import java.util.ArrayList;

import java.util.Scanner;

// Student Class

class Student {

private int studentID;

private String name;

private int age;

private String address;

public Student(int studentID, String name, int age, String address) {

this.studentID = studentID;

this.name = name;

this.age = age;

this.address = address;

}

// Getters and Setters

public int getStudentID() {

return studentID;

}

public void setStudentID(int studentID) {

this.studentID = studentID;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public void viewGrades() {

// Mock method to simulate viewing grades

System.out.println("Grades for " + name + ": A");

}

public void enrollCourse(Course course) {

// Mock method to simulate enrolling in a course

System.out.println(name + " has enrolled in course: " + course.getCourseName());

}

}

// Course Class

class Course {

private int courseID;

private String courseName;

private int credits;

public Course(int courseID, String courseName, int credits) {

this.courseID = courseID;

this.courseName = courseName;

this.credits = credits;

}

// Getters and Setters

public int getCourseID() {

return courseID;

}

public void setCourseID(int courseID) {

this.courseID = courseID;

}

public String getCourseName() {

return courseName;

}

public void setCourseName(String courseName) {

this.courseName = courseName;

}

public int getCredits() {

return credits;

}

public void setCredits(int credits) {

this.credits = credits;

}

}

// Administrator Class for Backend Management

class Administrator {

private ArrayList<Student> students;

public Administrator() {

students = new ArrayList<>();

}

public void addStudent(Student student) {

students.add(student);

System.out.println("Student " + student.getName() + " added.");

}

public void updateStudent(int studentID, String newName, String newAddress) {

for (Student student : students) {

if (student.getStudentID() == studentID) {

student.setName(newName);

student.setAddress(newAddress);

System.out.println("Student " + studentID + " updated.");

return;

}

}

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

}

public void removeStudent(int studentID) {

students.removeIf(student -> student.getStudentID() == studentID);

System.out.println("Student " + studentID + " removed.");

}

public void listStudents() {

if (students.isEmpty()) {

System.out.println("No students available.");

} else {

for (Student student : students) {

System.out.println("ID: " + student.getStudentID() + ", Name: " + student.getName());

}

}

}

// Getter for students list

public ArrayList<Student> getStudents() {

return students;

}

}

// Main Class (Front-end and Back-end)

public class StudentManagementSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Administrator admin = new Administrator();

Course course1 = new Course(101, "Mathematics", 3); // Sample course

while (true) {

// Front-end menu options

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

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

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

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

System.out.println("4. List Students");

System.out.println("5. Enroll in Course");

System.out.println("6. View Grades");

System.out.println("7. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline left-over

switch (choice) {

case 1:

// Add student (Back-end)

System.out.print("Enter Student ID: ");

int id = scanner.nextInt();

scanner.nextLine(); // Consume newline left-over

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

String name = scanner.nextLine();

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

int age = scanner.nextInt();

scanner.nextLine(); // Consume newline

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

String address = scanner.nextLine();

Student newStudent = new Student(id, name, age, address);

admin.addStudent(newStudent); // Call backend method

break;

case 2:

// Update student details

System.out.print("Enter Student ID to Update: ");

int updateID = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter New Name: ");

String newName = scanner.nextLine();

System.out.print("Enter New Address: ");

String newAddress = scanner.nextLine();

admin.updateStudent(updateID, newName, newAddress); // Call backend method

break;

case 3:

// Remove student

System.out.print("Enter Student ID to Remove: ");

int removeID = scanner.nextInt();

admin.removeStudent(removeID); // Call backend method

break;

case 4:

// List all students

admin.listStudents(); // Call backend method

break;

case 5:

// Enroll student in course

System.out.print("Enter Student ID to Enroll: ");

int enrollID = scanner.nextInt();

for (Student student : admin.getStudents()) { // Access students using getter

if (student.getStudentID() == enrollID) {

student.enrollCourse(course1); // Call backend method

break;

}

}

break;

case 6:

// View student grades

System.out.print("Enter Student ID to View Grades: ");

int viewID = scanner.nextInt();

for (Student student : admin.getStudents()) { // Access students using getter

if (student.getStudentID() == viewID) {

student.viewGrades(); // Call backend method

break;

}

}

break;

case 7:

// Exit

System.out.println("Exiting system...");

scanner.close();

System.exit(0);

default:

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

}

}

}

}