

Program ideal state



Sonny Shi

Introduction

This report will show the design concept and completion effect of the program, but it is more as a student's summary of the whole process of the program. There's always a difference between reality and dream, but it doesn't stop you from showing the most perfect thing to others. This report will also show the most perfect prototype of the program in my mind and what it should be able to achieve. Then compared with the program I have completed.

List

1 Screen 1 (Initial Interface)

1.1 Perfect state

1.2 Reality

2 Screen 2 (View Timetable)

2.1 Perfect state

2.2 Reality

3 Screen 3 (Add New Student data, Add New Subject data, Create New Timetable)

3.1 Perfect state

3.2 Reality

4 Summary

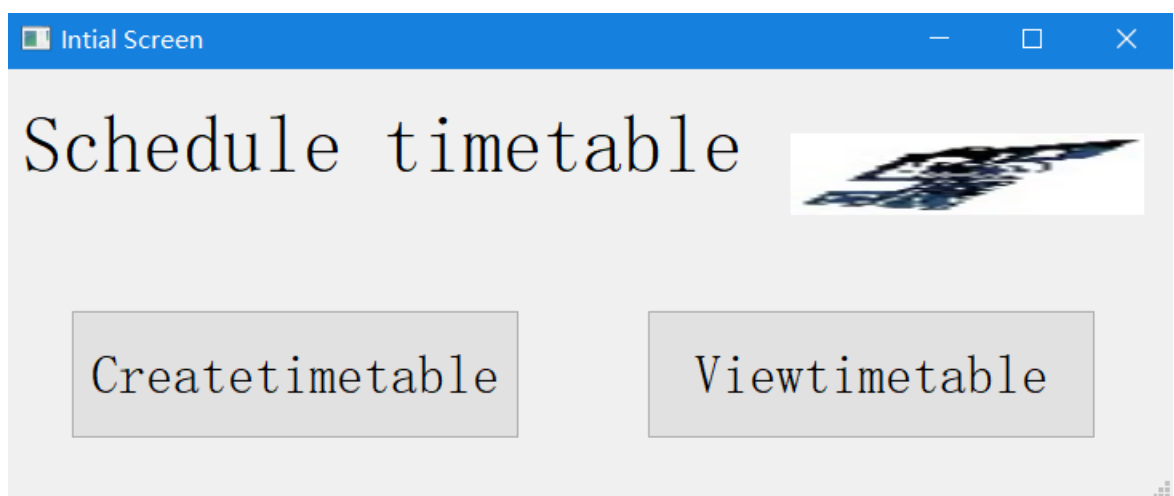
1 Screen 1 (Initial Interface)

1.1 Perfect state

As the initial display screen of the whole software, only two buttons were designed at the beginning of the design, namely: Create Timetable, View Timetable. And the logo of the company is in the upper right corner. The two buttons are also closely related, the button "View Timetable" can only be opened after a timetable has been created or it will prompt user first create a timetable.

1.2 Reality

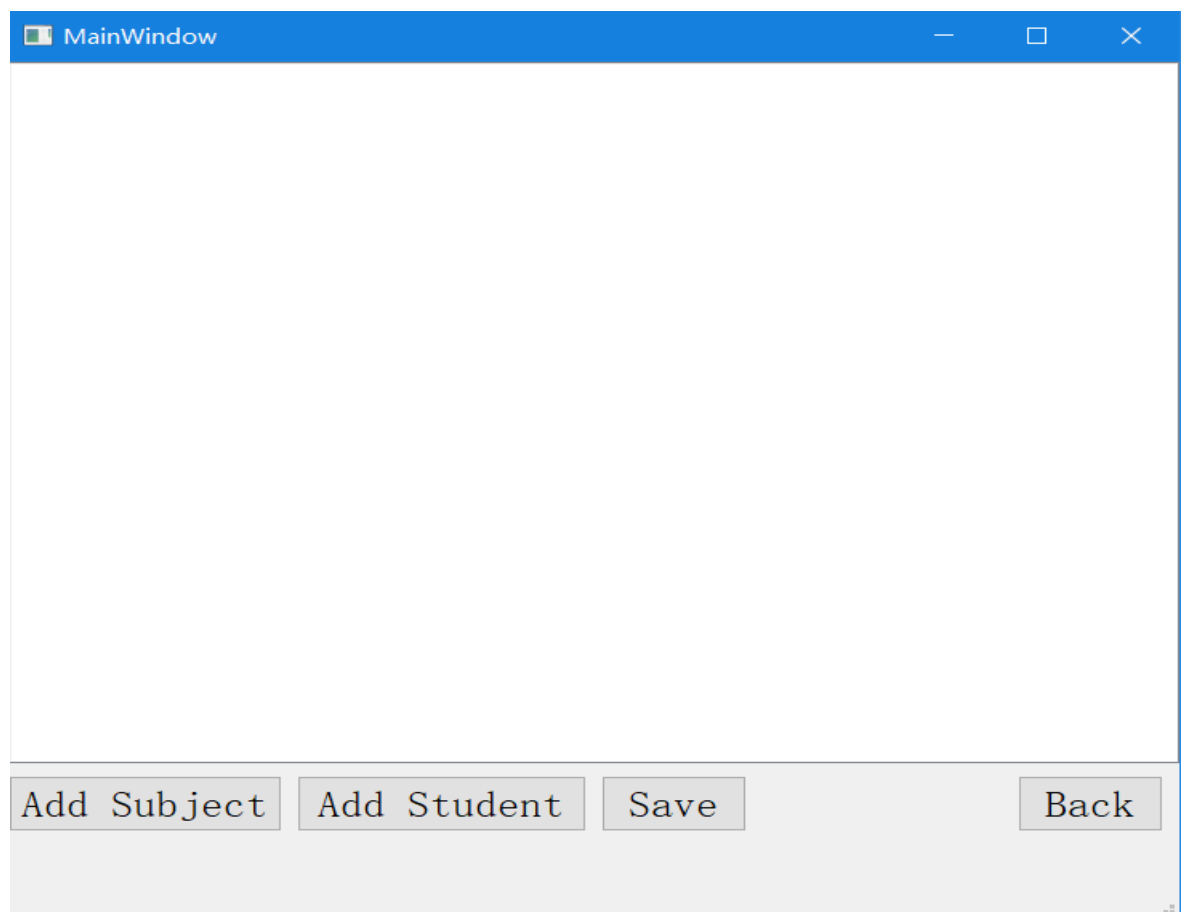
In the whole appearance, the program is in line with the original design, except for the wrong proportion of logo. But it doesn't conform to the application. The two buttons can be pressed in any situation, even if user have not created a timetable.



2 Screen 2 (View Timetable)

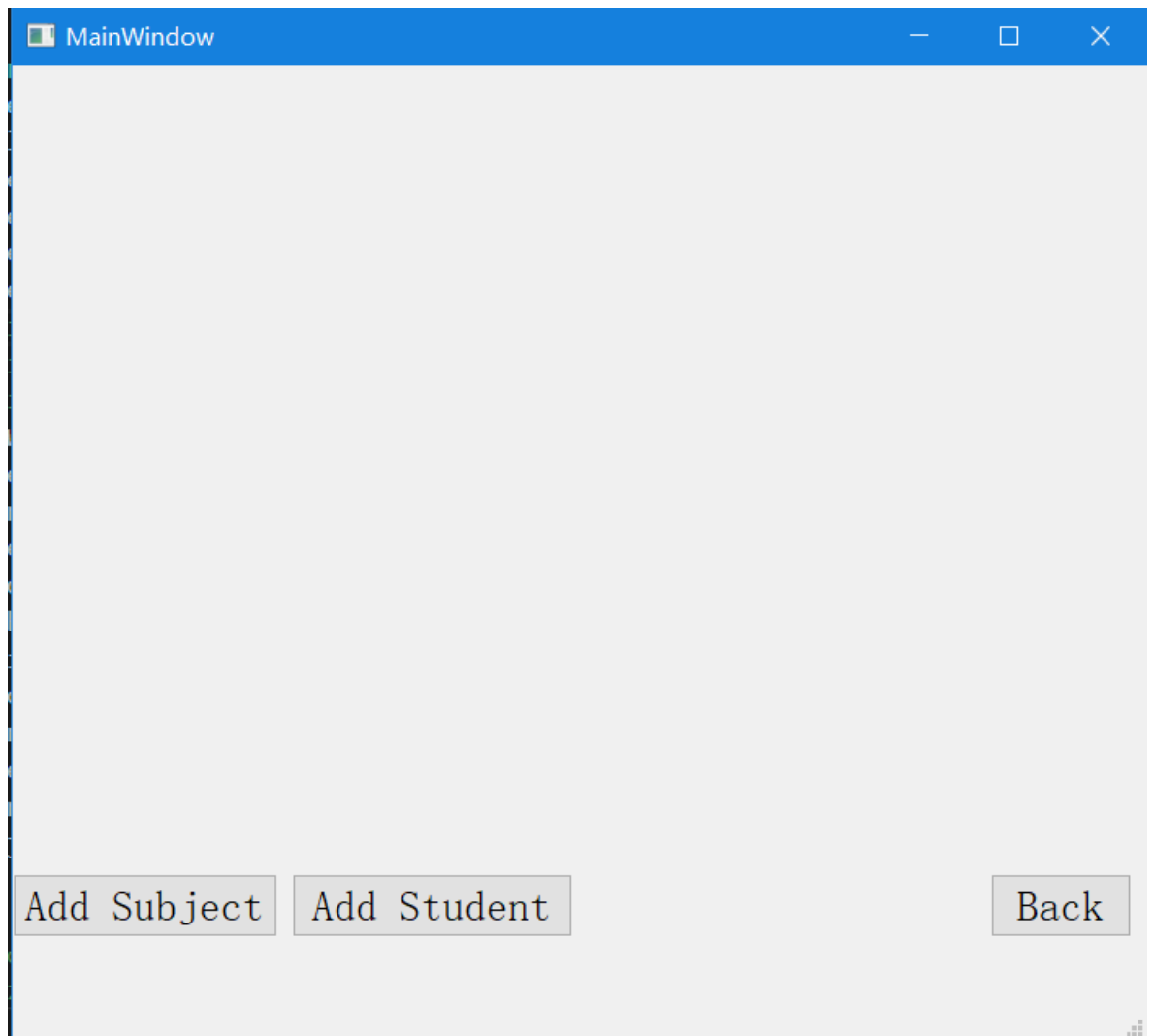
2.1 Perfect state

The second screen has four buttons, which are: Add Subject, Add Student, Save and Back. Their functions are as follows: Go to Add Subject screen, Go to Add Student screen, Save the timetable and Go back to initial screen. In the middle of the screen there is a special program to display the timetable, which can display the timetable that the user has made.



2.2 Reality

There are only three buttons actually achieved. Display and save button are not completed, can only be removed in the program, and the following screenshot is the effect of removal.

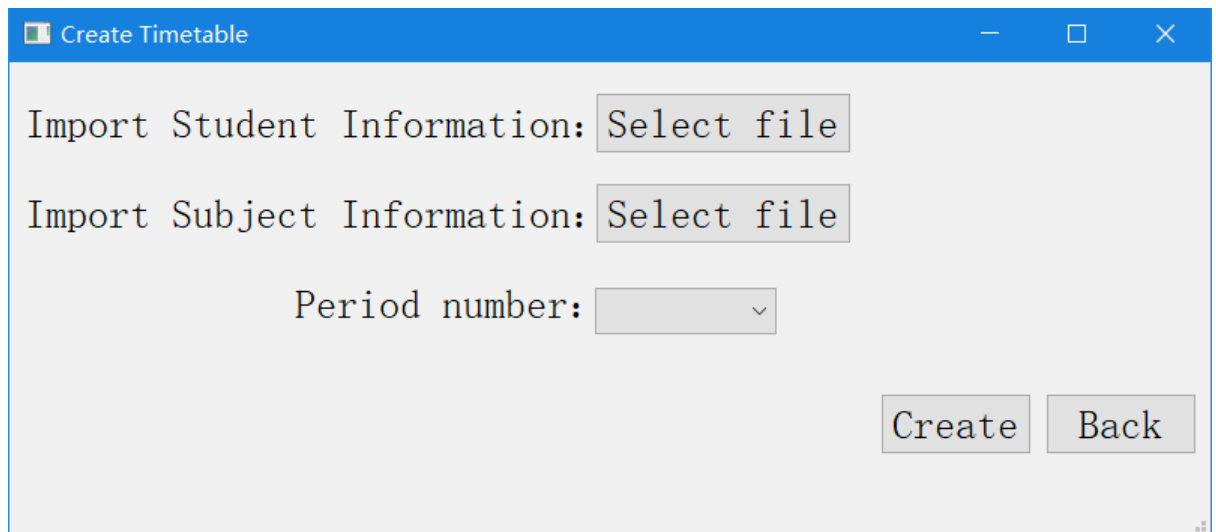


3 Screen 3,4,5 (Add New Student data, Add New Subject data, Create New Timetable)

3.1 Perfect state

Put the three screens together because they are closely related.

As the original data core, screen5 allows users to set the source of files, and also to find the database for adding data form of screen3 and 4.

The image shows a software window titled "Create Timetable". It has a blue header bar with standard window controls (minimize, maximize, close). The main area is light gray and contains three input fields. The first is labeled "Import Student Information:" followed by a button labeled "Select file". The second is labeled "Import Subject Information:" followed by another button labeled "Select file". Below these is a label "Period number:" followed by a dropdown menu with a downward arrow. At the bottom right, there are two buttons: "Create" and "Back".

By the two “Select” button, the data the location of the CSV file will transferred to the other two form. Then the two form can add new data into the database. The “Select” button will also test weather the file that user choose is csv file or not. The add new student screen will read the data in the subject csv file and give different choice in the combo box for user to choose.

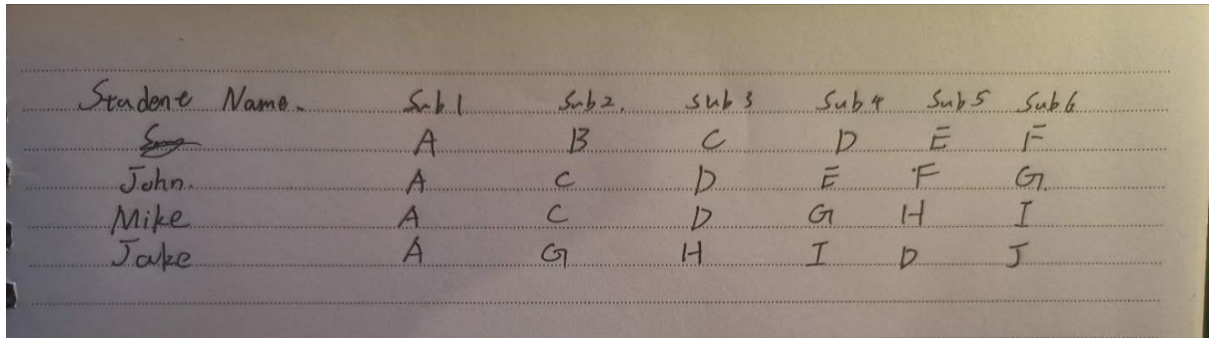
3.2 Reality

The actual results are completely out of line with the design, and the database location selected on screen 5 cannot be communicated to the other two forms. Only can add an initial into the program in order to let two form read the data.

However, good new is that the combo box work. The new subject add into the csv file can be read and print in the combo box but it needs to restart the whole program or the new subject will not be display in the combo box.

4 Summary

In the process of programming, I found that there is an important question that I have not considered clearly. How to arrange different courses?



Student Name	Sub 1	Sub 2	Sub 3	Sub 4	Sub 5	Sub 6
Sam	A	B	C	D	E	F
John	A	C	D	E	F	G
Mike	A	C	D	G	H	I
Jake	A	G	H	I	D	J

Take this picture as an example. There are four students in total. Each student has A course and five different courses. My idea is to divide the remaining five courses into two groups according to their similarity. However, it only can use in a small number of students Mike can be targeted. Once the number of normal students in a grade is brought in, the grouping will become extremely large, and each group will continue to be divided into several different groups. After that, it is necessary to make a more detailed sorting, which courses can be carried out at the same time for those students to attend classes together. These are all things to consider. You can't just imagine that all the students are taking the same course. In that case, it's just a random order of 1 to 6. This problem has been bothering me for nearly a

week, and I haven't been able to come up with a solution. Near the deadline, I can only choose to give up these features and spend more time perfecting the rest.

I found a lot of problems from this project. First, I mistakenly estimated what I could achieve. Secondly, I chose a very long and difficult project. But it also gave me an idea that I still have a lot to learn. In perfecting the whole project, I didn't regard it as an interest, a want to complete. It is regarded as a learning task, just to show the understanding of knowledge. But after this time, I have a different idea about the course selection in the future university. Should I continue to choose computer related majors in the university? If I continue to choose computer science, can I devote myself to it? This is what I need to think deeply and make a decision in the next few months