public class Student {

private String name;

private int rollNumber;

private String grade;

private String email;

// Constructor

public Student(String name, int rollNumber, String grade, String email) {

this.name = name;

this.rollNumber = rollNumber;

this.grade = grade;

this.email = email;

}

// Getters and Setters

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public int getRollNumber() { return rollNumber; }

public void setRollNumber(int rollNumber) { this.rollNumber = rollNumber; }

public String getGrade() { return grade; }

public void setGrade(String grade) { this.grade = grade; }

public String getEmail() { return email; }

public void setEmail(String email) { this.email = email; }

@Override

public String toString() {

return "Student{" +

"name='" + name + '\'' +

", rollNumber=" + rollNumber +

", grade='" + grade + '\'' +

", email='" + email + '\'' +

'}';

}

}import java.util.ArrayList;

import java.util.List;

public class StudentManagementSystem {

private List<Student> students;

public StudentManagementSystem() {

this.students = new ArrayList<>();

}

// Add a student

public void addStudent(Student student) {

if (validateStudent(student)) {

students.add(student);

System.out.println("Student added successfully!");

}

}

// Remove a student

public void removeStudent(int rollNumber) {

students.removeIf(student -> student.getRollNumber() == rollNumber);

System.out.println("Student removed successfully!");

}

// Search for a student

public Student searchStudent(int rollNumber) {

return students.stream()

.filter(student -> student.getRollNumber() == rollNumber)

.findFirst()

.orElse(null);

}

// Display all students

public void displayAllStudents() {

if (students.isEmpty()) {

System.out.println("No students found!");

return;

}

students.forEach(System.out::println);

}

// Validate student data

private boolean validateStudent(Student student) {

if (student.getName().isEmpty() || student.getGrade().isEmpty() ||

student.getEmail().isEmpty()) {

System.out.println("All fields must be filled!");

return false;

}

if (searchStudent(student.getRollNumber()) != null) {

System.out.println("Student with this roll number already exists!");

return false;

}

return true;

}

}import java.util.Scanner;

public class Main {

public static void main(String[] args) {

StudentManagementSystem sms = new StudentManagementSystem();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\n=== Student Management System ===");

System.out.println("1. Add Student");

System.out.println("2. Remove Student");

System.out.println("3. Search Student");

System.out.println("4. Display All Students");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter name: ");

String name = scanner.nextLine();

System.out.print("Enter roll number: ");

int rollNumber = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter grade: ");

String grade = scanner.nextLine();

System.out.print("Enter email: ");

String email = scanner.nextLine;

Student student = new Student(name, rollNumber, grade, email);

sms.addStudent(student);

break;

}

case 2:

System.out.print("Enter roll number to remove: ");

int rollToRemove = scanner.nextInt();

sms.removeStudent(rollToRemove);

break;

case 3:

System.out.print("Enter roll number to search: ");

int rollToSearch = scanner.nextInt();

Student foundStudent = sms.searchStudent(rollToSearch);

if (foundStudent != null) {

System.out.println("Student found: " + foundStudent);

} else {

System.out.println("Student not found!");

}

break;

case 4:

sms.displayAllStudents();

break;

case 5:

System.out.println("Thank you for using Student Management System!");

scanner.close();

System.exit(0);

default:

System.out.println("Invalid choice! Please try again.");

}

}

}