Hong Kong Diploma of Secondary Education

Examination 2016

Information and Communication Technology

Option D: Software Development

Title: Extra-curricular Activities System

Contents

Chapter 1 Introduction

1.1 Problem & Situation

Students in the Kaw school, a secondary school, find it uneasy to choose the extra-curricular activities in the existing system. And there are some problems, for example, the existing system may not let the students cancel their choices. In a bid to address the problem, the Kaw school wants to set up a new system so that the students can choose the activities by the program and the data can be organized systematically. As an I.T. monitor, I was assigned to develop the extra-curricular activities system.

1.2 Objective

The school would like to provide an extra-curricular activities system to let students have a better and efficient system to select extra-curricular activities that they are interested in.

I am now responsible for the project and I am working on this written report to provide solutions for the school. It is hoped that the well-organized new extra-curricular activities system can be beneficial to both teachers and students.

Chapter 2 Design of solution

2.1 Brief Description

In order to develop an extra-curricular activities system, I am going to set up the program with the following functions.

It means that in the program, not only is selection included in it, but also functions like login, display activities, update chooses, adding new club, 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 students use only and they have different functions. For students, they can select their activities very clear and they can change afterward. As for student who took the teacher code, they can add a new club or activities for students to choose.

For the extra-curricular activities, there are total 3 blocks and each block has 5-8 activities for students choose. And it shows as the table below.

.

Block1	Block2	Block3
1. Chinese Society 2. English Society 3. Maths Society 4. Economics Society 5. Science Club 6. Computer Club 7. Putonghua Club 8. Geography Club	1. Volleyball Team 2. Football Team 3. Basketball Team 4. Badminton Team 5. Table Tennis Team	1. Drama Club 2. Dancing Society 3. Music Society 4. Choir 5. Chinese Orchestra

2.2 General Function

The program contains some functions for students. The system may let the student choose their activities and display what are there for students to choose. Apart from that, the students who having the teacher code, they may add a new club for students.

Login & Logout System

To make the system become more users friendly and creatively, I design the program with changing color if the users enter wrong their code or their password.

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.

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 extra-curricular activities

The display extra-curricular activities function can let the students have a look about the activities of election in different block or the remaining quota of each activity before they select their activities.

Display Choices Function

This function is used to let students to check what their choices are and they can also confirm their last update activities to ensure they have choose their interested activities correctly. Besides, they can check their personal information here.

Choose Activities/Update Choices Function

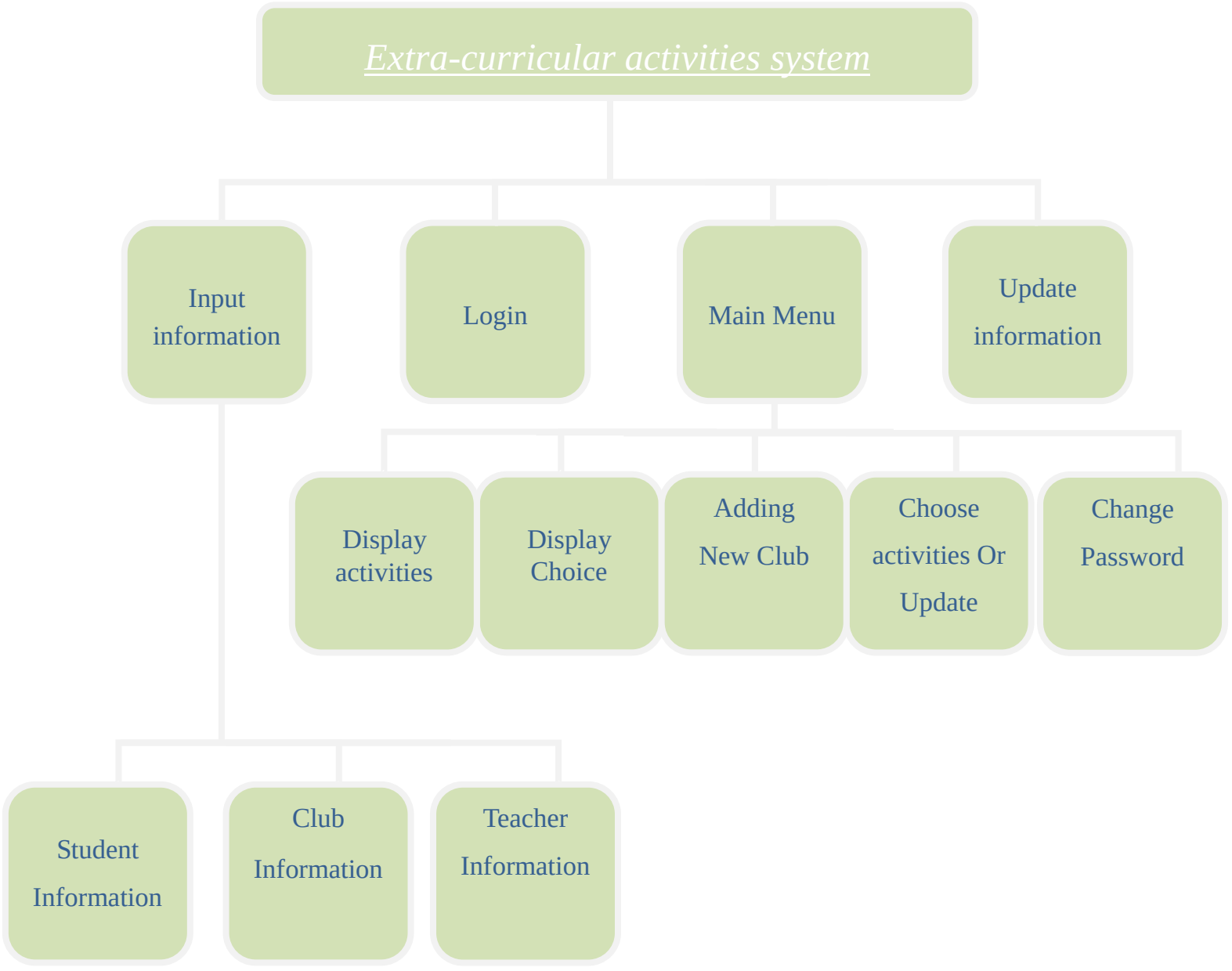
The main function of the system, for students to choose extra-curricular activities they interested. In this function, student can choose the activities block by block and they no need to choose the activities orderly. They can choose the activities in order which they want. Since some students want to join one activity only, this function is wonderful to them.

Display Activities List Function

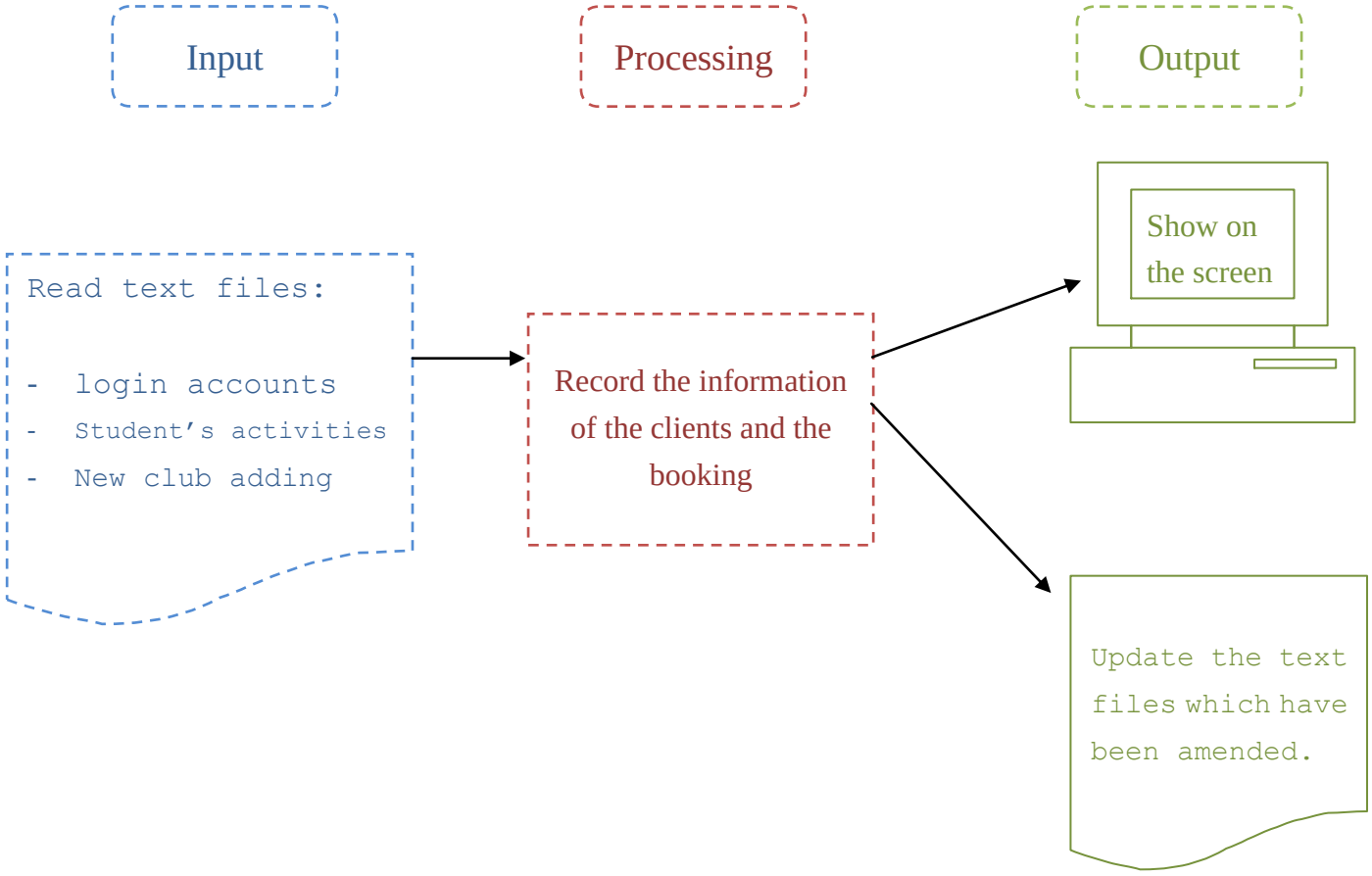
It is a function to convene students to let them know what clubs or activities are giving out for them to choose. So students can know which activities are popular and how many codas le

System Design

Design of main program



The Below Diagram Describe the Data Flow:



File Structure

There are three files to store the data. Record will use to store the data.

1. clubs.txt (store clubs record)
2. students.txt (store students record)
3. teachers.txt (store teachers record)

Data file for storing students' information:

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

1. Student ID (4 characters)
2. Student password (4 character)
3. Student name (25 character)
4. Indicator of activity
5. Selection of Activity1 (integer)
6. Selection of Activity2 (integer)
7. Selection of Activity3 (integer)

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

e.g.

S002	Student ID (4 characters)
1234	Student password (4 character)
Au Yue, Joanne	Student name (25 character)
Y	Indicator of activity
5	Selection of Activity1 (integer)
12	Selection of Activity2 (integer)
18	Selection of Activity3 (integer)

Sample file:
Students.txt

```
s0011234Au Sham Ki, Bobby          Y1 12 17
s0021234Au Yue, Joanne              Y5 12 18
s0031234Chan Kai Bong               Y1 11 21
s0041234Chan Man Cheun             N0 0 0
s0051234Chan Mei Ling              N0 0 0
s0061234Chan Shui Wah, Shirley     N0 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:

1. Teacher code (3 characters)
2. Teacher password (4 characters)

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

e.g.

SHC	Teacher code (3 characters)
1234	Teacher password (4 characters)

Sample File:
Teachers.txt

SHC1234
CKH1234
WCY1234
CTN1234
LTY1234
CYT1234

Data file for storing activities' information:

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

1. Activity name (17 characters)
2. Teacher name (3 characters)
3. Block number (integer)
4. Number of quota (integer)

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

e.g.

Chinese Society	Activity name (17 characters)
SHC	Teacher name (3 characters)
1	Block number (integer)
8	Number of quota (integer)

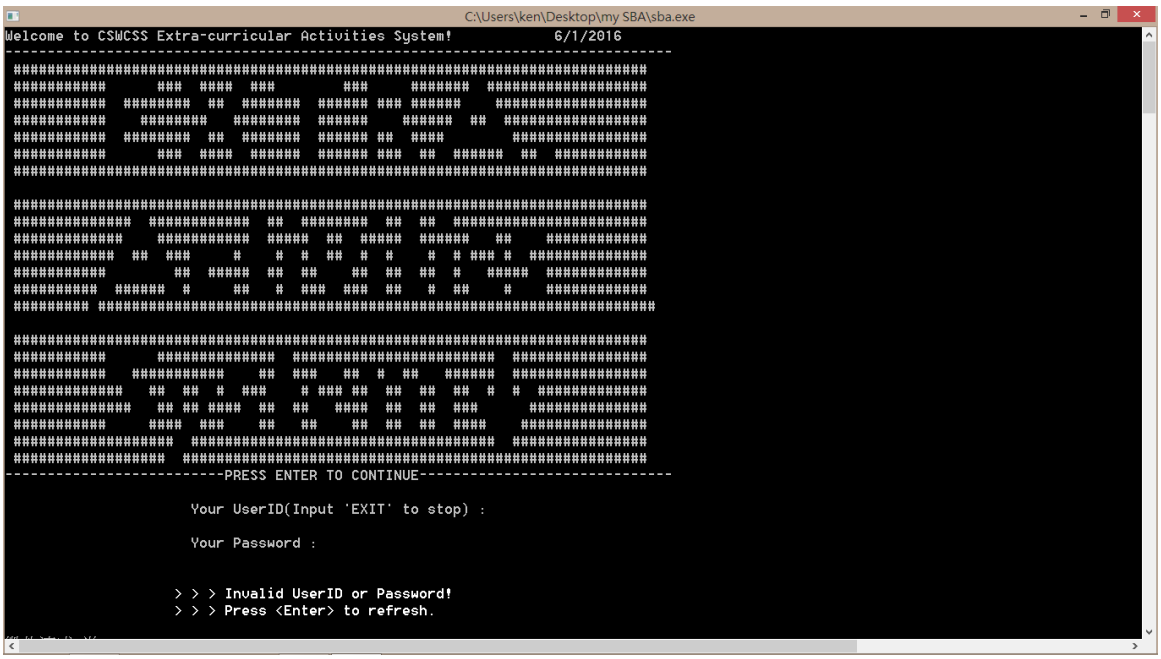
Sample File:
Club.txt

NIL NIL0 0
Chinese Society SHC1 8
English Society CKH1 10
Maths Society WCY1 10
Economics SocietyCTN1 10
Science Club LTY1 9
Computer Club CYT1 10

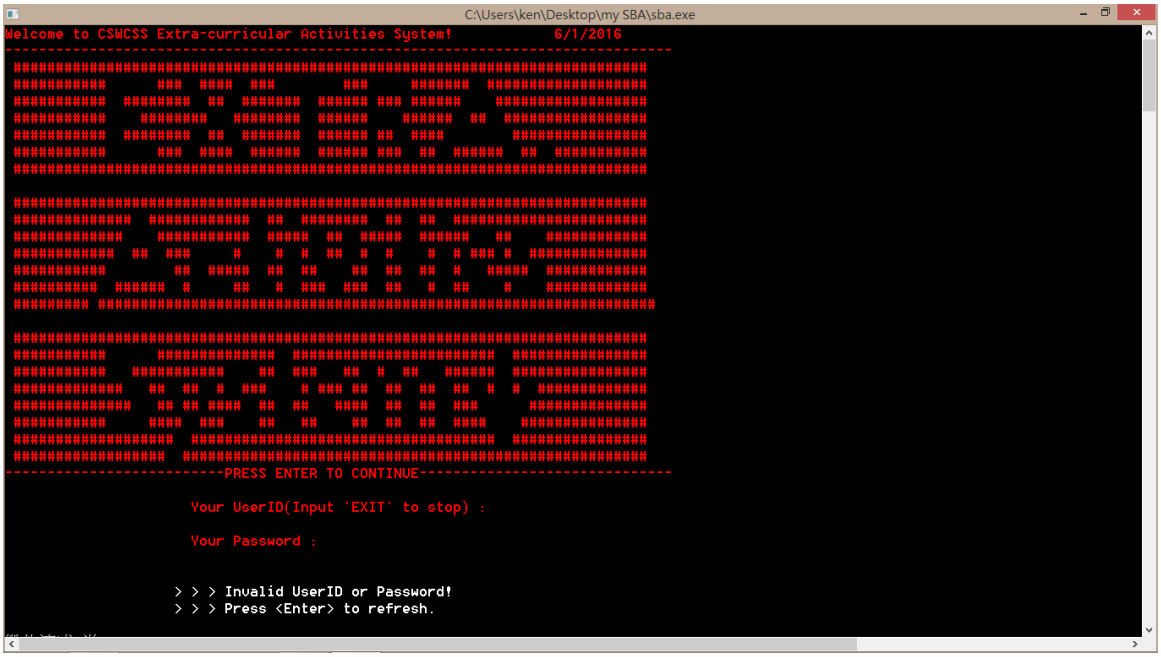
*Remark: 'NIL in clubs' means the students did not choose activity in that block.

2.4 Design of user interface

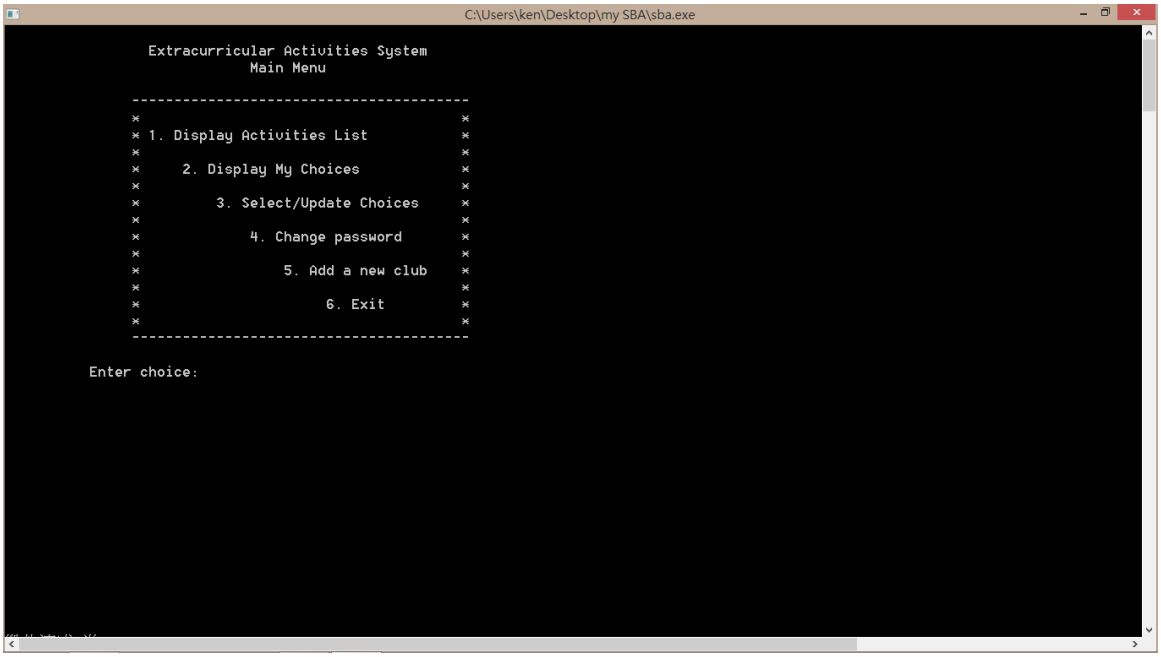
Login Screen(1)



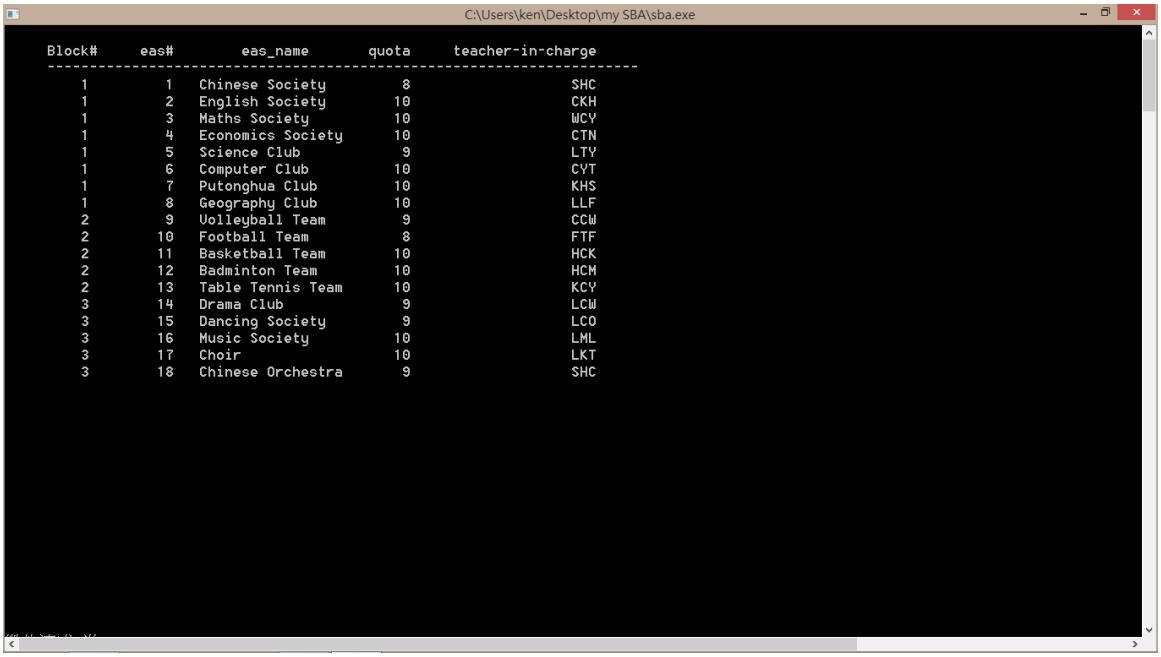
Login Screen(2)



Main Menu

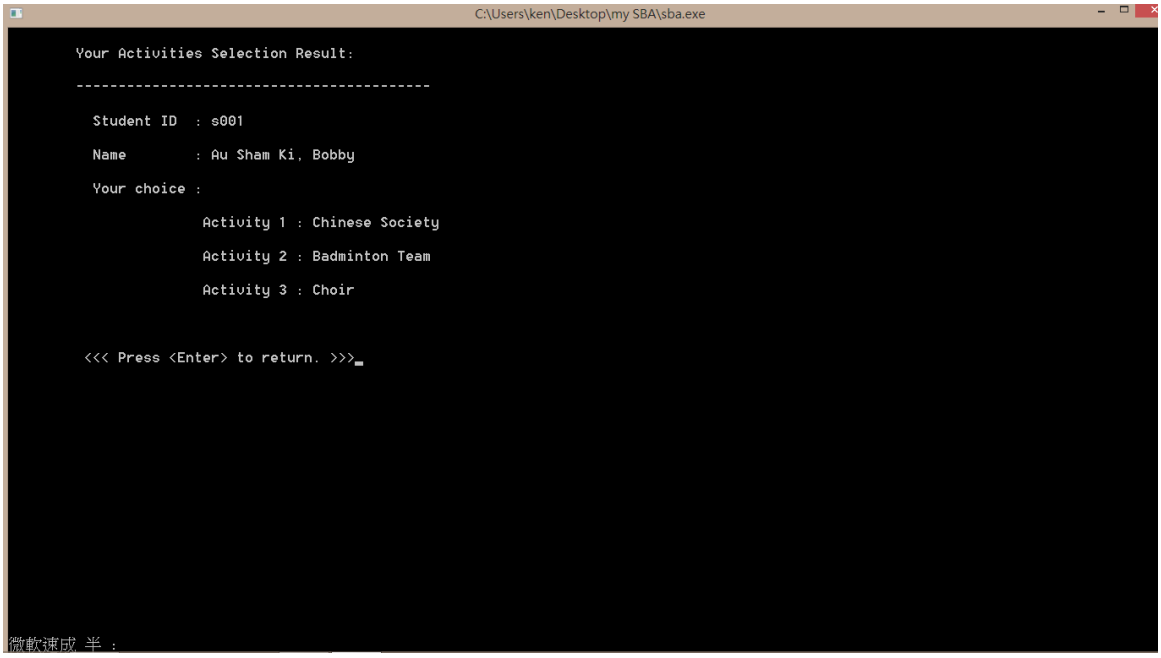

C:\Users\ken\Desktop\my SBA\sba.exe
Extracurricular Activities System
Main Menu
1. Display Activities List
2. Display My Choices
3. Select/Update Choices
4. Change password
5. Add a new club
6. Exit
Enter choice:

Display activities

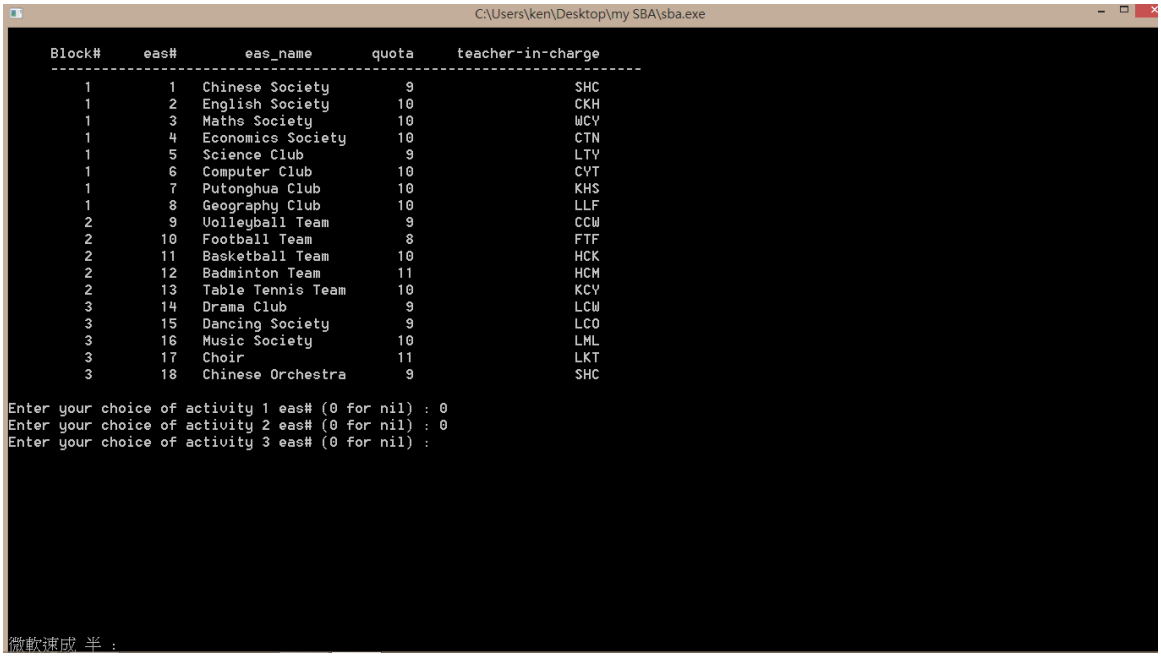



```
Block#     eas#       eas_name       quota     teacher-in-charge
--------------------------------------------------------------------------
    1         1   Chinese Society         8                   SHC
    1         2   English Society        10                   CKH
    1         3   Maths Society          10                   WCY
    1         4   Economics Society      10                   CTN
    1         5   Science Club            9                   LTY
    1         6   Computer Club          10                   CYT
    1         7   Putonghua Club         10                   KHS
    1         8   Geography Club         10                   LLF
    2         9   Volleyball Team         9                   CCW
    2        10   Football Team           8                   FTF
    2        11   Basketball Team        10                   HCK
    2        12   Badminton Team         10                   HCM
    2        13   Table Tennis Team      10                   KCY
    3        14   Drama Club              9                   LCW
    3        15   Dancing Society         9                   LCO
    3        16   Music Society          10                   LML
    3        17   Choir                  10                   LKT
    3        18   Chinese Orchestra       9                   SHC
```

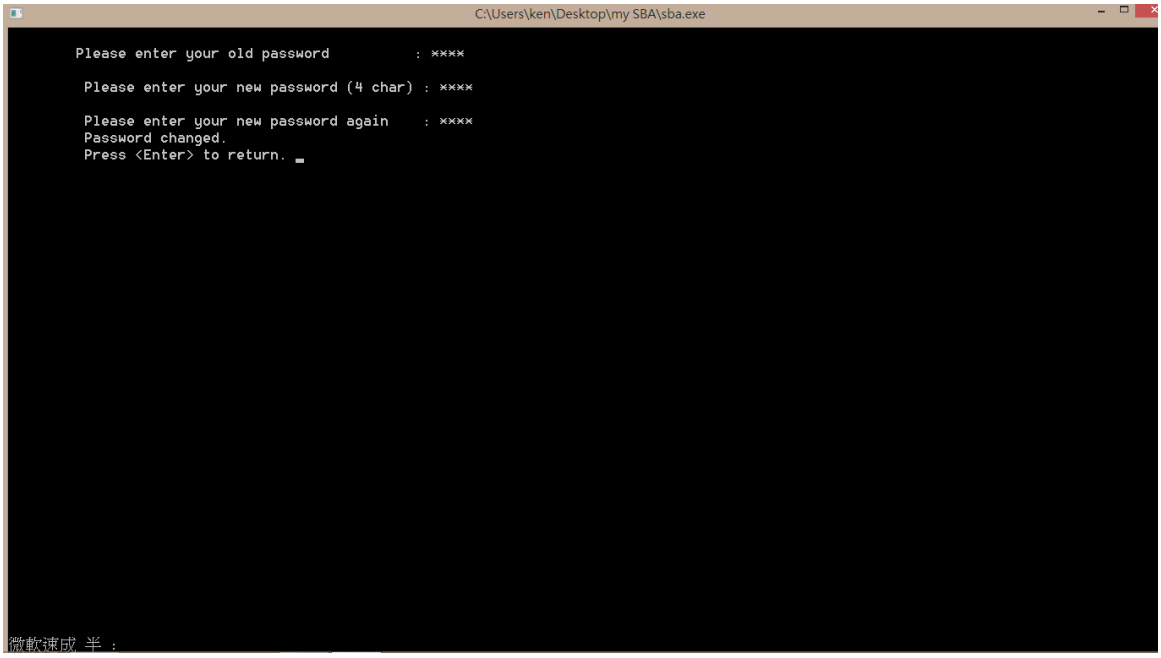
Display own activities



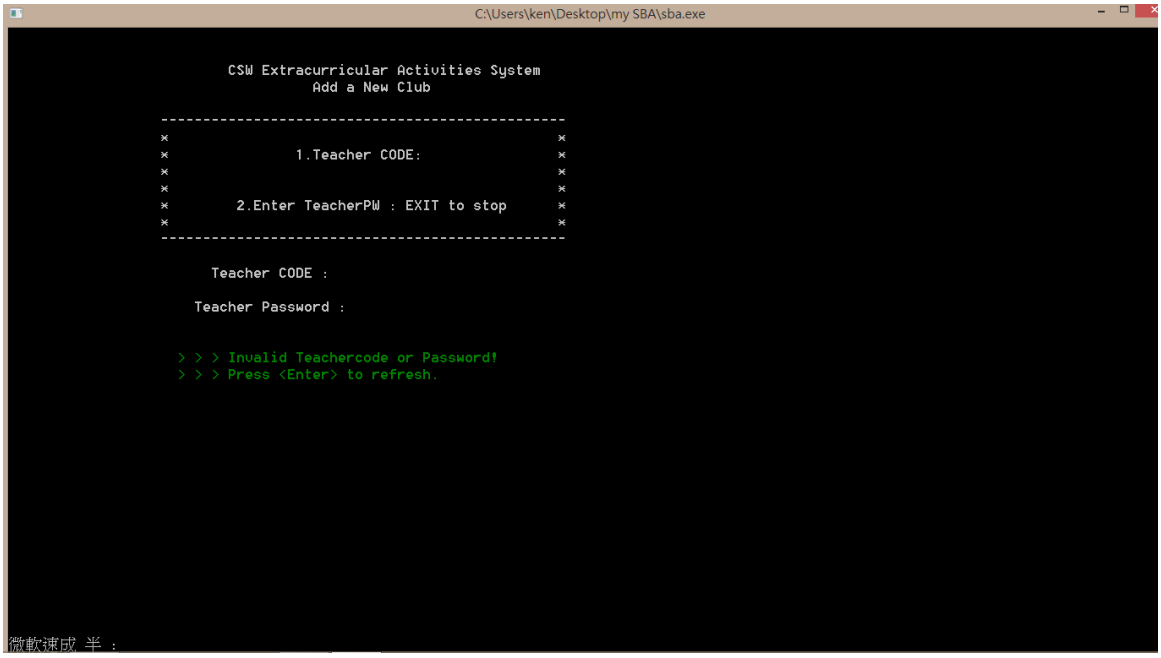
Select / Updata Choice



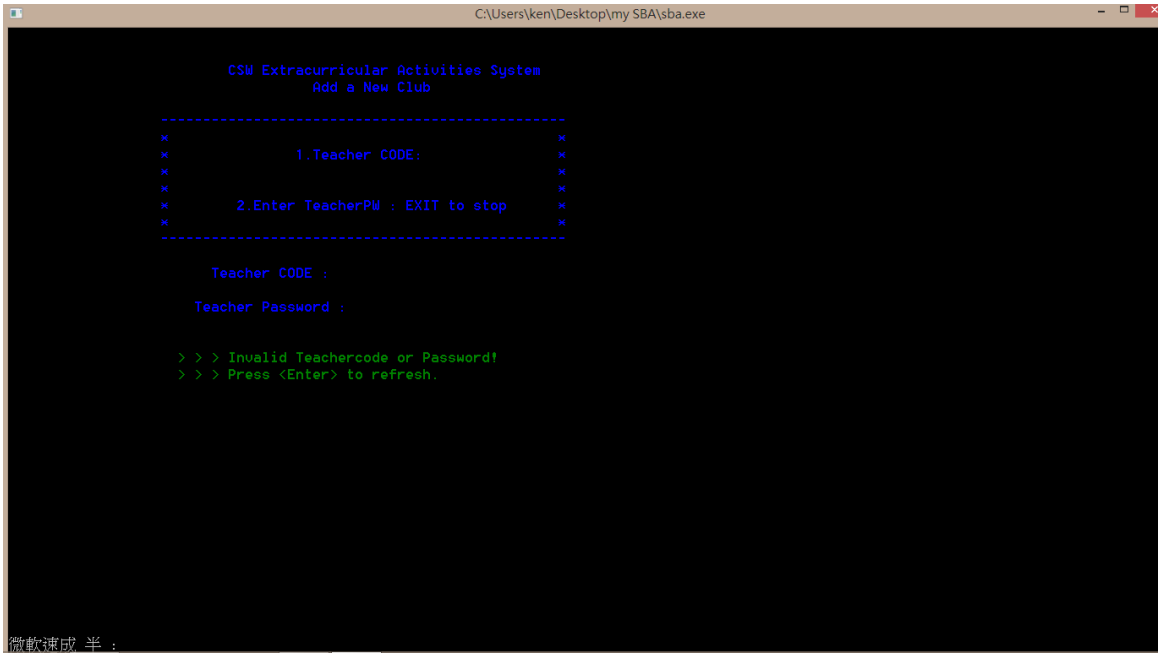
Change Password



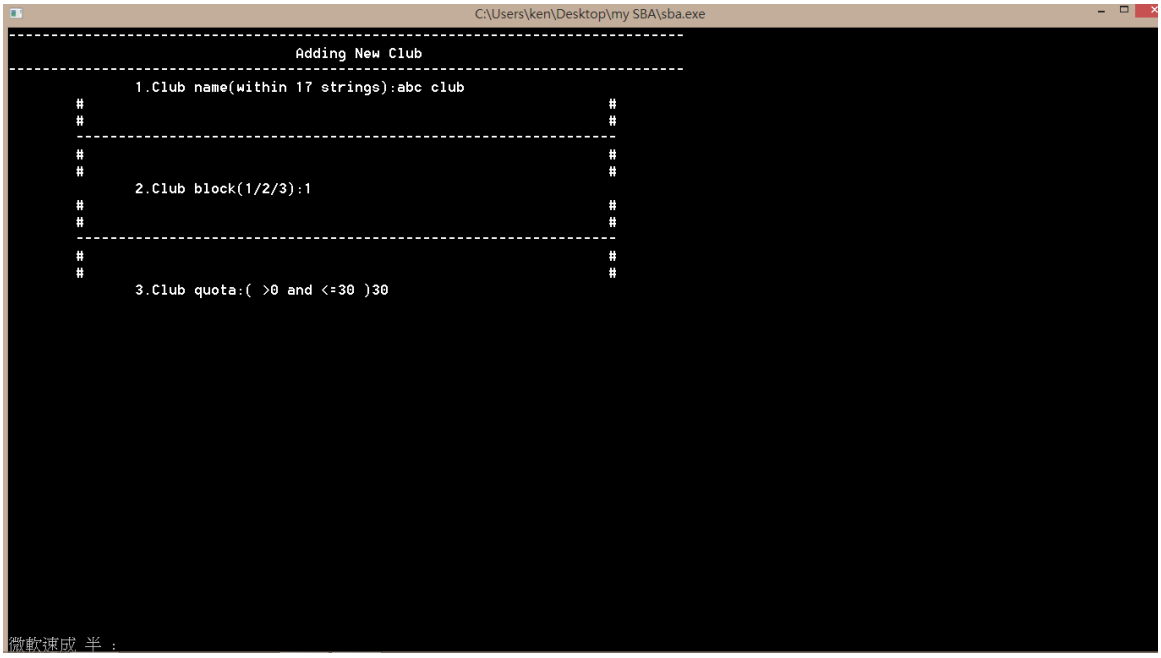
Add a new club(1)



Add a new club(2)



Add a new club(3)



Chapter 3 Implementation

3.1 Brief Description

The Dev-pascal is our choice to implement the NSS Extra-Circular activities 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.

Program coding

In term of program code, I will introduce some of the code which I have been used in the program.

1. Record

e.g.

```
var
  studid:array[1..max_stud]of string[4];
  studpw:array[1..max_stud]of string [4];
  studname:array[1..max_stud]of string[25];
  studdone:array[1..max_stud]of char;
  studeas1:array[1..max_stud]of integer;
  studeas2:array[1..max_stud]of integer;
  studeas3:array[1..max_stud]of integer;
  teacode:array[1..max_tea]of string[3];
  teapw:array[1..max_tea]of string[4];
```

Record is a useful program code which can store data having same data type, for example the " Studid" and the "Studpw" being stored in the text file. (students.txt)

2. Text file

e.g.

```
begin
    assign(f, 'students.txt');
    reset(f);
    i:=0;
```

Text file can help we store data. It may collect lot of data and also let user read and use the data.

3. Case

e.g.

```
writeln;
      case choice of
         1 : display_activities_return;
         2 : display_choices(stud_index);
         3 : choose_activities(stud_index);
         4 : change_password(stud_index);
         5 : Add_a_new_club;
      end;
   until choice = 6;
```

First, the case function will get the user inputted. If the choice is one of the providing number. It will start that case. Such as 1. If user enter (1) it will show the list of the activities which are providing.

4. Clrscr

e.g.

```
begin
      clrscr;
      only:=true;
      textcolor(white);
      assign(f,'clubs.txt');
```

(Clrscr) is a function which will clear all the words in the screen. Since there will be lot of words on the screen if we don't clear them away. So (Clrscr) is a useful function which can convince the user.

The above are some important function in the program. And the else function will show in the program.

3.3 Producer in the program

1. procedure moment

It is a procedure used to show the data which we on the program. Let user easily to know what are the data.

2. procedure read_students

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

3. procedure store_students

It is a procedure used to store the information of students from the text file which stored student' information,

4. procedure read_teachers

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

5. procedure store_teachers

It is a procedure used to store the information of teachers from the text file which stored teacher' information,

6. **procedure read_activities**

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

7. **procedure store_activities**

It is a procedure used to store the information of activities from the text file which stored club' information,

8. **procedure display_activities**

It is a procedure used to display the information of activities from the text file which stored club' information.

9. **procedure display_activities_return**

It is a procedures used to tell user, they have entre wrong information and ask them enter again.

10. **procedure display_choices(stud_index : integer)**

It is a procedure used to display the user activities choice from the text file which stored club' information.

11. procedure choose_activities(stud_index : integer)

It is a procedure used to let the user choose activities from the text file which stored club' information.

12. procedure change_password(stud_index : integer)

It is a procedure used to let the user change their password from the text file which stored student' information.

13. procedure New_Club_Adding(teacher:string[3])

It is a procedure used to let the users which have teacher code to add new club for students from the text file which stored club' information.

14. procedure Add_a_new_club

It is a procedure used to let the users which have teacher code to add detail information about the new club which will stored club' information.

15. procedure main_menu(stud_index : integer)

It is a procedure used to show to main menu to the user. Let them know what function are there.

16. procedure login(var stud_index : integer)

It is a procedure used to let user login their account by input their user ID and password which stored in student file.

3.4 Main Body of the Program

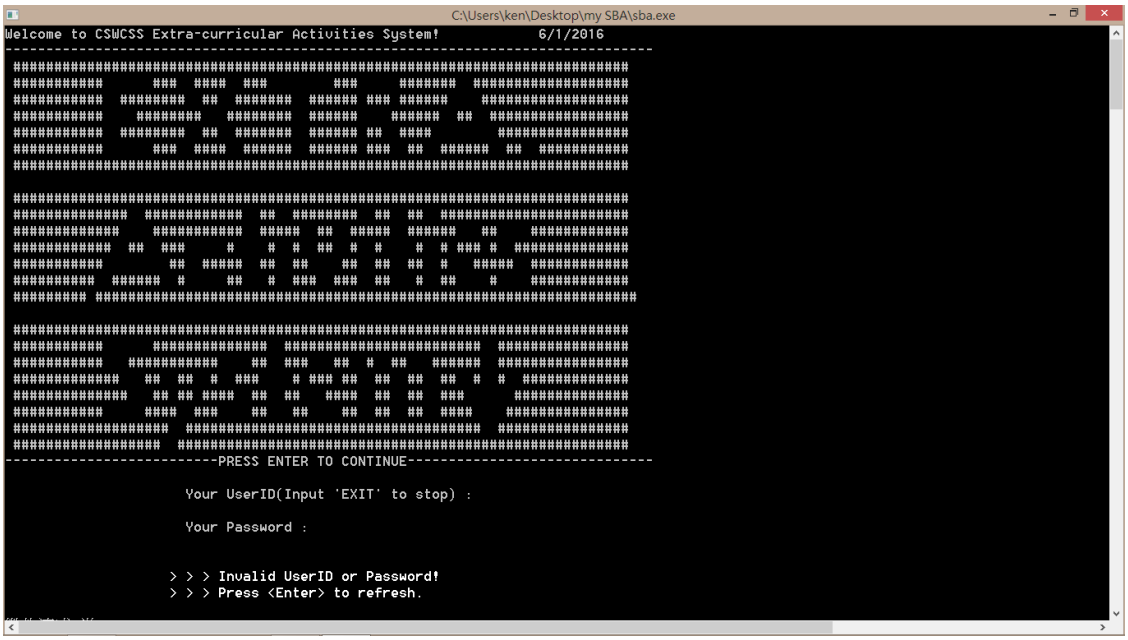
The follow is the main program:

```
begin {main body}
    exit1:=false;
    moment;
    read_students;
    read_teachers;
    read_activities;
    repeat
      login(stud_index);
      if stud_index <> 0 then
        main_menu(stud_index)
    until exit1;
    readln
     end.
```

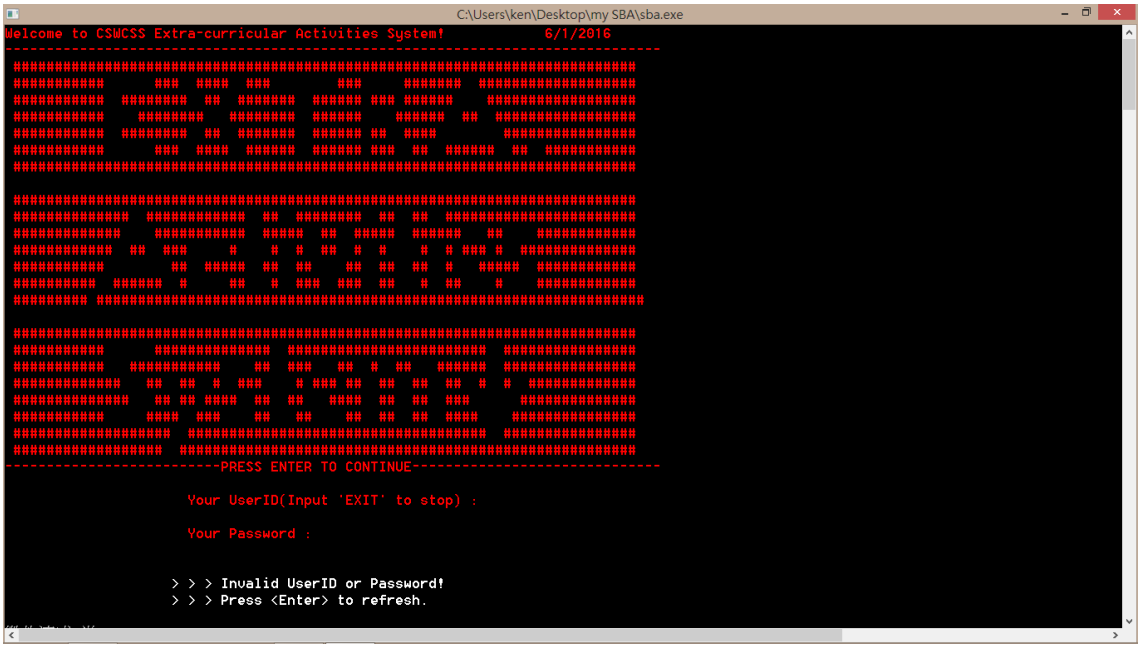
3.5 Program Execution

All the text file will be save in the same folder to let them run correctly.

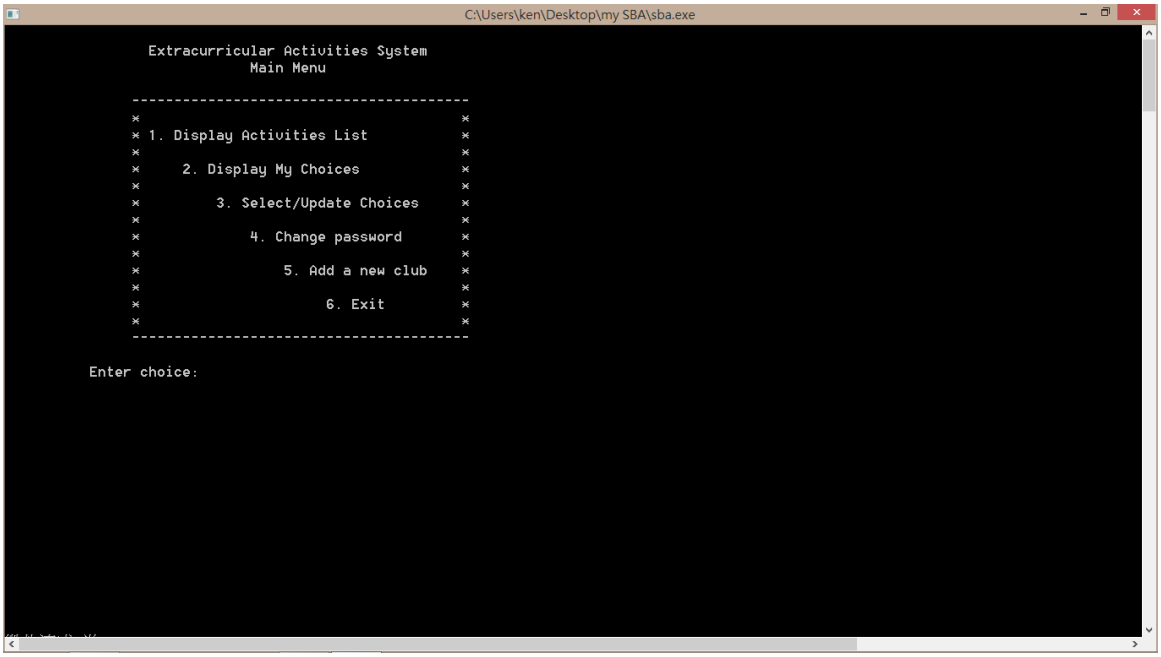
1. Login in pages



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 enter their user ID and password if enter wrong the page will turn into red.

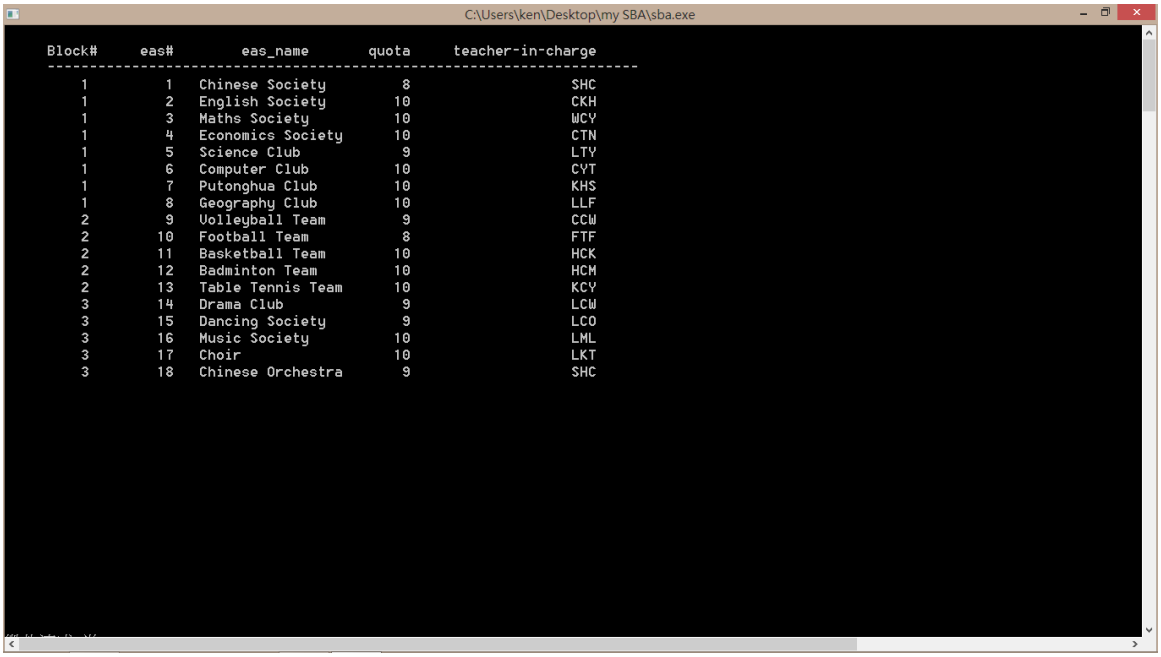


2. Main Menu



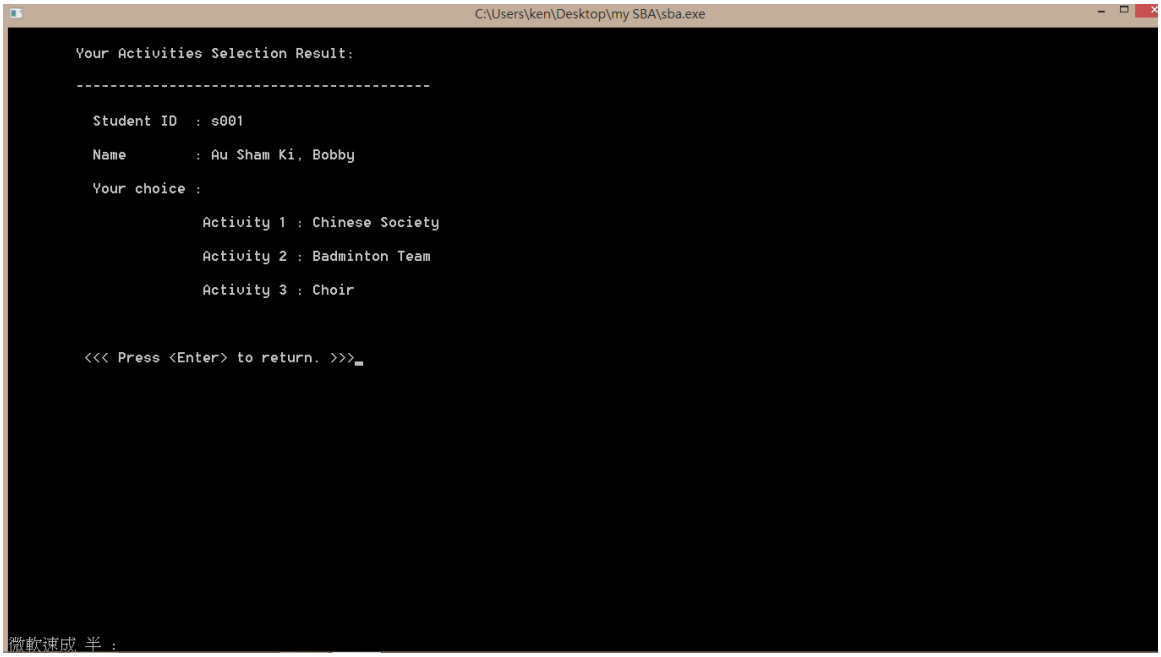
It is the main menu. Students can use the system in this page. In the middle part, student can choose want they want to do by input number from 1-6. E.g (1. Display Activities List)

3. Display activities List



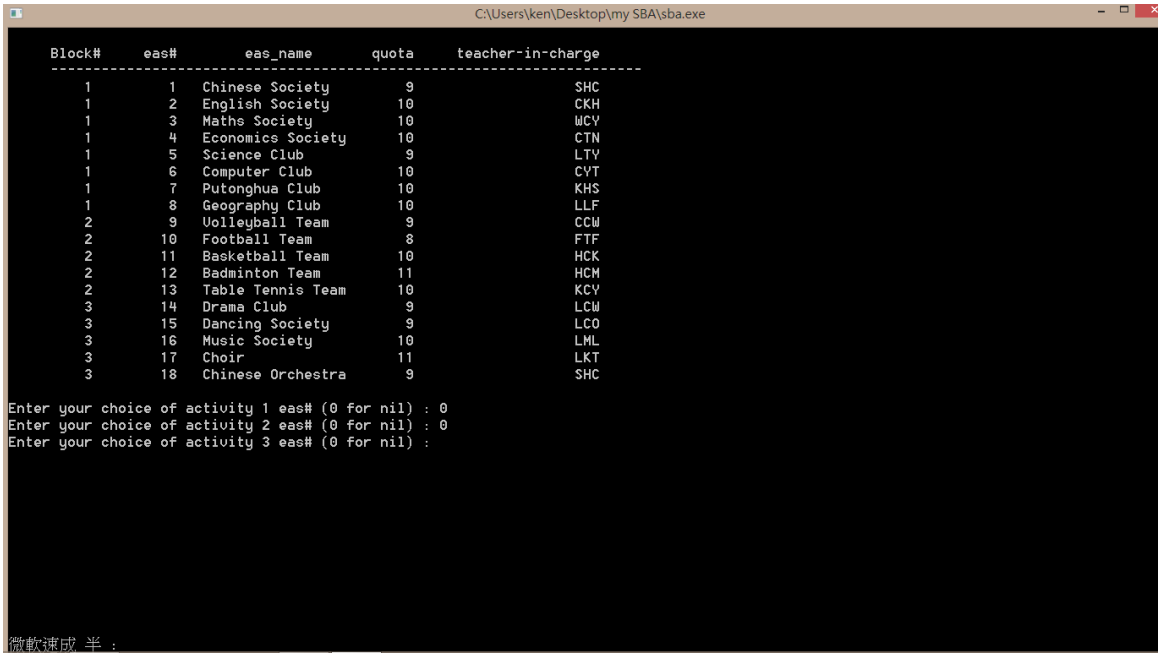
The display activities function can let the students have a look about the Choices of activities in different block or the remaining quota of each activity before they select their extra-curricle activities .And when user press <Enter> will return to the main menu.

4. Display Own choices



This function is used to let students to check what their choices are. Besides, they can check their personal information here.

5. Select or Update own choices



The main function of the system, for students to choose activities they want. In this function, students can choose the activities block by block. They can don't choose any activity in block 1 and choose activity in block 2. Since some students want to choose one activity only. So this function is wonderful for them. They can also display their choices to have a reminder or return to the main menu.

Chapter 4 Testing & Evaluation

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 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.

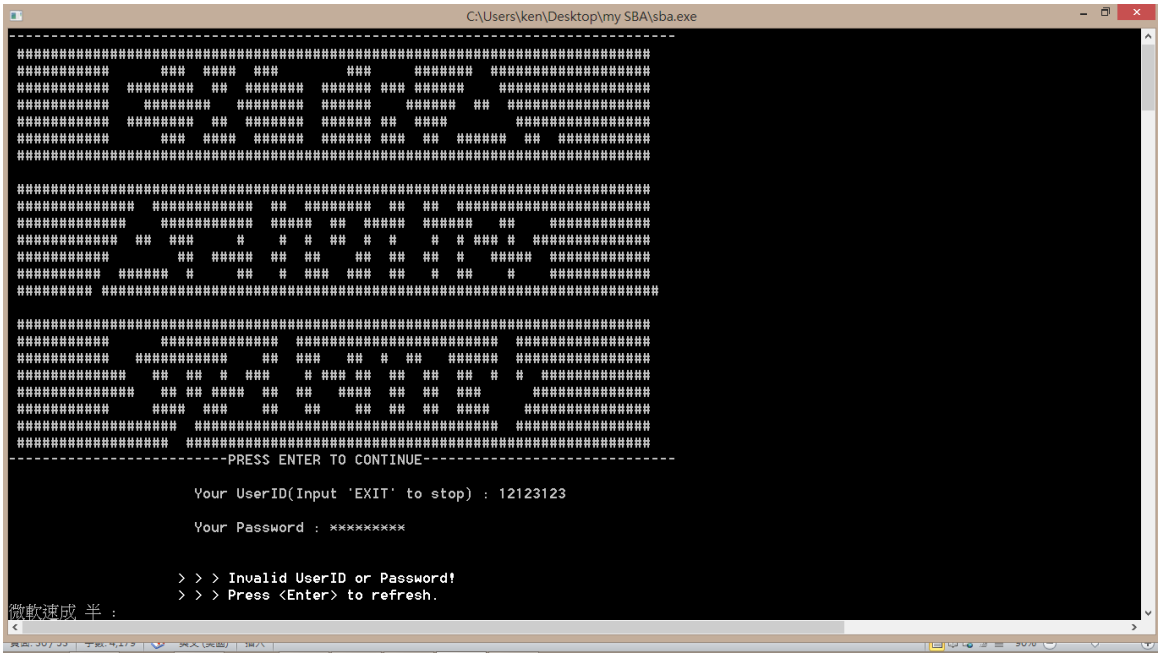
After that, as one of the main reasons in this chapter is to make the program more user-friendly. Some debug works are important to make the program become more user-friendly. So debugging are important.

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 the program. So that bugs may find out while using.

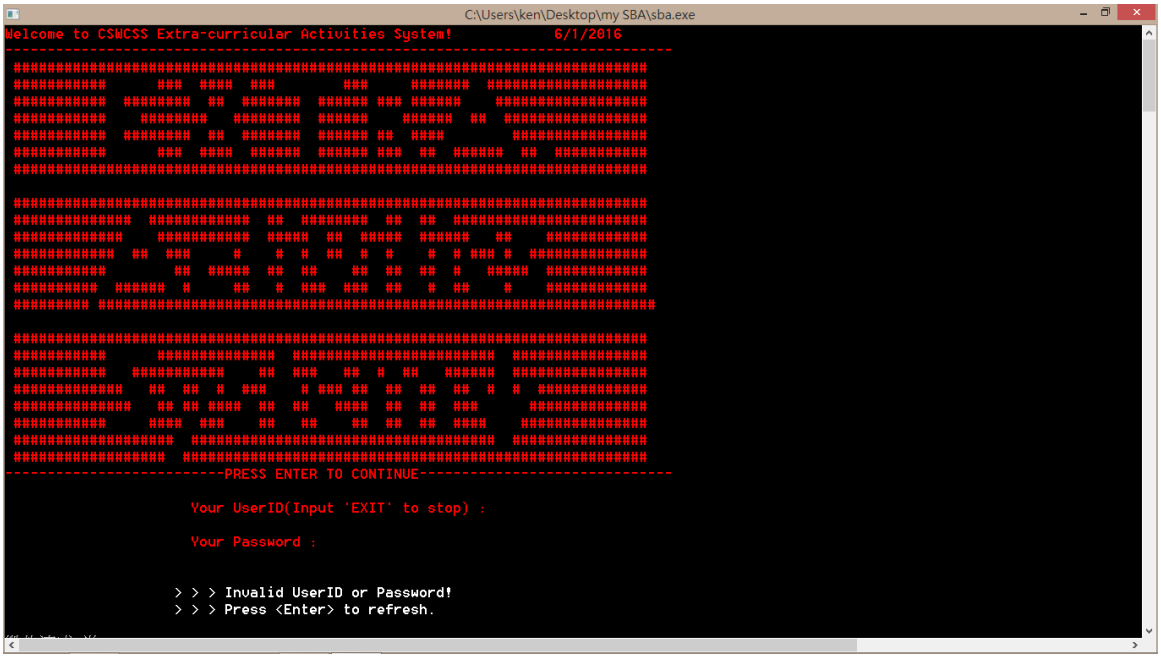
Last but not least, I will modify the program to make it perfect for users to use.

4.1 Internal Testing

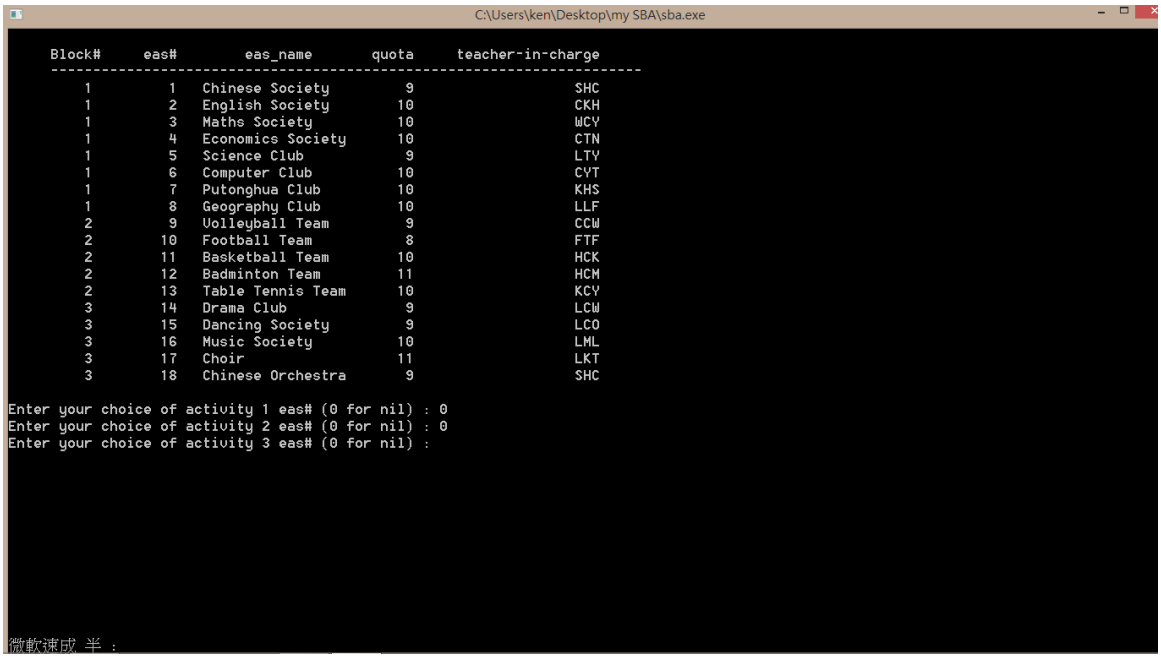
Below is the finial program (After debug)



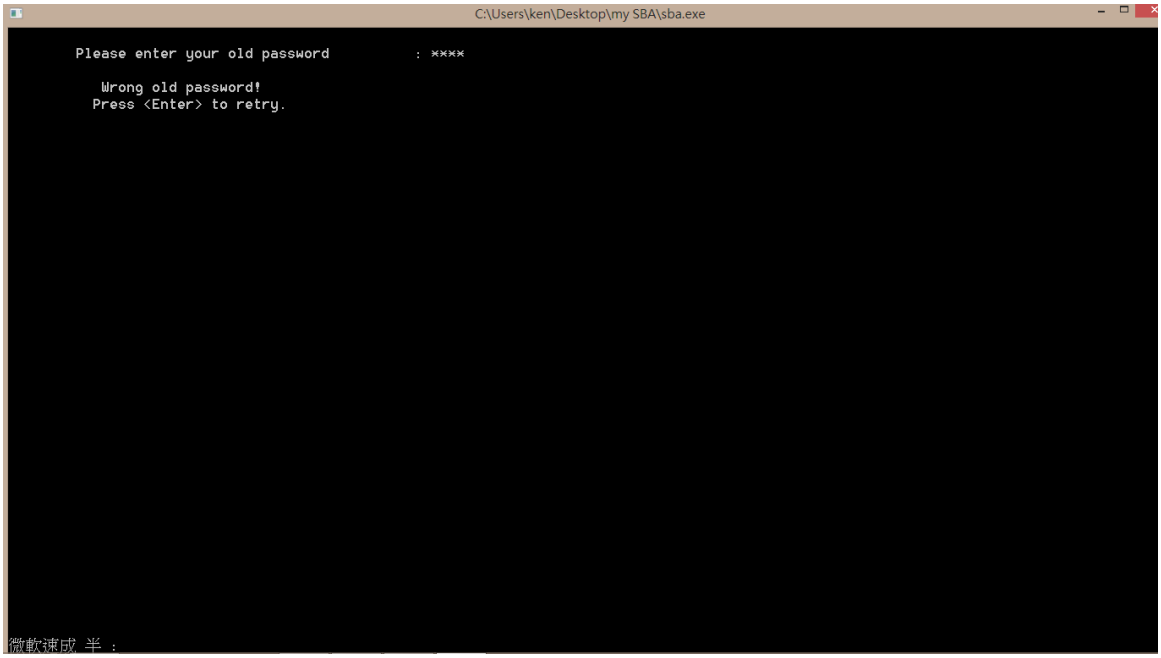
Purpose:	Check the program can detect the wrong input
Input:	Choice a wrong ID or Password
Expected Output:	Invalid User ID or Password
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil



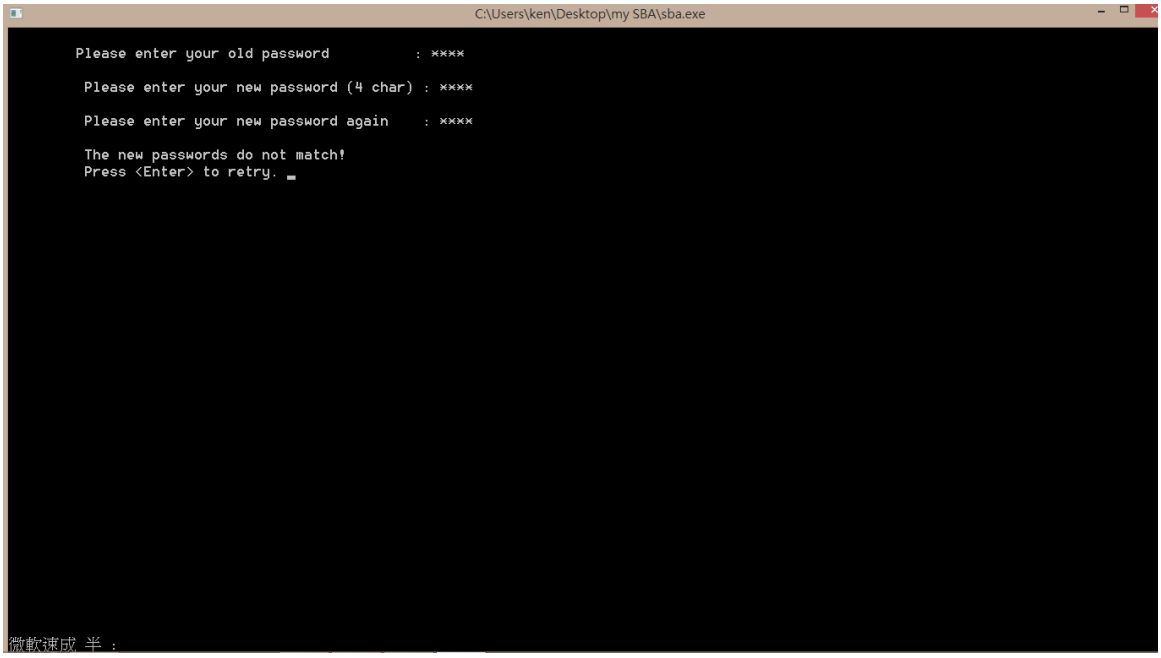
Purpose:	Check the program can detect the wrong input
Input:	Choice a wrong ID or Password
Expected Output:	Invalid User ID or Password & the background will become red
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil



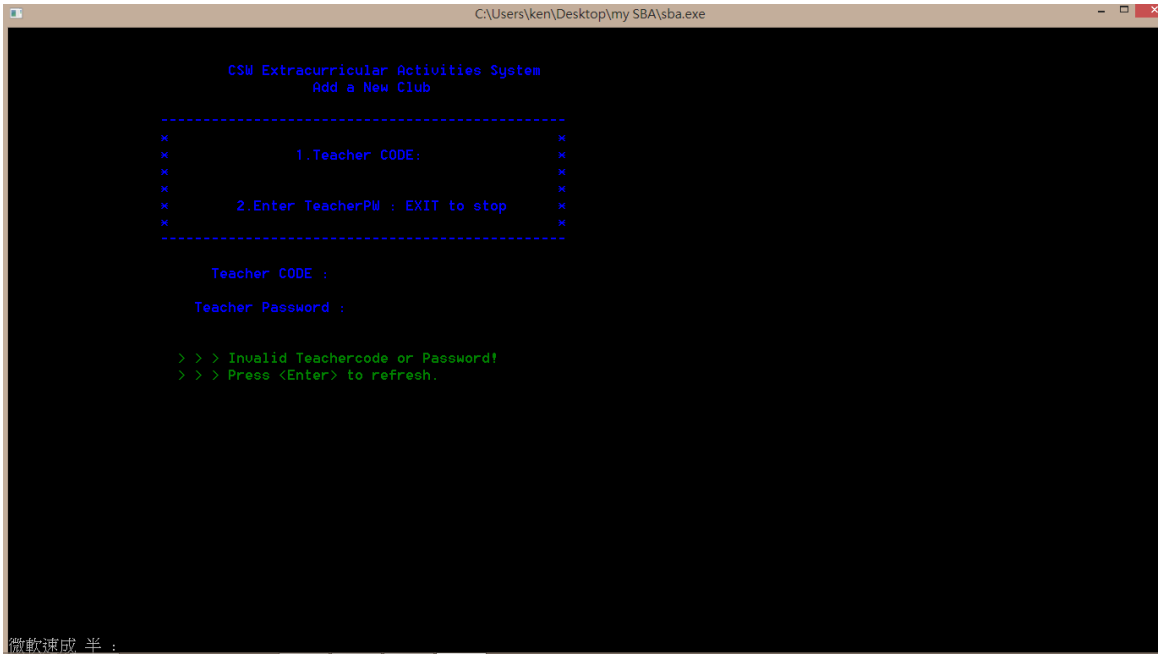
Purpose:	Check the program can detect the wrong input
Input:	Choice some activities which out of the block
Expected Output:	Incorrect choice
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil



Purpose:	Check the program can detect the wrong input
Input:	Wrong 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
Follow-up Action:	Nil

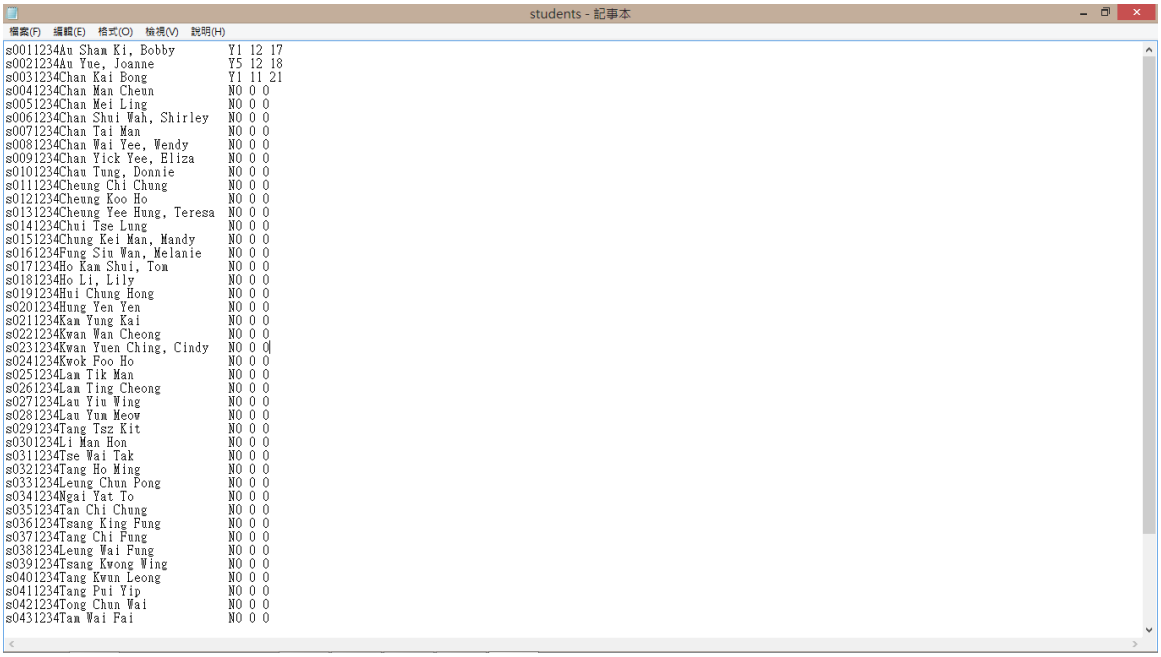


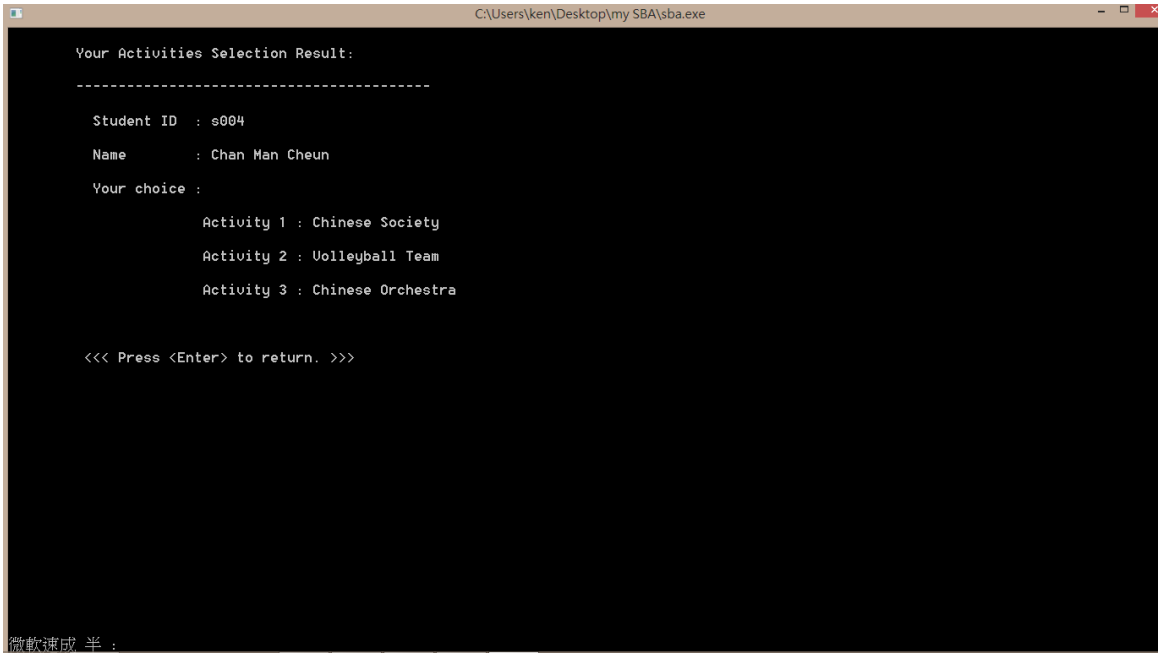
Purpose:	Check the program can detect the wrong input
Input:	A wrong password which not match with the old
Expected Output:	The new passwords do not match! Press <Enter> to retry.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil



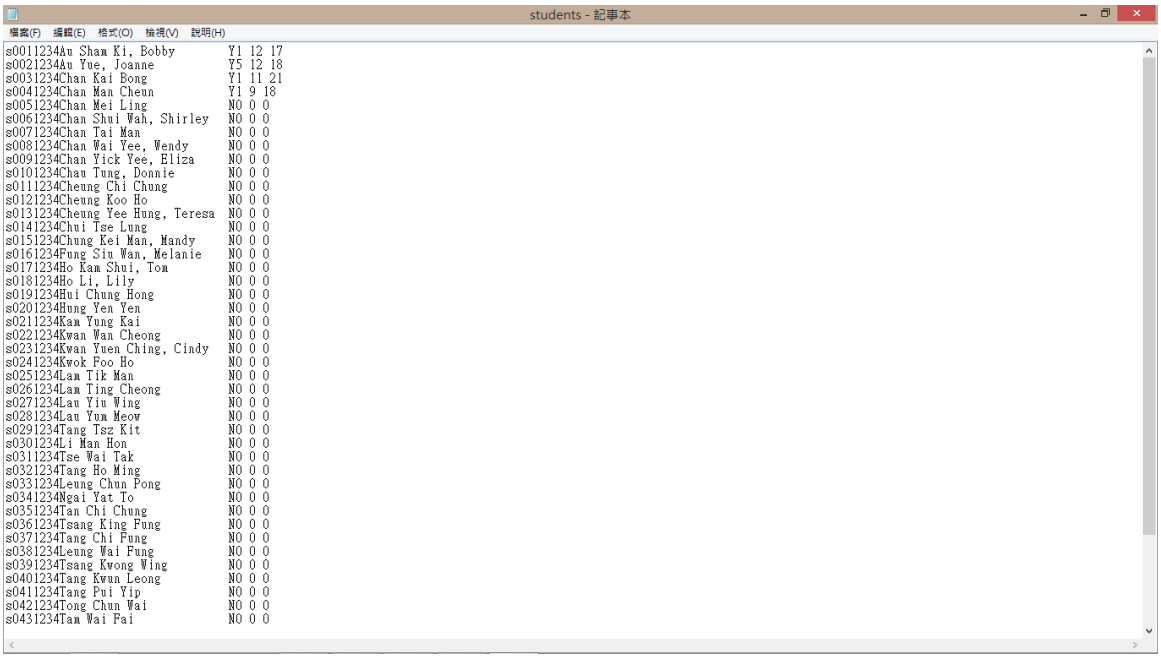
Purpose:	Check the program can detect the wrong input
Input:	Wrong teacher code or password
Expected Output:	> > > Invalid Teachercode or Password > > > Press <Enter> to refresh.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil

Before:


students - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
s0011234Au Sham Ki, Bobby Y1 12 17
s0021234Au Yue, Joanne Y5 12 18
s0031234Chan Kai Bong Y1 11 21
s0041234Chan Man Cheun NO 0 0
s0051234Chan Mei Ling NO 0 0
s0061234Chan Shui Wah, Shirley NO 0 0
s0071234Chan Tai Man NO 0 0
s0081234Chan Wai Yee, Wendy NO 0 0
s0091234Chan Yick Yee, Eliza NO 0 0
s0101234Chau Tung, Donnie NO 0 0
s0111234Cheung Chi Chung NO 0 0
s0121234Cheung Koo Ho NO 0 0
s0131234Cheung Yee Hung, Teresa NO 0 0
s0141234Chui Tse Lung NO 0 0
s0151234Chung Kei Man, Mandy NO 0 0
s0161234Fung Siu Wan, Melanie NO 0 0
s0171234Ho Kam Shui, Tom NO 0 0
s0181234Ho Li, Lily NO 0 0
s0191234Hui Chung Hong NO 0 0
s0201234Hung Yen Yen NO 0 0
s0211234Kam Yung Kai NO 0 0
s0221234Kwan Wan Cheong NO 0 0
s0231234Kwan Yuen Ching, Cindy NO 0 0
s0241234Kwok Foo Ho NO 0 0
s0251234Lam Tik Man NO 0 0
s0261234Lam Ting Cheong NO 0 0
s0271234Lau Yiu Wing NO 0 0
s0281234Lau Yum Meow NO 0 0
s0291234Tang Tsz Kit NO 0 0
s0301234Li Man Hon NO 0 0
s0311234Tse Wai Tak NO 0 0
s0321234Tang Ho Ming NO 0 0
s0331234Leung Chun Pong NO 0 0
s0341234Ngai Yat To NO 0 0
s0351234Tan Chi Chung NO 0 0
s0361234Tsang King Fung NO 0 0
s0371234Tang Chi Fung NO 0 0
s0381234Leung Wai Fung NO 0 0
s0391234Tsang Kwong Wing NO 0 0
s0401234Tang Kwun Leong NO 0 0
s0411234Tang Pui Yip NO 0 0
s0421234Tong Chun Wai NO 0 0
s0431234Tam Wai Fai NO 0 0


C:\Users\ken\Desktop\my SBA\sba.exe
Your Activities Selection Result:
Student ID : s004
Name : Chan Man Cheun
Your choice :
Activity 1 : Chinese Society
Activity 2 : Volleyball Team
Activity 3 : Chinese Orchestra
<<< Press <Enter> to return. >>>
微軟速成 半 :

After:



Purpose:	Check if the program can display correct output in the text file
Input:	The choices of the three activities
Expected Output:	Activity 1 : Chinese Society Activity 2 : Volleyball Team Activity 3 : Chinese Orchestra
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil

4.2 External Testing and Evaluation

I have invited 20 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.

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

Testing and Evaluation Form

Please help to evaluate the program: Multiple-Choice Analysis. Thanks!

Instruction:

- Please execute the program according to the instructions in the ReadMe.txt file in the program folder.

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 program is user-friendly	5	4	3	2	1
2.	The design of the input screen is good	5	4	3	2	1
3.	The format of the report is good	5	4	3	2	1
4.	The ……. functions are useful	5	4	3	2	1

Which features you like most?

_____

What other functions should be provided by the program?

_____

_____

_____

Other Suggestions

_____

_____

_____

Evaluator's Name: _____ **Date:** _____

Chapter 5 Conclusion & Discussion

5.1 Future Improvement

In future, it is hoped that the program can be updated each year, such as the activity 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.

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

5.2 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 extracurricular_activities_selection;

uses Crt,dos;

const
    max_eas= 100;
    max_stud= 100;
    max_tea= 20;



var
  studid:array[1..max_stud]of string[4];
  studpw:array[1..max_stud]of string [4];
  studname:array[1..max_stud]of string[25];
  studdone:array[1..max_stud]of char;
  studeas1:array[1..max_stud]of integer;
```

```
    studeas2:array[1..max_stud]of integer;
    studeas3:array[1..max_stud]of integer;
    teacode:array[1..max_tea]of string[3];
    teapw:array[1..max_tea]of string[4];

    neweas1:array[1..max_tea]of string[30];
    year,month,day,timeofday:word;



    activity:array[0..max_eas]of string[17];
    block:array[0..max_eas]of integer;
    quota:array[0..max_eas]of integer;
    teaname:array[0..max_eas]of string[3];


    num_eas,num_stud,stud_index,num_tea,tea_index:integer;
    exit1:boolean;

procedure moment;
begin
   getdate(year,month,day,timeofday);
end;

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;
                  readln(f, studid[i], studpw[i], studname[i], studdone[i], studeas1[i], studeas2[i], studeas3[i]);
              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
              writeln(f, studid[i], studpw[i], studname[i], studdone[i], studeas1[i],' ', studeas2[i],' ', studeas3[i]);
           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;
                  readln(f, teacode[i], teapw[i]);
              end;
      num_tea:=i;
      close(f);
end;

procedure store_teachers;
var i : integer;
      f : text;
begin
        assign(f, 'teachers.txt');
        rewrite(f);
        for i := 1 to num_stud do
```

```
            begin
                writeln(f, teacode[i], teapw[i]);
            end;
    close(f);
end;



procedure read_activities;
var i : integer;
      f : text;
begin
        assign(f, 'clubs.txt');
        reset(f);
        i := -1;
        while not eof(f) do
            begin
                i := i+1;
                readln(f, activity[i], teaname[i],block[i], quota[i]);
            end;
        num_eas := i;
        close(f);
end;

procedure store_activities;
var i : integer;
      f : text;
begin
        assign(f, 'clubs.txt');
        rewrite(f);
        for i := 0 to num_eas do
            begin
                writeln(f, activity[i], teaname[i],block[i], ' ', quota[i]);
            end;
        close(f);
end;
```

```
procedure display_activities;
var i : integer;
begin
   clrscr;
   writeln;
   writeln(':5,'Block#         eas#              eas_name            quota         teacher-in-charge');
   writeln(':5,'-------------------------------------------------------------------------');
   for i := 1 to num_eas do
      writeln(':5,block[i]:5, i:10, activity[i]:20, quota[i]:8,teaname[i]:22);
   readln();
end;



procedure display_activities_return;
begin
   display_activities;
   write('                <<< Press <Enter> to return. >>>');
end;

procedure display_choices(stud_index : integer);
begin
   clrscr;
   writeln;
   writeln('              Your Activities Selection Result: ');
   writeln;
   writeln('              ------------------------------------------ ');
   writeln;
   writeln('                 Student ID    : ', studid[stud_index]);
   writeln;
   writeln('                 Name              : ', studname[stud_index]);
   writeln;
   writeln('                 Your choice : ');
   writeln;
   writeln('                                    Activity 1 : ', Activity[studeas1[stud_index]]);
   writeln;
   writeln('                                    Activity 2 : ', Activity[studeas2[stud_index]]);
   writeln;
   writeln('                                    Activity 3 : ', Activity[studeas3[stud_index]]);
```

```
    writeln;
    writeln;
    writeln;
    write('                <<< Press <Enter> to return. >>>');
    readln();
end;



procedure choose_activities(stud_index : integer);
var
        i : integer;
        choice : array[1..3] of integer;
        choice_done : boolean;
begin
      {Reset the quota}
      quota[studeas1[stud_index]] := quota[studeas1[stud_index]] + 1;
      studeas1[stud_index] := 0;
      quota[studeas2[stud_index]] := quota[studeas2[stud_index]] + 1;
      studeas2[stud_index] := 0;
      quota[studeas3[stud_index]] := quota[studeas3[stud_index]] + 1;
      studeas3[stud_index] := 0;

      quota[0] := 0;
      studdone[stud_index] := 'N';

      display_activities;

      for i := 1 to 3 do
          begin
              choice_done := false;
              repeat
                  write('Enter your choice of activity ', i, ' eas# (0 for nil) : ');
                  readln(choice[i]);
                  if (choice[i] = 0) then

                      choice_done := true
                  else
```

```
                    if (block[choice[i]] <> i) then
                       writeln('Incorrect choice!')
                    else
                       if (quota[choice[i]] = 0) then
                          writeln('Not enougt quota!')
                    else
                       choice_done := true
                until choice_done;
          end;
  studeas1[stud_index] := choice[1];
  quota[choice[1]] := quota[choice[1]] - 1;
  studeas2[stud_index] := choice[2];
  quota[choice[2]] := quota[choice[2]] - 1;
  studeas3[stud_index] := choice[3];
  quota[choice[3]] := quota[choice[3]] - 1;

    quota[0] := 0;
    studdone[stud_index] := 'Y';

    { Update data files }
    store_students;
    store_activities;

    display_choices(stud_index);
end;



{ Ref: http://computer-programming-forum.com/29-pascal/7af4f3f05f738777.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 <> studpw[stud_index] 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 pf 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
                    studpw[stud_index] := newpass1;
                    store_students;
                    pwchanged := true;
                    writeln('              Password changed.');
                    write('              Press <Enter> to return. ');
                    readln
                  end
              end
          end
  until pwchanged;
end;


procedure New_Club_Adding(teacher:string[3]);
var club_n:string[17];
      club_b,club_q,i,j,temp:integer;
      only:boolean;
```

```
    f:text;
begin
    clrscr;
    only:=true;
    textcolor(white);
    assign(f,'clubs.txt');
    writeln('-------------------------------------------------------------------------------');
    writeln('                                          Adding New Club
');
    writeln('-------------------------------------------------------------------------------');
    repeat
        write('                    1.Club name(within 17 strings):');
        readln(club_n);
        if length(club_n)>17
            then writeln('The length of the club name should be within 17 strings.');
    until length(club_n)<=17;

    writeln('          #                                                                    #');
    writeln('          #                                                                    #');
    writeln('          ---------------------------------------------------------------');
    writeln('          #                                                                    #');
    writeln('          #                                                                    #');
    repeat
    write('                    2.Club block(1/2/3):');
    readln(club_b);
    if (club_b <> 1) and (club_b <> 2) and (club_b <> 3)
        then writeln('Not available! Enter again!')
    until (club_b <> 1) or (club_b <> 2) or (club_b <> 3);
    writeln('          #                                                                    #');
    writeln('          #                                                                    #');
    writeln('          ---------------------------------------------------------------');
    writeln('          #                                                                    #');
    writeln('          #                                                                    #');
    repeat
    write('                    3.Club quota:( >0 and <=30 )');
    readln(club_q);
    until (club_q >0) and (club_q <=30);
    writeln('          #                                                                    #');
```

```
writeln('              #                                                                #');
writeln('              ----------------------------------------------------------------');
writeln('              #                                                                #');
writeln('              #                                                                #');
case    club_b of
1:begin
          rewrite(f);
          for i:= 0 to num_stud do
               if (block[i]=2) and only then begin temp:=i;only:=false; end;
          for j:= 0 to temp-1 do
               writeln(f,activity[j],teaname[j],block[j],' ',quota[j]);
          for j:=1 to (17-length(club_n)) do
               club_n:=club_n+' ';
          writeln(f,club_n,teacher,club_b,' ',club_q);
          for j:= (temp+1) to num_eas do
               writeln(f,activity[j],teaname[j],block[j],' ',quota[j]);
          num_eas:=num_eas+1;
   end;
2:begin
          rewrite(f);
          for i:= 0 to num_stud do
               if (block[i]=3) and only then begin temp:=i;only:=false; end;
          for j:= 0 to temp-1 do
               writeln(f,activity[j],teaname[j],block[j],' ',quota[j]);
          for j:=1 to (17-length(club_n)) do
               club_n:=club_n+' ';
          writeln(f,club_n,teacher,club_b,' ',club_q);
          for j:= (temp+1) to num_eas do
               writeln(f,activity[j],teaname[j],block[j],' ',quota[j]);
          num_eas:=num_eas+1;
   end;
3:begin
          append(f);
          for j:=1 to (17-length(club_n)) do
               club_n:=club_n+' ';
          writeln(f,club_n,teacher,club_b,' ',club_q);
          num_eas:=num_eas+1;
   end;
```

```
        end;
        close(f);
        writeln('The club is successfully created! Press "Enter" to return to main menu!!');
        readln();
end;




procedure Add_a_new_club;
var
    userid, password : string;
    found : boolean;
    i : integer;
begin
        clrscr;
        writeln;
        writeln;
        writeln('                              CSW Extracurricular Activities System                    ');
        writeln('                                        Add a New Club                                  ');
        writeln;
        writeln('                    ------------------------------------------------        ');
        writeln('                    *                                                 *        ');
        writeln('                    *                1.Teacher CODE:                  *        ');
        writeln('                    *                                                 *        ');
        writeln('                    *                                                 *        ');
        writeln('                    *          2.Enter TeacherPW : EXIT to stop       *        ');
        writeln('                    *                                                 *        ');
        writeln('                    ------------------------------------------------        ');
        writeln;
        write('                          Teacher CODE : ');
        readln(userid);
        writeln;
        write('                        Teacher Password : ');
        readln(password);
        writeln;
```

```
      writeln;
   found:=false;
   for i:=1 to max_tea do
        if userid=teacode[i] then
                                          if password=teapw[i] then found:=true;


 if not found then
      if userid='EXIT'
          then begin tea_index :=0; found:=true; end
               else
                    begin
                         tea_index :=0;
                         textcolor(green);
                         writeln(':20,'> > > Invalid Teachercode or Password!');
                         writeln(':20,'> > > Press <Enter> to refresh.');
                         textcolor(lightblue);
                         readln
                         end;
    if (not found )
        then Add_a_new_club;
    if found
    then New_Club_Adding(userid);
end;




procedure main_menu(stud_index : integer);
var
   choice : integer;
begin
   repeat
      clrscr;
      read_activities;
```

```
    textcolor(lightgray);
    writeln;
    writeln('                    Extracurricular Activities System ');
    writeln('                                Main Menu');
    writeln;
    writeln('                  ----------------------------------------');
    writeln('                  *                                      *');
    writeln('                  * 1. Display Activities List           *');
    writeln('                  *                                      *');
    writeln('                  *      2. Display My Choices           *');
    writeln('                  *                                      *');
    writeln('                  *          3. Select/Update Choices    *');
    writeln('                  *                                      *');
    writeln('                  *              4. Change password      *');
    writeln('                  *                                      *');
    writeln('                  *                   5. Add a new club  *');
    writeln('                  *                                      *');
    writeln('                  *                        6. Exit       *');
    writeln('                  *                                      *');
    writeln('                  ----------------------------------------');
    if studdone[stud_index] = 'N' then
      begin
        textcolor(green);
        writeln('            You have not chosen the activities!');
        textcolor(lightgray)
      end;
    writeln;
    write('            Enter choice: ');
    readln(Choice);
    writeln;
    case choice of
      1 : display_activities_return;
      2 : display_choices(stud_index);
      3 : choose_activities(stud_index);
      4 : change_password(stud_index);
      5 : Add_a_new_club;
    end;
  until choice = 6;
```

```
end;


procedure login(var stud_index : integer);
var
   userid, password : string;
   found : boolean;
   i : integer;
begin

      clrscr;
writeln('Welcome to CSWCSS Extra-curricular Activities System!','                ',day,'/',month,'/',year);
writeln('--------------------------------------------------------------------------------');
writeln(' ################################################################################');
writeln(' ###########         ###   ####   ###          ###       #######   ##################');
writeln(' ###########   ########   ##   #######   ###### ### ######      ##################');
writeln(' ###########      ########    ########   ######     ######   ##   ##################');
writeln(' ###########   ########   ##   #######   ###### ##   ####          ################');
writeln(' ###########         ###   ####   ######   ###### ###   ##   ######   ##   ###########');
writeln(' ################################################################################');
writeln('                                                                                ');
writeln(' ################################################################################');
writeln(' ##############   ############   ##   ########   ##   ##   ######################');
writeln(' #############      ###########   #####   ##   #####   ######    ##     ###########');
writeln(' ############   ##   ###       #     #   #   ##   #   #     #   # ### #   ##############');
writeln(' ###########           ##   #####   ##   ##     ##   ##   ##   #    #####   ###########');
writeln(' ##########   ######   #       ##    #   ###   ###   ##    #   ##     #     ###########');
writeln(' ######### ##############################################################################');
writeln('                                                                                ');
writeln(' ################################################################################');
writeln(' ###########         ##############   ########################   ################');
writeln(' ###########    ###########      ##   ###    ##   #   ##    ######   ################');
writeln(' #############    ##   ##   #   ###      # ### ##   ##   ##   ##   #   #   #############');
writeln(' ##############    ## ## ####   ##   ##    ####   ##   ##   ###        ##############');
writeln(' ###########       ####   ###     ##    ##     ##   ##   ##   ####     ##############');
writeln(' ###################   ###################################   ################');
writeln(' ##################   ########################################################');
writeln('--------------------------PRESS ENTER TO CONTINUE-----------------------------');
```

```
      readln();
      write('                          Your UserID(Input "EXIT" to stop) : ');
      readln(userid);
      writeln;
      write('                          Your Password : ');
      password := GetPWord;
      writeln;
      writeln;

   found := false;
   i := 0;
   while (i < num_stud) and (not found) do
      begin
         i := i + 1;
         if (userid = studid[i]) and (password = studpw[i]) then
            begin
               found := true;
               stud_index := i
            end
      end;
  if not found then
      if userid='EXIT'
          then begin stud_index :=0; exit1:=true; end
                else
                    begin
                          stud_index :=0;
                          textcolor(white);
                          writeln(':20,'> > > Invalid UserID or Password!');
                          writeln(':20,'> > > Press <Enter> to refresh.');
                          textcolor(lightred);                    readln

                          end
end;


begin {main body}
   exit1:=false;
   moment;
```

```
    read_students;
    read_teachers;
    read_activities;
    repeat
       login(stud_index);
       if stud_index <> 0 then
          main_menu(stud_index)
    until exit1;
    readln
end.
```

Appendix2: 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

THE END