Hong Kong Diploma of Secondary Education Examination xxxx

Information and Communication Technology (Coursework)

Option D: Software Development

Title: NSS student-subject allocation system

Candidate Name: XXXXX

Candidate ID No: XXXXX

Contents

Chapter 1 Introduction

1.1 Background

Situation

Nowadays, technology has been well-developed into a more advanced tool than before among the world. Therefore, more and more schools would like to use computer system to produce a better quality.

Moreover, under the New Senior Secondary (NSS) curriculum, students in Hong Kong are required to select elective subjects from among the various subjects offered by schools. It is a must of all school to have an allocation system for student to select elective subject they want.

As a result, more and more subjection allocation system is being produced.

Problem

As an IT project manager for a secondary school, school would like to provide a student-subject allocation system to let student to have a better quality system to select elective subject that they are interest.

I am now responsible for the project. That’s why I am working on this written report to provide solutions for the school. It is hoped that the student-subject allocation system can bring benefit to both school and students

Chapter 2 Design

2.1 Brief Description

As I am aimed at doing a student-subject allocation system, I am going to set up the student-subject allocation system with different functions.

It means that in the program, not only is selection included in it, but also functions like login, display subject, update choose, etc. are built so that the program can be comprehensive enough and reach the real situation of an allocation system.

To make the system better in image, I use different on the font and the background so that it seems to be more clearly. Followed by it is a rounded login function and password changing function. When the users login to the system, they will see a main menu with different commands so that the program can be well-arranged.

The system can let both teachers and students use and they have different functions. For students, they can select their subject very clear and they can change afterward. As for teachers, they can check out the details and information such as students' choices and also the name list of each subject.

For the elective subject, there are total 3 elective blocks and each block has 3-7 subjects for students choose. And it shows as the table below.

Block1	Block2	Block3
1. *Physics* 2. *Chemistry* 3. *Economics*	1. *Physics* 2. *Economics* 3. *Biology* 4. *Chinese History* 5. *ICT* 6. *V.A* 7. *E&RS*	1. *Chemistry* 2. *Geography* 3. *Biology* 4. *History* 5. *NSS P.E* 6. *BAFS*

2.2 General Function

The main program contains two parts which for students and teachers. The function of the two program are different that can provide different use for different users. User can choose their role before they login the system.

Login & Logout System

To make the system become more user friendly, I allow the users login to the system to identify them. It can also identify who are they, students or teacher.

Also, the login function has another proposes which is to save the selection of the particular student. So they can review their choices and also update it. For those teachers, they can use the system to find out the situation of the selection.

At the same time, a logout function is also set for users to logout and let other users to use the system instead of re-opening the system again.

Password Changing Function

To make the system more convenient, a password changing function is needed. Users can change their password so that they will not forget it easily. It is a user-friendly function.

Display Elective Function

The display elective function can let and students have a look about the choices of election in different block or the remaining quota of each subject before they select their elective subject.

Display Choices Function

This function is used to let students to check what their choices are and they can also confirm their last update time to ensure others will not using their account. Besides, they can check their personal information here.

As for teachers, they can use this function to check a particular student's choices. They can find personal information of students in this function too.

Choose Elective/Update Choices Function

The main function of the system, for students to choose elective subject they want. In this function, students can choose the subject block by block and they no need to choose subject form block1 first. They can choose the subjects in order which they want. Since some students want to choose elective subject form block2 or block3 first, this function is wonderful for them.

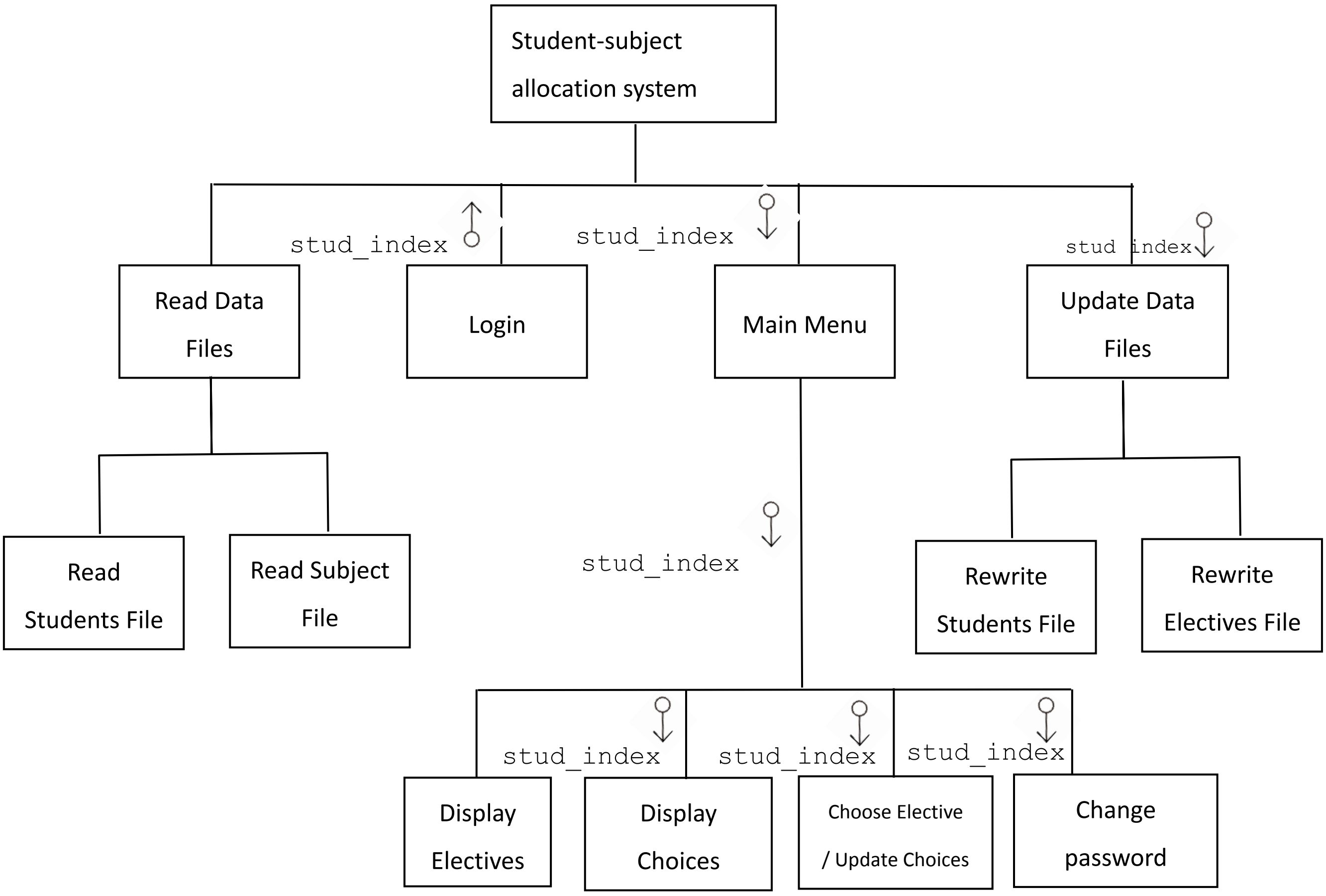
Display Subject Name List Function

It is a unique function for teacher, the aim of this function is for teacher check which students choose a particular subject. So, teacher can know more about the interest of student or that can use this data to do other analysis.

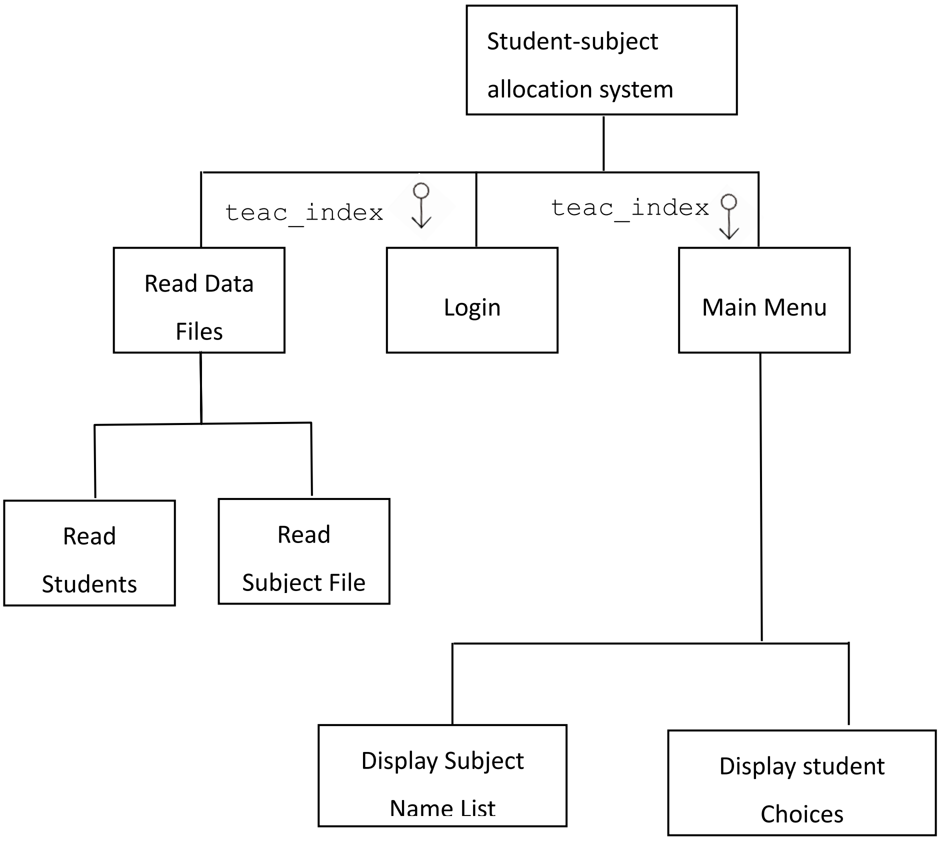
2.3 System Design

Flow Chart of the System

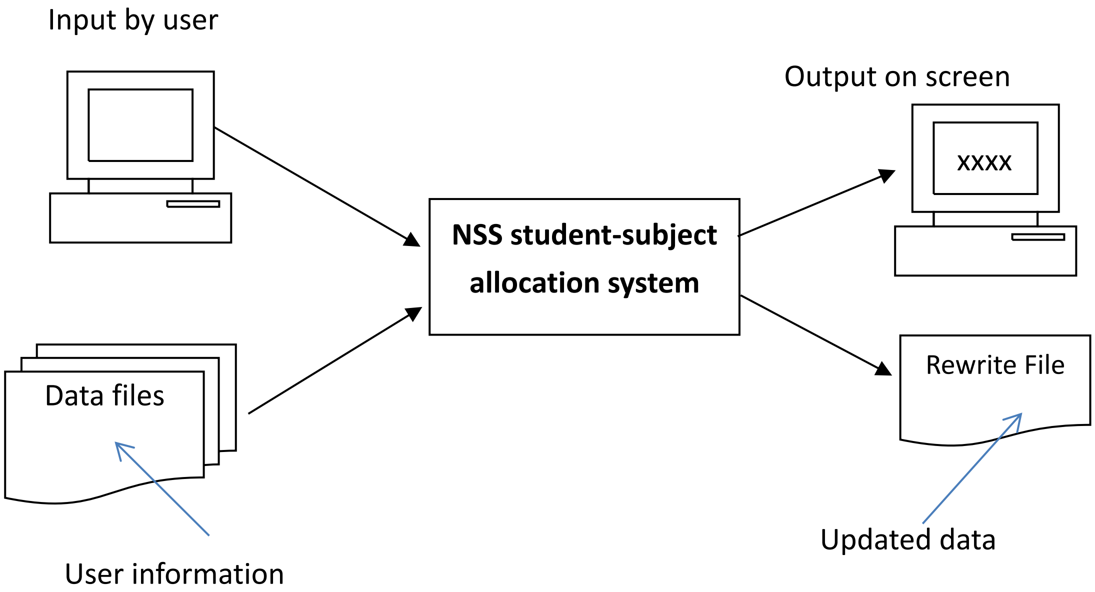
For Students:



For Teacher:


Student-subject allocation system
teac_index
teac_index
Read Data Files
Login
Main Menu
Read Students
Read Subject File
Display Subject Name List
Display student Choices

The below diagram describe the data flow:



- File structure

- There are 5 text files to store the data. Record will use to store the data.

1. Students.txt ← Store students record
2. Teachers.txt ← Store teachers record
3. Elective1.txt ← Store elective1 record
4. Elective2.txt ← Store elective2 record
5. Elective3.txt ← Store elective3 record

1. Data file for storing students' information:

For student record, the data file stores the record of each student, which includes the following information:

- ○ Student ID (4 characters)
- ○ student Password (4 characters)
- ○ Student Name (25 characters)
- ○ Indicator of elective1 (1 characters)
- ○ Indicator of elective2 (1 characters)
- ○ Indicator of elective3 (1 characters)
- ○ Time of last update (16 characters)
- ○ Selection of elective1 (integer)
- ○ Selection of elective2 (integer)
- ○ Selection of elective3 (integer)

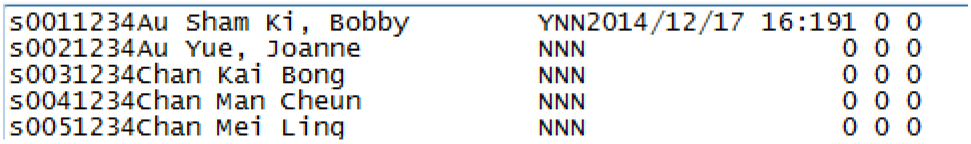
Each line of the data file stores the record of one participant with the following format:

e.g

S001	Student ID (4 characters)
1234	Student Password (4 characters)
Au Sham Ki, Bobby	Student Name (25 characters)
Y	Indicator of elective1 (1 characters)
N	Indicator of elective2 (1 characters)
N	Indicator of elective3 (1 characters)
2014/12/17 16:19	Time of last update (16 characters)
1	Selection of elective1 (integer)
0	Selection of elective2 (integer)
0	Selection of elective3 (integer)

Sample File:

Students.txt



```
s0011234Au Sham Ki, Bobby        YNN2014/12/17 16:191 0 0
s0021234Au Yue, Joanne           NNN                0 0 0
s0031234Chan Kai Bong            NNN                0 0 0
s0041234Chan Man Cheun           NNN                0 0 0
s0051234Chan Mei Ling            NNN                0 0 0
```

2. Data file for storing teachers' information:

For teacher record, the data file stores the record of each teacher, which includes the following information:

○ Teacher ID (4 characters)
○ Teacher Password (4 characters)
○ Teacher Name (25 characters)

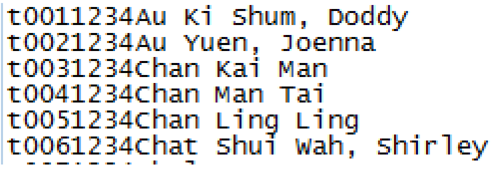
Each line of the data file stores the record of one participant with the following format:

e.g

t001	Teacher ID (4 characters)
1234	Teacher Password (4 characters)
Chan Ling Ling	Teacher Name (25 characters)

Sample File:

teachers.txt



```
t0011234Au Ki Shum, Doddy
t0021234Au Yuen, Joenna
t0031234Chan Kai Man
t0041234Chan Man Tai
t0051234Chan Ling Ling
t0061234Chat Shui Wah, Shirley
```

3. Data file for storing elective1’s information:

For elective1 record, the data file stores the record of elective1, which includes the following information:

○ Subject Name (16 characters)

○ Block Number (integer)

○ Elective Number (integer)

○ Number of quota (integer)

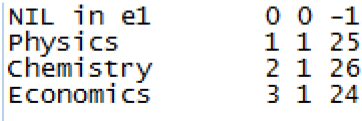
Each line of the data file stores the record of one participant with the following format:

e.g

Physics	Subject Name (16 characters)
1	Block Number (integer)
1	Elective Number (integer)
25	Number of quota (integer)

Sample File:

Elective1.txt



```
NIL in e1          0 0 -1
Physics            1 1 25
Chemistry          2 1 26
Economics          3 1 24
```

*Remark: ‘NIL in e1’ means the student did not choose elective in elective1

4. Data file for storing elective2's information:

For elective2 record, the data file stores the record of elective2, which includes the following information:

○ Subject Name (16 characters)

○ Block Number (integer)

○ Elective Number (integer)

○ Number of quota (integer)

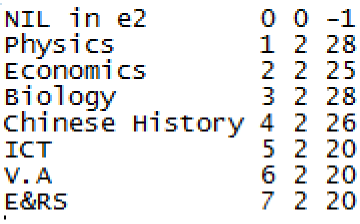
Each line of the data file stores the record of one participant with the following format:

e.g

Economics	Subject Name (16 characters)
2	Block Number (integer)
2	Elective Number (integer)
25	Number of quota (integer)

Sample File:

Elective2.txt



```
NIL in e2       0 0 -1
Physics         1 2 28
Economics       2 2 25
Biology         3 2 28
Chinese History 4 2 26
ICT             5 2 20
V.A             6 2 20
E&RS            7 2 20
```

*Remark: 'NIL in e2' means the student did not choose elective in elective2

5. Data file for storing elective3's information:

For elective3 record, the data file stores the record of elective3, which includes the following information:

○ Subject Name (16 characters)
○ Block Number (integer)
○ Elective Number (integer)
○ Number of quota (integer)

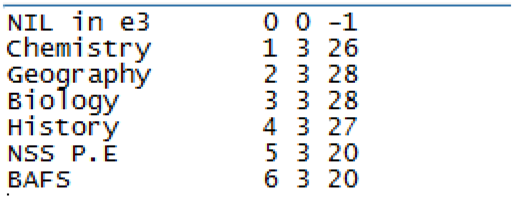
Each line of the data file stores the record of one participant with the following format:

e.g

Chemistry	Subject Name (16 characters)
1	Block Number (integer)
3	Elective Number (integer)
26	Number of quota (integer)

Sample File:

Elective3.txt

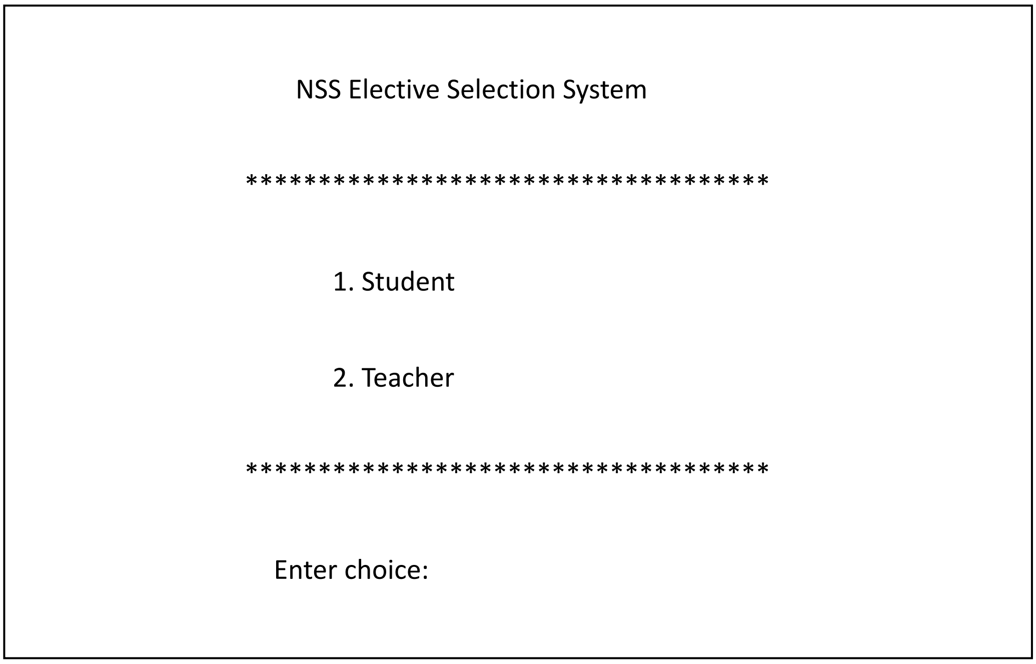


```
NIL in e3        0 0 -1
Chemistry        1 3 26
Geography        2 3 28
Biology          3 3 28
History          4 3 27
NSS P.E          5 3 20
BAFS             6 3 20
```

*Remark: 'NIL in e3' means the student did not choose elective in elective3

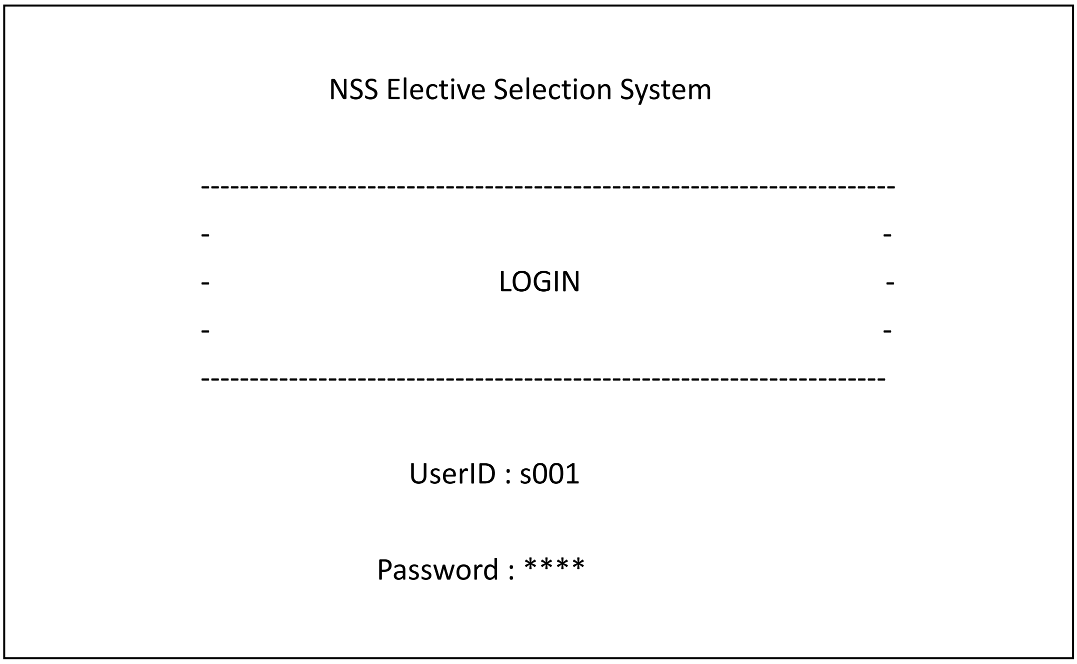
2.4 Design of User Interface

- **Logon Screen (1)**



```
              NSS Elective Selection System

        ************************************

                1. Student

                2. Teacher

        ************************************

          Enter choice:
```

- **Logon Screen (2)**



```
                    NSS Elective Selection System

          ---------------------------------------------------------------------
          -                                                                   -
          -                              LOGIN                                -
          -                                                                   -
          --------------------------------------------------------------------

                          UserID : s001

                        Password : ****
```

- **Students Main Menu**

```
Welcome
                         NSS Elective Selection System
                                  Main Menu
                    ************************************
                         1.   Display Elective

                         2. Display My Choices

                         3. Choose Elective/Update Choices

                         4. Change password

                         5. Quit

                    ************************************

                    Enter choice:
```

- **Display Elective**

```
Elective Name          Elective#      Block#   Quota
  ----------------------------------------------------------------
Physics                        1           1     25
Chemistry                      2           1     26
Economics                      3           1     24

Physics                        1           2     28
Economics                      2           2     25
Biology                        3           2     28
Chinese History                4           2     26
ICT                            5           2     20
V.A                            6           2     20
E&RS                           7           2     20

Chemistry                      1           3     26
Geography                      2           3     28
Biology                        3           3     28
History                        4           3     27
NSS P.E                        5           3     20
BAFS                           6           3     20



            <<< Press <Enter> to return. >>>
```

- **Display Elective**

```
      Your Choices:
         ----------------------------------------------------
            Student ID    : s001

            Name              : Au Sham Ki, Bobby

            Your Choices :

                                 Elective1 : Physics

                                 Elective2 : NIL in e2

                                 Elective3 : NIL in e3

            Last Update : 2014/12/17 16:19
          <<< Press <Enter> to return. >>>
```

- **Main of Choose/Update Elective**

```
          NSS Elective Selection System
        Choose Electice/Update Choices
    ************************************
              1. Electice1

              2. Electice2

              3. Elective3

              4. Display My Choices

              5. Return
    ************************************
        Enter choice:
```

- **Choose Elective1 / Elective2/ Elective3**

```
Elective Name    Elective#    Block#    Quota
 ----------------------------------------------------------


Physics                  1          1    26
Chemistry                2          1    26
Economics                3          1    24


 ------------------------------------------


Enter your choice of Block 1 elective no. (0 for nil) :
```

- **Change Password**

```
Please enter your old password                 : ****

Please enter your new password (4 char)     :    ****

Please enter your new password again        : ****

Password changed.
Press <Enter> to return.
```

- **Tachers Main Menu**

```
Welcome,Au Ki Shum, Doddy
                         NSS Elective Selection System
                                        Main Menu
                      ************************************
                               1. Display Subject Name List


                               2. Display Students Choices


                               3. Quit
                      ************************************
                         Enter choice:
```

- **Display Subject Name List**

```
NSS Elective Selection System
Name list of student taking Physics
-------------------------------------------
Au Sham Ki, Bobby
Chan Shui Wah, Shirley
Chan Yick Yee, Eliza
Kwan Wan Cheong
Lam Ting Cheong

<Press <Enter> to return>
```

- **Display Students Choices**

```
                         NSS Elective Selection System
                            Display Students Choices
                ---------------------------------------------------------


                         Student ID : s001
```

Chapter 3 Implementation

3.1 Brief Description

The Dev-pascal is our choice to implement the NSS student-subject allocation system which I mentioned above. We are going to make a source program and compile it to an executable program.

To begin with, I will design the simply structure of the program by considering the procedure in the program. Besides the procedure, the sample output from above should be well-considered too as I want to make the appearance of the program better.

After that, I will start on producing the program by using the Dev-pascal with different program codes, procedures, functions, etc.

Last but not least, I will have an explanation on the execution of the program so that I can see clearly about the program flow.

3.2 Program Coding

In the term of program code, I will introduce some of those and all of the program code will show in my complete program in the Appendix Section, so you may refer to the Appendix Section 1 for the reference.

1) Record

Example:

```
stud_record = record
    studid : string[4];
    studpw : string[4];
    chooseelective1 : char;
    chooseelective2 : char;
    chooseelective3 : char;
    lastupdate : string[16];
    elective1 : integer;
    elective2 : integer;
    elective3 : integer;
var
    student : array[1..max_stud] of stud_record;
```

Record is a useful program code which can store data having same data type, for example the" Student ID" and the "Student password" being stored in the text file, words from the dictionary.

2) Text file

Example:

```
assign(f, 'students.txt');
reset(f);
```

A text file can help to store data from the program so that more data can be stored. Besides, the program can also read and get data from the text file.

3) Clrscr

Example:

```
clrscr:
```

'Clrscr' is used to clear all the words in the screen to make the appearance better. With the problem that text displayed will be accumulated and the screen will get more and more words, it will look very chaos. So, 'Clrscr' is useful to solve this issue.

4) Case

Example:

```
case choice of
   1 : display_elective;
   2 : display_choices(stud_index);
   3 : choose_elective(stud_index);
   4 : change_password(stud_index);
```

First, the case function will get the character which the users inputted. If the character is match with one of the case, it will do that case. Such as the character is 1 in the above example, the program will go to procedure display_elective.

The above are some of the important program code used in the program and the remaining program code will be shown in the program.

3.3 Procedure in the Program

- **Procedure read_students**

  It is a procedure used to read the information of students from the text file which stored students' information.

- **Procedure store_students**

  It is a procedure used to store the information of students to the text file from the program.

- **Procedure read_teachers**

  It is a procedure used to read the information of teachers from the text file which stored students information.

- **Procedure read_elective1**

  It is a procedure used to read the information of elective1 subject from the text file which stored elective1's information.

- **Procedure store_elective1**

  It is a procedure used to store the information of elective1 to the text file from the program.

- **Procedure read_elective2**

  It is a procedure used to read the information of elective1 subject from the text file which stored elective1's information.

- **Procedure store_elective2**

  It is a procedure used to store the information of elective1 to the text file from the program.

- **Procedure read_elective3**

  It is a procedure used to read the information of elective1 subject from the text file which stored elective1's information.

- **Procedure store_elective3**

  It is a procedure used to store the information of elective1 to the text file from the program.

- **Procedure display_elective**

  It is a procedure used to display all the electives which students can choose. Include electives in both Block1,2 and 3.

- **Procedure display_elective1_choice**

  It is a procedure used to display the electives which students can choose in elective block1.

- **Procedure display_elective2_choice**

  It is a procedure used to display the electives which students can choose in elective block2.

- **Procedure display_elective3_choice**

  It is a procedure used to display the electives which students can choose in elective block3.

- **Procedure display_choices**

  It is a procedure used to display student’s choices on the elective. It also included some information of students such as student ID, student name and the last update time.

- **Procedure choose_elective1**

  It is a procedure used to choose the elective which students choose in elective block1.

- **Procedure choose_elective2**

  It is a procedure used to choose the elective which students choose in elective block2.

- **Procedure choose_elective3**

  It is a procedure used to choose the elective which students choose in elective block3.

- **Procedure main_menu**

  It is a procedure display the main menu of students. After student login their account, they will go to this main menu.

- **Procedure display_students_choices**

  It is a procedure let teachers choose which student’s choices they want to see.

- **Procedure display_subject_namelist**

  It is a procedure let teachers choose which subject name list they want to see.

  And this procedure also included 3 procedures; they are searchblock1, searchblock2 and searchblock3. These 3 procedures are used to find out which student chooses the subject that the teacher wants to see.

- **Procedure main_menut**

  It is a procedure display the main menu of teachers. After teacher login their account, they will go to this main menu.

- **Function GetPWord**

  It is a procedure to make the password type be the user become '*' to keep the users' privacy. And the reference is from: http://computer-programming-forum.com/29-pascal/7af4f3f05f738777.htm.

- **Procedure loginstudent**

  It is a procedure used to let students login their account by input their student ID and password.

- **Procedure loginteacher**

  It is a procedure used to let teachers login their account by input their teachers ID and password.

- **Procedure choose_role**

  It is a procedure used to let user to choose their role, teacher or student.

3.4 Main Body of the Program

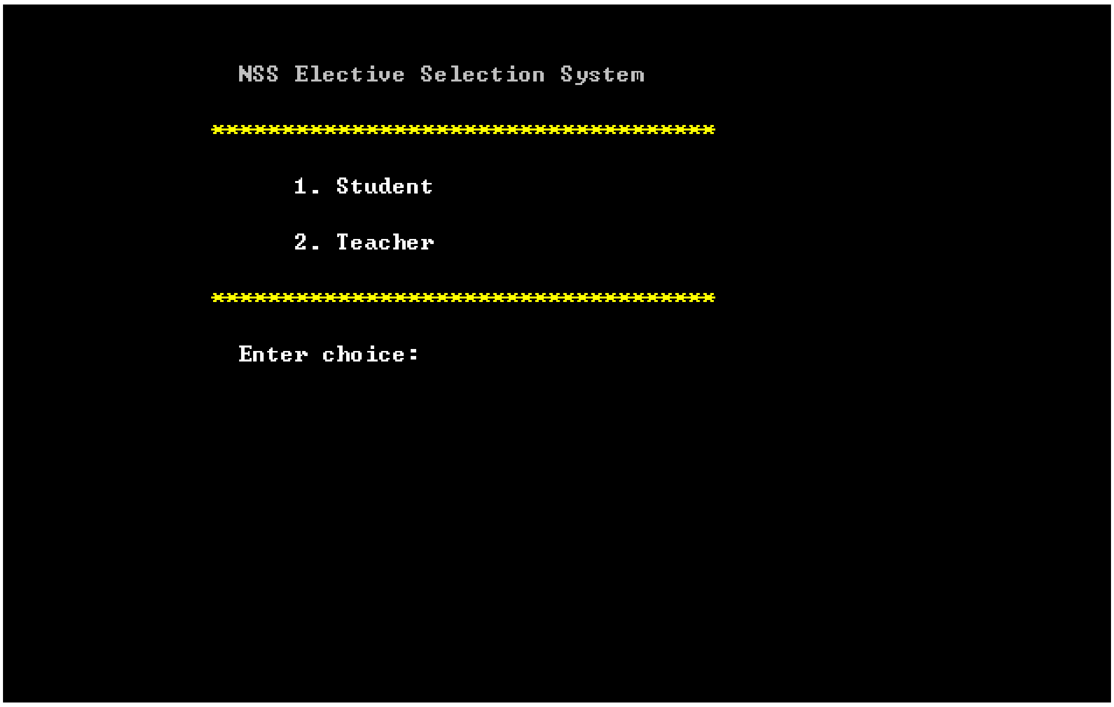
Here is the main body of the program:

```
begin  (* main body *)
   read_students;
   read_teachers;
   read_elective1;
   read_elective2;
   read_elective3;
   choose_role;
end.
```

3.5 Program Execution

All the text files should be placed at the same folder to make the program run correctly.

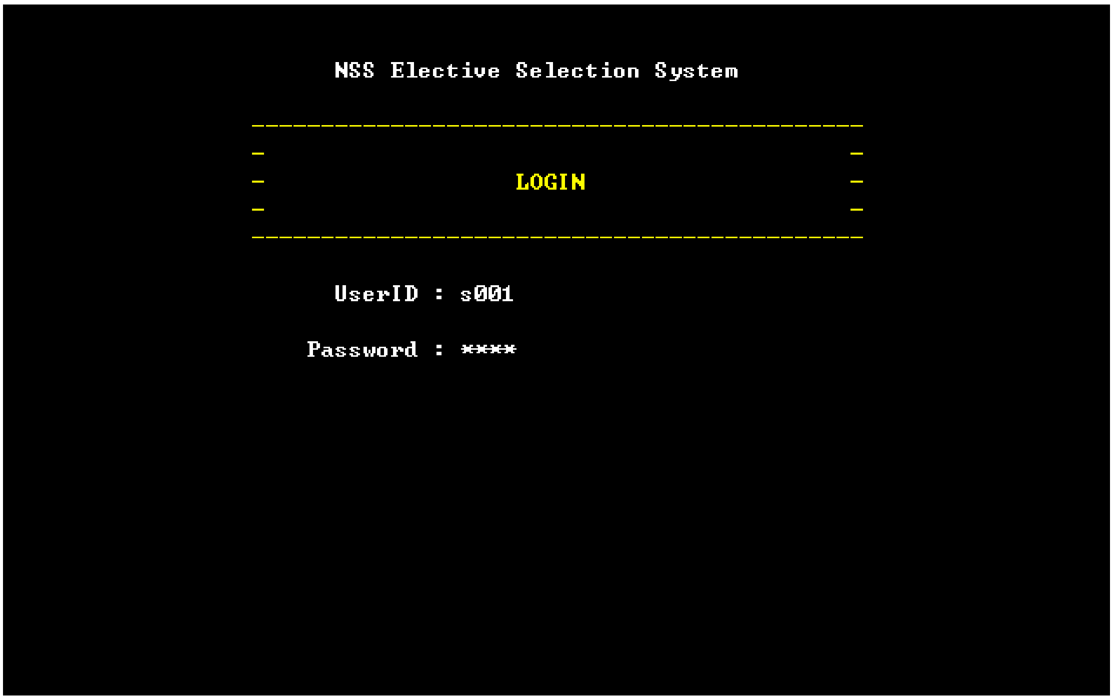
1) Enter page



```
  NSS Elective Selection System

************************************

      1. Student

      2. Teacher

************************************

  Enter choice:
```

The above picture is the enter page of the system. That means user will first go into this page when they use this system. In this page, user needs to choose their role (teacher or student) by typing 1 or 2.

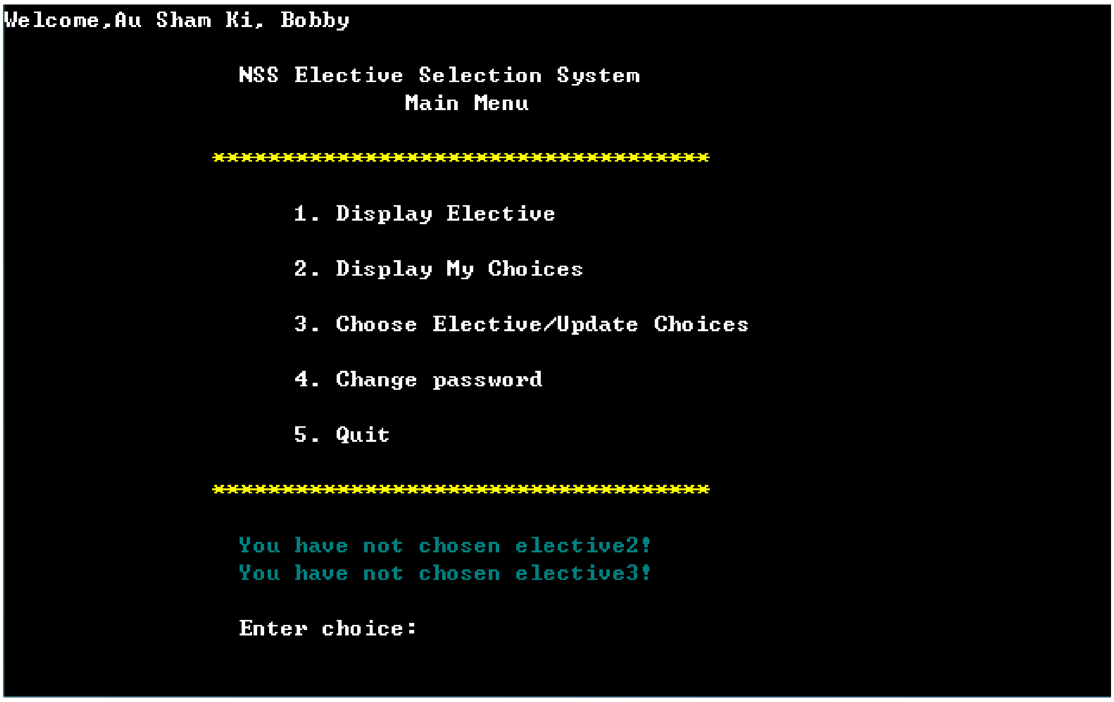
2) Login Page



```
      NSS Elective Selection System

--------------------------------------------
-                                          -
-                  LOGIN                   -
-                                          -
--------------------------------------------

      UserID : s001

    Password : ****
```

The above picture is the login page of the system. User needs to enter their ID and password to login their account. After that, they can use the system.

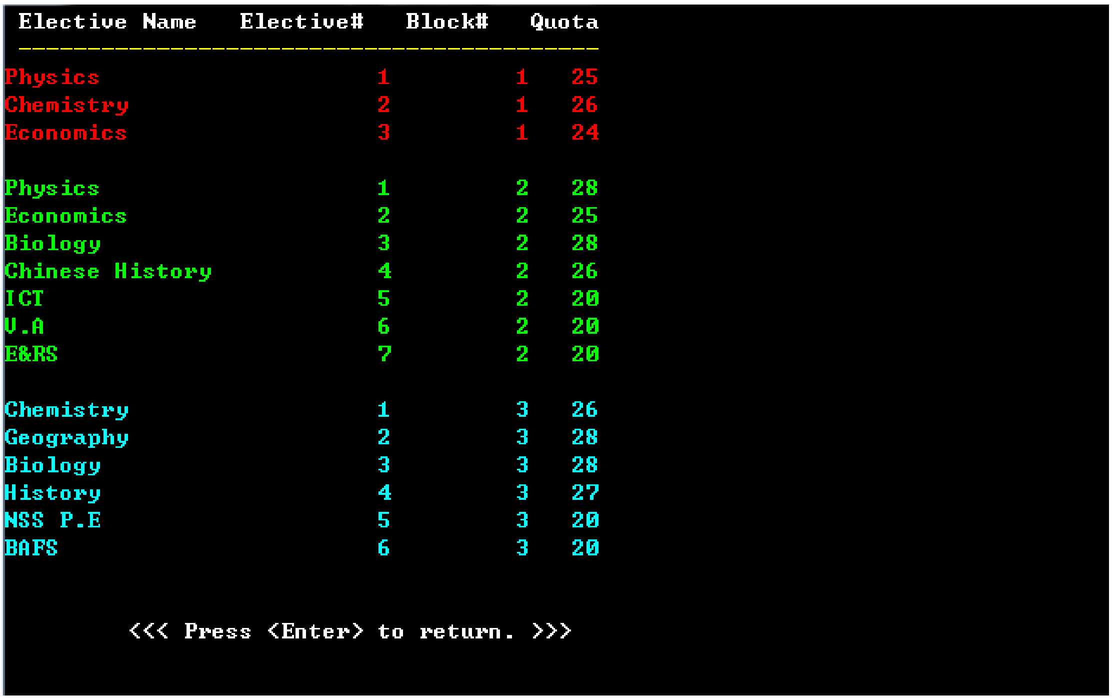
3) Students' Main Menu



```
Welcome,Au Sham Ki, Bobby

                 NSS Elective Selection System
                             Main Menu

               ************************************

                     1. Display Elective

                     2. Display My Choices

                     3. Choose Elective/Update Choices

                     4. Change password

                     5. Quit

               ************************************

                 You have not chosen elective2!
                 You have not chosen elective3!

                 Enter choice:
```

It is the students' main menu. Students can use the system in this page. In the left corner, it shows the name of the student to ensure the user is using their own account. At the below part of the page, it shows the elective which that students have not chosen. In the middle part, student can choose want they want to do by input number from 1-5.

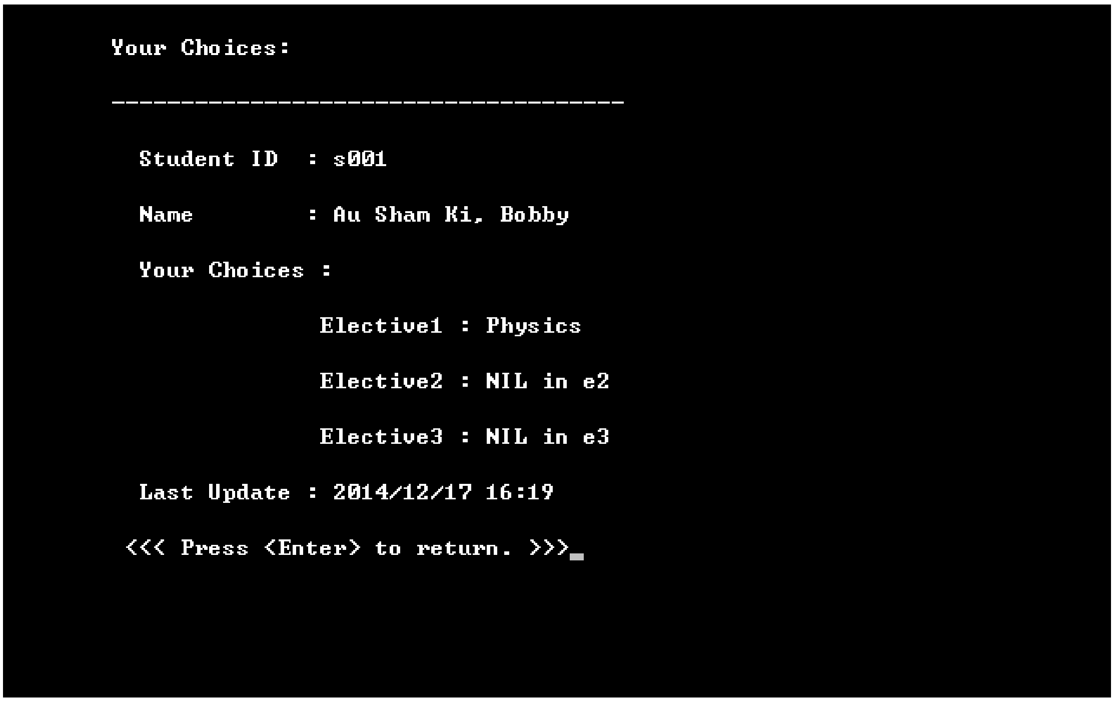
4) Display Elective



```
 Elective Name    Elective#    Block#    Quota
 ---------------------------------------------
Physics                    1          1    25
Chemistry                  2          1    26
Economics                  3          1    24

Physics                    1          2    28
Economics                  2          2    25
Biology                    3          2    28
Chinese History            4          2    26
ICT                        5          2    20
V.A                        6          2    20
E&RS                       7          2    20

Chemistry                  1          3    26
Geography                  2          3    28
Biology                    3          3    28
History                    4          3    27
NSS P.E                    5          3    20
BAFS                       6          3    20


         <<< Press <Enter> to return. >>>
```

The display elective function can let and students have a look about the choices of election in different block or the remaining quota of each subject before they select their elective subject. Press <Enter> will return to the main menu.

5) Display My Choice

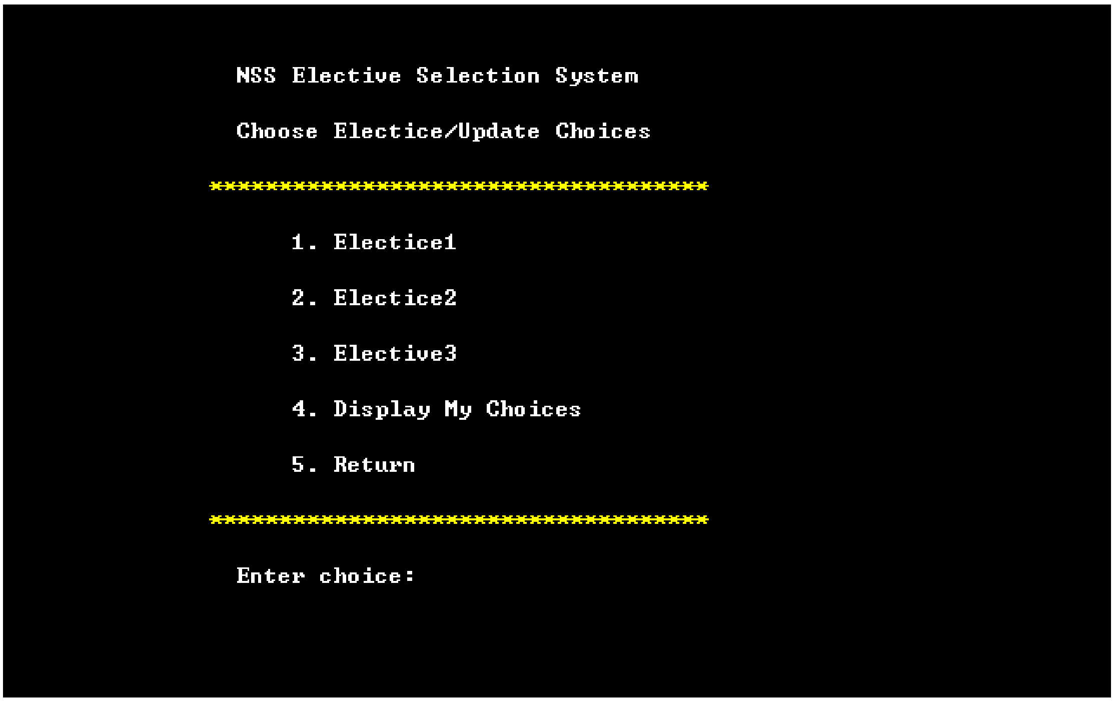


```
Your Choices:

-------------------------------------

  Student ID  : s001

  Name        : Au Sham Ki, Bobby

  Your Choices :

               Elective1 : Physics

               Elective2 : NIL in e2

               Elective3 : NIL in e3

  Last Update : 2014/12/17 16:19

 <<< Press <Enter> to return. >>>_
```

This function is used to let students to check what their choices are and they can also confirm their last update time to ensure others will not using their account. Besides, they can check their personal information here.

As for teachers, they can use this function to check a particular student's choices. They can find personal information of students in this function too.

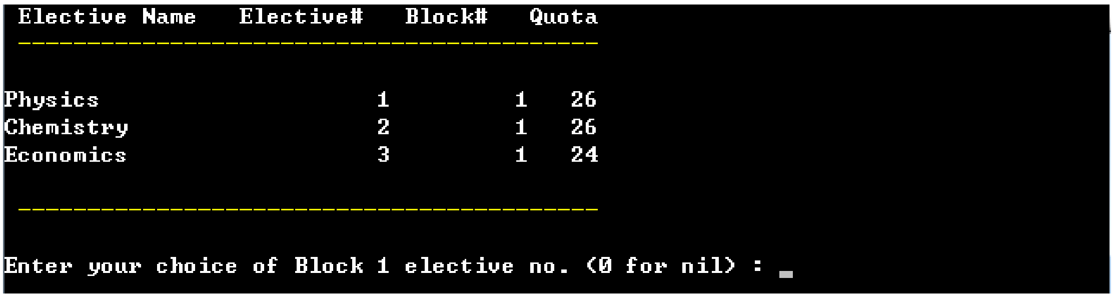
6) Choose Elective/Update Choices



```
  NSS Elective Selection System

  Choose Electice/Update Choices

************************************

      1. Electice1

      2. Electice2

      3. Elective3

      4. Display My Choices

      5. Return

************************************

  Enter choice:
```

The main function of the system, for students to choose elective subject they want. In this function, students can choose the subject block by block and they no need to choose subject form block1 first. They can choose the subjects in order which they want. Since some students want to choose elective subject form block2 or block3 first, this function is wonderful for them. They can also display their choices to have a reminder or return to the main menu.

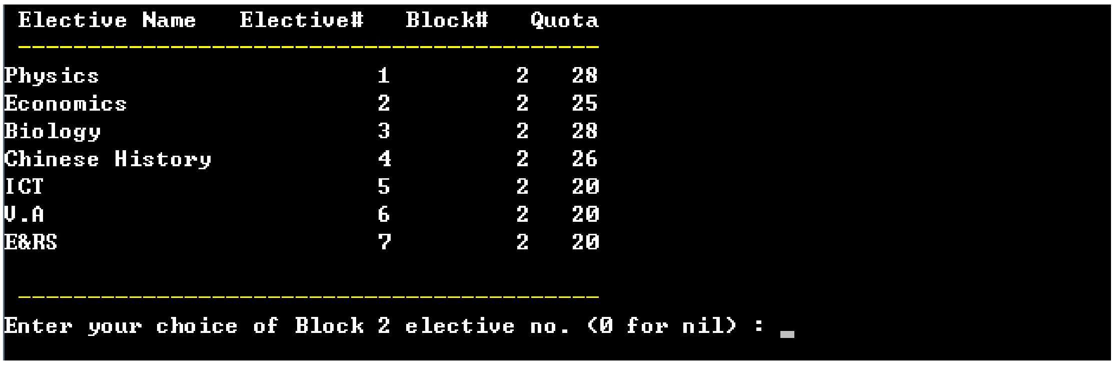
6.1) Choose Elective1



```
 Elective Name   Elective#   Block#   Quota
 ------------------------------------------


Physics                    1         1   26
Chemistry                  2         1   26
Economics                  3         1   24

 ------------------------------------------

Enter your choice of Block 1 elective no. (0 for nil) :
```

Students can choose subject they want in the elective block2 in this page by typing the elective number. After that, they will back to the' Display My Choice' and 'Choose Elective/Update Choices'.

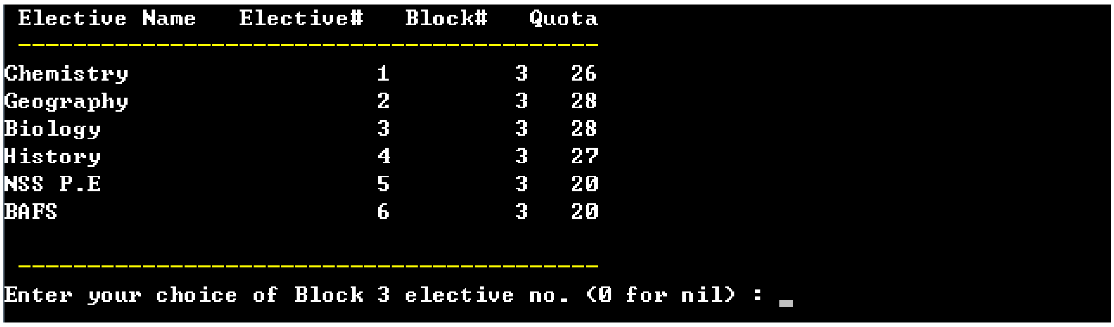
6.2) Choose Elective2



```
 Elective Name   Elective#   Block#   Quota
 ------------------------------------------
Physics                    1         2   28
Economics                  2         2   25
Biology                    3         2   28
Chinese History            4         2   26
ICT                        5         2   20
V.A                        6         2   20
E&RS                       7         2   20

 ------------------------------------------
Enter your choice of Block 2 elective no. (0 for nil) :
```

Students can choose subject they want in the elective block2 in this page by typing the elective number. After that, they will back to the' Display My Choice' and 'Choose Elective/Update Choices'.

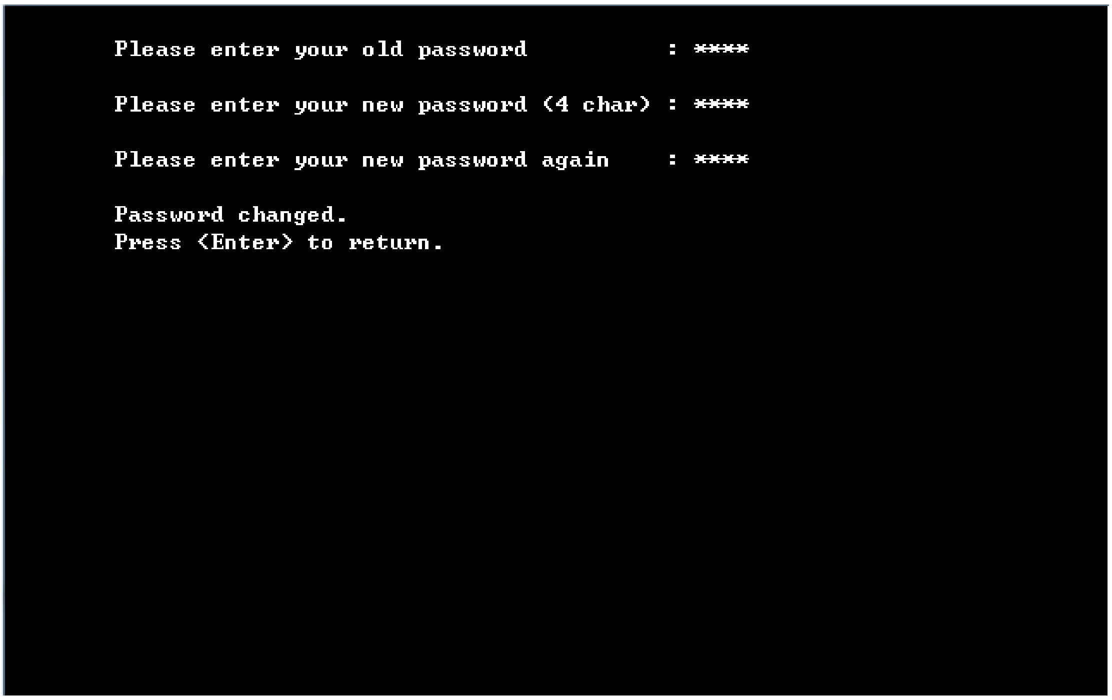
6.3) Choose Elective3



```
 Elective Name   Elective#   Block#   Quota
 ------------------------------------------
Chemistry                  1         3   26
Geography                  2         3   28
Biology                    3         3   28
History                    4         3   27
NSS P.E                    5         3   20
BAFS                       6         3   20

 ------------------------------------------
Enter your choice of Block 3 elective no. (0 for nil) :
```

Students can choose subject they want in the elective block3 in this page by typing the elective number. After that, they will back to the' Display My Choice' and 'Choose Elective/Update Choices'.

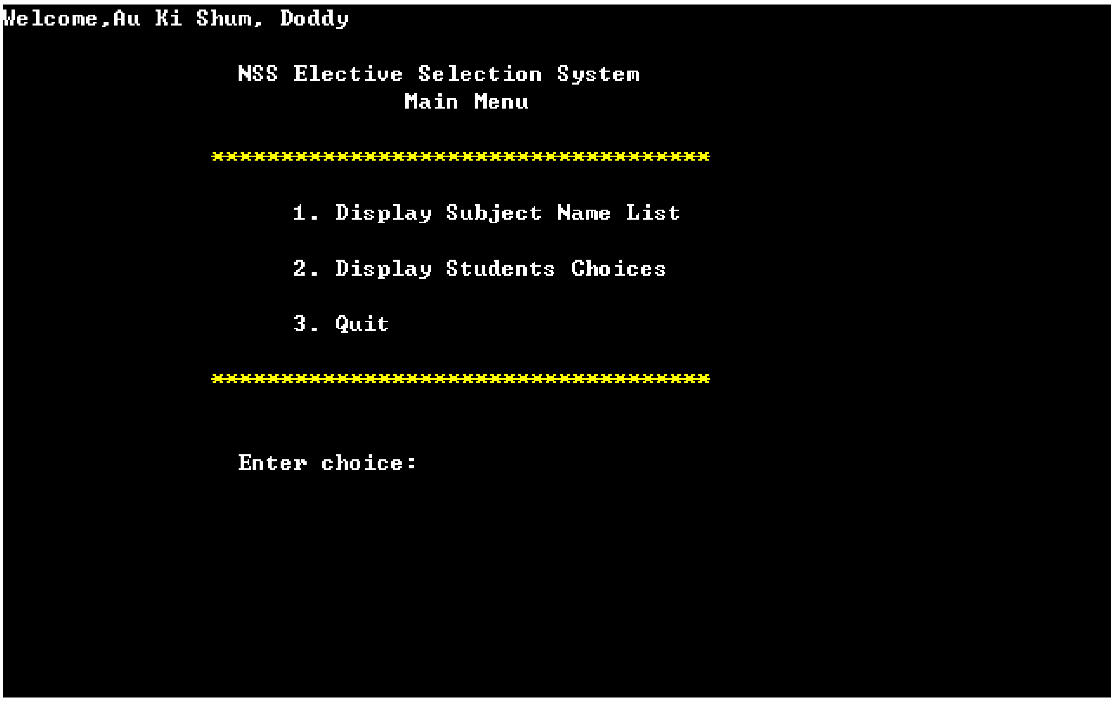
7) Choose Elective/Update Choices



```
Please enter your old password          : ****

Please enter your new password (4 char) : ****

Please enter your new password again    : ****

Password changed.
Press <Enter> to return.
```

To make the system more convenient, a password changing function is needed. Users can change their password so that they will not forget it easily. It is a user-friendly function.

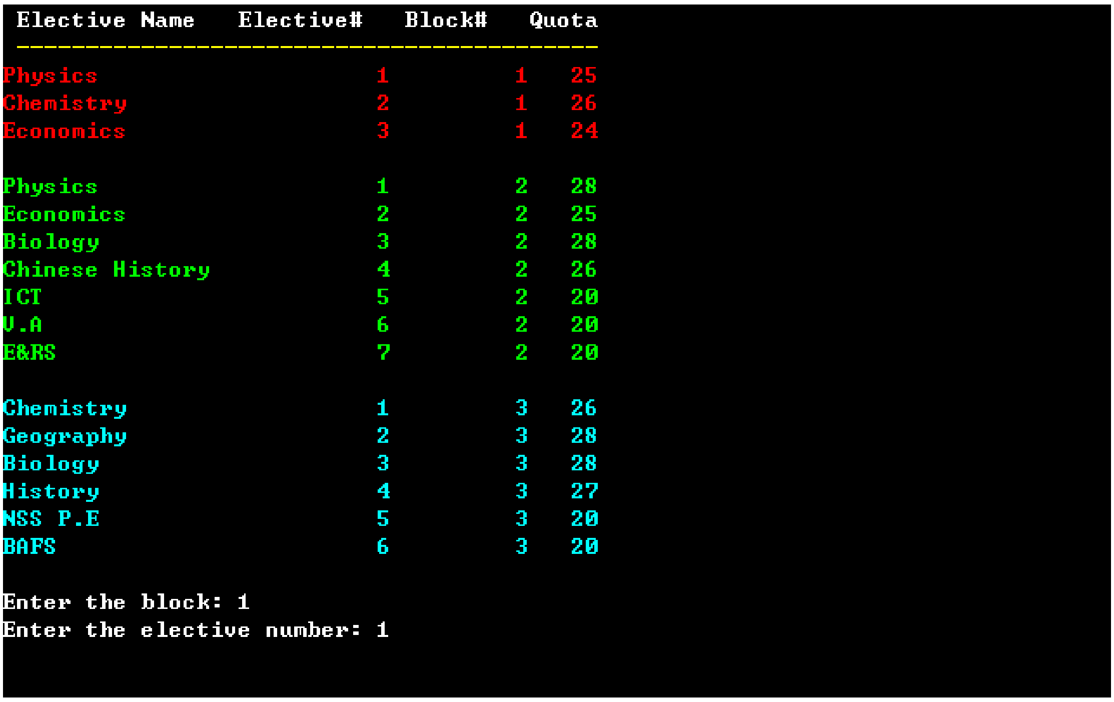
8) Teachers' Main Menu



```
Welcome,Au Ki Shum, Doddy

                 NSS Elective Selection System
                             Main Menu

               ************************************

                     1. Display Subject Name List

                     2. Display Students Choices

                     3. Quit

               ************************************


                 Enter choice:
```

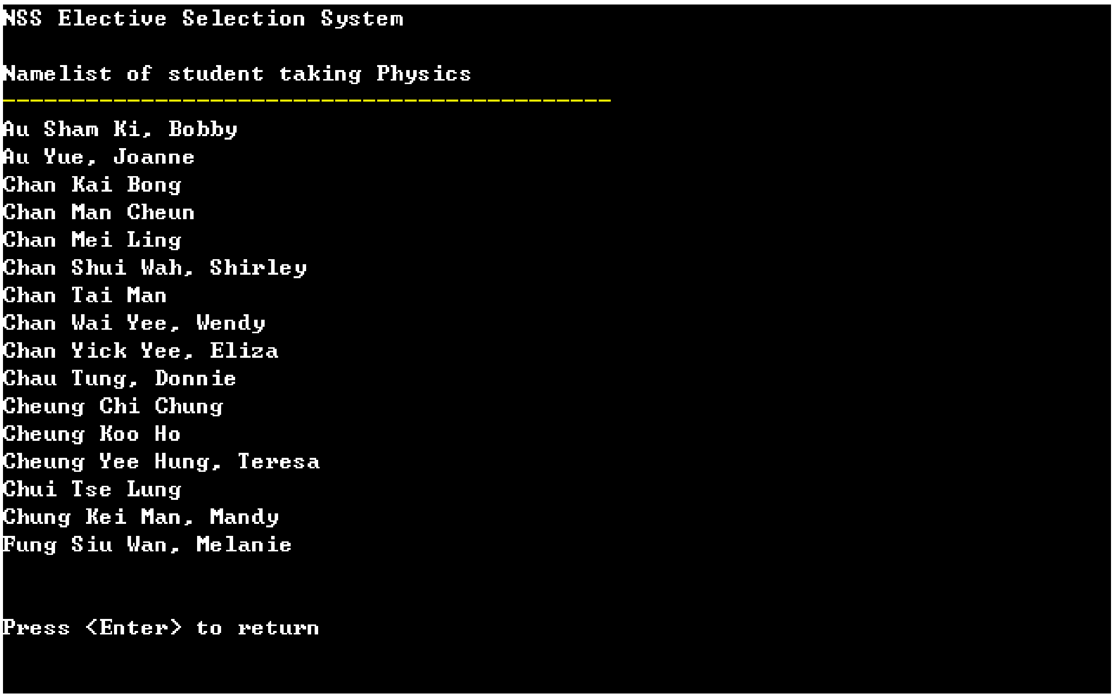
It is the teachers' main menu. Teachers can use the system in this page. In the left corner, it shows the name of the teacher to ensure the user is using their own account. In the middle part, teacher can choose want they want to do by input number from 1-3.

9) Display Subject Name List



```
 Elective Name   Elective#   Block#   Quota
 ------------------------------------------
Physics                    1         1   25
Chemistry                  2         1   26
Economics                  3         1   24

Physics                    1         2   28
Economics                  2         2   25
Biology                    3         2   28
Chinese History            4         2   26
ICT                        5         2   20
V.A                        6         2   20
E&RS                       7         2   20

Chemistry                  1         3   26
Geography                  2         3   28
Biology                    3         3   28
History                    4         3   27
NSS P.E                    5         3   20
BAFS                       6         3   20

Enter the block: 1
Enter the elective number: 1
```

Teacher can find the name list of each subject in this page. First, enter the block. Then enter the elective number. Finally, teacher can see the name list. And the below is the example of the page showing the name list.



```
NSS Elective Selection System

Namelist of student taking Physics
 -------------------------------------------
Au Sham Ki, Bobby
Au Yue, Joanne
Chan Kai Bong
Chan Man Cheun
Chan Mei Ling
Chan Shui Wah, Shirley
Chan Tai Man
Chan Wai Yee, Wendy
Chan Yick Yee, Eliza
Chau Tung, Donnie
Cheung Chi Chung
Cheung Koo Ho
Cheung Yee Hung, Teresa
Chui Tse Lung
Chung Kei Man, Mandy
Fung Siu Wan, Melanie


Press <Enter> to return
```

Chapter 4 Testing & Evaluation

4.1 Brief Description

As every program is not perfect at all, testing and evaluation is a must for all program. There will be some loopholes and bugs in the program such as logical or run-time error will be also included. With this reason, this part is used to check can the program run properly and meet the aim.

To begin with, I have to check whether the program can meet all the structure from the design part. I have to look at the function which in design part and test can the program can achieve the mission and has all these function.

After that, as one of the main reasons in this chapter is to make the program more user-friendly. Some debug works are important to make the program become more user-friendly. So debugging will be the next step after testing.

Besides, although some debugging had been done by me, there may still contain bugs that I cannot find out at all, so what I am going to do is let others to try and use my program. So that bugs may find out while using.

Last but not least, I will modify the program to make it perfect for users to use.

4.2 Testing and Evaluation Plan

The program will be tested and evaluated under the plan below.

I will test the program first at all. It is important for me to test it first because there may be bugs that users cannot discover easily. For all the functions, at least five times testing should have been done. As there will be a handful of possible bugs, both correct and incorrect data should be tried to ensure the accuracy of the program. After finding all the bugs, debug is required but at the same time.

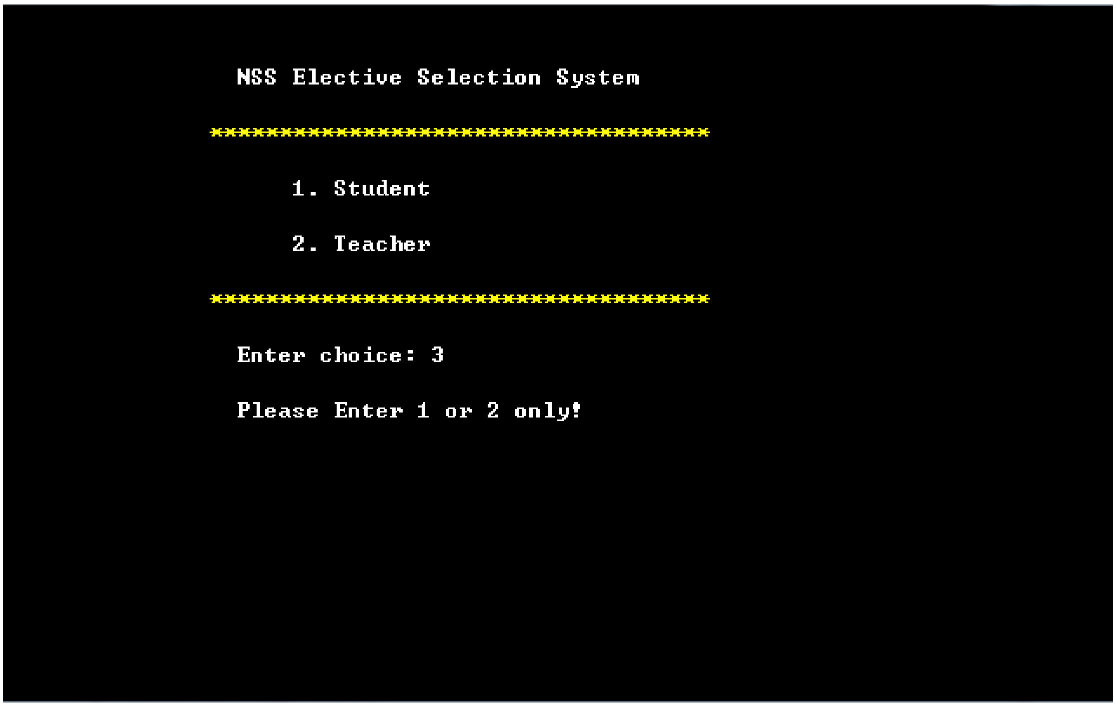
My own test will be focus on logic and run-time error. Since all the syntax error were debugged after finished the program. I have to check logic and run-time error this time. Run-time error is quite brothering. Since it is harder to debug, as Pascal do not warn you where is the error. As for logic error, it is the hardest error for debugging. Since neither Pascal nor the program will show errors to you. Only if enter the testing data can find the errors. So I will focus on these two errors

However, in view of a programmer, I may not find out all the error as people will be subjective on works done by them. So, I will let 5 people who do not have much knowledge in ICT and 5 people who have studied ICT to try my program and see whether any bugs can be exist.

I will try to modify the program according to the reported bugs and suggestions. The new version of program after making modifications will be shown in Appendix 1.

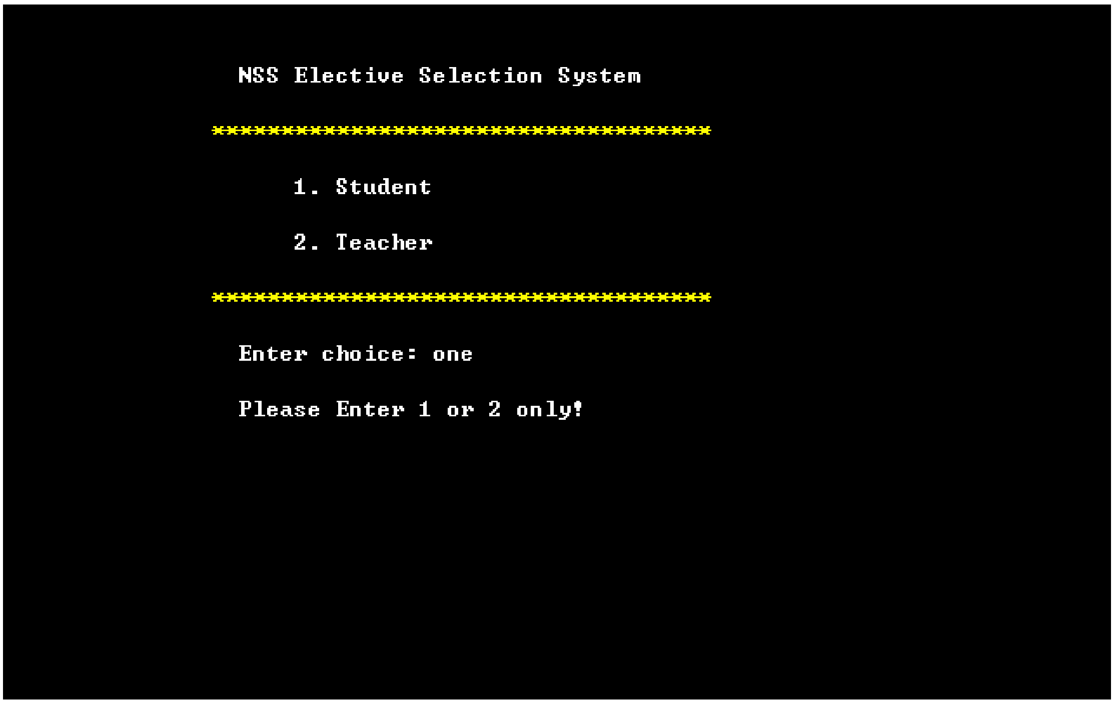
4.3 Internal Testing

Below is the testing record after the final debug for the program:



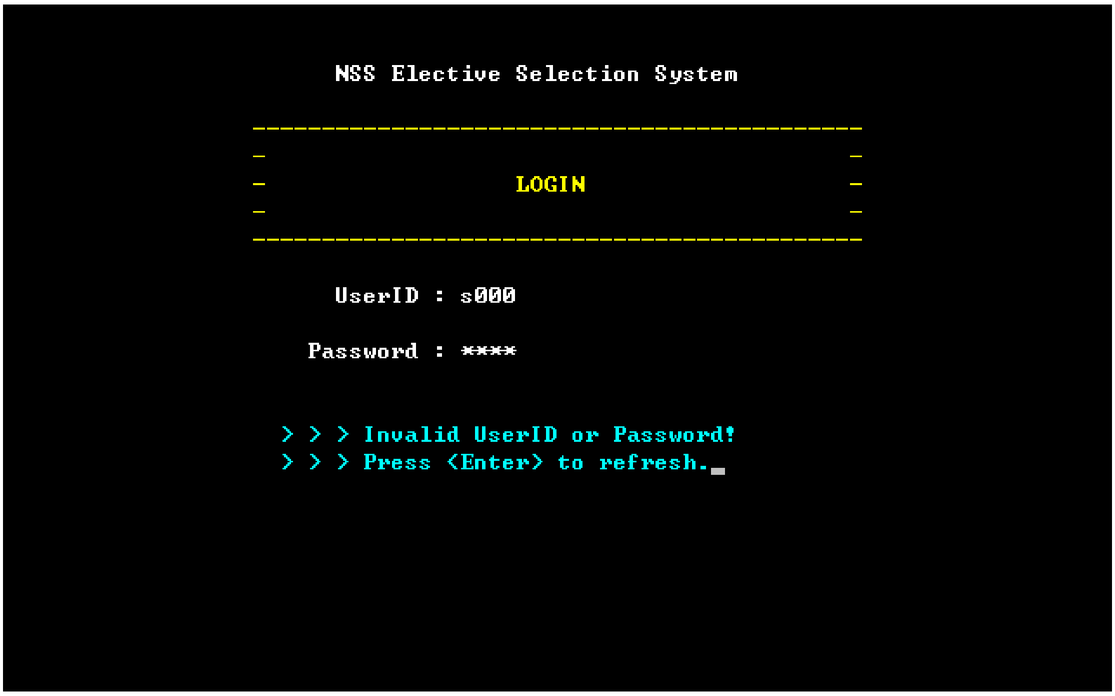
```
      NSS Elective Selection System

    **************************************

            1. Student

            2. Teacher

    **************************************

      Enter choice: 3

      Please Enter 1 or 2 only!
```

Purpose:	To check if the program can detect the invalid input
Input:	Choice which is out of the range
Expected Output:	Please Enter 1 or 2 only!
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



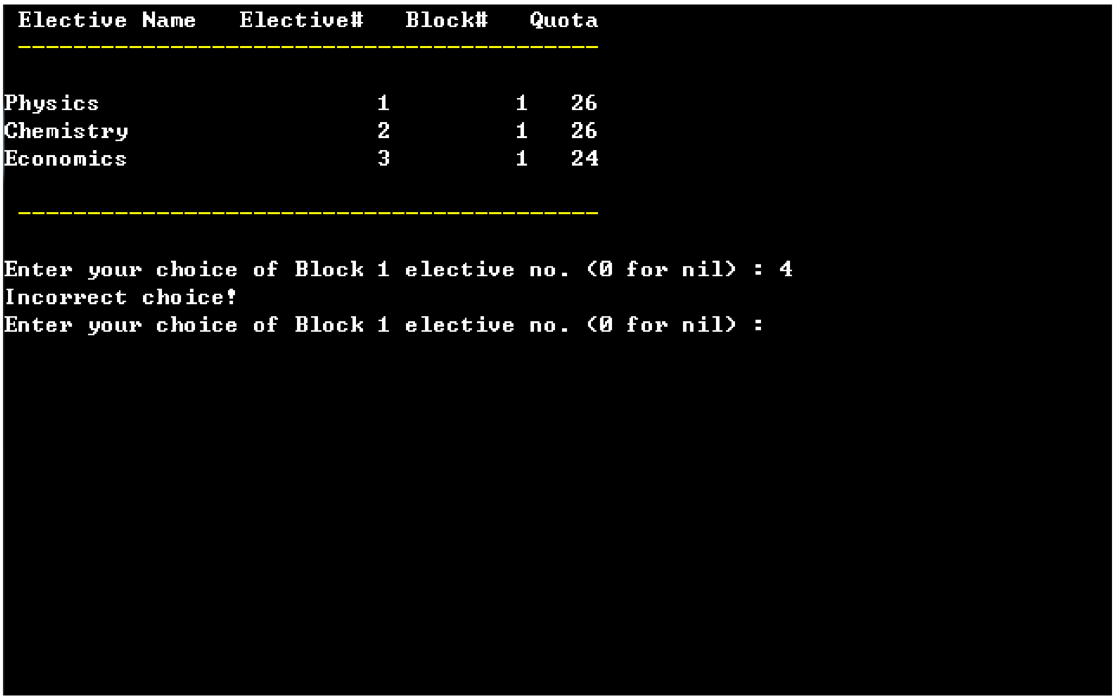
```
    NSS Elective Selection System

  ************************************

        1. Student

        2. Teacher

  ************************************

    Enter choice: one

    Please Enter 1 or 2 only!
```

Purpose:	To check if the program can detect the invalid input
Input:	Choice which is in another type
Expected Output:	Please Enter 1 or 2 only!
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



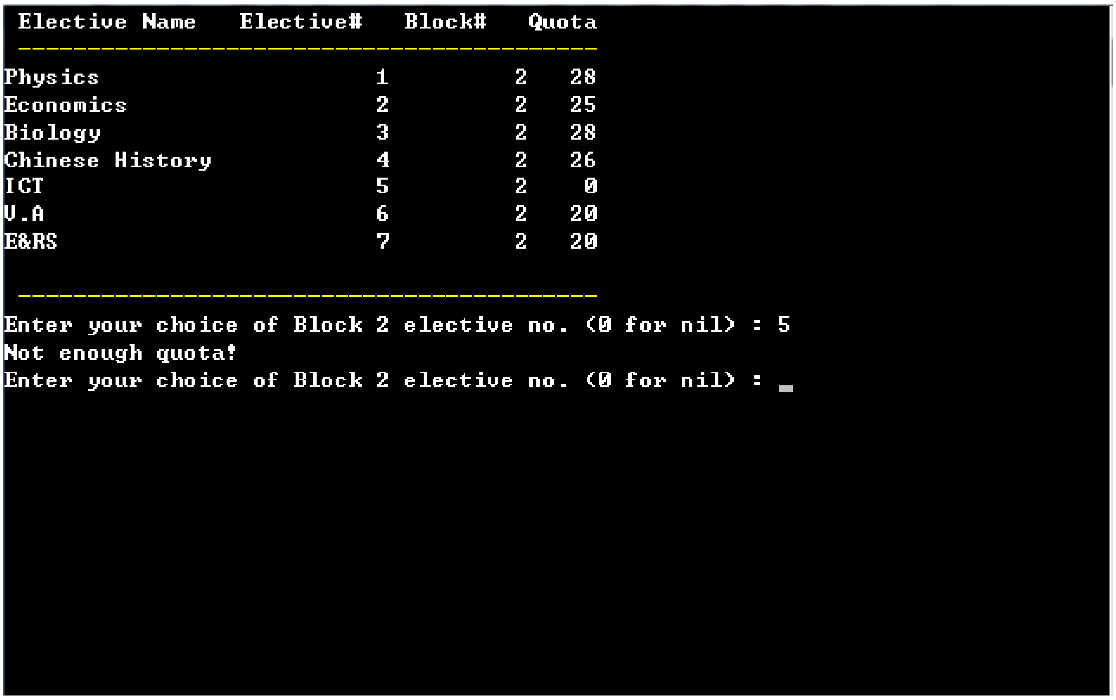
```
                NSS Elective Selection System

          --------------------------------------------
          -                                          -
          -                   LOGIN                  -
          -                                          -
          --------------------------------------------

                UserID : s000

              Password : ****

            > > > Invalid UserID or Password!
            > > > Press <Enter> to refresh._
```

Purpose:	To check if the program can detect the invalid input
Input:	Incorrect User ID or Password
Expected Output:	> > > Invalid UserID or Password! > > > Press <Enter> to refresh.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



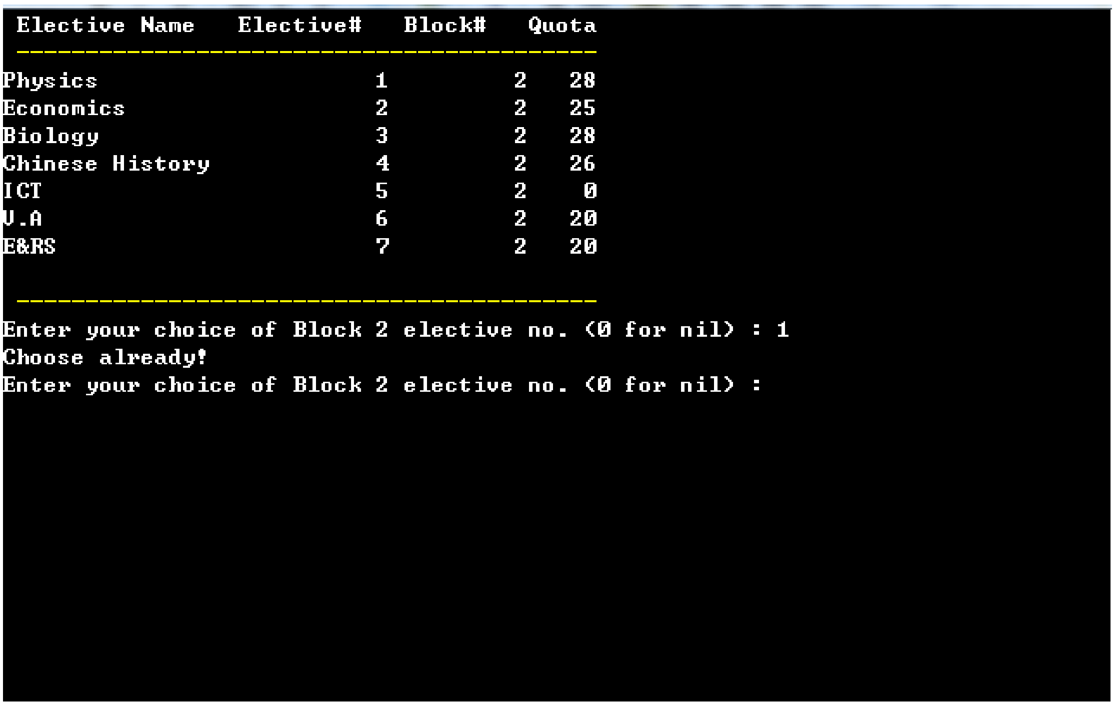
```
 Elective Name   Elective#   Block#   Quota
 -------------------------------------------

Physics                    1         1   26
Chemistry                  2         1   26
Economics                  3         1   24

 -------------------------------------------

Enter your choice of Block 1 elective no. (0 for nil) : 4
Incorrect choice!
Enter your choice of Block 1 elective no. (0 for nil) :
```

Purpose:	To check if the program can detect the invalid input
Input:	Choice which is out of the range
Expected Output:	Incorrect choice! Enter your choice of Block 1 elective no. <0 for nil> :
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



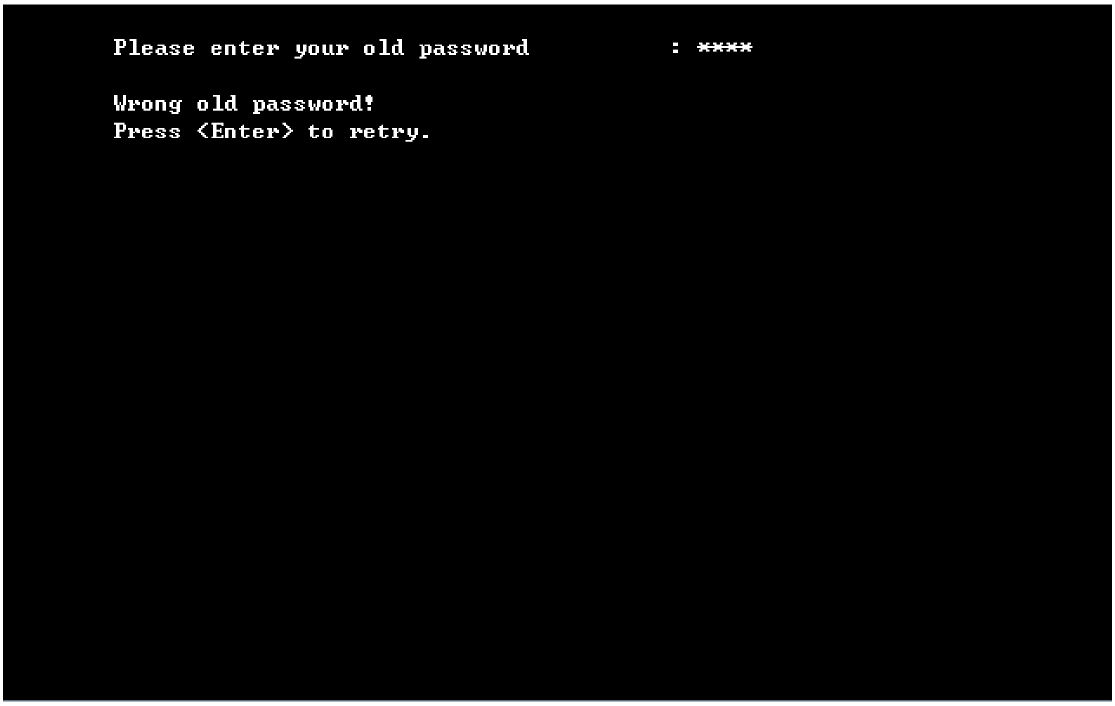
```
 Elective Name    Elective#   Block#    Quota
 ------------------------------------------
Physics                      1         2    28
Economics                    2         2    25
Biology                      3         2    28
Chinese History              4         2    26
ICT                          5         2     0
V.A                          6         2    20
E&RS                         7         2    20

 ------------------------------------------
Enter your choice of Block 2 elective no. (0 for nil) : 5
Not enough quota!
Enter your choice of Block 2 elective no. (0 for nil) : _
```

Purpose:	To check if the program can detect the invalid input
Input:	Choice which do not have enough quota
Expected Output:	Not enough quota! Enter your choice of Block 1 elective no. <0 for nil> :
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



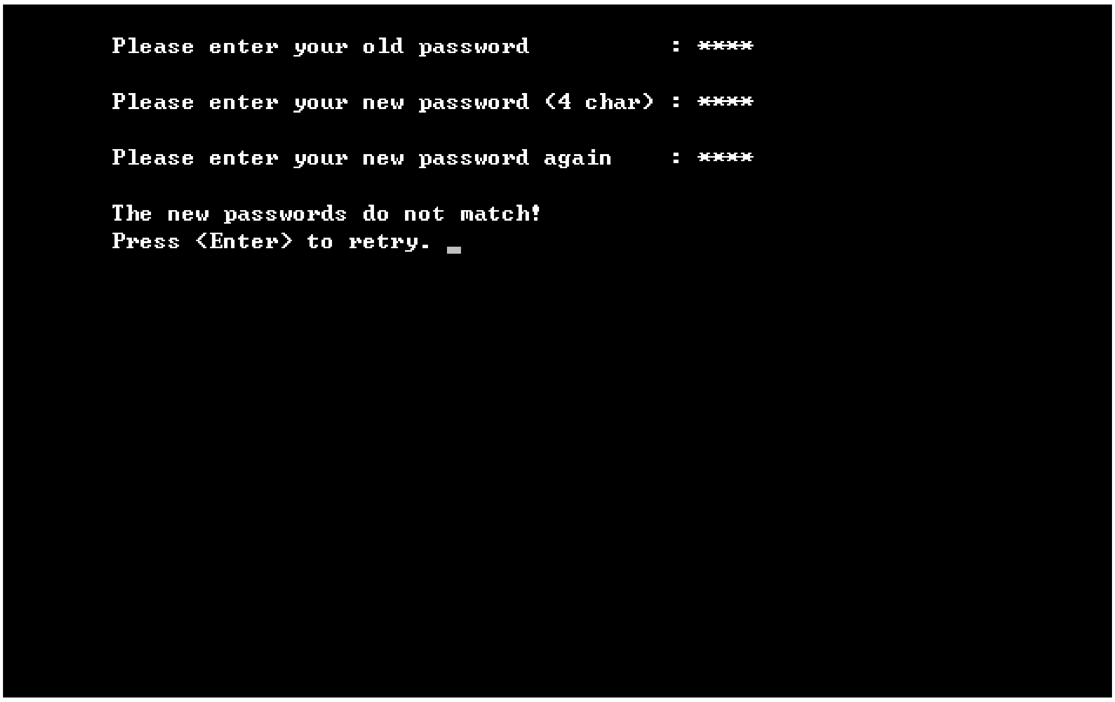
```
 Elective Name    Elective#    Block#    Quota
 ------------------------------------------
Physics                    1         2    28
Economics                  2         2    25
Biology                    3         2    28
Chinese History            4         2    26
ICT                        5         2     0
V.A                        6         2    20
E&RS                       7         2    20

 ------------------------------------------
Enter your choice of Block 2 elective no. <0 for nil> : 1
Choose already!
Enter your choice of Block 2 elective no. <0 for nil> :
```

Purpose:	To check if the program can detect the invalid input
Input:	Choice which choose already in other block
Expected Output:	Choose already! Enter your choice of Block 1 elective no. <0 for nil> :
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



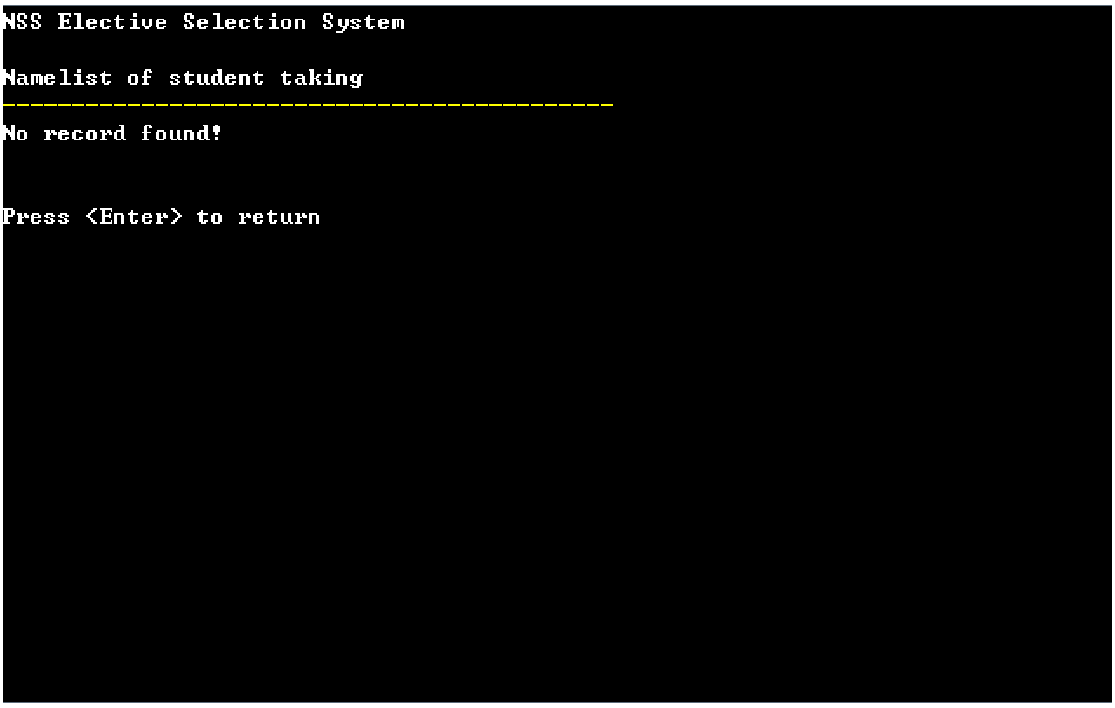
```
        Please enter your old password          : ****

        Wrong old password!
        Press <Enter> to retry.
```

Purpose:	To check if the program can detect the invalid input
Input:	Incorrect old Password
Expected Output:	Wrong old password! Press <Enter> to retry.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



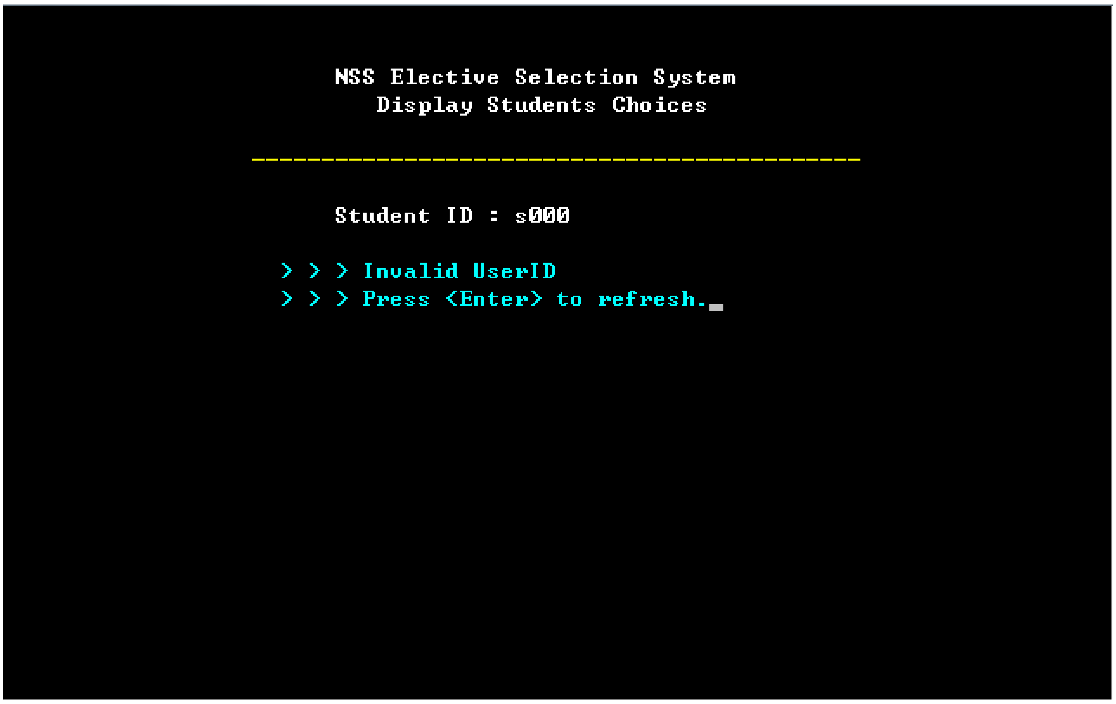
```
Please enter your old password          : ****

Please enter your new password (4 char) : ****

Please enter your new password again    : ****

The new passwords do not match!
Press <Enter> to retry. _
```

Purpose:	To check if the program can detect the invalid input
Input:	New password which do not match
Expected Output:	The new passwords do not match! Press <Enter> to retry.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



```
NSS Elective Selection System

Namelist of student taking
-------------------------------------------
No record found!


Press <Enter> to return
```

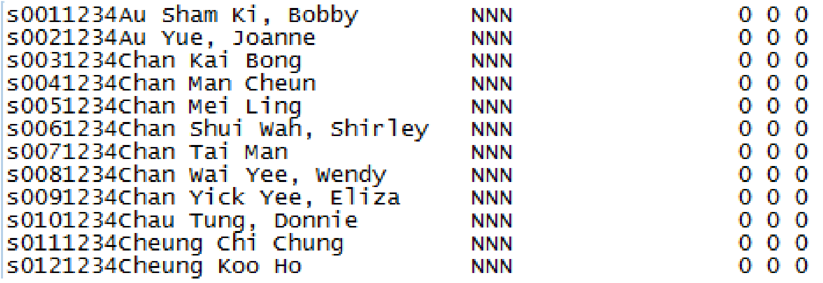
Purpose:	To check if the program can display correct data
Input:	Subject which no one choose
Expected Output:	('No record found!') ('<Press <Enter> to return>')
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



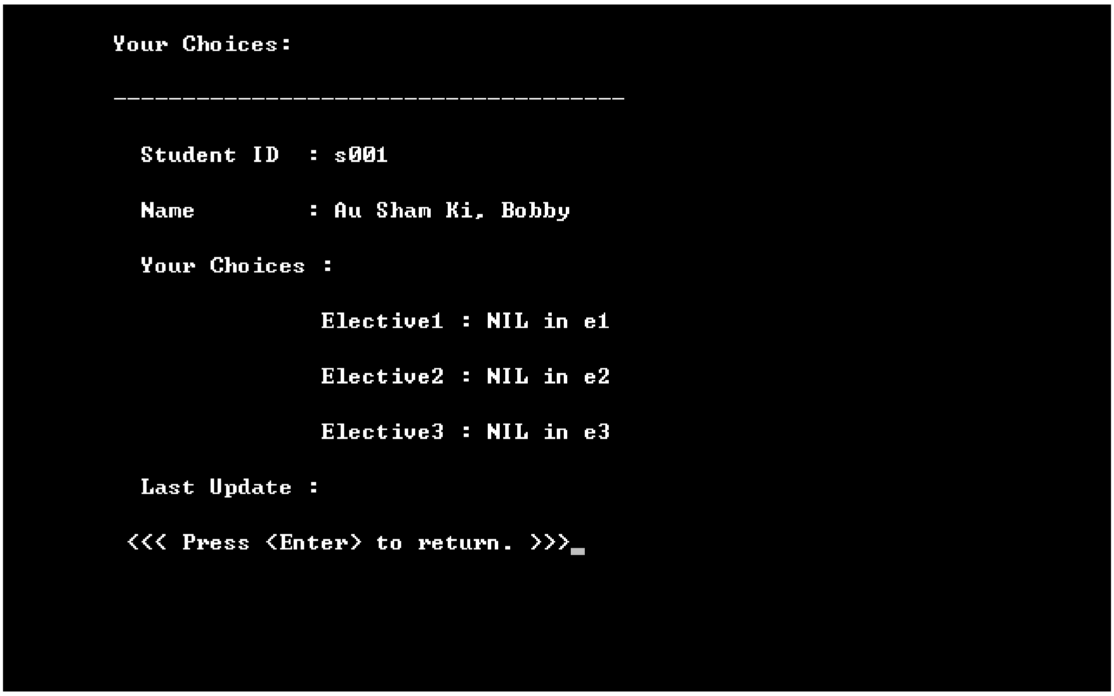
```
                NSS Elective Selection System
                   Display Students Choices

         --------------------------------------------

                Student ID : s000

              > > > Invalid UserID
              > > > Press <Enter> to refresh._
```

Purpose:	To check if the program can detect the invalid input
Input:	Incorrect User ID
Expected Output:	> > > Invalid UserID > > > Press <Enter> to refresh.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

Before

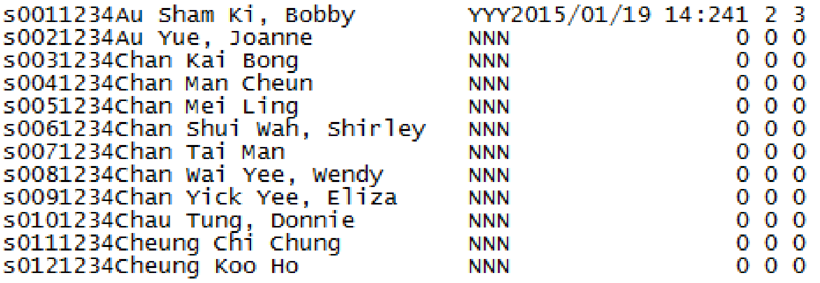


```
s0011234Au Sham Ki, Bobby        NNN                 0 0 0
s0021234Au Yue, Joanne           NNN                 0 0 0
s0031234Chan Kai Bong            NNN                 0 0 0
s0041234Chan Man Cheun           NNN                 0 0 0
s0051234Chan Mei Ling            NNN                 0 0 0
s0061234Chan Shui Wah, Shirley   NNN                 0 0 0
s0071234Chan Tai Man             NNN                 0 0 0
s0081234Chan Wai Yee, Wendy      NNN                 0 0 0
s0091234Chan Yick Yee, Eliza     NNN                 0 0 0
s0101234Chau Tung, Donnie        NNN                 0 0 0
s0111234Cheung Chi Chung         NNN                 0 0 0
s0121234Cheung Koo Ho            NNN                 0 0 0
```

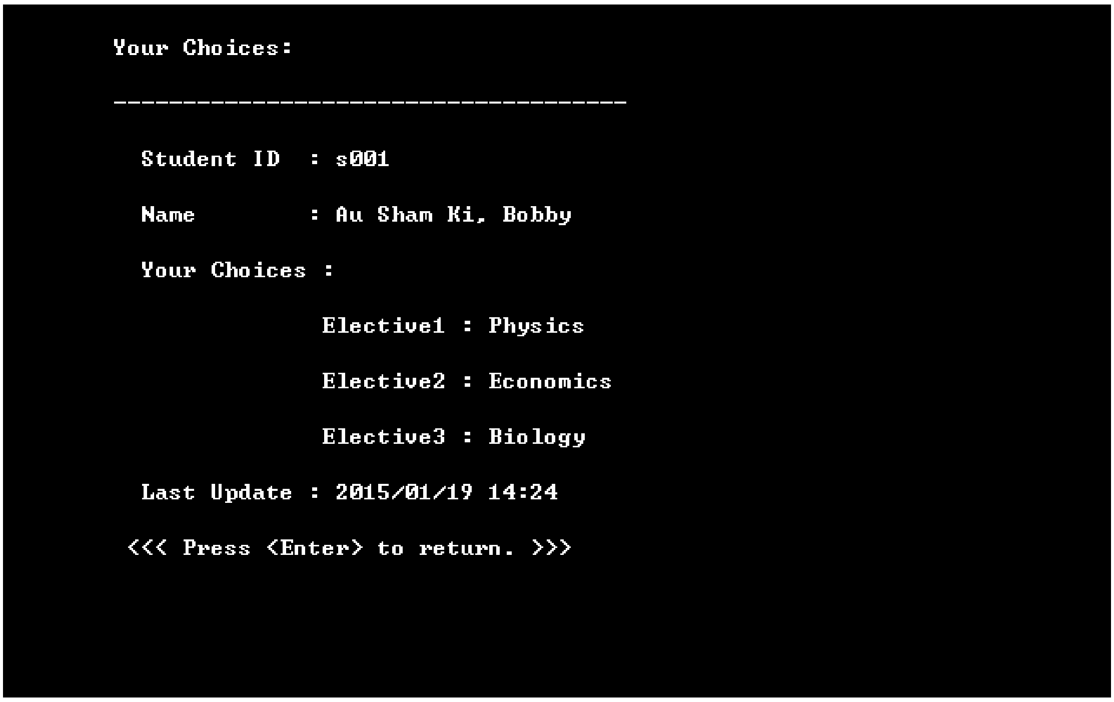


```
       Your Choices:

       -------------------------------------

         Student ID  : s001

         Name        : Au Sham Ki, Bobby

         Your Choices :

                      Elective1 : NIL in e1

                      Elective2 : NIL in e2

                      Elective3 : NIL in e3

         Last Update :

        <<< Press <Enter> to return. >>>_
```

After



```
s0011234Au Sham Ki, Bobby        YYY2015/01/19 14:241 2 3
s0021234Au Yue, Joanne           NNN                 0 0 0
s0031234Chan Kai Bong            NNN                 0 0 0
s0041234Chan Man Cheun           NNN                 0 0 0
s0051234Chan Mei Ling            NNN                 0 0 0
s0061234Chan Shui Wah, Shirley   NNN                 0 0 0
s0071234Chan Tai Man             NNN                 0 0 0
s0081234Chan Wai Yee, Wendy      NNN                 0 0 0
s0091234Chan Yick Yee, Eliza     NNN                 0 0 0
s0101234Chau Tung, Donnie        NNN                 0 0 0
s0111234Cheung Chi Chung         NNN                 0 0 0
s0121234Cheung Koo Ho            NNN                 0 0 0
```



```
       Your Choices:

       -------------------------------------

         Student ID  : s001

         Name        : Au Sham Ki, Bobby

         Your Choices :

                      Elective1 : Physics

                      Elective2 : Economics

                      Elective3 : Biology

         Last Update : 2015/01/19 14:24

        <<< Press <Enter> to return. >>>
```

Purpose:	To check if the program can display correct output and rewrite text file
Input:	Choices of 3 electives
Expected Output:	Elective 1 : Physics Elective 2 : Economics Elective3 : Biology Last Update : 2015/01/19 14:24 And also rewrite the first line in 'students.txt' s0011234Au Sham Ki, Bobby YYY2015/01/19 14:361 2 3
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

4.4 Self-Evaluation

As the restriction of a NSS student-subject allocation system is quite large, everyone can only do similar works, so I am not only focusing on the process when choosing the subject, but also on the realistic of it. I pay more effort on those functions which can make the program become more user-friendly can provide more function such as let teachers use the system too.

The Teachers System is the unique part of my program. It can let teachers can have some analysis on the students' choice of the elective. So, they can have some changes on the elective next year after concern about the more famous subject. For example, teacher can host one more class on the subject which is the fastest full.

The most important thing is that my program can even provide function to the teachers, it is a function that creates by me. So, it can provide more function but not just a normal subject allocation system.

4.5 External Testing and Evaluation

I have invited 10 of my classmates to use my program so that while they are playing on it, they may discover bugs and loopholes in my program which I cannot find out in the previous sections.

To make them clear on what they should do when they are playing on the program, I have set up a simple testing form for them to follow by. In here, I want to thank all people who are helping me on finding out the problems in my program.

After collecting all the data and opinion from the classmates, I have summarized all of them and do improvement.

Thanks for helping me to play the program and find out whether there are bugs in my program or not.

Instruction:

Try the system and functions.

Report on Bugs:

No.	Description of errors

Program Evaluation:

Please answer the following questions by circling the numbers on the right hand side.

				Rating		
		Agree		Average		Disagree
1.	The appearance is good.	5	4	3	2	1
2.	The design of the input screen is good	5	4	3	2	1
3.	The functions are useful	5	4	3	2	1
4	The game is too difficult	5	4	3	2	1

Which function do you like the most?

____

_

What other functions should be provided by the program?

____

_

Other Suggestions

____

After summarizing the opinion, the result is below.

		Rating
1.	The appearance is good.	4.8
2.	The design of the input screen is good	4.4
3.	The functions are useful	4
4	The program is user-friendly	5

From the above result, I found out that people almost think that the program is user-friendly and the appearance is good.

However, the functions are not very useful. It is because the restriction of a NSS student-subject allocation system is quite large, everyone can only do similar works. So I am focusing on the unique function and make the program more user- friendly.

Chapter 5 Conclusion & Discussion

Pros and cons of my Program

In this program, I have paid a lot of effort in order to make the program more perfect. Planning, designing, implementing is some part that I work hard most. However, I still cannot make the program zero mistake even tried my best. But I will do it at my best. Moreover, I am sure that the program is well in quality.

My program is a NSS student-subject allocation system for secondary school students. I have well-designed the appearance of the program as the colorful and many reminder of the program can ensure students will input data carefully.

The look is only the surface of my program, but more importantly is that the content of the program is wonderful too. The program consists two parts for different users, teachers and students, it is different than other subject allocation system which only let student to choose their wanted subject but do not have other function.

In the student part, some function such as display subject, display their choices are included. So students can not only use the program to choose elective subject, but they can also have other information to make the program more user-friendly.

In the teacher part, the most unique function is the display name list function. The aim of this function is to let teacher have more detail on the information regarding the choices of the students. So, they can have a look before the name list is confirmed or they can arrange the choices of the elective subject next year.

And I think that the most important feature of my program is the teacher part. Not every subject allocation system allows more than one type of user. I think that a rounded system should contain different type of user, not just only for students or teachers.

Unfortunately, the program still has shortcomings since every program are not perfect. The information of the user that display on the program may not enough.

Also, it cannot ensure other people may use the program but do not use their own account. So anyone get someone’s ID and password can easily use the program and change the password.

But consider all the system, the program can work properly and smoothly which can make the users to use it with a high efficiency. Also, it can already for a secondary school to use for the NSS student-subject allocation system that it is a user-friendly system.

Finally, I have confidence that the program will be useful for the schools and all secondary students can be benefit by the system.

Future Improvement

In future, it is hoped that the program can be updated each year, such as the subject students can choose can be updated, it is just an easy step by changing the text in the text files.

Moreover, after users use the program. They can have some feedback to me and I will try my best the make the program more user-friendly and may add more functions.

I will start working on more functions focus on the teacher part of program so that teachers can have more detail about the choices of students. So they can have better arrangement of their policy. It can benefit both students and teachers.

Last but not least, I will keep on receiving more comment of the program so that I can have better improvement.

Self-Reflection

I have learnt a handful of skill and knowledge when I am working on the program. At first, I thought that it was a simple task for me. However, it is not as easy as I think, there were a lot of bugs in the program which challenge me and it is one of the biggest difficulties. But I learnt to ask others opinions and learn from others and teachers especially my ICT teacher.

Also, the program codes that I have learnt are only some of the codes that I learn in the lesson, so I surfed the internet to search for more wonderful and skillful code to make my program more convenient for users to use and more user-friendly.

Time management is also one of the problems. Because there is time limit for the design of the system, I need to work on it every day to make the program have a well design. So, in the report part, the time is running short and I need to work very rush and more carefully. I hope I can have better time management next time.

Chapter 6 Reference and Acknowledgement

Appendix 1: Program Code

```
program ChooseElective;

uses crt, sysutils;

const
  max_stud = 100;
  max_teac = 30;
  max_elective = 10;


type
{record for storing students' information }
stud_record = record
               studid : string[4];
               studpw : string[4];
               studname : string[25];
               chooseelective1 : char;
               chooseelective2 : char;
               chooseelective3 : char;
               lastupdate : string[16];
               elective1 : integer;
               elective2 : integer;
               elective3 : integer;
             end;


{record for storing teachers' information }
teac_record = record
               teacid : string[4];
               teacpw : string[4];
               teacname : string[25];
             end;


{record for storing elective1's information }
elective1_record = record
               e1_name :  string[16]  ;
               e1_num : integer ;
               e1_block : integer ;
               e1_quota : integer ;
             end;


{record for storing elective2's information }
elective2_record = record
               e2_name :  string[16]  ;
               e2_num : integer ;
               e2_block : integer ;
               e2_quota : integer ;
             end;


{record for storing elective3's information }
elective3_record = record
               e3_name :  string[16]  ;
```

```
                  e3_num : integer ;
                  e3_block : integer ;
                  e3_quota : integer ;
                end;



var
  student : array[1..max_stud] of stud_record;
  teacher : array[1..max_teac] of teac_record;
  elective1 : array[0..max_elective] of elective1_record;
  elective2 : array[0..max_elective] of elective2_record;
  elective3 : array[0..max_elective] of elective3_record;
  num_stud, stud_index, num_teac, teac_index, num_elective1,
num_elective2, num_elective3 : integer;



procedure read_students;
var
  i : integer;
  f : text;
begin
  assign(f, 'students.txt');
  reset(f);
  i := 0;
  while not eof(f) do
    begin
      i := i + 1;
      with student[i] do
        begin
        readln(f, studid, studpw, studname, chooseelective1,
chooseelective2, chooseelective3, lastupdate, elective1,
elective2, elective3);
        end;
    end;
  num_stud := i;
  close(f)
```

```
end;

procedure store_students;
var  i : integer;
     f : text;
begin
  assign(f, 'students.txt');
  rewrite(f);
  for i := 1 to num_stud do
    begin
      with student[i] do
        begin
          writeln(f,     studid,     studpw,     studname,
chooseelective1,    chooseelective2,    chooseelective3,
lastupdate, elective1, ' ', elective2, ' ', elective3)
        end;
    end;
  close(f)
end;


procedure read_teachers;
var
  i : integer;
  f : text;
begin
  assign(f, 'teachers.txt');
  reset(f);
  i := 0;
  while not eof(f) do
    begin
      i := i + 1;
      with teacher[i] do
        begin
        readln(f, teacid, teacpw, teacname);
        end;
    end;
  num_teac := i;
  close(f)
end;



procedure read_elective1;
var
  i : integer;
  f : text;
begin
  assign(f, 'elective1.txt');
  reset(f);
  i := -1;
  while not eof(f) do
    begin
      i := i + 1;
      with elective1[i] do
        begin
          readln(f, e1_name, e1_num, e1_block ,e1_quota);
        end;
    end;
  num_elective1 := i;
  close(f)
end;



procedure store_elective1;
var  i : integer;
     f : text;
begin
  assign(f, 'elective1.txt');
  rewrite(f);
  for i := 0 to num_elective1 do
    begin
      with elective1[i] do
        begin
          writeln(f,  e1_name,  e1_num,  '  '  ,  e1_block,  '
' ,e1_quota);
        end;
    end;
  close(f)
end;

procedure read_elective2;
var
  i : integer;
  f : text;
begin
  assign(f, 'elective2.txt');
  reset(f);
  i := -1;
  while not eof(f) do
    begin
      i := i + 1;
      with elective2[i] do
        begin
          readln(f, e2_name, e2_num, e2_block ,e2_quota);
        end;
    end;
  num_elective2 := i;
  close(f)
end;



procedure store_elective2;
var  i : integer;
     f : text;
begin
  assign(f, 'elective2.txt');
  rewrite(f);
  for i := 0 to num_elective2 do
    begin
      with elective2[i] do
        begin
```

```
          writeln(f,  e2_name,  e2_num,  '  '  ,  e2_block,  '
' ,e2_quota);
        end;
    end;
  close(f)
end;

procedure read_elective3;
var
  i : integer;
  f : text;
begin
  assign(f, 'elective3.txt');
  reset(f);
  i := -1;
  while not eof(f) do
    begin
      i := i + 1;
      with elective3[i] do
        begin
          readln(f, e3_name, e3_num, e3_block ,e3_quota);
        end;
    end;
  num_elective3 := i;
  close(f)
end;



procedure store_elective3;
var  i : integer;
     f : text;
begin
  assign(f, 'elective3.txt');
  rewrite(f);
  for i := 0 to num_elective3 do
    begin
      with elective3[i] do
      begin
        writeln(f, e3_name, e3_num, ' ' , e3_block, '
' ,e3_quota);
      end;
    end;
  close(f)
end;




procedure display_elective;
var
   i : integer;
begin
  clrscr;
  writeln(' Elective Name   Elective#   Block#   Quota');
  textcolor(yellow);
  writeln(' -----------------------------------------');
  textcolor(Lightred);
  for i := 1 to num_elective1 do
    writeln(elective1[i].e1_name:15,
elective1[i].e1_num:12,          elective1[i].e1_block:10,
elective1[i].e1_quota:5);
    writeln;
  textcolor(Lightgreen);
  for i := 1 to num_elective2 do
    writeln(elective2[i].e2_name:15,
elective2[i].e2_num:12,          elective2[i].e2_block:10,
elective2[i].e2_quota:5);
    writeln;
  textcolor(Lightcyan);
  for i := 1 to num_elective3 do
    writeln(elective3[i].e3_name:15,
elective3[i].e3_num:12,          elective3[i].e3_block:10,
elective3[i].e3_quota:5);
    writeln;
  textcolor(white);
  writeln;
  write('        <<< Press <Enter> to return. >>>');
  readln
end;

procedure display_elective1_choice;
var
   i : integer;
begin
  clrscr;
  writeln(' Elective Name   Elective#   Block#   Quota');
  textcolor(yellow);
  writeln(' ------------------------------------------');
  textcolor(white);
  writeln;
  for i := 1 to num_elective1 do
    writeln(elective1[i].e1_name:15,
elective1[i].e1_num:12,          elective1[i].e1_block:10,
elective1[i].e1_quota:5);
  writeln;
  textcolor(yellow);
  writeln(' ------------------------------------------');
  textcolor(white);
  writeln
end;

procedure display_elective2_choice;
var
   i : integer;
begin
  clrscr;
  writeln(' Elective Name   Elective#   Block#   Quota');
  textcolor(yellow);
  writeln(' ------------------------------------------');
  textcolor(white);
  for i := 1 to num_elective2 do
    writeln(elective2[i].e2_name:15,
elective2[i].e2_num:12,          elective2[i].e2_block:10,
elective2[i].e2_quota:5);
  writeln;
  textcolor(yellow);
  writeln(' ------------------------------------------');
  textcolor(white);
end;

procedure display_elective3_choice;
var
   i : integer;
begin
  clrscr;
  writeln(' Elective Name   Elective#   Block#   Quota');
  textcolor(yellow);
  writeln(' ------------------------------------------');
  textcolor(white);
  for i := 1 to num_elective3 do
    writeln(elective3[i].e3_name:15,
elective3[i].e3_num:12,          elective3[i].e3_block:10,
elective3[i].e3_quota:5);
  writeln;
  textcolor(yellow);
  writeln(' ------------------------------------------');
  textcolor(white);
end;

procedure display_choices(stud_index : integer);
begin
with student[stud_index] do
  clrscr;
  writeln;
  writeln('       Your Choices: ');
  writeln;
  writeln('          -------------------------------------
');
  writeln;
  writeln('                        Student   ID     :   ',
student[stud_index].studid);
```

```
  writeln;
  writeln('                    Name                 :   ',
student[stud_index].studname);
  writeln;
  writeln('        Your Choices : ');
  writeln;
  writeln('                              Elective1  :  ',
elective1[student[stud_index].elective1].e1_name);
  writeln;
  writeln('                              Elective2  :  ',
elective2[student[stud_index].elective2].e2_name);
  writeln;
  writeln('                              Elective3  :  ',
elective3[student[stud_index].elective3].e3_name);
  writeln;
  writeln('                        Last   Update   :   ',
student[stud_index].lastupdate);
  writeln;
  write('       <<< Press <Enter> to return. >>>');
  readln
end;


procedure choose_elective1(stud_index : integer);
var
   choice : integer;
   choice_done : boolean;
begin
 { Reset the quota of the elective }
  elective1[student[stud_index].elective1].e1_quota     :=
elective1[student[stud_index].elective1].e1_quota + 1;
  student[stud_index].elective1 := 0;
  elective1[0].e1_quota := 0;
  student[stud_index].chooseelective1 := 'N';
  display_elective1_choice;


   begin
```

```
      choice_done := false;
      repeat
        write('Enter your choice of Block 1 elective no. (0 for
nil) : ');
        readln(choice);

        if (choice = 0) then
          choice_done := true
        else
          if (elective1[choice].e1_block <> 1) then
            writeln('Incorrect choice!')
          else
            if(elective1[choice].e1_quota = 0) then
              writeln('Not enough quota!')
            else
              if(elective1[choice].e1_name                     =
elective2[student[stud_index].elective2].e2_name)
                or          (elective1[choice].e1_name      =
elective3[student[stud_index].elective3].e3_name) then
                writeln('Choose already!')
              else
              choice_done := true
        until choice_done;
    end;

  student[stud_index].elective1 := choice;
  elective1[choice].e1_quota := elective1[choice].e1_quota
- 1;
  student[stud_index].lastupdate := DateTimeToStr(now);

  student[stud_index].chooseelective1 := 'Y';

  { Update data files }
  store_students;
  store_elective1;

  display_choices(stud_index);
end;
```

```
procedure choose_elective2(stud_index : integer);
var
    choice : integer;
    choice_done : boolean;
begin
 { Reset the quota of the elective }
  elective2[student[stud_index].elective2].e2_quota      :=
elective2[student[stud_index].elective2].e2_quota + 1;
  student[stud_index].elective2 := 0;
  elective2[0].e2_quota := 0;
  student[stud_index].chooseelective2 := 'N';
  display_elective2_choice;


    begin
      choice_done := false;
      repeat
        write('Enter your choice of Block 2 elective no. (0 for
nil) : ');
        readln(choice);
        choice := choice;
        if (choice = 0) then
          choice_done := true
        else
          if (elective2[choice].e2_block <> 2) then
            writeln('Incorrect choice!')
          else
            if(elective2[choice].e2_quota = 0) then
              writeln('Not enough quota!')
            else
              if(elective2[choice].e2_name                    =
elective1[student[stud_index].elective1].e1_name)
                or          (elective2[choice].e2_name      =
elective3[student[stud_index].elective3].e3_name) then
                writeln('Choose already!')
              else
              choice_done := true

        until choice_done;
    end;

  student[stud_index].elective2 := choice;
  elective2[choice].e2_quota := elective2[choice].e2_quota
- 1;
  student[stud_index].lastupdate := DateTimeToStr(now);

  student[stud_index].chooseelective2 := 'Y';

  { Update data files }
  store_students;
  store_elective2;

  display_choices(stud_index);
end;

procedure choose_elective3(stud_index : integer);
var
    choice : integer;
    choice_done : boolean;
begin
 { Reset the quota of the elective }
  elective3[student[stud_index].elective3].e3_quota     :=
elective3[student[stud_index].elective3].e3_quota + 1;
  student[stud_index].elective3 := 0;
  elective3[0].e3_quota := 0;
  student[stud_index].chooseelective3 := 'N';
  display_elective3_choice;


    begin
      choice_done := false;
      repeat
        write('Enter your choice of Block 3 elective no. (0 for
nil) : ');
```

```
       readln(choice);
       choice := choice;
       if (choice = 0) then
         choice_done := true
       else
         if (elective3[choice].e3_block <> 3) then
           writeln('Incorrect choice!')
         else
           if(elective3[choice].e3_quota = 0) then
             writeln('Not enough quota!')
           else
             if(elective3[choice].e3_name                =
elective1[student[stud_index].elective1].e1_name)
              or          (elective3[choice].e3_name      =
elective1[student[stud_index].elective1].e1_name) then
              writeln('Choose already!')
            else
            choice_done := true
       until choice_done;
   end;

  student[stud_index].elective3 := choice;
  elective3[choice].e3_quota := elective3[choice].e3_quota
- 1;
  student[stud_index].lastupdate := DateTimeToStr(now);

  student[stud_index].chooseelective3 := 'Y';

  { Update data files }
  store_students;
  store_elective3;

  display_choices(stud_index);
end;

procedure choose_elective(stud_index : integer);
 var
  chooseelective : char;
```

```
begin
  repeat
    clrscr;
    writeln;
    writeln;
    writeln('              NSS Elective Selection System ');
    writeln;
    writeln('                       Choose  Electice/Update
Choices');
    writeln;
    textcolor(yellow);
    writeln('
************************************');
    textcolor(white);
    writeln;
    writeln('                  1. Electice1');
    writeln;
    writeln('                  2. Electice2');
    writeln;
    writeln('                  3. Elective3');
    writeln;
    writeln('                  4. Display My Choices');
    writeln;
    writeln('                  5. Return');
    writeln;
    textcolor(yellow);
    writeln('
************************************');
    textcolor(white);
    writeln;
    write('              Enter choice: ');
    readln(chooseelective);
    writeln;
    case chooseelective of
     '1' : choose_elective1(stud_index);
     '2' : choose_elective2(stud_index);
     '3' : choose_elective3(stud_index);
     '4' : display_choices(stud_index);
    end;
  until chooseelective = '5';
 end;



{                                                        Ref:
http://computer-programming-forum.com/29-pascal/7af4f3f05
f738777.htm }
function GetPWord : string;    (* A function for hiding
password *)
var
  S : string;
  C : Char;
begin
  S := '';
  repeat
    C := ReadKey;
    if (C <> #10) and (C <> #13) and (C <> #8) then
      begin
        S := S + C;
        write('*');
      end
    else if C = #8 then
      begin
        S[0] := Chr(Length(S) - 1);
        GotoXY(WhereX - 1, WhereY);
        write(' ');
        GotoXY(WhereX - 1, WhereY);
      end;
  until (C = #10) or (C = #13);
  GetPWord := S;
  writeLn;
end;

procedure change_password(stud_index : integer);
var
  oldpass, newpass1, newpass2 : string;
  pwchanged : boolean;
begin
  pwchanged := false;
  repeat
    clrscr;
    writeln;
    write('       Please enter your old password          : ');
    oldpass := GetPword;
    if oldpass <> student[stud_index].studpw then
      begin
        writeln;
        writeln('       Wrong old password!');
        write('       Press <Enter> to retry. ');
        readln
      end
    else
      begin
        writeln;
        write('       Please enter your new password (4 char) :
');
        newpass1 := GetPword;
        if length(newpass1) <> 4 then
          begin
            writeln;
            writeln('       The length of password must be
4!');
            write('       Press <Enter> to retry. ');
            readln
          end
        else
          begin
            writeln;
            write('       Please enter your new password
again    : ');
            newpass2 := GetPword;
            if newpass1 <> newpass2 then
              begin
                writeln;
                writeln('          The  new  passwords  do  not
match!');
                write('       Press <Enter> to retry. ');
                readln
              end
            else
              begin
                student[stud_index].studpw := newpass1;
                store_students;
                pwchanged := true;
                writeln;
                writeln('       Password changed.');
                write('       Press <Enter> to return. ');
                readln
              end
          end
      end
  until pwchanged
end;

procedure main_menu(stud_index : integer);
var
  choice : char;
begin
  repeat
    clrscr;
    writeln('Welcome,', student[stud_index].studname);
    writeln;
    writeln('              NSS Elective Selection System ');
    writeln('                         Main Menu');
    writeln;
    textcolor(yellow);
    writeln('
************************************');
    textcolor(white);
    writeln;
    writeln('                  1. Display Elective');
    writeln;
```

```
   writeln('                  2. Display My Choices');
   writeln;
   writeln('                    3. Choose Elective/Update
Choices');
   writeln;
   writeln('                  4. Change password');
   writeln;
   writeln('                  5. Quit');
   writeln;
   textcolor(yellow);
   writeln('
************************************');
   textcolor(white);
   writeln;

   if student[stud_index].chooseelective1 = 'N' then
     begin
       textcolor(cyan);
       writeln('                      You have not chosen
elective1!');
       textcolor(white)
     end;

   if student[stud_index].chooseelective2 = 'N' then
     begin
       textcolor(cyan);
       writeln('                      You have not chosen
elective2!');
       textcolor(white)
     end;

   if student[stud_index].chooseelective3 = 'N' then
     begin
       textcolor(cyan);
       writeln('                      You have not chosen
elective3!');
       textcolor(white)
     end;
```

```
    writeln;
    write('               Enter choice: ');
    readln(Choice);
    writeln;
    case choice of
      '1' : display_elective;
      '2' : display_choices(stud_index);
      '3' : choose_elective(stud_index);
      '4' : change_password(stud_index);
    end;
  until choice = '5';
end;

procedure display_students_choices(stud_index : integer);
var
  studid : string;
  found : boolean;
  i : integer;
begin
  clrscr;
  writeln;
  writeln;
  writeln('                           NSS Elective Selection
System           ');
  writeln('                         Display Students Choices
');
  writeln;
  textcolor(yellow);
  writeln('
-----------------------------------------      ');
  textcolor(white);
  writeln;
  write('                     Student ID : ');
  readln(studid);
  writeln;
  found := false;
  i := 0;
  while (i < num_stud) and (not found) do
```

```
    begin
      i := i + 1;
      if (studid = student[i].studid)then
        begin
          found := true;
          stud_index := i
        end
    end;
  if not found then
    begin
      stud_index := 0;
      textcolor(Lightcyan);
      writeln('':20,'> > > Invalid UserID ');
      write('':20,'> > > Press <Enter> to refresh.');
      textcolor(white);
      readln
    end

  else
  display_choices(stud_index);
end;


procedure  display_subject_namelist ;
var
   i, block, enum : integer;

procedure searchblock1(enum : integer);
var k : integer;
    found : boolean;
  begin
  clrscr;
  found := false;
  writeln('NSS Elective Selection System          ');
  writeln;
  writeln('Namelist        of        student        taking
',elective1[enum].e1_name);
  textcolor(yellow);
  writeln('--------------------------------------------
');
  textcolor(white);
    for k :=1 to num_stud do
     if student[k].elective1 = enum then
      begin
      writeln(student[k].studname);
      found := true;
      end;
    if not found then writeln('No record found!');
    writeln;
    writeln;
    write('Press <Enter> to return');
    readln;
  end;

procedure searchblock2(enum : integer);
var k : integer;
    found : boolean;
  begin
  clrscr;
  found := false;
  writeln('NSS Elective Selection System          ');
  writeln;
  writeln('Namelist of student taking
',elective2[enum].e2_name);
  textcolor(yellow);
  writeln('--------------------------------------------
');
  textcolor(white);
    for k :=1 to num_stud do
     if student[k].elective2 = enum then
      begin
      writeln(student[k].studname);
      found := true;
      end;
    if not found then writeln('No record found!');
    writeln;
    writeln;
    write('Press <Enter> to return');
    readln;
  end;


procedure searchblock3(enum : integer);
var k : integer;
    found : boolean;
  begin
  clrscr;
  found := false;
  writeln('NSS Elective Selection System           ');
  writeln;
  writeln('Namelist        of        student        taking
',elective3[enum].e3_name);
  textcolor(yellow);
  writeln('--------------------------------------------
');
  textcolor(white);
    for k :=1 to num_stud do
      if student[k].elective3 = enum then
       begin
       writeln(student[k].studname);
       found := true;
       end;
    if not found then writeln('No record found!');
    writeln;
    writeln;
    write('<Press <Enter> to return>');
    readln;
  end;


begin
  clrscr;
  writeln(' Elective Name   Elective#   Block#   Quota');
  textcolor(yellow);
  writeln(' ----------------------------------------');
  textcolor(Lightred);
  for i := 1 to num_elective1 do
    writeln(elective1[i].e1_name:15,
elective1[i].e1_num:12,          elective1[i].e1_block:10,
elective1[i].e1_quota:5);
    writeln;
  textcolor(Lightgreen);
  for i := 1 to num_elective2 do
    writeln(elective2[i].e2_name:15,
elective2[i].e2_num:12,          elective2[i].e2_block:10,
elective2[i].e2_quota:5);
    writeln;
  textcolor(Lightcyan);
  for i := 1 to num_elective3 do
    writeln(elective3[i].e3_name:15,
elective3[i].e3_num:12,          elective3[i].e3_block:10,
elective3[i].e3_quota:5);
    writeln;
  textcolor(white);
  write('Enter the block: ');
  readln(block);
  write('Enter the elective number: ');
  readln(enum);

  case block of
  1 :searchblock1(enum);
  2 :searchblock2(enum);
  3 :searchblock3(enum);
  end;





end;
```

```
procedure main_menut(teac_index : integer);
var
  choice : char;
begin
  repeat
    clrscr;
    writeln('Welcome,', teacher[teac_index].teacname);
    writeln;
    writeln('              NSS Elective Selection System ');
    writeln('                        Main Menu');
    writeln;
    textcolor(yellow);
    writeln('
************************************');
    textcolor(white);
    writeln;
    writeln('                      1. Display Subject Name
List');
    writeln;
    writeln('                         2. Display Students
Choices');
    writeln;
    writeln('                 3. Quit');
    writeln;
    textcolor(yellow);
    writeln('
************************************');
    textcolor(white);
    writeln;
    writeln;
    write('              Enter choice: ');
    readln(Choice);
    writeln;
    case choice of
      '1' : display_subject_namelist;
      '2' : display_students_choices(stud_index);
    end;
  until choice = '3';
end;
```

```
procedure loginstudent(var stud_index : integer);
var
  userid, password : string;
  found : boolean;
  i : integer;
begin
  clrscr;
  writeln;
  writeln;
  writeln('                        NSS Elective  Selection
System          ');
  writeln;
  textcolor(yellow);
  writeln('
--------------------------------------------     ');
  writeln('                                              -
-     ');
  writeln('                    -                    LOGIN
-     ');
  writeln('                                              -
-     ');
  writeln('
--------------------------------------------     ');
  textcolor(white);
  writeln;
  write('                    UserID : ');
  readln(userid);
  writeln;
  write('                  Password : ');
  password := GetPword;
  writeln;
  writeln;
  found := false;
  i := 0;
  while (i < num_stud) and (not found) do
    begin
      i := i + 1;
      if (userid = student[i].studid) and (password =
student[i].studpw) then
        begin
          found := true;
          stud_index := i
        end
    end;
  if not found then
    begin
      stud_index := 0;
      textcolor(Lightcyan);
      writeln('':20,'> > > Invalid UserID or Password!');
      write('':20,'> > > Press <Enter> to refresh.');
      textcolor(white);
      readln
    end;
end;


procedure loginteacher(var teac_index : integer);
var
  userid, password : string;
  found : boolean;
  i : integer;
begin
  clrscr;
  writeln;
  writeln;
  writeln('                        NSS Elective Selection
System          ');
  writeln;
  textcolor(yellow);
  writeln('
--------------------------------------------     ');
  writeln('                                            -
-     ');
  writeln('                    -                   LOGIN
-     ');
  writeln('                                            -
-     ');
  writeln('
--------------------------------------------     ');
  textcolor(white);
  writeln;
  write('                     UserID : ');
  readln(userid);
  writeln;
  write('                     Password : ');
  password := GetPword;
  writeln;
  writeln;
  found := false;
  i := 0;
  while (i < num_teac) and (not found) do
    begin
      i := i + 1;
      if  (userid  =  teacher[i].teacid)  and  (password  =
teacher[i].teacpw) then
        begin
          found := true;
          teac_index := i
        end
    end;
  if not found then
```

```
   begin
     teac_index := 0;
     textcolor(Lightcyan);
     writeln('':20,'> > > Invalid UserID or Password!');
     write('':20,'> > > Press <Enter> to refresh.');
     textcolor(white);
     readln
   end;
end;




procedure choose_role;
 var
  choice : char;
begin
  repeat
   clrscr;
   writeln;
   writeln;
   writeln('              NSS Elective Selection System ');
   writeln;
   textcolor(yellow);
   writeln('
************************************');
   textcolor(white);
   writeln;
   writeln('                 1. Student');
   writeln;
   writeln('                 2. Teacher');
   writeln;
   textcolor(yellow);
   writeln('
************************************');
   textcolor(white);
   writeln;
```

```
    write('                Enter choice: ');
    readln(choice);
    writeln;

if choice = '1' then
 begin
    repeat
    loginstudent(stud_index);
    if stud_index <> 0 then
      main_menu(stud_index)
    until false;
  readln;
 end


 else if choice = '2' then
  begin
    repeat
    loginteacher(teac_index);
    if teac_index <> 0 then
      main_menut(teac_index)
    until false;
  readln;
 end


 else
 writeln('                Please Enter 1 or 2 only!');
 readln;

until (choice = '1') or (choice = '2') ;

end;
```

```
begin  (* main body *)

  read_students;
  read_teachers;
  read_elective1;
  read_elective2;
  read_elective3;
  choose_role;

end.
```

Appendix 2: Working schedule

Date	Event
3/2014	Choice of Topic
5/2014	Complete the introduction and background
	Complete the Analysis
6/2014	Start to have briefly design of the program
8/2014	Finish the login function
8/2014	Start working the procedure
9/2014	Start implement the program
	Start testing the program
10/2014	Evaluation of the program
12/2014	Finish all of the program
1/2015	Work on the report
	Learn and use more program code
	Finalize the report
	Finish debugging the program

Appendix3: Reference

1. http://www.i-garden.org/blog/tag/hkeaa
2. http://www.cswcss.edu.hk

Readkey function :

3. http://www.freepascal.org/docs-html/rtl/crt/readkey.html

4. http://www.programmersheaven.com/mb/pasprog/Board.aspx?S=B20000

5. .Microsoft Corporation

6. Dev Pascal

7. Mr. Chu Kin Fung

8. Longman Hong Kong Education

End of the Project