**Presented By:**

**Mai Moheb Mahmoud Hussein**

**2023/08305**

**Hamza Bassem Adly Said Hussein**

**2023/02032**

**Seif Ahmed Hussien Badawy**

**2023/08433**

**Amira Ahmed Mohamed Tawfik Khalifa**

**2023/04705**

**Documentation**

* **Overview of system’s design and architecture:**
* This system is designed to manage student grade records across multiple subjects. It allows users to add students, input grades, calculate averages, display individual student data, and delete student records. The architecture is built around a single student structure and a menu-driven interface that guides user interaction.
* **Core Components and Functionality**

Student Structure

string name: Stores the student's name

int grades[NUM\_SUBJECTS]: Stores grades for each subject

* **Functions used and their purposes:**

**Data Entry and Management**

* add\_name(Student&): Prompts user to enter a student's name
* add\_grades(Student&): Prompts user to enter grades for each subject
* display\_grades(const Student&): Displays student name, grades, and average
* calc\_average(const Student&): Calculates and returns the average grade

**File Operations**

* save\_data(const Student[], const string&, int): Saves all student data to a file
* load\_data(Student[], const string&, int): Loads student data from a file

**Menu and Control Flow**

* display\_menu(): Displays the main menu options
* main(): Handles user input and controls program flow
* **Instructions for program running and compiling:**

* **User interaction with the main menu:**

**Upon running the program, the user is presented with a menu containing five options:**

1. **Add Student  
   Prompts for a new student name and add it to the system.**
2. **Add Student Grades  
   Prompts for a student name, then allows grade entry for that student.**
3. **Display Student Grades  
   Prompts for a student name and displays their grades and average.**
4. **Delete Student  
   Prompts for a student name and remove them from the system.**
5. **Exit Application  
   Saves all data to ’”grades.txt “and exits the program.**

* **Code Architecture:**

**A black screen with white text

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A computer code on a black background

AI-generated content may be incorrect.**

**A computer code on a black background

AI-generated content may be incorrect.A computer screen shot of text

AI-generated content may be incorrect.**

**A computer screen with text on it

AI-generated content may be incorrect.**

**A computer screen with white text and green text

AI-generated content may be incorrect.**A computer screen with text and numbers

AI-generated content may be incorrect.

* **Sample input/output to demonstrate system functionality:**

**Main MenuA black screen with a black background

AI-generated content may be incorrect.**

Figure 1: The main menu that prompts to the user when running the program

**Example: Adding a Student**

**A black background with white text

AI-generated content may be incorrect.**

**Example: Adding Grades**

**A black screen with white text

AI-generated content may be incorrect.**

**Example: Displaying Grades**

Figure 2: All shops and movies available are displayed ascendingly according to prices when user enters choice number "1"

**A black background with white text

AI-generated content may be incorrect.**

Figure 4: User asks to return a movie by entering choice number “3” then the system asks user to input the number of the shop he rented the movie from and number of the movie and gives a confirmation message that the movie is successfully returned.

Figure 3: User asks to rent a movie by entering choice number “2” then the system asks user to input the number of the preferred shop and movie and gives a confirmation message that the movie is successfully rented

A black background with white text

AI-generated content may be incorrect.**Example: Deleting a student**