```
import java.io.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Student {
    private String name;
    private int rollNumber;
    private String grade;

    public Student(String name, int rollNumber, String grade) {
        this.name = name;
        this.rollNumber = rollNumber;
        this.grade = grade;
    }

    public String getName() {
        return name;
    }

    public int getRollNumber() {
        return rollNumber;
    }

    public String getGrade() {
        return grade;
    }

    @Override
    public String toString() {
        return "Name: " + name + "\nRoll Number: " + rollNumber + "\nGrade: " + grade + "\n";
```

```
    }
}

class StudentManagementSystem {
    private List<Student> students;

    public StudentManagementSystem() {
        students = new ArrayList<>();
    }

    public void addStudent(Student student) {
        students.add(student);
    }

    public void removeStudent(int rollNumber) {
        students.removeIf(student -> student.getRollNumber() == rollNumber);
    }

    public Student searchStudent(int rollNumber) {
        for (Student student : students) {
            if (student.getRollNumber() == rollNumber) {
                return student;
            }
        }
        return null;
    }

    public void displayAllStudents() {
        for (Student student : students) {
            System.out.println(student);
        }
```

```
    }

    public void saveStudentDataToFile(String filename) {
        try (PrintWriter writer = new PrintWriter(new FileWriter(filename))) {
            for (Student student : students) {
                writer.println(student.getName() + "," + student.getRollNumber() + "," +
student.getGrade());
            }
            System.out.println("Student data saved to " + filename);
        } catch (IOException e) {
            System.out.println("Error saving student data to file: " + e.getMessage());
        }
    }

    public void loadStudentDataFromFile(String filename) {
        students.clear();
        try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 3) {
                    String name = parts[0];
                    int rollNumber = Integer.parseInt(parts[1]);
                    String grade = parts[2];
                    students.add(new Student(name, rollNumber, grade));
                }
            }
            System.out.println("Student data loaded from " + filename);
        } catch (IOException e) {
            System.out.println("Error loading student data from file: " + e.getMessage());
        }
```

```
    }
}

public class Task_5_Student_management_system {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        StudentManagementSystem system = new StudentManagementSystem();

        while (true) {
            System.out.println("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. Save Student Data");
            System.out.println("6. Load Student Data");
            System.out.println("7. Exit");
            System.out.print("Select an option: ");

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

            switch (choice) {
                case 1:
                    System.out.print("Enter student name: ");
                    String name = scanner.nextLine();
                    System.out.print("Enter roll number: ");
                    int rollNumber = scanner.nextInt();
                    scanner.nextLine(); // Consume newline
                    System.out.print("Enter grade: ");
                    String grade = scanner.nextLine();
```

```
            system.addStudent(new Student(name, rollNumber, grade));
            break;
        case 2:
            System.out.print("Enter roll number of student to remove: ");
            int rollNumberToRemove = scanner.nextInt();
            system.removeStudent(rollNumberToRemove);
            break;
        case 3:
            System.out.print("Enter roll number of student to search: ");
            int rollNumberToSearch = scanner.nextInt();
            Student searchedStudent = system.searchStudent(rollNumberToSearch);
            if (searchedStudent != null) {
                System.out.println("Student found:\n" + searchedStudent);
            } else {
                System.out.println("Student not found.");
            }
            break;
        case 4:
            System.out.println("All Students:");
            system.displayAllStudents();
            break;
        case 5:
            System.out.print("Enter filename to save student data: ");
            String saveFilename = scanner.nextLine();
            system.saveStudentDataToFile(saveFilename);
            break;
        case 6:
            System.out.print("Enter filename to load student data: ");
            String loadFilename = scanner.nextLine();
            system.loadStudentDataFromFile(loadFilename);
            break;
```

```
            case 7:
                System.out.println("Exiting Student Management System. Goodbye!");
                scanner.close();
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please select a valid option.");
        }
    }
  }
}
```