Student Management System on C++

Yuchen Song Student No.201830360498

I. Introduction

The project is aimed at building a student management system to store students' information. The basic functions for the system are required as adding records, modifying records, showing all records, finding individual record (based on name, number, etc), remove record(s), etc. To make the system more useful, I added two more functions(locating function to locate student's order and plotting histogram function to draw a histogram of students' scores) and one security system (a password system). To achieve this goal, I applied OOP(Object Oriented Programming) method as well as the Procedural Programming and chose linked list to store information.

II. Methodology

A. Data structure

1) Structure Student has 8 members, they are ID number, name, gender, age, grade of math, computer and English and a pointer to next Student structure (that's what we call next node in a linked list). I originally added a pointer to the last Student structure node, but it seems little use for the project, so I have deleted it now.

```
7   struct Student
8   {
9       int id;
10      string name;
11      string gender;
12      int age;
13      int math;
14      int computer;
15      int English;
16      Student *next;
17      //Student *last;
18  };
```

Figure 1. Structure Specification

2) There are 15 functions(main function included) in the source code. And I will cover them in a single running test.

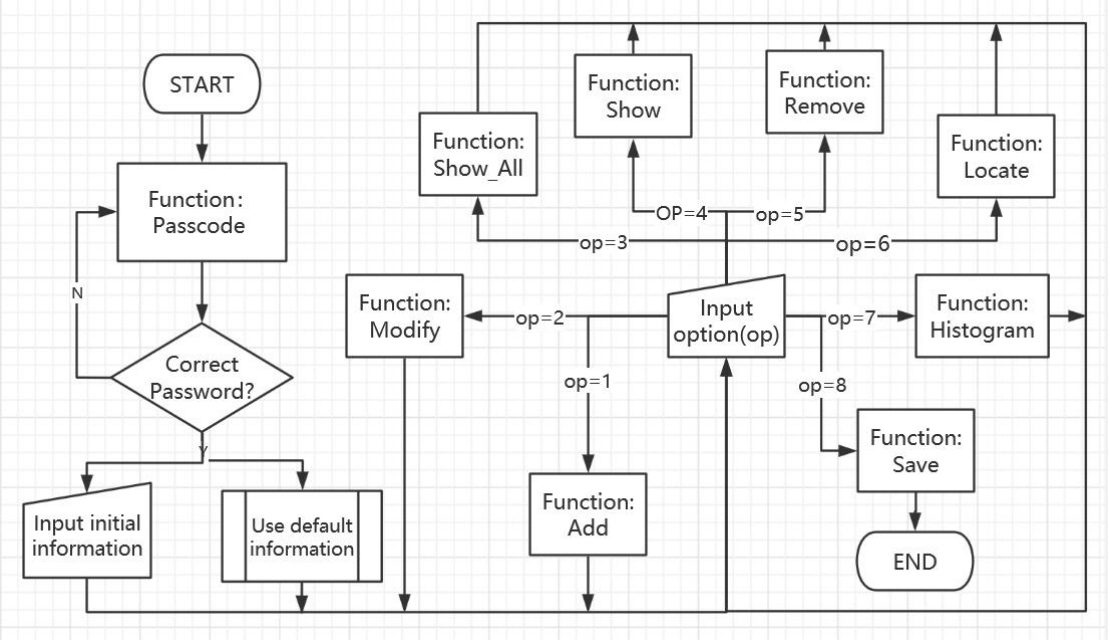

Figure 2. Flow Chart

In the beginning, the program will check the password and let user go to the next step only if the user input the correct password. The password system also has a security procedure: every 5 times user input wrong password, it will be locked for 1 minute.

Then if the user input the correct password, he will go to the second step: type in students' information. The user can also skip this step by input 0 to pass in the default student, Bill (his information is stored in "default data.txt").

After that, the user finally enter the system. The program will ask the user to input options [1~8], different options will activate different functions respectively. Details are shown as follows.

option	1	2	3	4	5	6	7	8
function	Add	modify	Show_all	show	remove	locate	histogram	logout
detail	Add one student	Modify the information	Show all data	Show one student	Remove data	Return the order	Plot histogram	Save data and logout

Table 1. Functionalities Introduction

B. Technical Problems

I encountered several technical problems on my way to accomplishing the tasks. The first one was about delay function. Considering the requirement that not mentioned libraries are forbidden, I cannot use Windows.h and built-in function “Delay”. I finally tackled the problem with the help of loop and number countering.

The second obstacle was how to restrict illegal input. I thought about endless loop, and only under certain circumstances can the program break out of the loop, so I used while(1) together with if-else sentence to ensure the input is legal.

The third trouble was about the linked list. In reference to different tutorials, it’s difficult to keep the list uniformed. For example, some articles may say you should use *head and *last as well as pointer to both next node and last node, others may say you only need to use *head and pointers to next node. I finally unified the form to *head->*normal student->*end->NULL. That brings advantages to the porting of separating defined functions.

The last problem I met was very tricky. When I almost finished the whole project, I corrected some codes in the source code rather than in the solution. When I figured it out, I copied the codes to the solution, then something strange happened: All the functions threw a warning and all cins and couts are said to be “not clear”. Actually there are two little problems here: one is owing to the slow response of VS, when I delete “using namespace std;” and rewrote it, all warnings of cin and cout have gone. The second problem was due to a miss of “}”, my own carelessness was to blame.

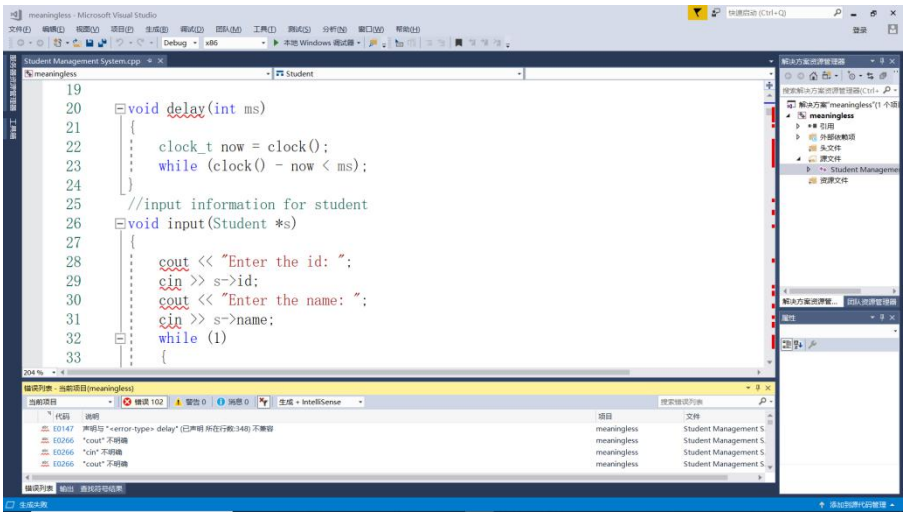

Figure 3. A tricky Problem

C. Testing of Program

The program went well and different functionalities can be implemented.

III. Results

A. Security System

Before entering the system, the program will first check the identification. Only if the user input the right passcode(“passcode”), the system will unlock.

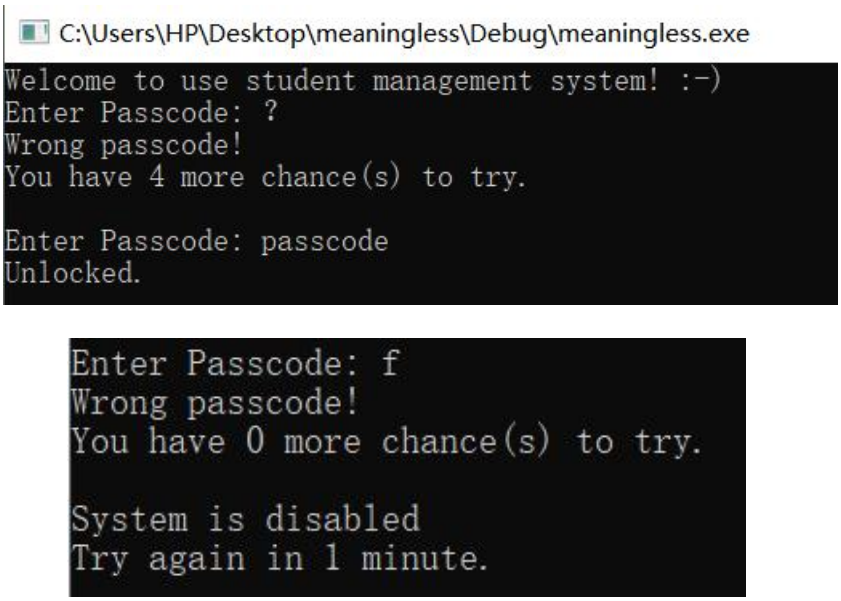

Figure 4-1 Security System

B. Create a Linked List.

There are two ways to initialize a linked list. Use the default information(input 0) or input your information at hand manually.

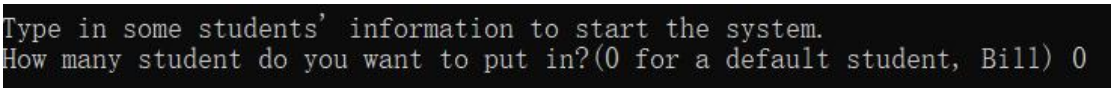

Figure 4-2 Create a Linked List

C. Select your options.

After the initialization, you can use the system now. Input your options to get to the next step.

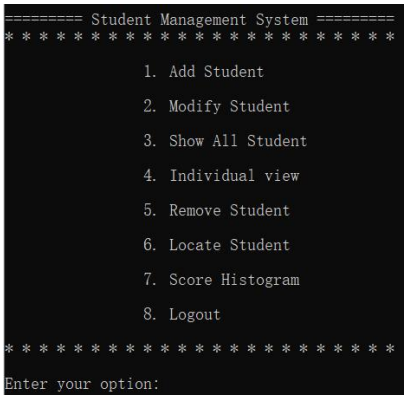

Figure 4-3 Select an Option

D. Add a student.

If you input 1 here, you will be in the adding mode. The system will ask you to input the information of the newly added student. (Note. gender will be restricted to “male” or “female” only)

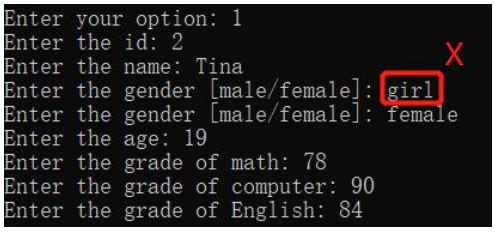

Figure 4-4 Add a Student

E. Modify a Student.

If you input 2 here, you will be in the modifying mode. The system will first ask you whether to use option 6 to locate the student. Here we can remember Tina is student 2 (Bill is student 1), so we can skip the procedure. Then you can modify the wrong input. (Note.category is restricted to the given ones)

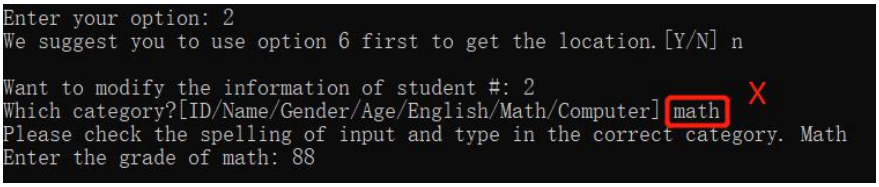

Figure 4-5 Modify a Student

F. Show All Students.

If you input 3 here, you will see the data of all students.(Note that the wrong data has already been modified.)

Figure 4-6 Show All Student

G. Individual View.

If you input 4 here, you can see the data of a single student. Like option 2, the system will first ask you whether to use option 6 to locate the student.

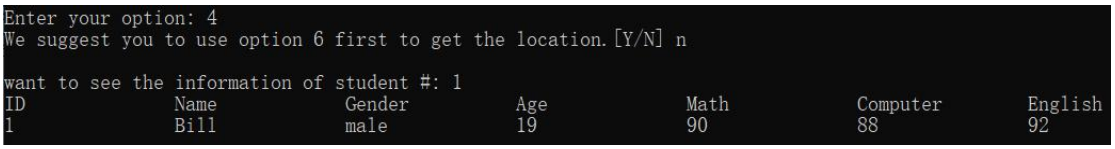

Figure 4-7 Show All Student

H. Remove Record

If you input 5 here, you can make a choice to remove the record of a single student or all students' record. The program also contains a double check in case of regretting.

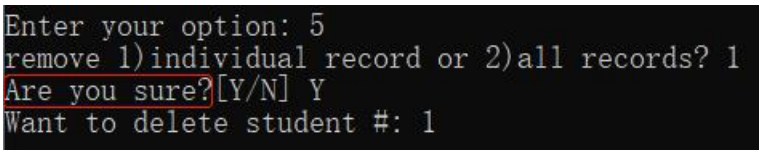

Figure 4-7 Remove Record

I. Locate Student

If you input 6 here, you are able to locate a student with his/her name or id.(Note the Bill's record has been deleted, Tina is expected to be located at #1).

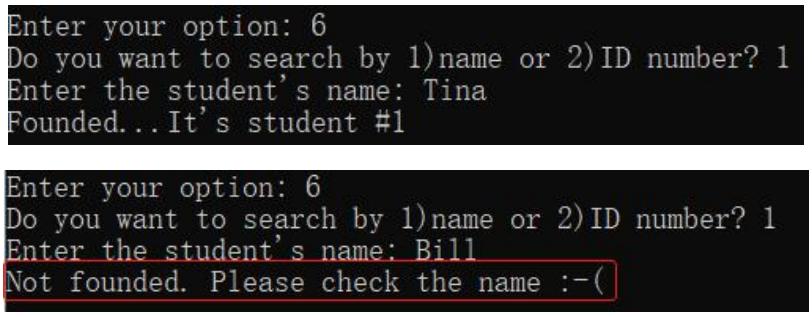

Figure 4-7 Locate Record

J. Score Histogram

If you input 7 here, you can see the histogram of each subject.

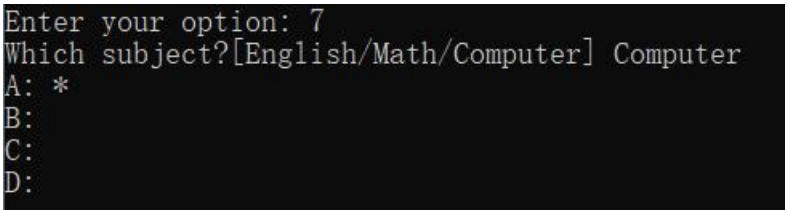

Figure 4-7 Score Histogram

K. Save Data and Logout

If you input 8 here, you will store the data and log out the system.

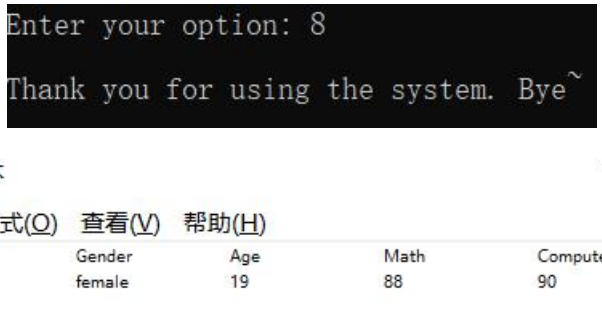

Figure 4-8 Save Data and Logout

IV. Conclusion and future development

In this project, I built a student management system that can meet the basic requirement. In addition, I added two more functionalities and a password system. From this project, I have a better understanding of linked list and the usage of endless loop.

Future works include illegal input detecting and auto-correcting, designing a nicer user interface, developing a staff management system, implement a filter function and so on.

Appendix

A. User Manual

I. Read the “readme.txt” file before running the code.

II. Copy the source code to the solution folder.

III. Copy the data file “default data.txt” to the same folder as source code.

IV. Open and run the source code in solution.

V. If cin and cout cannot be recognized, delete and rewrite “using namespace std;” at line 5.

VI. If you want to reset the password, you can do it in line 408.

VII. If you input illegal data type, for instance, input “Tom” to id category, the program may die. In this case, you should shut the program and restart it.

B. Data

The default student data used in the program is a virtual student named Bill, more information can be found in file “default data.txt”