Student Grade Management System

1. Project Requirements and Objectives

1.1 Functional Requirements

1. The system must allow adding new students with their names and unique IDs.
2. The system must allow assigning grades to students for different subjects.
3. The system must calculate the average grade for each student.
4. The system must display student records with their grades.

1.2 Non-Functional Requirements

1. The system should provide clear and user-friendly console prompts.
2. The system should validate inputs (e.g., IDs must be unique, grades within 0-100).
3. The system should be structured with methods to ensure modularity and avoid repetition.
4. The program should use loops and control structures to handle multiple records and user choices efficiently.
5. The system should be maintainable and easy to extend in the future.

1.3 Project Objectives

- To develop a simple console-based program for managing student grades.
- To allow adding students, assigning subject grades, calculating averages, and displaying student records.
- To demonstrate programming skills through the use of control structures, loops, and methods.

2. Design Outline

2.1 Main Components

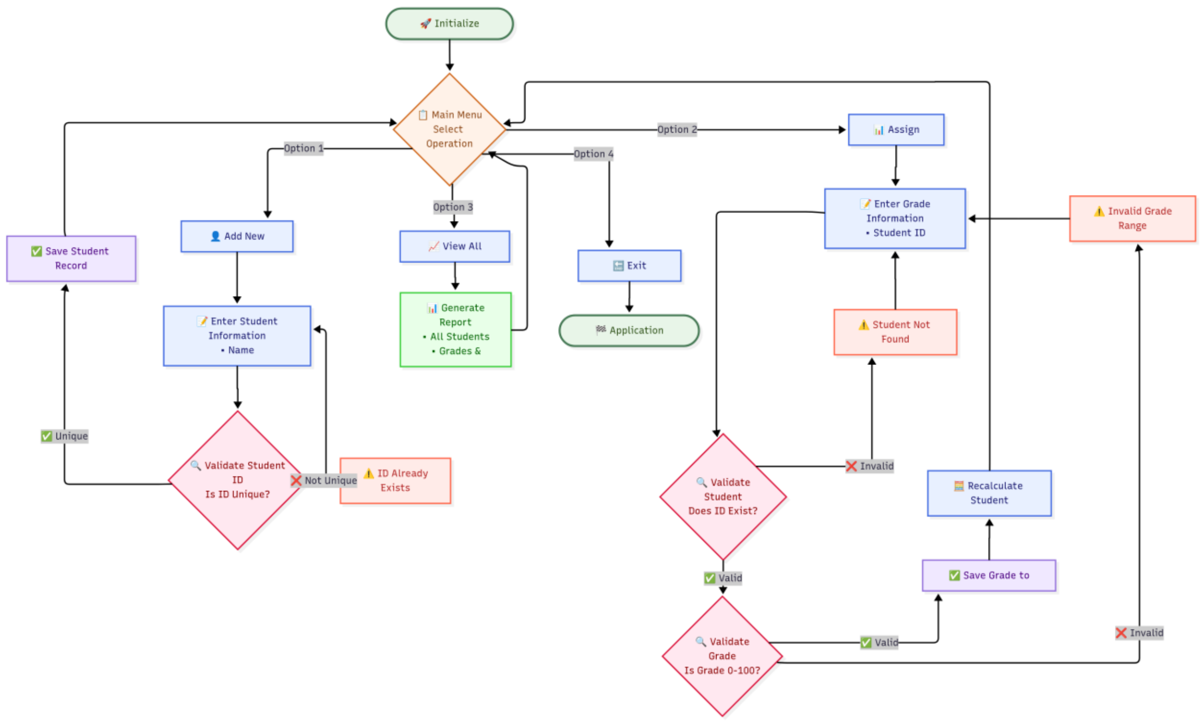
- **Student Data Storage:** A dictionary structure to store student names, IDs, and grades.
- Core Methods (directly based on the key features):
    - `AddStudent()` → Register a new student with name and ID.
    - `AssignGrade()` → Add grades for different subjects for a student.
    - `CalculateAverage()` → Compute the average grade of a student.
    - `DisplayStudents()` → Show student records with their grades and averages.

2.2 Control Structures

- **If-Else:** Input validation (e.g., grade range, unique IDs).
- **Switch / If-Else:** Main menu navigation (Add Student, Assign Grade, Display Records, Exit).

2.3 Loops

- **For / Foreach:** Iterate through students and their subjects/grades.
- **While:** Keep the menu running until the user chooses Exit.


🚀 Initialize
📋 Main Menu Select Operation
Option 1
Option 2
Option 3
Option 4
👤 Add New
📈 View All
🔚 Exit
🏁 Application
📊 Assign
📝 Enter Grade Information • Student ID
⚠️ Invalid Grade Range
✅ Save Student Record
📝 Enter Student Information • Name
📊 Generate Report • All Students • Grades &
⚠️ Student Not Found
✅ Unique
🔍 Validate Student ID Is ID Unique?
❌ Not Unique
⚠️ ID Already Exists
🔍 Validate Student Does ID Exist?
❌ Invalid
🧮 Recalculate Student
✅ Save Grade to
✅ Valid
🔍 Validate Grade Is Grade 0-100?
✅ Valid
❌ Invalid