders	• Student: Needs accurate records • Warden: Requires edit permissions • Admin: Audits changes	• Warden: Needs verification authority • Admin: Ensures data compliance
Preconditions	1. Student exists in system 2. Warden has "Edit" privileges 3. Original records accessible	1. Student marked for departure 2. All dues cleared 3. Backup exists
Postconditions	1. Records updated with timestamp 2. Change log entry created 3. Notifications sent if critical field modified	1. Record moved to archive 2. Room status updated 3. All accesses revoked
Main Success Scenario	1. Search student by ID 2. Select "Edit" 3. Modify fields (contact/room) 4. System validates changes 5. Save with digital signature	1. Verify student eligibility for deletion 2. Confirm room vacated 3. Execute soft-delete 4. Update inventory
Alternative Flows	A1. Room transfer → Update both room records A2. Contact change → SMS verification	A1. Outstanding fees → Block deletion A2. Shared facilities → Cascade update
Exception Flows	E1. Concurrent edit conflict → Merge protocol E2. Invalid data format → Rejection	E1. Dependent records exist → Partial delete E2. System archive full → Alert admin

NAME: UZAIR ARIF REG NO :SP23-BSE-168



Student Delete & update



Code:

```
import java.util.*;

public class StudentManager {
  static class Student {
    String id, name, email, course;

    public Student(String name, String email, String course) {
      this.id = UUID.randomUUID().toString();
      this.name = name;
      this.email = email;
      this.course = course;
    }

    @Override
    public String toString() {
      return String.format("ID: %s\nName: %s\nEmail: %s\nCourse: %s\n",
                  id, name, email, course);
    }
  }

  static List<Student> students = new ArrayList<>();
  static Scanner scanner = new Scanner(System.in);

  public static void main(String[] args) {
    while (true) {
      System.out.println("\nSTUDENT MANAGEMENT SYSTEM");
```

```
            System.out.println("1. Add Student");
            System.out.println("2. View All Students");
            System.out.println("3. Update Student");
            System.out.println("4. Delete Student");
            System.out.println("5. Exit");
            System.out.print("Enter choice: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1 -> addStudent();
                case 2 -> viewStudents();
                case 3 -> updateStudent();
                case 4 -> deleteStudent();
                case 5 -> System.exit(0);
                default -> System.out.println("Invalid choice!");
            }
        }
    }

    static void addStudent() {
        System.out.print("Enter name: ");
        String name = scanner.nextLine();
        System.out.print("Enter email: ");
        String email = scanner.nextLine();
        System.out.print("Enter course: ");
        String course = scanner.nextLine();

        students.add(new Student(name, email, course));
        System.out.println("Student added successfully!");
    }

    static void viewStudents() {
        if (students.isEmpty()) {
            System.out.println("No students found!");
            return;
        }
        students.forEach(System.out::println);
    }

    static void updateStudent() {
        System.out.print("Enter student ID to update: ");
        String id = scanner.nextLine();

        students.stream()
            .filter(s -> s.id.equals(id))
            .findFirst()
            .ifPresentOrElse(s -> {
                System.out.print("Enter new name: ");
                s.name = scanner.nextLine();
                System.out.print("Enter new email: ");
                s.email = scanner.nextLine();
                System.out.print("Enter new course: ");
            s.course = scanner.nextLine();
            System.out.println("Student updated!");
        }, () -> System.out.println("Student not found!"));
    }
    static void deleteStudent() {
        System.out.print("Enter student ID to delete: ");
        String id = scanner.nextLine();
        if (students.removeIf(s -> s.id.equals(id))) {
            System.out.println("Student deleted!");
        } else {
            System.out.println("Student not found!");
        }
    }
}
```