Computerized Grading System
for
Nuestra Señora De Aranzazu Parochial School

Barbo, Michael Yuri D.
Bautista, Mark Joneil S.
Bautista, Troy Jerron A.
Geronimo, Joncel Zairho G

Technological Institute of the Philippines
Quezon City

November 2025

Table of Contents

Introduction

Nuestra Senora De Aranzazu is a private school located in San Mateo, Rizal. The school was founded in 1968, and started as a pre-school and operated in a makeshift building. The school offers a complete pre-school (nursery and kindergarten), and basic education that consists of elementary (Grades 1-6), and junior and senior high school (Grade 7-12).

Today, the school uses computer based-tools like the E-class record, but the recording of grades is still manually done. Some teachers still record the grades of quizzes and exams manually before inputting into a computer, but each teacher has different ways of recording the grades.

As mentioned before, some teachers already computed their grades through the E-class record, that inputs one's grades, and it will show the grades of a student, however the recording of each component of the grading system (eg, quizzes, assignments, exams) are still recorded manually, and put them into report cards

The current system of recording grades is traditional, but very inefficient for a number of reasons. It can be tedious and time-consuming, could be prone to errors, be very inconvenient, and among other related problems that could affect the accuracy and efficiency of the current grade management.

To address this issue, we propose a C++ based grading system computer program that not only automates the computations of grades, but also records and stores them too. This program calculates the final grades by inputting scores of each component, displaying the results, and keeping them in files for the teacher to access and retrieve previous grades efficiently.

To summarize, Nuestra Senora De Aranzazu still records the grade of each student manually, which are prone to various problems. To solve this, a proposed C++ grading system program addresses these issues by automated computation and recording of grades. This program will not only help in reducing time consumption and inconvenience, it will also help the students as it provides an easy way to record and retrieve grades with very minimal issues.

The Project

This Grading System allows authorized users (teachers and admin) to:

- Log in using a username and password.
- Add, edit, and remove student records.
- Enter grades for multiple assessment categories (e.g., quizzes, assignments, projects, exams).

- Specify weights for each category and compute weighted averages automatically.
- Assign remarks (Passed/Failed) based on grading policy (e.g., passing mark = 75).
- Save and retrieve student records from simple text files (users.txt, students.txt, grades.txt).
- Generate a per-student grade report (displayed on console and saved as a text excerpt if desired).

User roles:

Admin: Manage teacher accounts, view all records.

Teacher: Manage students in their class, input grades, generate reports.

Workflow (high-level)

The user starts the program.

Login or Register (admin account pre-provisioned).

After login, the teacher chooses from the menu: Add Student, Input Grades, Compute Grades, View Report, Logout.

When grades are input, the system computes weighted average and saves the record.

Teachers can export or print the displayed report.

Hardware and Software Requirements

Hardware requirements:

- Processor: Intel Core i3
- Monitor: Any display that supports text-based console output
- RAM: At least 2 GB of memory.
- Storage: 500 MB free disk space

Software requirements:

- compiler: Visual studio
- Operating System: Windows 10 (64-bit) or Windows 11.
- Programming language: C++

Statement of the Problem

Nuestra Senora De Aranzazu Parochial School still uses a manual method of computing and recording student grades. Teachers calculate grades using calculators or spreadsheets, then write them on paper forms or report cards. This process is time-consuming and often leads to errors in computation, misplaced records, and difficulty in tracking student performance over time.

Because of the manual process, retrieving previous grades or generating reports for multiple students also becomes inconvenient. Teachers spend more time managing grades instead of focusing on teaching their students. This shows the need for a computerized grading system that can make the process faster, more accurate, and easier to manage.

Objectives

To develop a simple and reliable C++ based Grading System that automates grade computations, secures records on specific subjects and makes report generation faster and accurate for Nuestra Senora De Aranzazu Parochial School

1. C++ coding
2. Automation of Grades
3. Login systems for teachers and admins

Flowchart of the System

This section presents the flowchart of Nuestra Señora De Aranzazu Parochial School, which graphically illustrates the logical structure, operational sequence, and decision-making processes of the proposed system. The flowchart serves as a comprehensive visual representation that demonstrates how various system components and user interactions are integrated to ensure the efficient management of grading operations. It also provides a clear understanding of the system's overall workflow, emphasizing the logical progression from user authentication to grade computation and report generation.

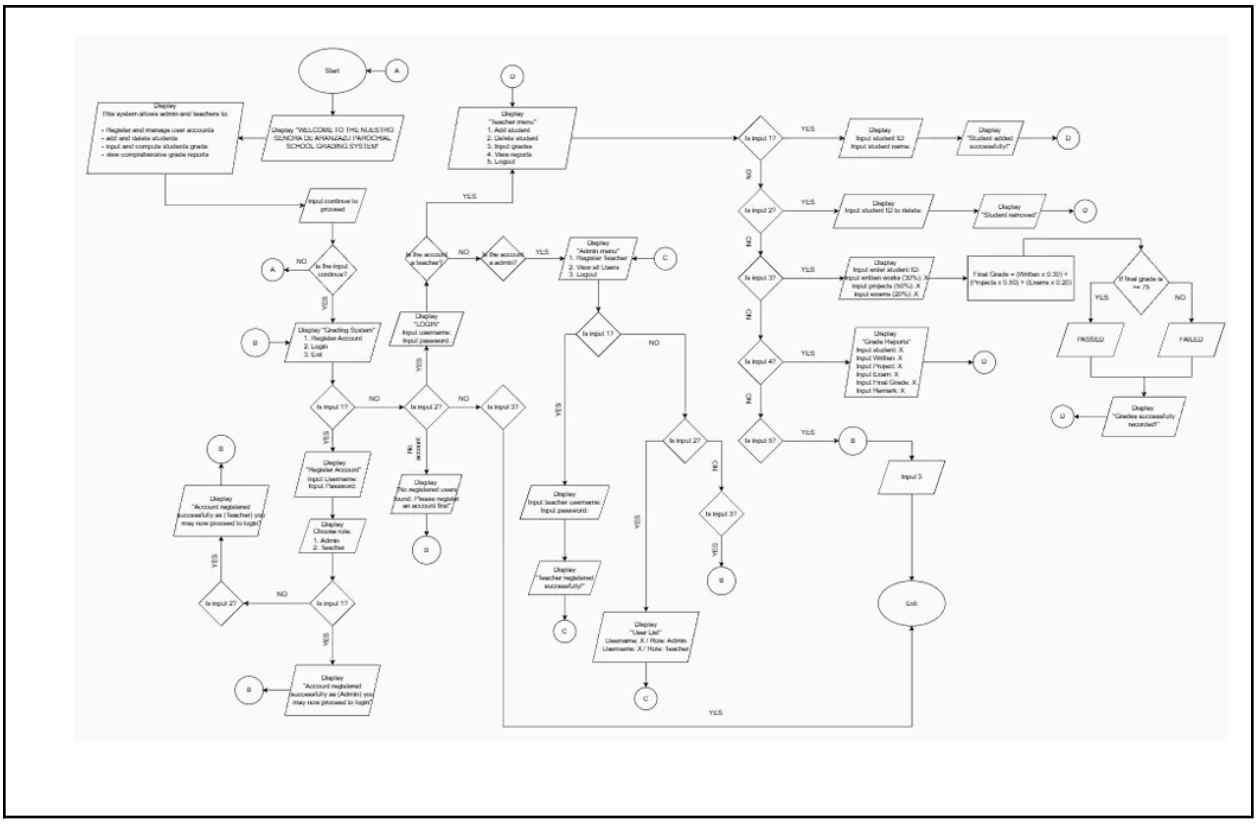

Figure 1: Nuestra Señora De Aranzazu Parochial School Flowchart

The flowchart begins with a welcome display and user authentication process. Depending on the user's role, either as an administrator or teacher, the system grants access to specific functionalities such as account registration, student record management, grade encoding, and report viewing.

- Administrators can register new users, manage accounts, and oversee grading operations.

- Teachers can input student grades, compute final grades based on given weight percentages, and view or print student reports. The process concludes with the computation of the final grade and a decision node determining whether the student Passed or Failed, followed by appropriate output messages.

Pseudocode

The pseudo code outlines the logical flow and major functionalities of the system, including user registration, login validation, student management, grade computation, and report viewing. It serves as the foundation for the program's structure before translating it into actual programming code. The goal is to provide a clear and structured representation of how the system operates and interacts with users, administrators, and teachers.

```
START
DISPLAY "WELCOME TO THE NUESTRO SENORA DE ARANZAZU PAROCHIAL SCHOOL GRADING SYSTEM"
DISPLAY  This system allows admin and teachers to:
DISPLAY    Register and manage user accounts
DISPLAY    add and delete students
DISPLAY    input and compute students grade
DISPLAY    view comprehensive grade reports

INPUT proceed
IF proceed = continue THEN
        GO TO Grading system
ELSE
        GO TO Start
 END IF

DISPLAY “Grading System”
DISPLAY “1. Register Account”
DISPLAY “2. Login“
DISPLAY “3. Exit“
INPUT Choice
IF Choice = 1 THEN
DISPLAY “Register Account”
```

```
DISPLAY “Input a username:”
INPUT Username:
DISPLAY “Input a password:”
INPUT Password:
DISPLAY “Choose role: “
DISPLAY 1.  Admin
DISPLAY 2. Teacher
INPUT role

IF role = 1 THEN
        DISPLAY “Account registered successfully as (Admin) you may now proceed to login”
        GO TO Grading System
ELSE IF role = 2 THEN
        DISPLAY “Account registered successfully as (Teacher) you may now proceed to login”
        GO TO Grading System
ELSE
        DISPLAY “Invalid role”
        GO TO Grading System
END IF

ELSE IF Choice = 2 THEN
        DISPLAY “Login”
        DISPLAY “Input the username:”
        INPUT Username:
        DISPLAY “Input the password:”
        INPUT Password:
```

```
IF account_exists THEN
        DISPLAY “Login successful!”
        IF role = “Admin” THEN
                GO TO Admin Menu
        ELSE IF role = “Teacher” THEN
                GO TO Teacher Menu
        END IF
ELSE
        DISPLAY “No registered users found. Please register an account first”
        GO TO “Grading System”
END IF

ELSE IF Choice = 3 THEN
        GO TO End
END IF

Admin Menu
DISPLAY “Admin Menu”
DISPLAY “1. Register Teacher”
DISPLAY “2. View all Users”
DISPLAY “3. Logout”
INPUT Choice

IF Choice = 1 THEN
        DISPLAY “Register Teacher”
        DISPLAY “Input teacher username:“
        INPUT teacherusername
        DISPLAY “Input teacher password:”
        INPUT teacherpassword
        DISPLAY “Teacher registered successfully!”
```

```
        GO TO Admin Menu
ELSE IF Choice = 2 THEN
        DISPLAY “User List”
        DISPLAY “Username: X / Role: Admin
        DISPLAY “Password: X / Role: Teacher
        GO TO Admin Menu

ELSE IF Choice = 3 THEN
        DISPLAY “Logging out”
        GO TO Grading System
ELSE
        DISPLAY “Invalid input. Try again.”
        GO TO Admin Menu
END IF

Teacher Menu
DISPLAY “Teacher Menu”
DISPLAY “1. Add Student”
DISPLAY “2. Delete Student”
DISPLAY “3. Input Grades”
DISPLAY “4. View Reports”
DISPLAY “5. Logout”
INPUT Choice

IF Choice = 1 THEN
        DISPLAY “Input Student ID: “
        INPUT StudentID
        DISPLAY “Input Student Name: “
        INPUT StudentName
        DISPLAY “Student added successfully!”
```

```
        GO TO Teacher Menu
ELSE IF Choice = 2 THEN
        DISPLAY “Input Student ID to delete: “
        INPUT StudentID
        DISPLAY “Student removed”
        GO TO Teacher Menu
ELSE IF Choice = 3 THEN
        DISPLAY “Input Student ID:”
        INPUT StudentID
        DISPLAY “Input Written works (30%): “
        INPUT Written
        DISPLAY “Input Projects (50%): “
        INPUTProjects
        DISPLAY “Input Exam (20%): “
        INPUT Exams
        Final Grade = (Written x 0.30) + (Projects x 0.50) + (Exam x 0.20)
                IF FinalGrade >= 75 THEN
                        DISPLAY “PASSED”
                ELSE
                        DISPLAY “FAILED”

END IF
        DISPLAY “Grades successfully recorded!”
        GO TO Teacher Menu

ELSE IF Choice = 4 THEN
        DISPLAY “Input Written: X”
        DISPLAY “Input Projects: X”
        DISPLAY “Input Exam: X”
        DISPLAY “Input Final Grade: X”
```

```
        DISPLAY “Remarks: X”
GO TO Teacher Menu

ELSE IF Choice = 5 THEN
        DISPLAY “Logging out…”
        GO TO Grading System
        DISPLAY “Grading System”
        DISPLAY “1. Register Account”
        DISPLAY “2. Login“
        DISPLAY “3. Exit“
        INPUT Choice

IF Choice = 3 THEN
        GO TO End
END IF

END PROGRAM
END
```

Figure 2: Pseudo code of Nuestra Señora De Aranzazu Parochial School Grading System

The pseudo code demonstrates the sequential process of how the grading system functions. It begins by displaying a welcome message and describing the system’s purpose. Users can either register an account, log in, or exit. Upon successful login, access rights are determined by the user’s role:

Admin users can register teachers, view all registered users, and manage teacher accounts. Teacher users can add or delete student records, input grades, compute final grades, and view student reports. The system computes the final grade using the weighted formula: Final Grade = (Written × 0.30) + (Project × 0.50) + (Exam × 0.20) It then determines whether the student has passed or failed based on a minimum grade requirement of 75. The pseudo code ends with options to log out or exit the system.

Data Dictionary

The Data Dictionary provides detailed information about all the data elements used in the Nuestro Señora de Aranzazu Parochial School Grading System. It defines the names, sizes, data types, and descriptions of variables, structures, and files used in the program. The following table describes the data used in the system, including user accounts, student information, and grade records. This information ensures consistent understanding of how data is stored and processed throughout the application.

Table 1: Data Dictionary

Data Name	Size	Data Type	Description
1. Username	Up to 20 characters allowed	String	Stores the user's login name
2. Password	Up to 20 characters allowed	String	Stores the user's password
3. Role	Up to 10 characters	String	Admin or Teacher
4. StudentID	Up to 10 characters	String	Unique ID for each student
5. StudentName	Up to 50 characters	String	Full name of the student
6. Remark	Up to 10 characters	String	"Passed" or "Failed" status

Code

This program presents the complete C++ implementation of the "Nuestro Señora de Aranzazu Parochial School Grading System." The code demonstrates how the system performs essential functions such as user registration, login authentication, student record management, grade computation, and report generation. It is written using standard C++ libraries to ensure compatibility and simplicity.

```
#include <iostream>
#include <fstream>
#include <string>
#include <chrono>
#include <thread>
using namespace std;

// STRUCT DEFINITIONS
struct User {
    string username;
    string password;
    string role;
};

struct Student {
    string id;
    string name;
};

struct Grade {
    string id;
    double written;
    double project;
```

```
    double exam;
    double finalGrade;
    string remark;
};

int main() {

    // WELCOME SCREEN
    cout <<
"\n#########################################################################
\n";
    cout << "  WELCOME TO NUESTRO SENORA DE ARANZAZU PAROCHIAL
SCHOOL GRADING SYSTEM   \n";
    cout <<
"#########################################################################
\n";
    cout << "\nThis system allows Admins and Teachers to:\n";
    cout << " - Register and manage user accounts\n";
    cout << " - Input and compute student grades\n";
    cout << " - View comprehensive grade reports\n";
    cout << " - Add or remove students\n";

    string input;
    do {
        cout << "\nType 'continue' to proceed: ";
        cin >> input;
        if (input != "continue" && input != "CONTINUE") {
            cout << "Invalid input. Please type 'continue'.\n";
        }
    } while (input != "continue" && input != "CONTINUE");
```

```
cout << "\nLoading";
for (int i = 0; i < 3; i++) {
    cout << ".";
    cout.flush();
    this_thread::sleep_for(chrono::milliseconds(500));
}
cout << "\n";

//MAIN MENU
while (true) {
    int mainChoice;
    cout << "\n############## \n";
    cout << "  MAIN MENU   \n";
    cout << "############## \n";
    cout << "1. Register Account\n";
    cout << "2. Login\n";
    cout << "3. Exit\n";
    cout << "Choose: ";
    cin >> mainChoice;

    //REGISTER ACCOUNT
    if (mainChoice == 1) {
        User user;
        int r;

        cout << "\n####################\n";
        cout << "  REGISTER ACCOUNT  \n";
        cout << "####################\n";
        cout << "Enter Username: ";
```

```
        cin >> user.username;
        cout << "Enter Password: ";
        cin >> user.password;

        cout << "\n##################\n";
        cout << "   CHOOSE A ROLE   \n";
        cout << "################## \n";
        cout << "1. Admin\n2. Teacher\nChoose role: ";
        cin >> r;

        if (r == 1) user.role = "admin";
        else user.role = "teacher";

        ofstream ufile("users.txt", ios::app);
        ufile << user.username << " " << user.password << " " << user.role << "\n";
        ufile.close();

        cout << "\nAccount created successfully as (" << user.role << ")!\nYou may now
LOGIN.\n";
    }

    //LOGIN
    else if (mainChoice == 2) {
        string inputUser, inputPass;
        User user;
        bool found = false;

        cout << "\n##################\n";
        cout << "  LOGIN ACCOUNT   \n";
        cout << "################## \n";
```

```
cout << "Username: ";
cin >> inputUser;
cout << "Password: ";
cin >> inputPass;

ifstream file("users.txt");
while (file >> user.username >> user.password >> user.role) {
    if (inputUser == user.username && inputPass == user.password) {
        found = true;
        break;
    }
}
file.close();

if (!found) {
    cout << "\nAccount not found. Please register first.\n";
    continue;
}

cout << "\nLogin successful (" << user.role << ")!\n";

// ADMIN MENU
if (user.role == "admin") {
    int a;
    while (true) {
        cout << "\n###############\n";
        cout << "   ADMIN MENU   \n";
        cout << "############### \n";
        cout << "1. Register Teacher\n";
        cout << "2. View All Users\n";
```

```
            cout << "3. Logout\n";
            cout << "Choose: ";
            cin >> a;

            if (a == 1) {
                User teacher;
                cout << "\nTeacher Username: ";
                cin >> teacher.username;
                cout << "Password: ";
                cin >> teacher.password;
                teacher.role = "teacher";

                ofstream ufile("users.txt", ios::app);
                ufile << teacher.username << " " << teacher.password << " " << teacher.role <<
"\n";
                ufile.close();

                cout << "\nTeacher registered!\n";
            }

            else if (a == 2) {
                ifstream uf("users.txt");
                User u;
                cout << "\n===== USER LIST =====\n";
                while (uf >> u.username >> u.password >> u.role) {
                    cout << "Username: " << u.username << " | Role: " << u.role << endl;
                }
                uf.close();
            }
```

```
        else break;
    }
}

//TEACHER MENU
else if (user.role == "teacher") {
    int t;
    while (true) {
        cout << "\n################\n";
        cout << "  TEACHER MENU   \n";
        cout << "################ \n";
        cout << "1. Add Student\n";
        cout << "2. Delete Student\n";
        cout << "3. Input Grades\n";
        cout << "4. View Reports\n";
        cout << "5. Logout\n";
        cout << "Choose: ";
        cin >> t;

        //ADD STUDENT
        if (t == 1) {
            Student s;
            cout << "\nEnter Student ID: ";
            cin >> s.id;
            cout << "Enter Name: ";
            cin.ignore();
            getline(cin, s.name);

            ofstream sf("students.txt", ios::app);
            sf << s.id << " | " << s.name << "\n";
```

```
    sf.close();
    cout << "Student added.\n";
}

//DELETE STUDENT
else if (t == 2) {
    string id;
    cout << "\nEnter ID to delete: ";
    cin >> id;

    ifstream sf("students.txt");
    ofstream temp("temp.txt");

    string line;
    while (getline(sf, line)) {
        if (line.substr(0, id.size()) != id)
            temp << line << "\n";
    }
    sf.close();
    temp.close();

    remove("students.txt");

    rename("temp.txt", "students.txt");

    cout << "Student deleted (if found).\n";
}

//INPUT GRADES
else if (t == 3) {
```

```
    Grade g;
    cout << "\nEnter Student ID: ";
    cin >> g.id;

    cout << "Written (30%): ";
    cin >> g.written;
    cout << "Project (50%): ";
    cin >> g.project;
    cout << "Exam (20%): ";
    cin >> g.exam;

    g.finalGrade = g.written * 0.30 + g.project * 0.50 + g.exam * 0.20;
    g.remark = (g.finalGrade >= 75 ? "PASSED" : "FAILED");

    ofstream gf("grades.txt", ios::app);
    gf << g.id << " " << g.written << " " << g.project << " "
       << g.exam << " " << g.finalGrade << " " << g.remark << "\n";
    gf.close();
    cout << "Grades saved.\n";
}

//VIEW REPORTS
else if (t == 4) {
    cout << "\n###########\n";
    cout << "  REPORTS   \n";
    cout << "########### \n";
    Grade g;
    ifstream gf("grades.txt");

    while (gf >> g.id >> g.written >> g.project >> g.exam >> g.finalGrade >>
```

```
g.remark) {
                    cout << "\nStudent ID: " << g.id << endl;
                    cout << "Written: " << g.written << endl;
                    cout << "Project: " << g.project << endl;
                    cout << "Exam: " << g.exam << endl;
                    cout << "Final Grade: " << g.finalGrade << endl;
                    cout << "Remark: " << g.remark << endl;
                }
                gf.close();
            }

            else break;
        }
    }

    //EXIT PROGRAM
    else if (mainChoice == 3) {
        cout << "Exiting program...\n";
        break;
    }

    else {
        cout << "Invalid choice.\n";
    }
  }

  return 0;
}
```

Figure 3. Nuestra Señora De Aranzazu Parochial School Grading System

Code discussion

The program begins with the declaration of structs for User, Student, and Grade, which serve as the primary data structures for user accounts, student records, and grading information. The main function introduces a welcome screen followed by a main menu offering options to register, log in, or exit.

- Admin users can register teachers and view all user accounts.
- Teacher users have access to student management (add/delete), grade input, and report viewing.
- Grades are calculated based on weighted components: Written (30%), Project (50%), and Exam (20%).
- The program uses file handling (users.txt, students.txt, grades.txt) to store data persistently between sessions.

Results and Discussion

This section represents the results generated by the Nuestro Señora de Aranzazu Parochial School Grading System and discusses how each function contributes to managing and processing student information efficiently.

Program Output:

When the program runs, it first displays a Welcome Screen showing the name of the school and an introduction to the system's features. The user can then:

- Register and manage accounts (Admin and Teacher roles)
- Add or delete students
- Input and compute grades
- View grade reports with remarks (PASSED/FAILED)

Conclusion

The developed C++ program successfully automates the grading process for Nuestro Señora de Aranzazu Parochial School. It simplifies tasks such as account management, student data handling, and grade computation. The system provides accurate results based on weighted grade components, ensures data storage, and offers role-based functionality for both administrators and teachers.

Recommendations:

- Implement input validation and error handling to prevent invalid data entries.
- Integrate a graphical user interface (GUI) for better usability.
- Add data encryption for secure account management.
- Enable automatic report generation (PDF or Excel format) for easier record-keeping.

References:

GeeksforGeeks. (2024, March 12). *File C++*.https://www.geeksforgeeks.org/file-handling-cpp/

JDoodle. (n.d.). *Online compiler and IDE for C, C++, Java, and more.* JDoodle. Retrieved November 10, 2025, from https://www.jdoodle.com

Microsoft. (2025). *Visual Studio Code (Version 1.95)* [Computer software]. Microsoft. https://code.visualstudio.com

"Software System for Automatic Grading of Paper Tests." (2023). *Electronics*, 12(19), 4080. https://doi.org/10.3390/electronics12194080

TutorialsPoint. (2023). *C++ file* https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm

W3Schools. (2024). *C++ tutorial*. https://www.w3schools.com/cpp/