Student Management System - Group Project

Group Members:  
Mehul Bhargav  
Sukhjit Singh  
Sahil  
Varisdeep Singh  
Shivam Bhargav

# Student.java  
  
public class Student {  
 private String studentId;  
 private String name;  
 private String grade;  
  
 public Student(String studentId, String name, String grade) {  
 this.studentId = studentId;  
 this.name = name;  
 this.grade = grade;  
 }  
  
 public String getStudentId() { return studentId; }  
 public String getName() { return name; }  
 public String getGrade() { return grade; }  
 public void setGrade(String grade) { this.grade = grade; }  
  
 @Override  
 public String toString() {  
 return "Student ID: " + studentId + ", Name: " + name + ", Grade: " + grade;  
 }  
}

# FileHandler.java  
  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class FileHandler {  
 private static final String FILE\_PATH = "students.txt";  
  
 public static void saveStudent(Student student) throws IOException {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE\_PATH, true))) {  
 writer.write(student.getStudentId() + "," + student.getName() + "," + student.getGrade());  
 writer.newLine();  
 }  
 }  
  
 public static List<Student> loadStudents() throws IOException {  
 List<Student> students = new ArrayList<>();  
 try (BufferedReader reader = new BufferedReader(new FileReader(FILE\_PATH))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] details = line.split(",");  
 if (details.length == 3) {  
 students.add(new Student(details[0], details[1], details[2]));  
 }  
 }  
 }  
 return students;  
 }  
  
 public static void saveAllStudents(List<Student> students) throws IOException {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE\_PATH))) {  
 for (Student student : students) {  
 writer.write(student.getStudentId() + "," + student.getName() + "," + student.getGrade());  
 writer.newLine();  
 }  
 }  
 }  
}

# CLI.java  
  
import java.io.IOException;  
import java.util.List;  
import java.util.Scanner;  
  
public class CLI {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 while (true) {  
 System.out.println("\nStudent Management System");  
 System.out.println("1. Add Student");  
 System.out.println("2. Update Student Grade");  
 System.out.println("3. View All Students");  
 System.out.println("4. Exit");  
 System.out.print("Enter your choice: ");  
 int choice = scanner.nextInt();  
 scanner.nextLine(); // Consume newline  
  
 switch (choice) {  
 case 1:  
 addStudent(scanner);  
 break;  
 case 2:  
 updateStudentGrade(scanner);  
 break;  
 case 3:  
 viewAllStudents();  
 break;  
 case 4:  
 System.out.println("Exiting...");  
 return;  
 default:  
 System.out.println("Invalid choice. Please try again.");  
 }  
 }  
 }  
  
 private static void addStudent(Scanner scanner) {  
 System.out.print("Enter Student ID: ");  
 String studentId = scanner.nextLine();  
 System.out.print("Enter Student Name: ");  
 String name = scanner.nextLine();  
 System.out.print("Enter Student Grade: ");  
 String grade = scanner.nextLine();  
  
 Student student = new Student(studentId, name, grade);  
 try {  
 FileHandler.saveStudent(student);  
 System.out.println("Student added successfully.");  
 } catch (IOException e) {  
 System.out.println("Error saving student: " + e.getMessage());  
 }  
 }  
  
 private static void updateStudentGrade(Scanner scanner) {  
 System.out.print("Enter Student ID to update: ");  
 String studentId = scanner.nextLine();  
  
 try {  
 List<Student> students = FileHandler.loadStudents();  
 boolean found = false;  
 for (Student student : students) {  
 if (student.getStudentId().equals(studentId)) {  
 System.out.print("Enter new grade: ");  
 String newGrade = scanner.nextLine();  
 student.setGrade(newGrade);  
 FileHandler.saveAllStudents(students);  
 System.out.println("Grade updated successfully.");  
 found = true;  
 break;  
 }  
 }  
 if (!found) {  
 System.out.println("Student ID not found.");  
 }  
 } catch (IOException e) {  
 System.out.println("Error updating grade: " + e.getMessage());  
 }  
 }  
  
 private static void viewAllStudents() {  
 try {  
 List<Student> students = FileHandler.loadStudents();  
 if (students.isEmpty()) {  
 System.out.println("No students found.");  
 } else {  
 System.out.println("Student List:");  
 for (Student student : students) {  
 System.out.println(student);  
 }  
 }  
 } catch (IOException e) {  
 System.out.println("Error loading students: " + e.getMessage());  
 }  
 }  
}