Name: Oweipadei Joshua Bayefa

Student ID: 500221880

**User Interface Code:**

*StudentManagementSystem.java*

package User\_Interface;

import File\_Handling.FileHandler;

import Object\_Oriented\_Concepts.Student;

import Object\_Oriented\_Concepts.GraduateStudent;

import Object\_Oriented\_Concepts.UndergraduateStudent;

import java.util.ArrayList;

import java.util.Scanner;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.io.\*;

public class StudentManagementSystem {

private static final String FILE\_PATH = "students.txt";

private static final ArrayList students = new ArrayList<>();

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("\nStudent Management System Menu:");

System.out.println("1. Add new student");

System.out.println("2. View student details");

System.out.println("3. Update Student Grade");

System.out.println("4. Exit");

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

int choice = scanner.nextInt();

switch (choice) {

case 1:

addStudent();

break;

case 2:

viewStudentDetails();

break;

case 3:

updateGrade();

break;

case 4:

System.out.println("Exiting the system. Goodbye!");

System.exit(0);

break;

default:

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

}

}

}

private static void addStudent() {

Scanner scanner = new Scanner(System.in);

// Enter student ID

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

int studentId = scanner.nextInt();

// Enter student Name

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

String name = scanner.nextLine();

// Enter student grade

double grade = 0.0;

boolean validGrade = false;

while (!validGrade) {

try {

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

grade = scanner.nextDouble();

validGrade = true; // Set to true if no exception is thrown

} catch (java.util.InputMismatchException e) {

System.out.println("Invalid input for grade. Please enter a valid number.");

scanner.next(); // Consume the invalid input to avoid an infinite loop

}

}

// Enter student type

System.out.println("Select student type:");

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

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

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

int studentType = scanner.nextInt();

if (studentType == 1) {

UndergraduateStudent undergraduateStudent = new UndergraduateStudent(studentId, name, grade);

FileHandler.writeToFile(undergraduateStudent, undergraduateStudent.getStudentType());

} else if (studentType == 2) {

GraduateStudent graduateStudent = new GraduateStudent(studentId, name, grade);

FileHandler.writeToFile(graduateStudent, graduateStudent.getStudentType());

} else {

System.out.println("Invalid student type. Student not added.");

return;

}

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

}

private static void viewStudentDetails() {

System.out.println("Student Details:");

FileHandler.readFromFile();

}

private static void updateGrade() {

Scanner scanner = new Scanner(System.in);

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

int studentId = scanner.nextInt();

Student foundStudent = null;

for (Student student : students) {

System.out.println(student);

if (student.getStudentId() == studentId) {

foundStudent = student;

break;

}

}

// if student with this id does not exist

if (foundStudent == null) {

System.out.println("Student with ID " + studentId + " does not exist.");

return;

}

System.out.println("Select attribute to update:");

System.out.println("1. Name");

System.out.println("2. Grade");

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

int updateChoice = scanner.nextInt();

switch (updateChoice) {

case 1:

updateStudentName(foundStudent);

break;

case 2:

updateStudentGrade(foundStudent);

break;

default:

System.out.println("Invalid choice. No updates performed.");

}

}

private static void updateStudentName(Student foundStudent) {

Scanner scanner = new Scanner(System.in);

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

String name = scanner.nextLine();

foundStudent.setName(name);

System.out.println("Student Name updated successfully!");

}

private static void updateStudentGrade(Student foundStudent) {

Scanner scanner = new Scanner(System.in);

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

double newGrade = scanner.nextDouble();

foundStudent.setGrade(newGrade);

System.out.println("Student Grade updated successfully!");

}

}

**Object-Oriented Concepts Code:**

*Student.java*

package Object\_Oriented\_Concepts;

public class Student {

private int studentId;

private String name;

private double grade;

public Student(int studentId, String name, double grade) {

this.studentId = studentId;

this.name = name;

this.grade = grade;

}

// Getters and Setters

public int getStudentId() {

return studentId;

}

public void setName(String name) {

this.name = name;

}

public void setGrade(double grade) {

this.grade = grade;

}

@Override

public String toString() {

return "Student ID: " + studentId + ", Name: " + name + ", Grade: " + grade;

}

}

*GraduateStudent.java*

package Object\_Oriented\_Concepts;

public class GraduateStudent extends Student {

// Additional attributes and methods specific to graduate students

public GraduateStudent(int studentId, String name, double grade) {

super(studentId, name, grade);

// Additional initialization for graduate students

}

public String getStudentType() {

return "Graduate";

}

}

*UndergraduateStudent.java*

package Object\_Oriented\_Concepts;

public class UndergraduateStudent extends Student {

// Additional attributes and methods specific to undergraduate students

public UndergraduateStudent(int studentId, String name, double grade) {

super(studentId, name, grade);

// Additional initialization for undergraduate students

}

public String getStudentType() {

return "Undergraduate";

}

}

**File Handling Code:**

*FileHandler.java*

package File\_Handling;

import Object\_Oriented\_Concepts.Student;

import java.io.\*;

public class FileHandler {

private static final String FILE\_PATH = "students.txt";

public static void writeToFile(Student student, String studentType) {

try (PrintWriter writer = new PrintWriter(new FileWriter(FILE\_PATH, true))) {

writer.println(student.toString() + ", " + "Student Type: " + studentType);

} catch (IOException e) {

e.printStackTrace();

}

}

public static void readFromFile() {

try (BufferedReader reader = new BufferedReader(new FileReader(FILE\_PATH))) {

String line;

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

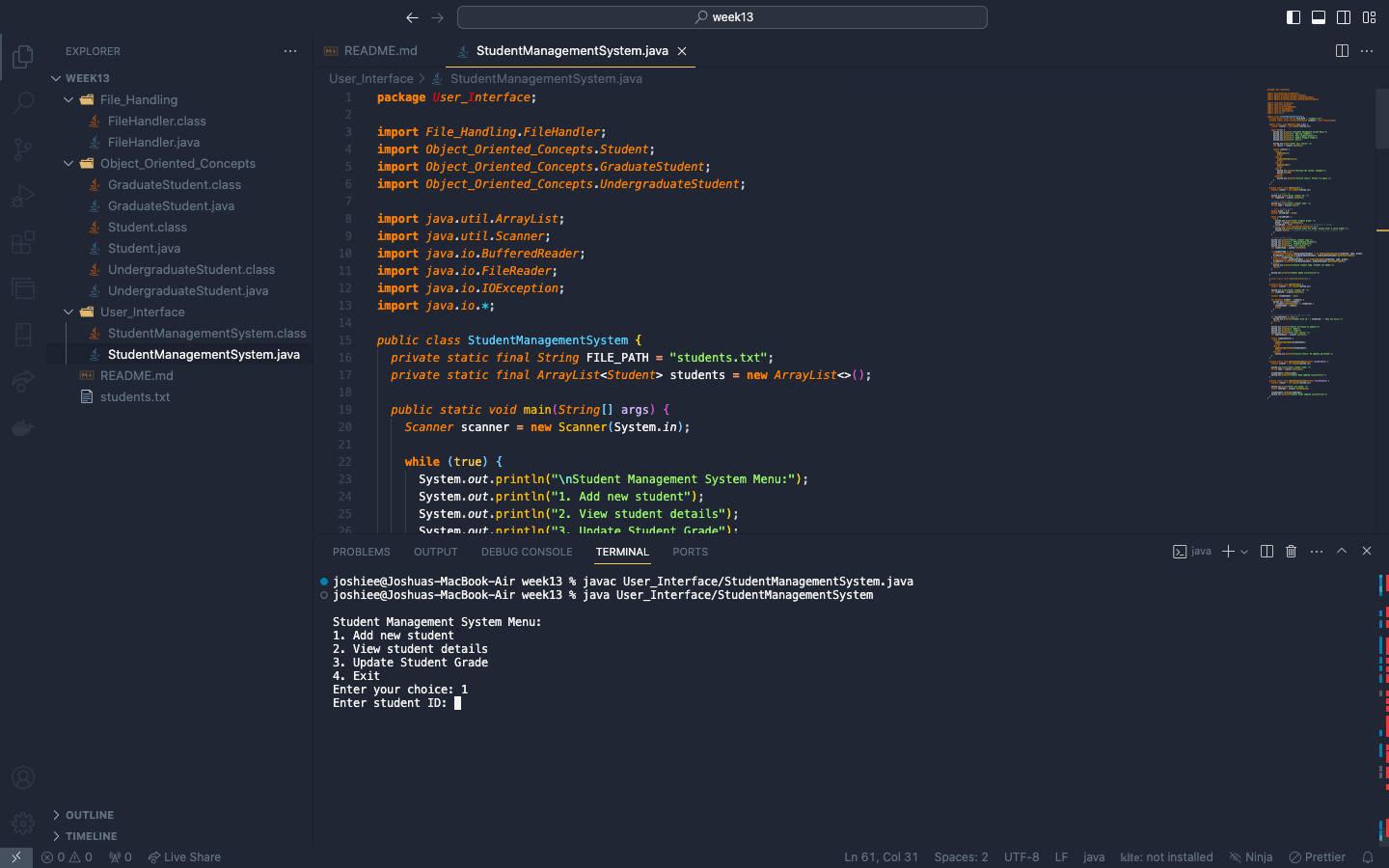
e.printStackTrace();

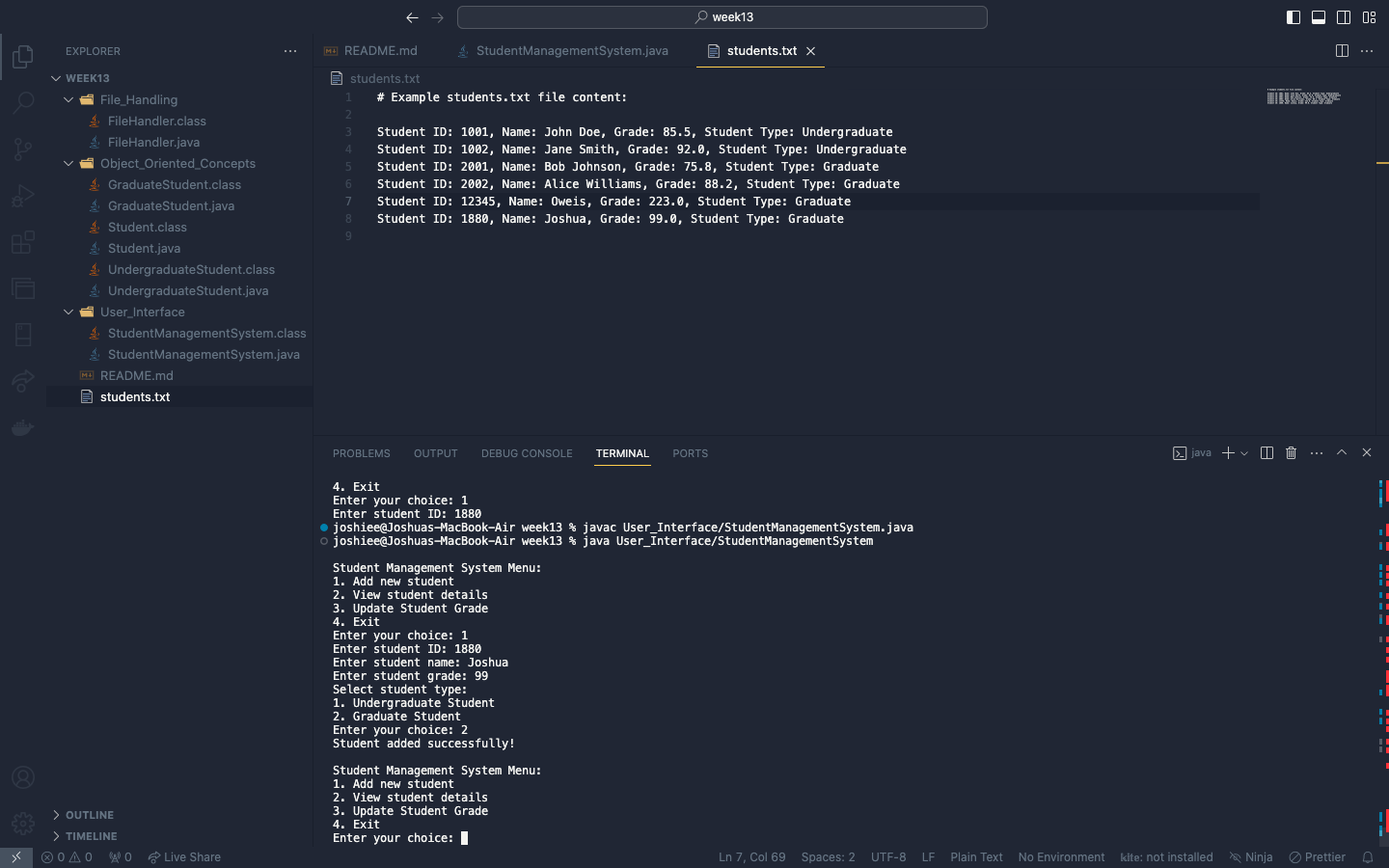
}

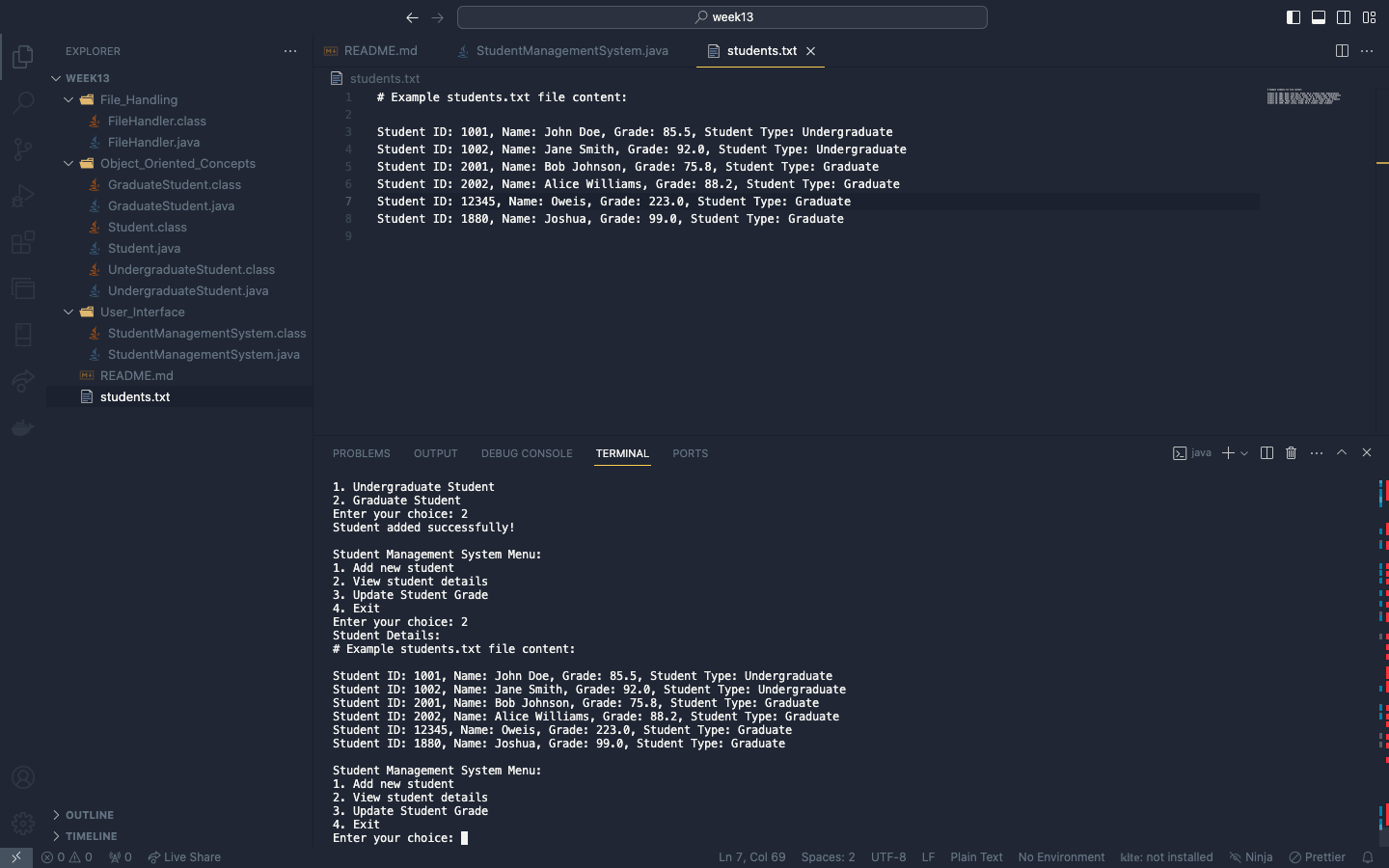
}

}

Screenshots

*Initial display of the application start up displaying the option t choose from*

*screenshot showing the result after adding new student*

*screenshot showing all the list of student data*