**STUDENT GRADE MANAGEMENT SYSTEM**

**PROJECT REQUIREMENTS AND OBJECTIVES:**

**Functional Requirements:**

* Users can add new students with their name and unique ID.
* Users can assign grades to students for different subjects.
* The system can calculate the average grade for each student.
* Users can view student records including names, IDs, grades, and average.

**Non-functional requirements:**

* The system should be simple and responsive in a console environment.
* Input/output operations must be clearly formatted and user-friendly.
* Data handling should avoid crashing on invalid inputs.

**Objectives:**

* Implement a console-based C# application to manage student grades.
* Demonstrate usage of C# control structures, loops, and methods.
* Allow interaction through a basic menu-driven interface.
* Ensure data is stored and manipulated properly during the session.

**DESIGN OUTLINE**

**Design Overview:**

The system is a console application with a menu interface. The core data structure will be a dictionary or list of Student objects. Each student has a name, ID, and a list of grades.

**Tasks and Code Elements:**

|  |  |
| --- | --- |
| **Task** | **Code Elements** |
| Add new student | Input handling, object creation, list or dictionary |
| Assign grades | Input, list manipulation, validation |
| Calculate average | Looping through grades, arithmetic, method call |
| Display student records | Loop, string formatting |
| Menu system | *switch* or *if-else*, loop for continuous interaction |

**Planned Methods:**

* AddStudent(string name)
* AddGrade(string id, int grade)
* CalculateAverage(string id): double
* DisplayStudentInfo()
* ShowMenu()

**Planned Control Structures:**

* *if-else* for input validation and flow control
* *switch* for menu selection
* *while* or *do-while* loop for persistent menu interaction
* *for* loop or foreach for iterating over grades

**Design Diagram:**

A diagram of a program

AI-generated content may be incorrect.