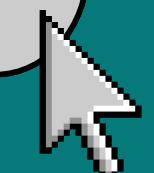


C++ Final Project

PROFESSOR: MA'AM MAY B. FIGUERODA

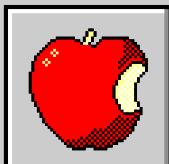
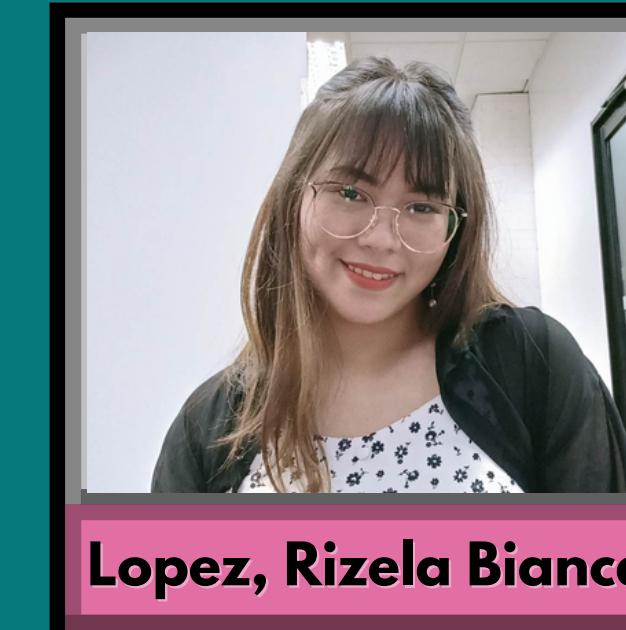
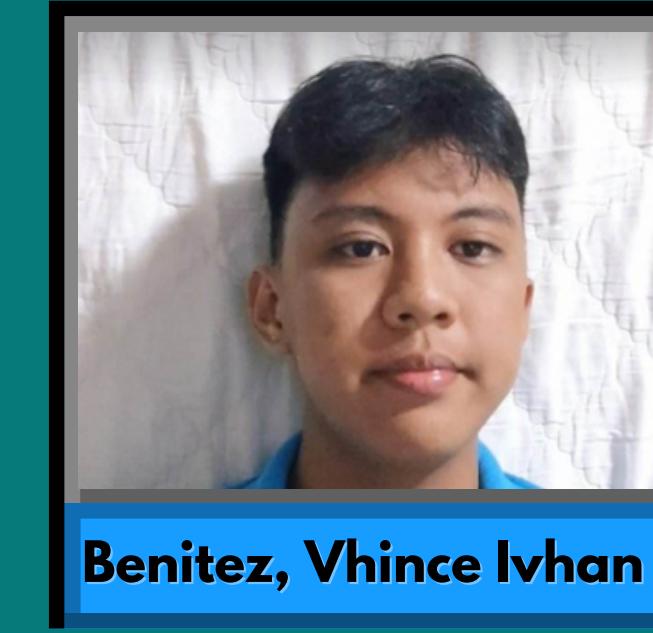


Hello World, Hello Ma'am
Sana Pumasa Ang Lahat



3:00PM

GROUP 3



Ba't si Kuya Melo
lahat gumawa?

Student Grade Tracker System



- =====
- 1. Add student
- 2. Display all students
- 3. Save students to file
- 4. Load students from file
- 5. Remove a student
- 6. Update a student's information
- 7. Sort students by grade
- 8. Exit
- =====

Command:

This system is a console application developed using the C++ programming language.

The purpose of this application is to provide a simple, efficient, and reliable tool for tracking and managing student grades.

Start



Student Grade Tracker System



- =====
- 1. Add student
- 2. Display all students
- 3. Save students to file
- 4. Load students from file
- 5. Remove a student
- 6. Update a student's information
- 7. Sort students by grade
- 8. Exit
- =====

Command:

In the modern educational system, we know that tracking student performance is a crucial task.

However, it can become a challenging one due to the sheer number of students and the array of assignments, tests, and exams that need to be tracked.

Start



Student Grade Tracker System



- =====
- 1. Add student
- 2. Display all students
- 3. Save students to file
- 4. Load students from file
- 5. Remove a student
- 6. Update a student's information
- 7. Sort students by grade
- 8. Exit
- =====

Command:

Our Student Grade Tracker System is designed to address this issue.

It provides an organized and accessible platform where professors, or even students themselves can easily monitor and manage grades.

Start



Student Grade Tracker System



- =====
- 1. Add student
- 2. Display all students
- 3. Save students to file
- 4. Load students from file
- 5. Remove a student
- 6. Update a student's information
- 7. Sort students by grade
- 8. Exit
- =====

Command:

Although our program design and functionalities are simple and limited, our focus was to demonstrate the importance of fundamental concepts in programming, how it works and in what way we can utilize those concepts.

Start



Student Grade Tracker System



- =====
- 1. Add student
- 2. Display all students
- 3. Save students to file
- 4. Load students from file
- 5. Remove a student
- 6. Update a student's information
- 7. Sort students by grade
- 8. Exit
- =====

Command:

It features a user-friendly interface that allows easy navigation and operation of the system.

User can add and remove students, update student information, view all student data, save and load data from a file, sort students by grades, and much more.

Start



Student Grade Tracker System



- =====
- 1. Add student
- 2. Display all students
- 3. Save students to file
- 4. Load students from file
- 5. Remove a student
- 6. Update a student's information
- 7. Sort students by grade
- 8. Exit
- =====

Command:

Our system incorporated some fundamental programming concepts that were taught from previous lectures such as;

- classes
- inheritance
- file handling
- linked lists
- setters and getters

to achieve our desired functionality.

Start



BACKGROUND

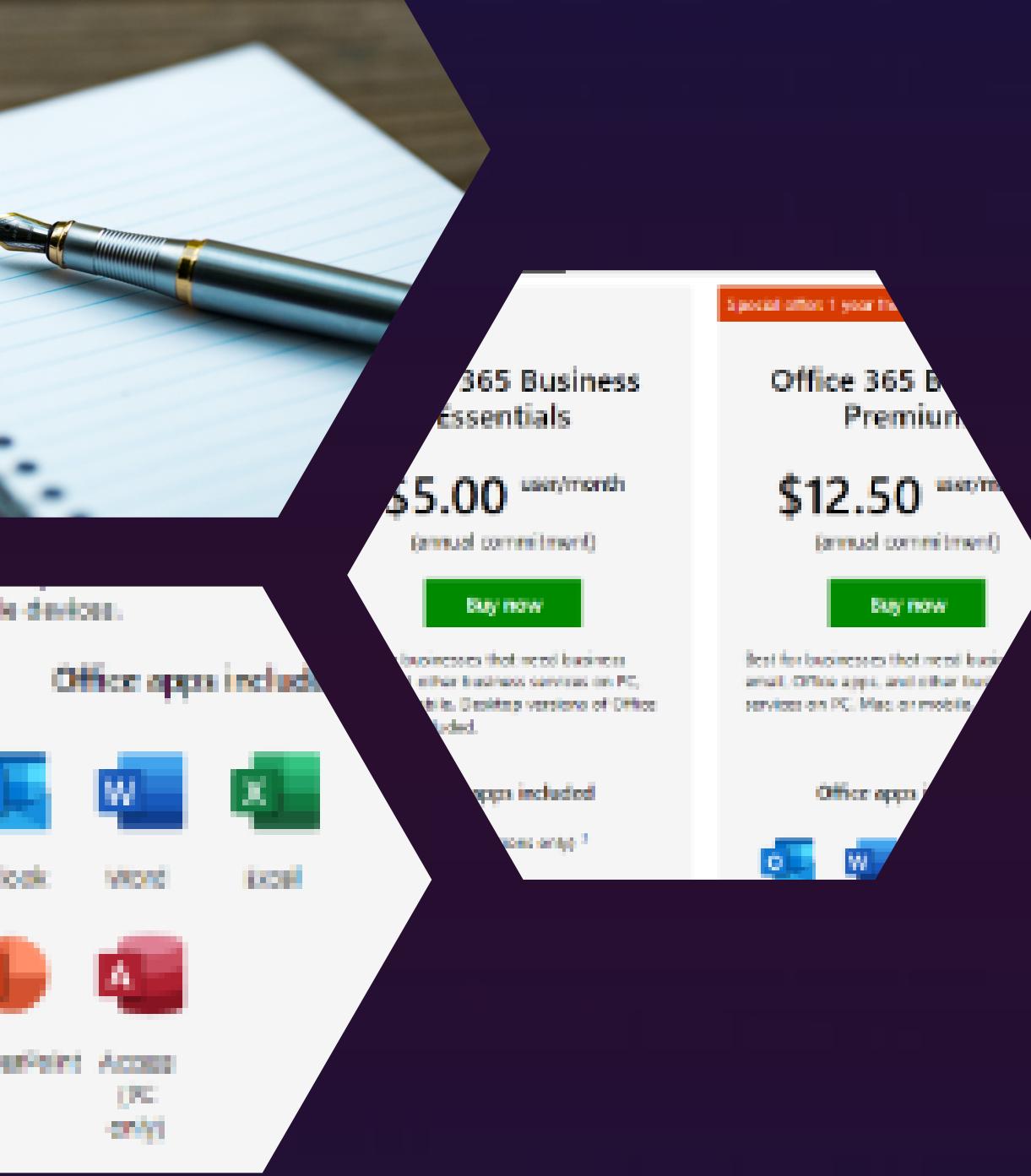


In the rapidly evolving world of education, efficiently managing and tracking student grades is more important than ever.

Traditionally, grades were tracked manually, often written down in grade books or Excel spreadsheets. While these methods served their purpose, they came with their own set of challenges, such as risk of data loss, difficulty in updating records, and time-consuming data retrieval and analysis in the case of written records.

In recent years, we saw that digital solutions have started to replace these manual methods, providing more reliable and efficient means for grade tracking.

BACKGROUND



However, for teachers that struggling to utilize their old laptop, many of these solutions are complex and overladen with features that aren't always necessary, which can make them intimidating to use, difficult to navigate and use effectively.

Additionally, these solutions often come with high costs for licensing and training, making them inaccessible for many schools, particularly those with limited financial resources to pay for subscription (Notably Microsoft Office Plan).



BACKGROUND



Recognizing these challenges, we saw a need for a simple, reliable, and cost-effective grade tracking solution that still offers the flexibility and efficiency of a digital platform.

This is where our Student Grade Tracker System comes in.

Our system is designed to be a user-friendly, straightforward tool that offers the core functionalities needed for efficient grade management, without the unnecessary complications and high costs.

BACKGROUND



The primary objective of our Student Grade Tracker System is to offer a simple, yet effective, digital solution for managing and tracking student grades.

This objective is driven by the need to enhance efficiency and accuracy in the management of student academic performance data, particularly in public schools.

BACKGROUND



One of the key areas of focus for our project is supporting public schools in remote areas. We know that these schools often have limited financial resources, making it difficult for them to afford sophisticated, expensive digital grade tracking systems.

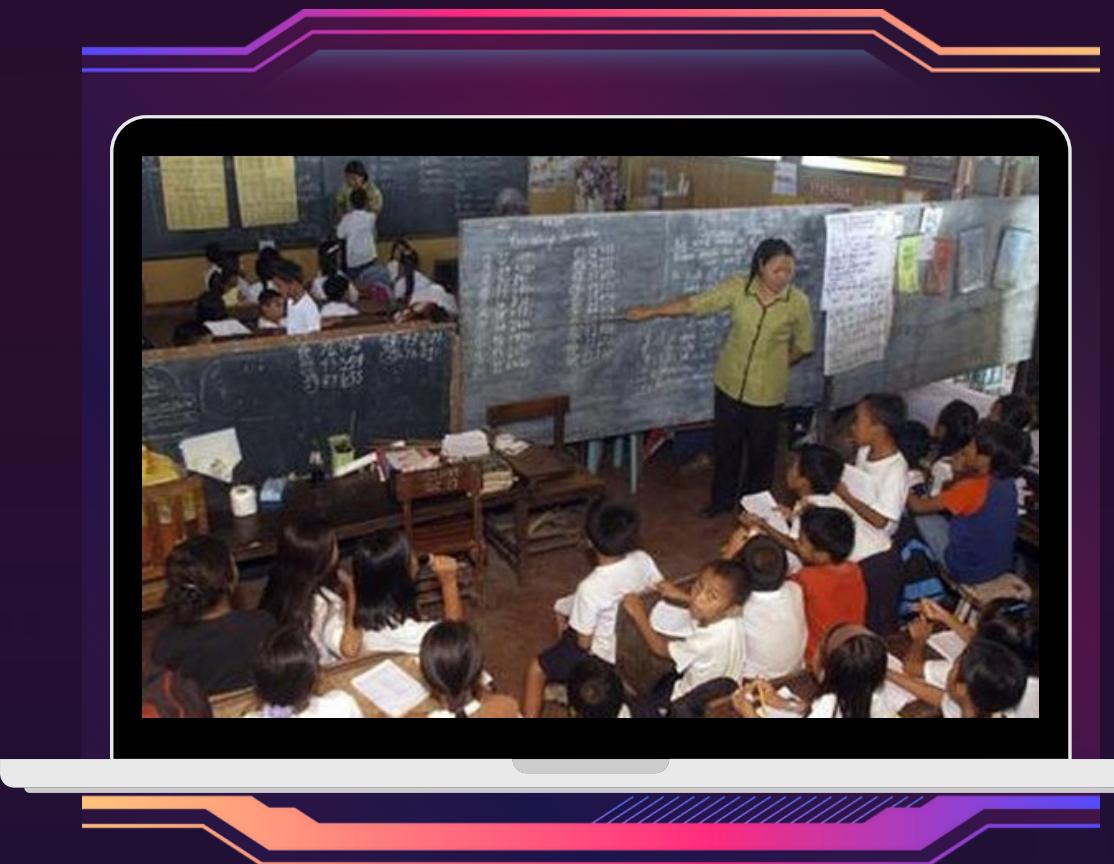
Additionally, the lack of reliable internet connectivity in these areas often makes web-based systems impractical.

BACKGROUND

Our Student Grade Tracker System is designed to address these challenges.

As a console-based application, it can run on virtually any computer without requiring high processing power or internet connectivity.

This makes it a suitable solution for remote schools with limited resources.

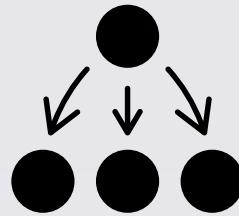


METHODOLOGY

Our Student Grade Tracker System is developed using the C++ programming language, chosen for its versatility and efficiency. It leverages several key programming concepts and methodologies to implement its functionalities.

Classes & Inheritance:

We've designed the system using Object-Oriented Programming (OOP).



A base class, Person, holds common attributes like name and age, while a derived class, Student, adds specific attributes like grades and student ID.

```
// Parent Class
class Person { ... };

// Child Class
class Student : public Person {
private:
    static int nextID;
    int id;
    float grade;

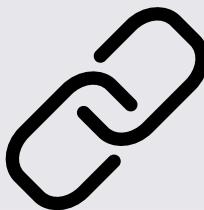
public:
    Student() : Person(), grade(0.0) {
        this->id = nextID++;
        if (nextID == 100) {
            nextID = 1;
        }
    }
}
```

METHODOLOGY

Our Student Grade Tracker System is developed using the C++ programming language, chosen for its versatility and efficiency. It leverages several key programming concepts and methodologies to implement its functionalities.

Linked List

We use a doubly linked list to manage the student data dynamically.



This enables efficient operations like adding, removing, and updating students.

```
List<Student> students;
```

```
cout << "Enter student name: ";
string name;
getline(cin, name);
student.setName(name);

cout << "Enter student age: ";
int age;
cin >> age;
student.setAge(age);

cout << "Enter student grade: ";
float grade;
cin >> grade;
student.setGrade(grade);

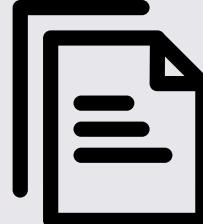
students.push_back(student);
break;
```

METHODOLOGY

Our Student Grade Tracker System is developed using the C++ programming language, chosen for its versatility and efficiency. It leverages several key programming concepts and methodologies to implement its functionalities.

File Handling

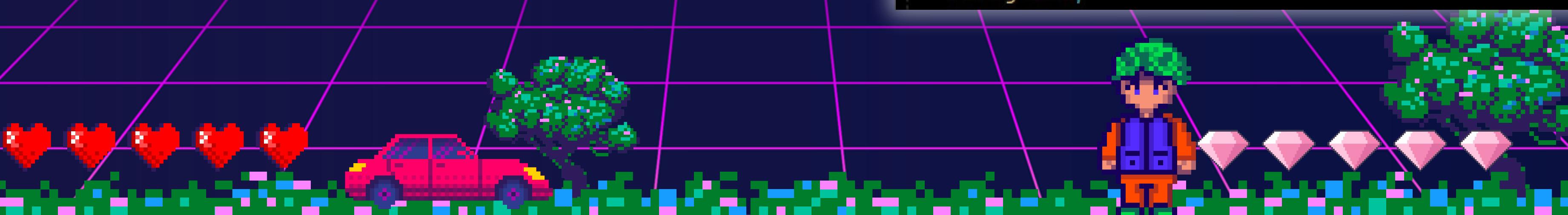
To ensure data persistence, we employ file handling techniques.



It reads from and writes to a text file from file directory, allowing the user to save and retrieve data across different sessions.

```
case 3: {
    ofstream file("C:/Users/user/Documents/C++ Current Dire
    for (const Student& student : students) {
        file << student.getName() << ", ";
        file << student.getAge() << ", ";
        file << student.getGrade() << "\n";
    }
    file.close();
    break;
}

case 4: {
    ifstream file("C:/Users/user/Documents/C++ Current Dire
    string line;
```



METHODOLOGY

Our Student Grade Tracker System is developed using the C++ programming language, chosen for its versatility and efficiency. It leverages several key programming concepts and methodologies to implement its functionalities.

Validation (Setters and Getters)



User inputs are validated to ensure data integrity, and invalid inputs are handled appropriately to enhance user experience and system robustness.

```
float getGrade() const {
    return grade;
}

void setGrade(float grade) {
    if (grade < 60 || grade > 99) {
        cout << "Invalid grade!. Grade should be ";
        return;
    }
    this->grade = grade;
}
```

METHODOLOGY

Our Student Grade Tracker System is developed using the C++ programming language, chosen for its versatility and efficiency. It leverages several key programming concepts and methodologies to implement its functionalities.

User Interface

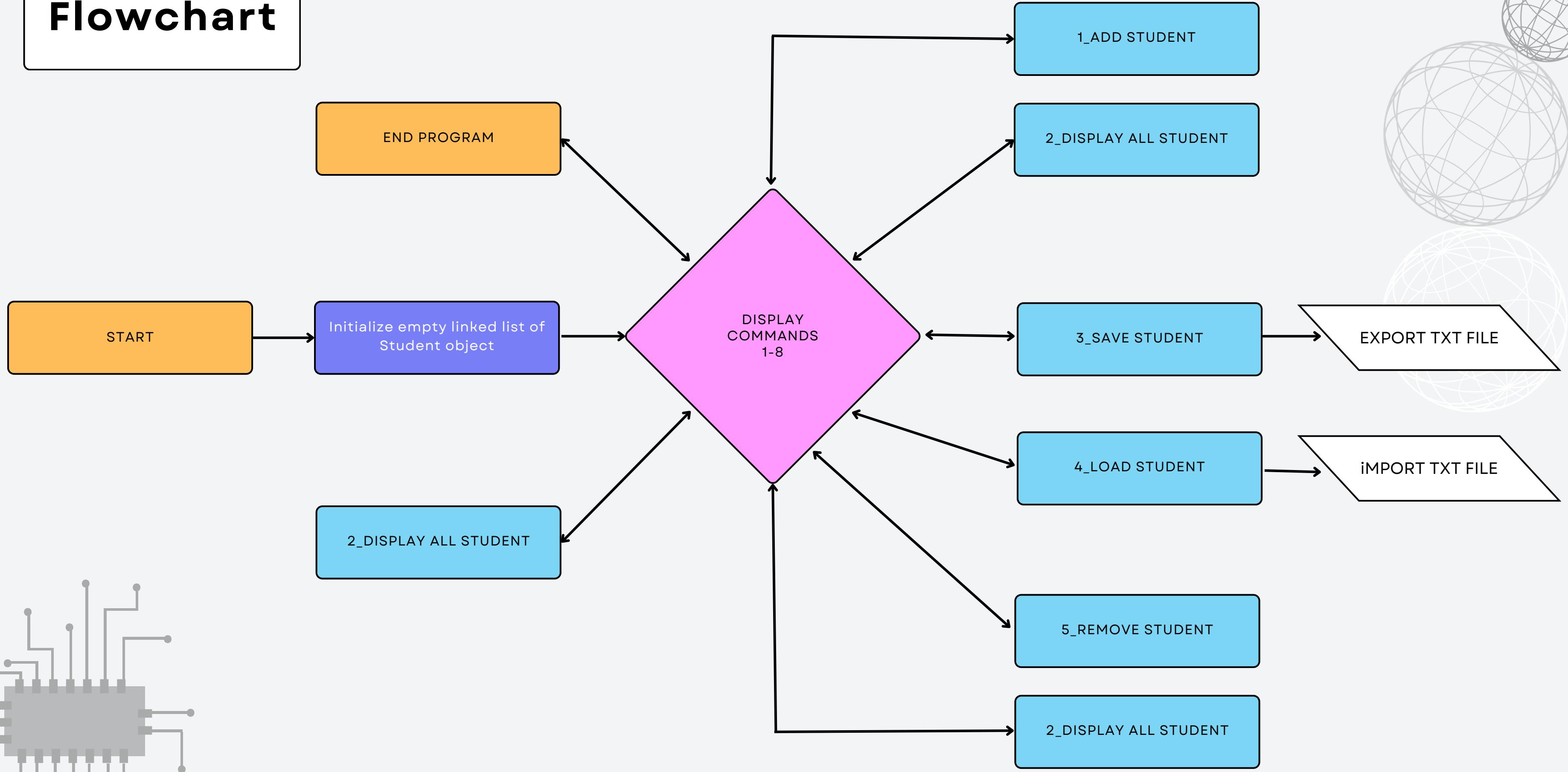
It features a simple user interface with a menu-driven system for ease of navigation.



User have to type a command from 1-8, each with its different function.

```
C:\Users\user\source\repos\2_Final C++ Project\x64\Debug\2_Fin...
=====
Student Grade Tracker System
=====
1. Add student
2. Display all students
3. Save students to file
4. Load students from file
5. Remove a student
6. Update a student's information
7. Sort students by grade
8. Exit
=====
Command:
```

Flowchart



EXPLAIN

01

PROGRAM FLOW

OPEN MO YUNG PSEUDOCODE TXT

02

CODE BREAKDOWN

DISCUSS MO LAHAT NG SECTIONS
NG CODE, AT CONCEPTS INVOLVED





SCOPE AND LIMITATIONS

SCOPE

1. The system allows users to add student data including name, age, and grade, which is then stored in a doubly linked list.
2. Users can display all students' data, providing a quick overview of all the records in the system.
3. The system offers file handling capabilities, allowing users to save the current student data into a text file and later load this data back into the system.
4. Users can remove a student from the system by specifying the student's ID.
5. If necessary, users can update a student's information in the system using the student's ID.
6. The system also provides a function to sort all student data by grades.

LIMITATION

1. The system currently supports a maximum of 99 students at a time, as IDs are allocated from 01 to 99.
2. There's no search functionality for finding a specific student based on certain criteria (like name or grade range). Users have to browse through all records to find a specific student.
3. Data validation is basic – it checks the validity of age and name only. Other data like grades are not validated.
4. Data security and privacy measures are currently not implemented. Any user can view, edit, or delete student records.

CONCLUSION

Our **Student Grade Tracker System** represents a step forward in efficiently managing and tracking student grades, particularly in public schools in remote areas.

By offering a user-friendly, versatile, and efficient console-based application, we have aimed to make grade management more accessible and straightforward.



CONCLUSION

Our **Student Grade Tracker System** show the potential of simple digital solutions to address specific challenges.

It serves as a reminder that technology, when we appropriately applied, can significantly enhance efficiency .

We hope that this system will be a useful tool for teachers on remote areas and look forward to exploring potential enhancements and improvements in the future.



GARCIA



GROUP 3



MACO

QUESTIONS?

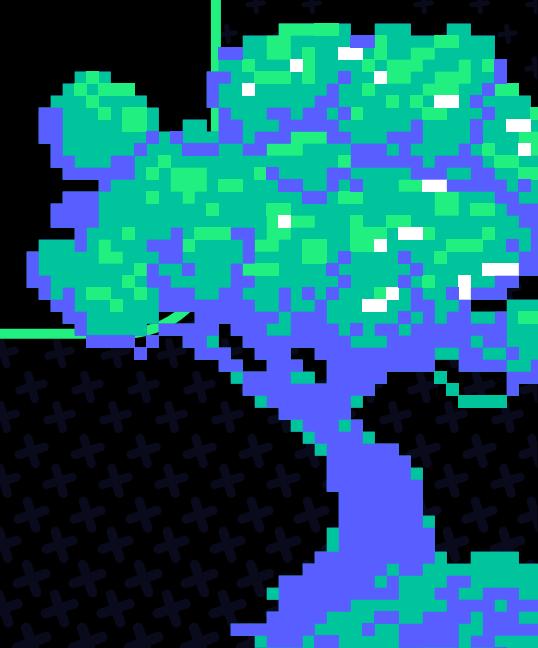
Regarding sa Program...
Suggestions, Feedback,
Violent Reactions?

Gisahan na haha...

```
C:\Users\user\source\repos\2_Final C++ Project\x64\Debug\2_Final ...
=====
Student Grade Tracker System
=====
1. Add student
2. Display all students
3. Save students to file
4. Load students from file
5. Remove a student
6. Update a student's information
7. Sort students by grade
8. Exit
=====
Command:
2
=====
ID: 0002
Name: Melo Ervy Garcia
Age: 19
Grade: 92
=====
ID: 0003
Name: Roxxie G
Age: 21
=====
```



KUYA MELO



GARCIA



LEVEL
UP

GROUP 3

MACO

END



GARCIA



MACO



BENITEZ



QUITORIANO



PRIETO



BALEAN

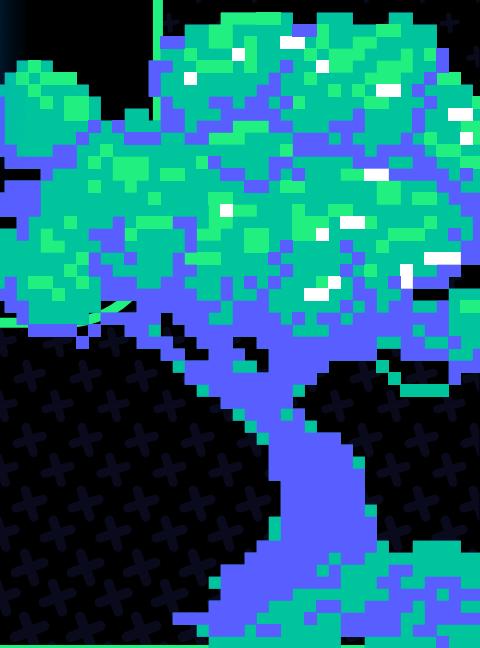


LOPEZ



IGANA

LEVEL
UP



I LOVE YOU
MEOW MEOW

KUYA MELO

