Hong Kong Diploma of Secondary Education Examination 2016

Information and Communication Technology (Coursework)

Option D: Software Development

Title: Venue Booking System

Contents

Chapter 1 Introduction

1.1 Background

Several secondary schools whose venue booking service are still operated manually would like to have a electronic venue booking system to enhance the cost-effectiveness of the venue booking service in their schools. The system should provide venue booking service that can handle the differences between schools (e.g. number of students). I am the IT project manager responsible for the project. I am going to provide a solution for their school.

As the information technology advances every day, more and more of our daily operations are completed on the Internet and computer system. Yet, electronic venue booking system for schools is still not popular. Therefore, the main goal of my study would be creating a flexible venue booking system that can suit different secondary school so that the accuracy and efficiency of venue booking service in a large number of schools can be enhanced.

1.2 Objectives

In this project, I am going to develop a venue booking system (aka VBS) for schools. The users of VBS would be selected by the administrator who is the first person launching VBS for the first time. User accounts created by the administrator have limited permission to the functions. The Venue and date available for booking is customizable, as well as the school name.

The system supports the following functions:

1. Registration (admin only) and Login for all the users and admin

2. Allow the admin to have full control on the venue, date and user customization

3. A full list of booking records for admin

4. Changing password for admin and users

5. Display the availability of the venues

6. Update new and outdated booking records

Chapter 2 Design of Solution

2.1 Proposed Functions of the System

In this chapter, I will design the program based on the functions I proposed in Chapter 1.

Designs of the VBS:

1. A general structure for each function page
2. A signup and login system
3. User-friendly menu pages
4. Functions provided for admin:
   i) Add/Remove users
   ii) Add/Remove venues
   iii) Check all booking records
   iv) Set School name for the school
5. Common Functions:
   i) Make a booking
   ii) Cancel a booking
   iii) Change password
6. Functions for VBS:
   i) Check the existence of the database

ii) Add/Remove new, outdated records to the database

7. Database formats

8. Divide the VBS program into 9 main parts and separated functions among all the main parts

Functions that enhance user-friendliness of the user-interface:

1. Password hidden in login, signup and change password sections

2. Data validation on all inputs

3. Escape button programmed for users to proceed to the previous page

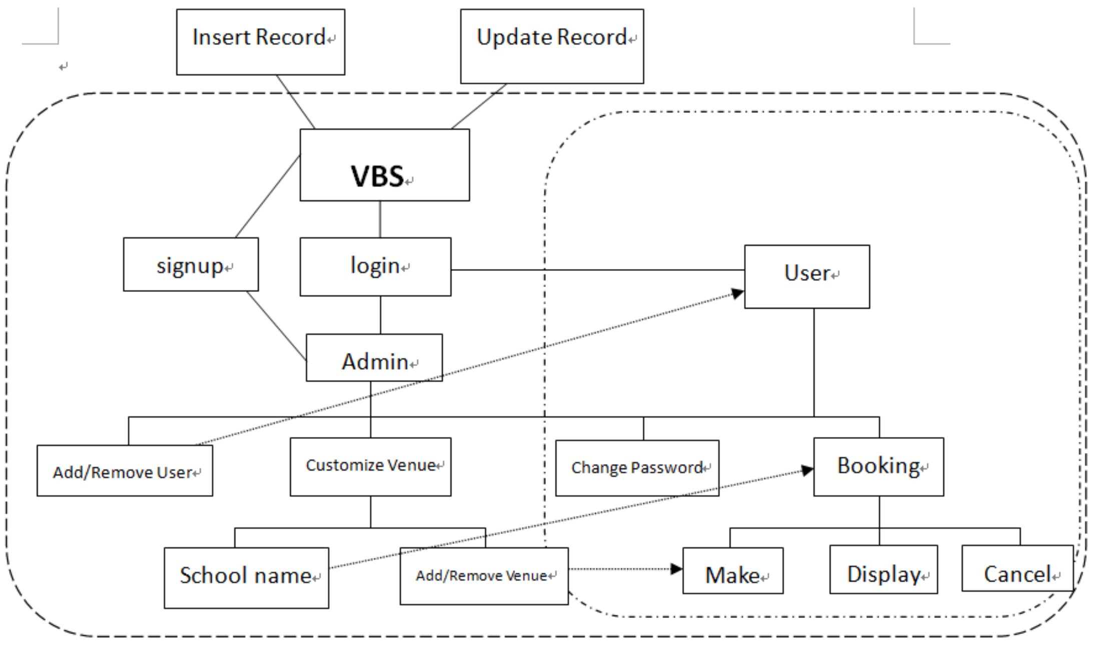
4. Extreme cases, e.g. no record/incomplete admins' customization for their schools

5. Prevent missing database from interfering the system's accuracy

2.2 Refinement

2.2.1 Design

The refinement to the design is as follows:



The first thing VBS would do when it is launched is to insert records from the database. If the database is missing, a warning message would pop up and VBS would not proceed until the database is presented. Once VBS starts its operation, it would update the records to the database whenever a confirmed change in data is made

For the first time the system is launched, the administrator would be enquired to register an "admin" account for himself. After that, anyone who launches the program would be asked to input valid user ID and password for further functions of VBS. And the user ID and password have to be created by the administrator in his account. Also, the administrator has o input the school name to start the venue booking service, but venues/rooms have to be added by the administrator. Otherwise, there wouldn't be any venue available for booking.

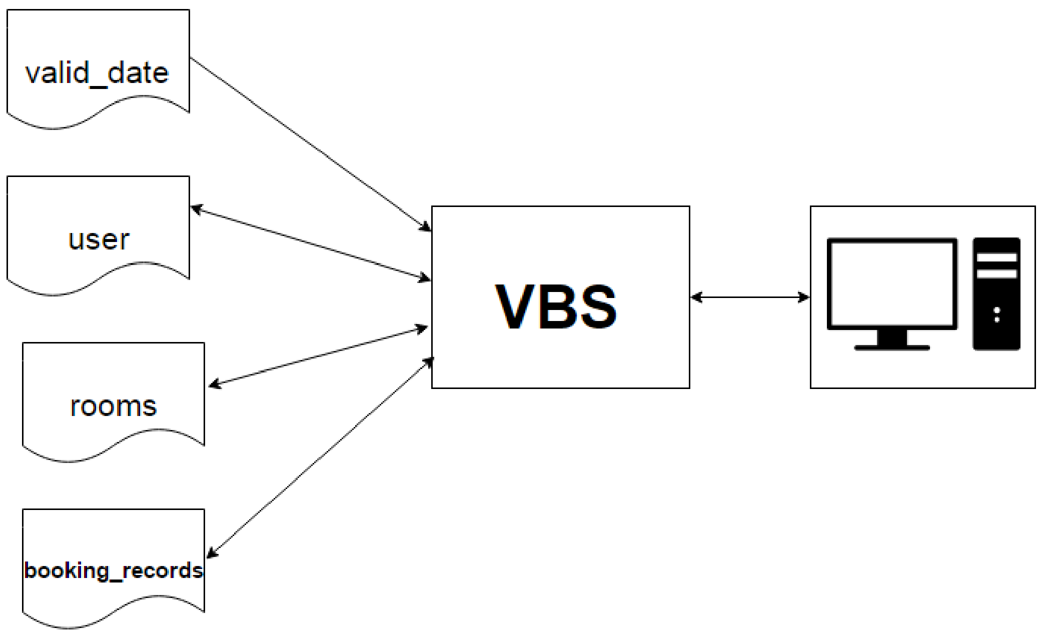
The administrator has access to all functions, while users only have access booking and change password. Also, only admin can read all the record while users can only read their own booking records. Indeed, administrator's operations on users and venue management would have certain influence to the booking record. For example, removing a user will also remove all his booking records.

2.2.2 user-interface

The refinement to the UI is as follows:

Situations	Solution
1. Hide password in login/change password/signup	Write '*' when the user press a key and apply other key functions (e.g. backspace, enter)
2. Invalid Inputs	-Show Error message -disallow further operations and ask the users to input again
3. 'Escape' function	Press 'Escape' key can go to the previous page/main menu/desktop (depends on the location pressed)
4. Nothing to remove / No school name	Disallow the remove Venue or User function / Disallow the booking function
5. Missing Database	If VBS cannot access the whole database at launch, error message and suggested solution will be displayed on screen

2.2.3 Data Flow



Except valid_date, VBS would update and insert data to all text files because the administrator can control the venues available for booking and select users to use VBS. VBS would only insert the data from valid_date for further process.

When VBS detects any missing files, the program would stop operating and display an error message until the consistency of the text files is restored.

2.3 Data File Formats

2.3.1 Valid date for booking

The file storing Valid date for booking – ‘valid_date’.

Valid_date is used to store all the valid date available for booking and VBS would filter all the futuristic date available for booking automatically.

It stores the following data per line of the file:

Available date:

i) Year (4 Words)
ii) Month (1-2 Words)
iii) Day (1-2 Words)

File structure:

Year (4 Words)	Month (1-2 Words)	Day (1-2 Words)
2016	3	11

Sample file (valid_date)

```
2016/01/23
2016/01/24
2016/01/25
2016/01/26
2016/01/27
2016/01/28
```

2.3.2 User Information

The file storing Valid date for booking – ‘user’.

User is used to store all the User ID, passwords and permissions all users of VBS.

It stores the following data per 3 lines of the file:

1) User ID (various length of string)
2) Password (various length of string)
3) Permission (1 integer) (To identify the user group of the users)

File Structure:

User ID (various length of string)	Password (various length of string)	Permission (1 integer)
Mr. To	12345	2
S107	1234	1
Ken	3689	1

Sample file (user)

```
Mr. To
12345
2
S107
1234
1
Ken
3689
1
```

2.3.3 Venues available for booking

The file storing Valid date for booking – ‘rooms’.

Rooms is used to store the school name and all the venues inputted by the admin available for booking.

It stores the following data per line of the file:

1) School name (various length ofstring)
2) Venue (various length of string)

File Structure:

Venue (various length of string)	School name (various length of sting)
Library	Cheung Sha Wan Catholic Secondary School

Sample file (rooms)

```
Cheung Sha Wan Catholic Secondary School
Room 201
Room 202
Room 203
Room 204
Library
```

2.3.4 Booking records

The file storing Valid date for booking – ‘booking_records’.

Booking_records is used to store the date, venue, time and booking user for every booking record.

It stores the following data per 4 lines of the file:

1) Date
   i) Year (4 words)
   ii) Month (1-2 words)
   iii) Day (1-2 words)
2) User (various length of string)
3) Venue (1 integer)
4) Time (1 integer)

File Structure

Date	User	Venue	Time
2016/2/3	Mr. To	2	3

Sample file (rooms)

```
2016/2/3
Mr. To
2
3
2016/1/5
Ken
1
3
```

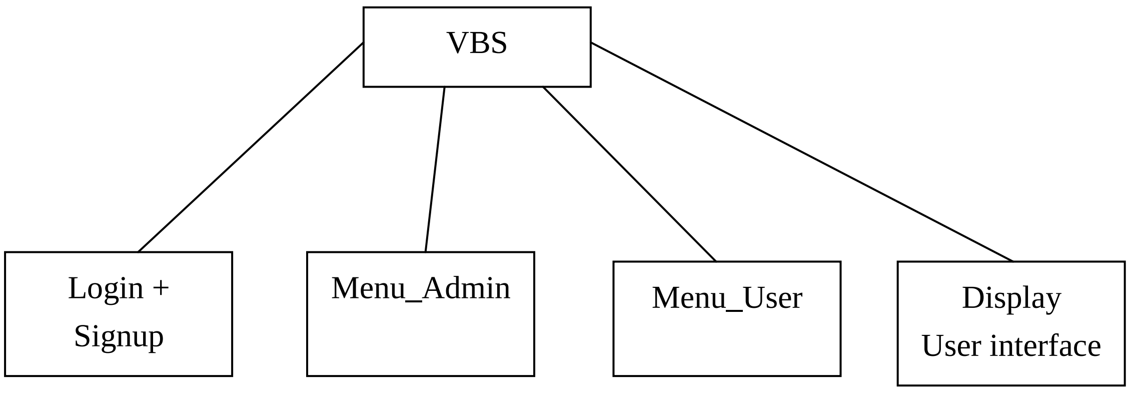
Chapter 3 Implementation

3.1 Description

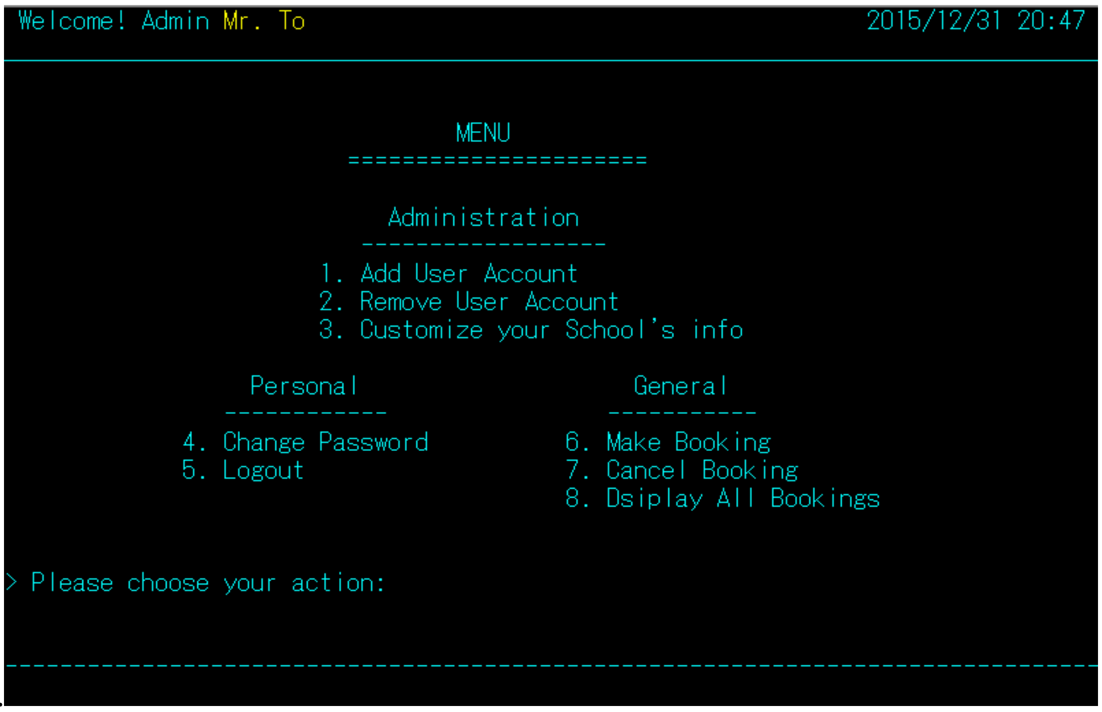
In this chapter, the implementation of the program VBS is going to be discussed in detail – program structure, procedures and functions, program coding and program execution.

3.2 Program Structure

The following figure shows the program outline:

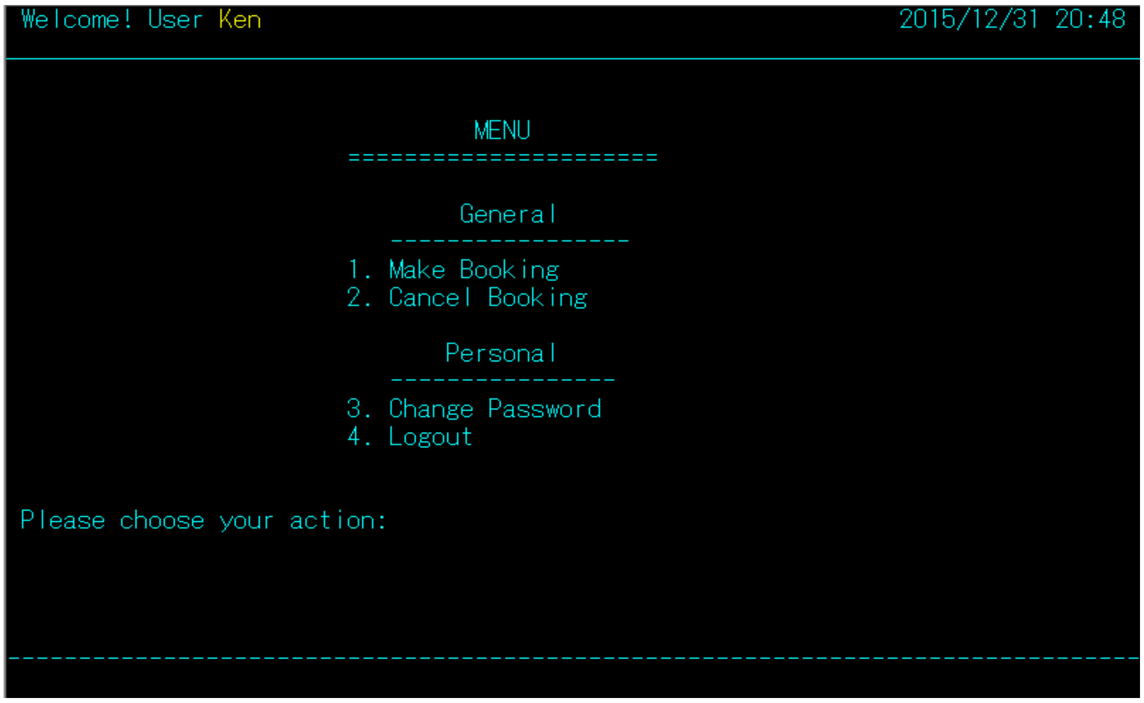


All the main functions of VBS is under Menu_Admin and Menu_User units depending on their permission level.



```
Welcome! Admin Mr. To                                    2015/12/31 20:47


                              MENU
                      ======================

                          Administration
                       ------------------
                    1. Add User Account
                    2. Remove User Account
                    3. Customize your School's info

               Personal                      General
              ------------                  -----------
          4. Change Password           6. Make Booking
          5. Logout                    7. Cancel Booking
                                       8. Dsiplay All Bookings


> Please choose your action:


-------------------------------------------------------------------------------
```

.



```
Welcome! User Ken                                          2015/12/31 20:48

                                   MENU
                          ======================

                                 General
                             -----------------
                          1. Make Booking
                          2. Cancel Booking

                                 Personal
                             ----------------
                          3. Change Password
                          4. Logout

Please choose your action:

---------------------------------------------------------------------------
```

Next, the following shows the basic structure of each program and unit;

1.VBS (Core)

a) Insert Records (user information, venues and date available for booking, booking records)

b) Write Text File (Update records to database when inputs detected)

c) Input (Hide Password, 'Escape Function')

2.Signup

a) Check if user information is empty

Yes > adopt registration procedure

2.1Login

Ask Admin and Users to login

3.Menu_Admin

a) Add User Account

b) Remove User Account

c) Customize School's info

d) Change Password

e) Logout

f) Make Bookings

g) Cancel Bookings

h) Display all bookings

4.Menu_User

a) Make Booking

b) Cancel Booking

c) Change Password

d) Logout

3.3 Data Structure

The following records are used to store all the booking records
and user informations:

```
UserData = record
                id : string;
                pw : string;
                permission : string;
             end;

 BookingData = record
                Year : word;
                Month : word;
                Day : word;
                user : string;
                venue : integer;
                time : integer;
             end;
```

The following one-dimensional arrays store the venues available for booking:
And valid dates inputted by the admin

```
room : array[1..500] of string;
yyyy, mm, dd : array[1..400] of word;
```

The following variables stores the user_id, password, permission
of the users logged in to the system:

```
user_id, password, permission: string;
```

3.4 Procedures & Functions

The program VBS can be divided into 13 parts, which contains 1 main program and 12 procedures/Functions. Meanwhile, there are 8 sub-programs and 4 sub-programs under Menu_Admin and Menu_User respectively . The following are the description of each part and how each part achieves the purposes of VBS.

[1a] – Insert Record

Variables:

- I : integer
- Total_booking, total_room, total, total-date : integer
- userR (records)
- Rooms :array[1..500] of string
- Schoolname : string
- Yyyy, mm, dd (3 arrays of words)
- Booking_record (records)

Features:

Contains three main while loops, reading all the data from the database Into appropriate arrays and records. All the variables with "total" count The total number of data inserted into each array and record. In inserting Booking records and valid dates, extra algorithm are added to extract the Year, month and day from the date string.

```
i := 0;
   reset(user);
   while not eof(user) do
   begin
        i := i + 1;
        with UserR[i] do
        begin
             readln(user, id);
             readln(user, pw);
             readln(user, permission);
        end;
        total := total + 1;
   end;
   close(user);
```

```
i := 1;
   reset(valid_date);
   While not eof(valid_date) do
   begin
      readln(valid_date, W);

      A := copy(W, 1, 4);
      val(A, yyyy[i], error);
      delete(W, 1, 5);
      A := copy(W, 1, (pos('/', W) - 1));
      val(A, mm[i], error);
      delete(W, 1, pos('/', W));
      A := copy(W, 1, length(W));
      val(A, dd[i], error);

      i := i + 1;

      total_date := total_date + 1;
   end;
   close(valid_date);
```

[1b] – WriteText

Variables:

- userR (records)
- Booking_record (records)
- Rooms :array[1..500] of string

Features:

Update and write user all information, booking records and venues available for booking into the database

```
rewrite(user);
   for i := 1 to total do
   begin
      with userR[i] do
      begin
         writeln(user, id);
         writeln(user, pw);
         writeln(user, permission);
      end;
   end;
   close(user);
```

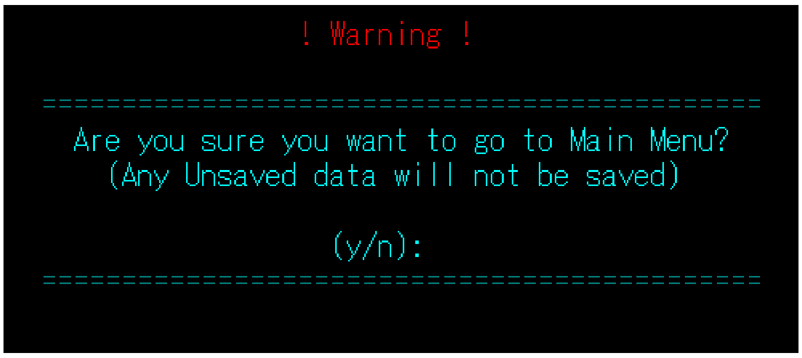
[1c] – Input

Variables:

- S, W, C : string
- Keypressed, done, hide_password : Boolean
- Quit : integer

Features

Detect the key pressed by the user during inputs. The hide_password function can either turn on and off. If turn on, the text inputted by users would be replaced by ‘*’. Also, it detects whether “Escape” key is pressed. If pressed, a message would pop up asking users to confirm the action. If user input yes, input would warp the user into the previous page/main menu/desktop based on the location of the inputs. If not, input would just refresh the page the user is located. Lastly, it also detects other keys like “Enter”, “Backspace” and “Space” to preserve basic input functions.



```
                    ! Warning !

 ==============================================
   Are you sure you want to go to Main Menu?
     (Any Unsaved data will not be saved)

                    (y/n):
 ==============================================
```

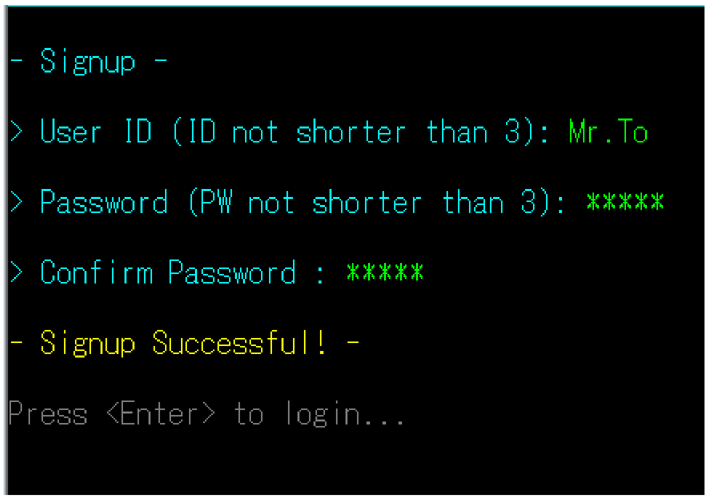
[2a] – Signup

Variables:

- UserR (Record)
- flag[1..3] : array of Boolean
- pw1, pw2, user_id : string

Features:

Detects whether user.txt is empty. If yes, that means there is no user accounts created and signup would launch to require the user to register his admin account with user ID and password. Data validation is featured to make sure the input is long enough to identify each data. The user have to input his chosen password once again to make sure he doesn't input the wrong password and able to login to VBS.



```
- Signup -

> User ID (ID not shorter than 3): Mr.To

> Password (PW not shorter than 3): *****

> Confirm Password : *****

- Signup Successful! -

Press <Enter> to login...
```

[2.1a] – Login

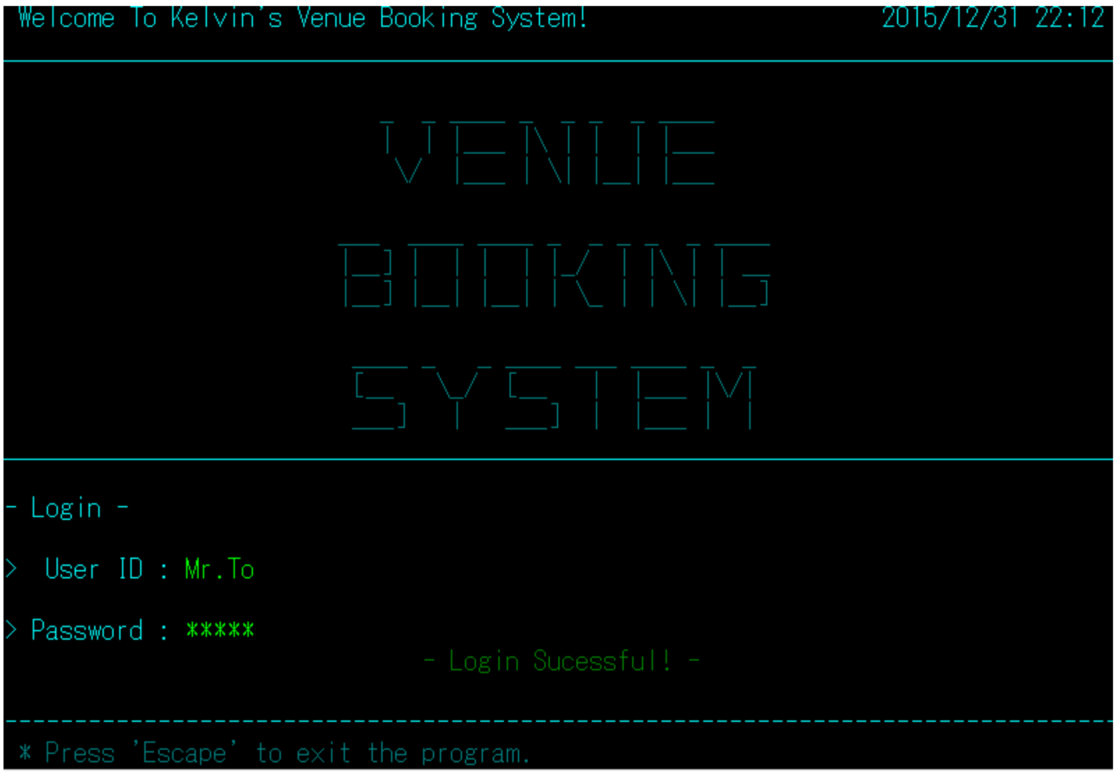
Variables:

- find : Boolean
- user_id, user_num, password, permission
- userR (record)

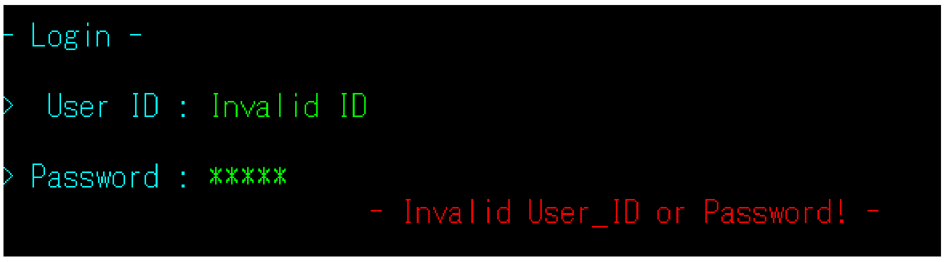
Features:

Once the admin account has be registered or the user information is detected in the database, login would launch to ask the user to login to their accounts with valid user

ID and password. After accepting the inputs, VBS would check the inputted user id and password with all the user ID and password of valid accounts stored in user. If the inputs matches with one of the records, the Boolean becomes true, log the user in into his account and store the user_id, password, permission and record number in user of the logged account. Otherwise, if the user fail to input valid user ID and password, a error message would display on screen and refresh the login
page.



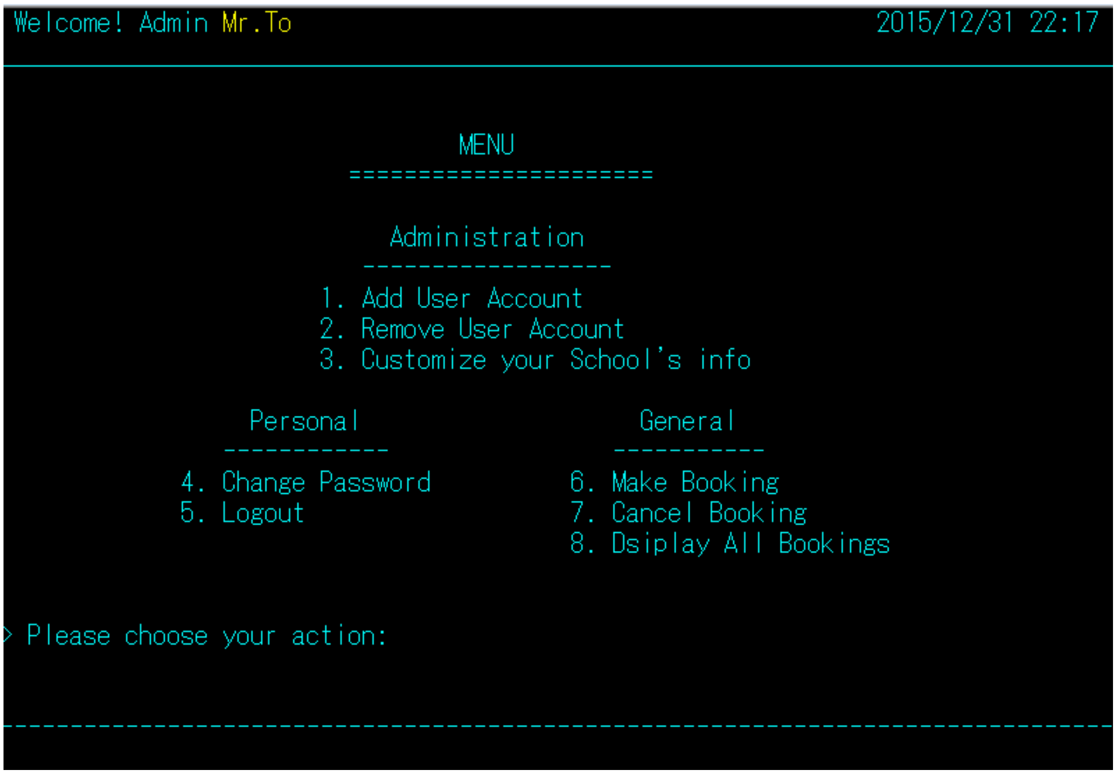
```
Welcome To Kelvin's Venue Booking System!                    2015/12/31 22:12
------------------------------------------------------------------------------

                            VENUE
                           BOOKING
                            SYSTEM
------------------------------------------------------------------------------
- Login -

>  User ID : Mr.To

> Password : *****
                              - Login Sucessful! -

------------------------------------------------------------------------------
 * Press 'Escape' to exit the program.
```



```
- Login -

>  User ID : Invalid ID

> Password : *****
                          - Invalid User_ID or Password! -
```

[3] – Admin Menu

The main menu displayed to admin logged in the system, listing all the functions available and requesting admin's choice on using which function.

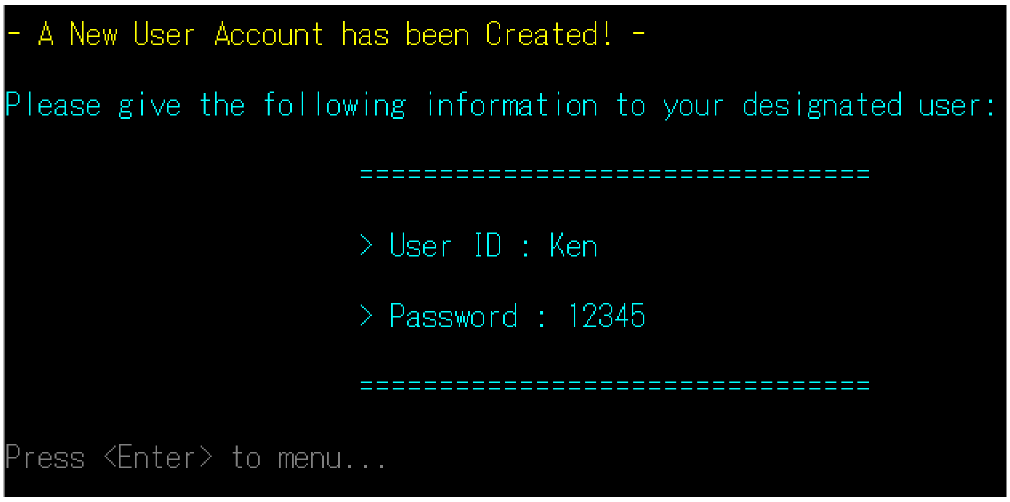


```
Welcome! Admin Mr.To                                  2015/12/31 22:17

                              MENU
                      ======================

                          Administration
                       ------------------
                    1. Add User Account
                    2. Remove User Account
                    3. Customize your School's info

          Personal                     General
         ------------                -----------
       4. Change Password          6. Make Booking
       5. Logout                   7. Cancel Booking
                                   8. Dsiplay All Bookings


> Please choose your action:

------------------------------------------------------------------
```

[3a] – Add User Account

Features:

Provide a signup procedure (user ID, password) for admin to create new user account for the booking service. Admin hs to input his chosen password for the users once again to make sure he doesn't input the wrong password and the users can login to VBS. Data Validation is included to make sure the length of the ID and password have appropriate lengths. Also, VBS would check for duplicate to ensure each user account can be identified. If the input pass the data validation(flag[1], flag[2], flag[3] = TRUE), a success message and the user id and password of the newly created account would be displayed on screen so that admin can distribute the user ID and password to his targeted user. Meanwhile, the input would be added into the userR record and then update the user information In the database. Else, error messages would be displayed on screen and the page would be refreshed.



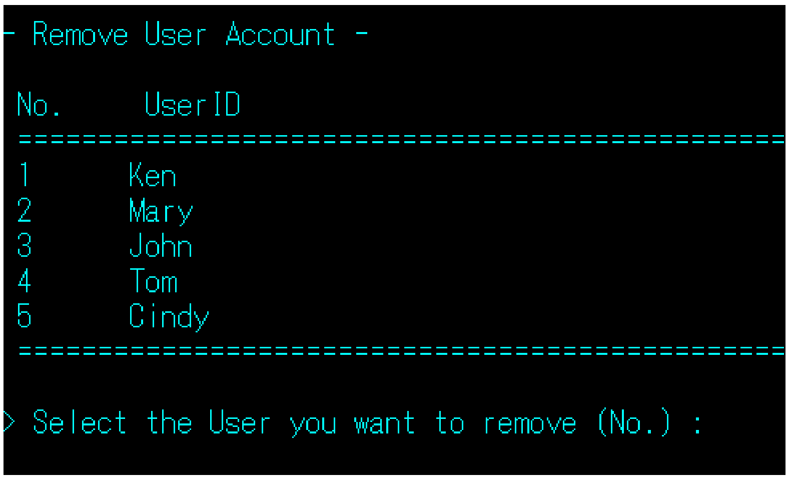
```
- A New User Account has been Created! -

Please give the following information to your designated user:

                       ================================

                       > User ID : Ken

                       > Password : 12345

                       ================================

Press <Enter> to menu...
```

[3b] – Remove User account

Features:

A full list of the user ID would be displayed on screen (except admin's) and the No. of the ID by a for loop from 1 to total. Admin are then requested to enter the No. of the ID to remove the unwanted accounts. If the input is within the total number of user accounts and is a integer, starting from unwanted user record, VBS would replace the record with the next record until the end of non-empty record with a while loop and the loop would end once the loop counter equal to (total + 1). The user information would be updated to the database finally.

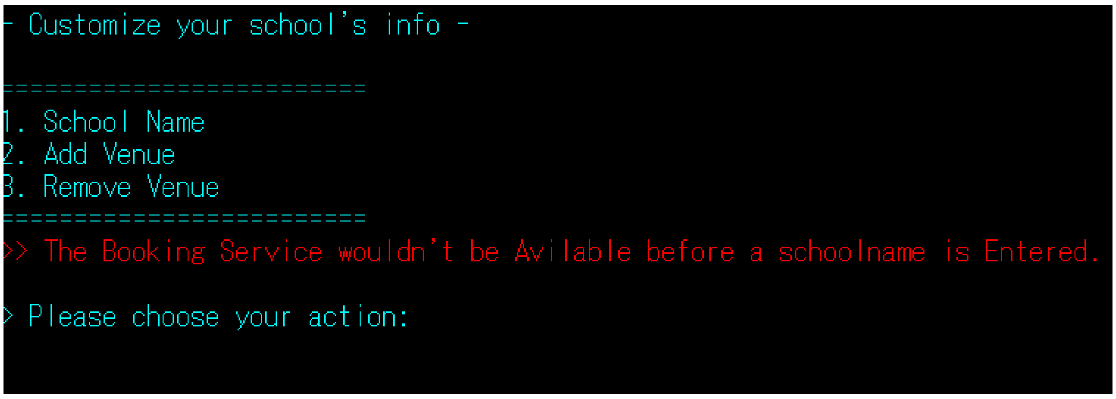
Also, VBS would check whether the unwanted user has any booking records. If VBS find the records, it would delete the booking record of the user the way VBS remove user accounts. Lastly, the booking record would be updated to the database. Otherwise, VBS would skip this part and remove the user account.



```
- Remove User Account -

 No.      UserID
 ================================================
 1       Ken
 2       Mary
 3       John
 4       Tom
 5       Cindy
 ================================================

> Select the User you want to remove (No.) :
```

[3c] – Customize school's info

Provide a interface for admin to input different school's information

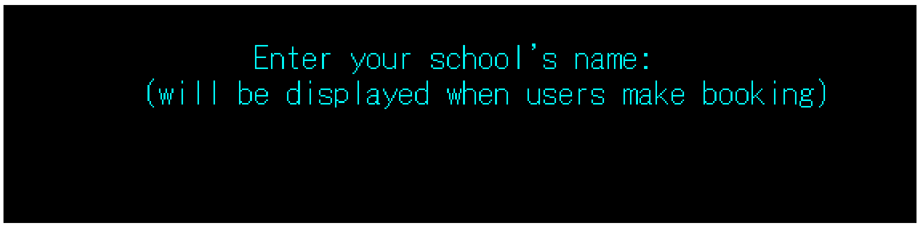


```
- Customize your school's info -

========================
1. School Name
2. Add Venue
3. Remove Venue
========================
>> The Booking Service wouldn't be Avilable before a schoolname is Entered.

> Please choose your action:
```

*Noted that the red color message will display before a school name is entered (would disappear after a school name is entered) and the booking service for admin and users would not start until admin enters a school name.

[3.1c] – school name

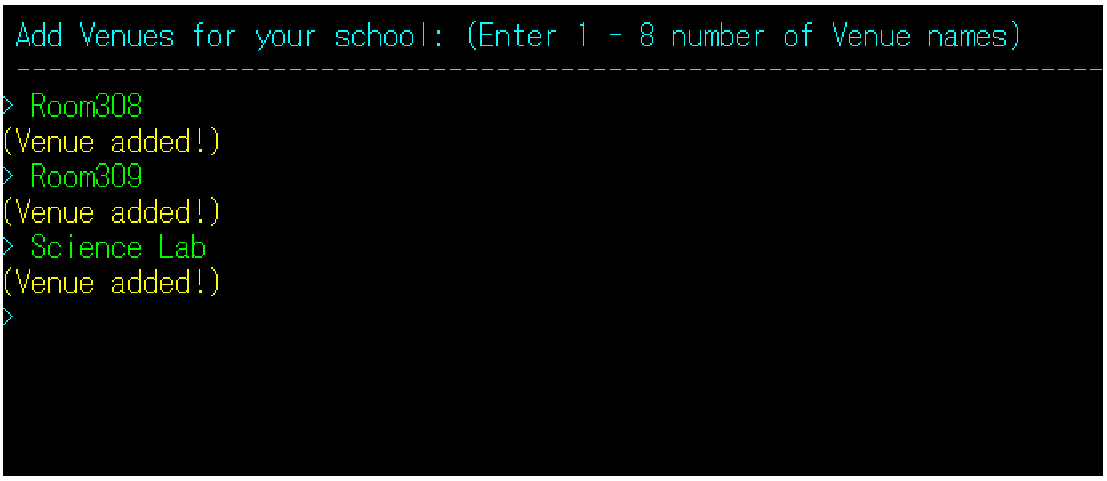
Ask the admin to input the school name adopting



```
            Enter your school's name:
       (will be displayed when users make booking)
```

[3.2c] – Add Venue

Allows the admin to input at most 8 venues at once and go to the interface freely using the "Escape" Function. Every time a venue name is entered, the input would be stored into Rooms array and be updated into the database. If 8 venues are entered, this sub-program would terminate and warp the user back to the Customization interface.



```
 Add Venues for your school: (Enter 1 - 8 number of Venue names)
 --------------------------------------------------------------------
> Room308
(Venue added!)
> Room309
(Venue added!)
> Science Lab
(Venue added!)
>
```

[3.3c] – Remove Venue

Provide a similar feature as Remove_User except removing venue

[3d] – Change password

Feature:

Require the user to input the password of his current account. Only proceed if the input match the password of the logged in account. If match, ask the user to input a new password. The user have to input his chosen password once again to make sure he doesn't input the wrong password and able to login to VBS. Error message world display if confirm password doesn't match password.

[3e] – Logout

Feature:

a message would pop up asking users to confirm logging out. If user input yes, input would warp the user into the login menu and log the user out of the system. If not, the user would stay on the main menu.

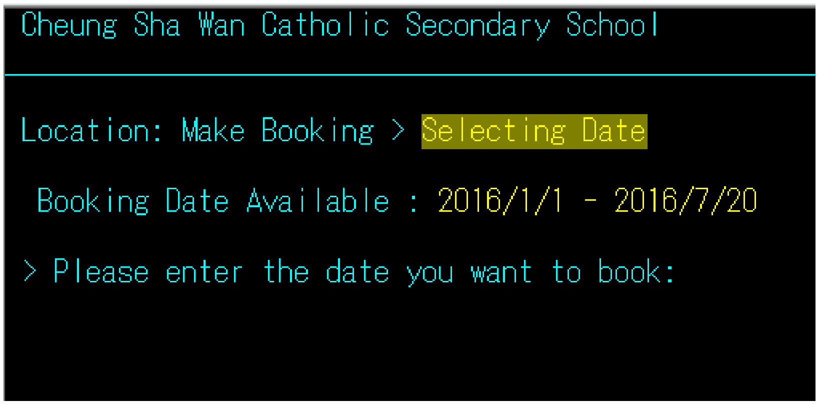
[3f] – Make Booking

Have to complete 3.1g, 3.2g, 3.3g before a booking can be made. Users can proceed to the previous section or main menu using the "Escape" function.

[3.1f] – Selecting Date For booking

Feature:

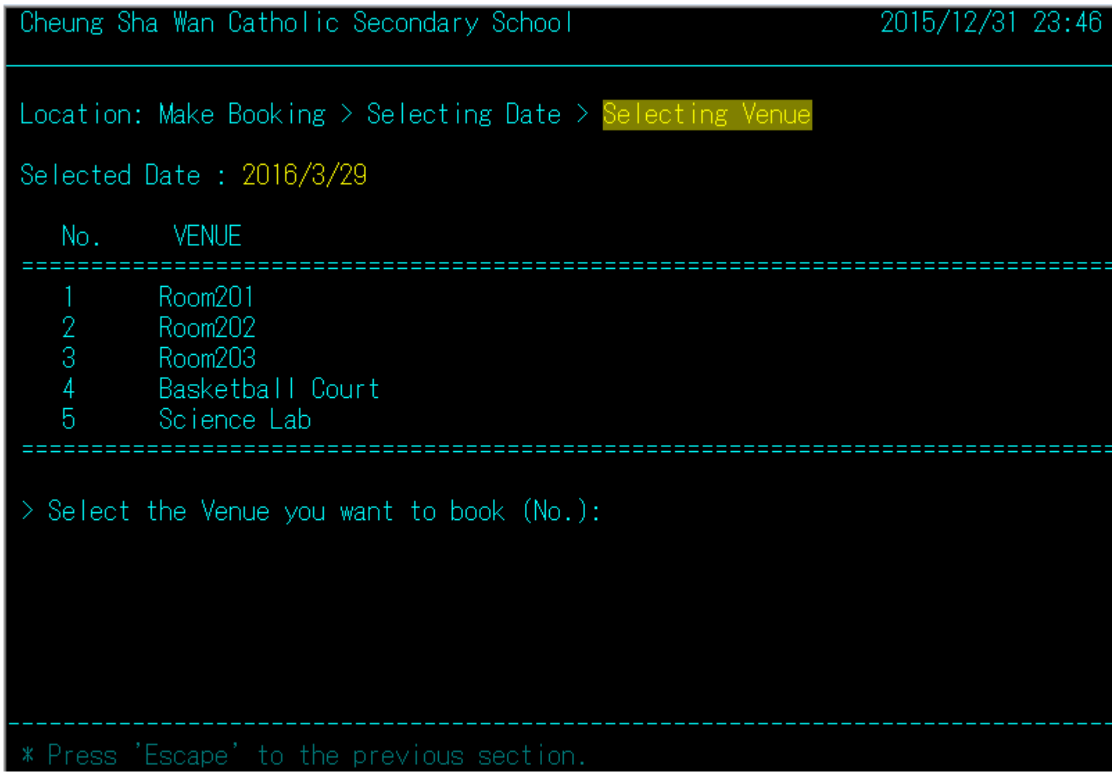
Call the Available_date procedure to get the first available date for booking which is the next day of the day the users make the booking. Then, the users would be asked to enter a valid date for booking. After that, VBS would check the existence of the inputted date in yyyy, mm and dd arrays starting from the first available date produced by Available_date. If the inputted date is found, the user can go to the next section. Otherwise, if the input is not in date format or within available period or not integer, error messages would be displayed and the page would be refreshed.


```
Cheung Sha Wan Catholic Secondary School

Location: Make Booking > Selecting Date

 Booking Date Available : 2016/1/1 - 2016/7/20

> Please enter the date you want to book:
```

[3.2f] – Selecting Venue

Feature:

The list of all venues available for booking would be displayed with No. aside. The users would be asked to enter a valid No. to select the venue they want to book. If the input is larger than the greatest No. or less than 1, an error message would be displayed and the page would be refreshed. If valid, the user can proceed to the next section.



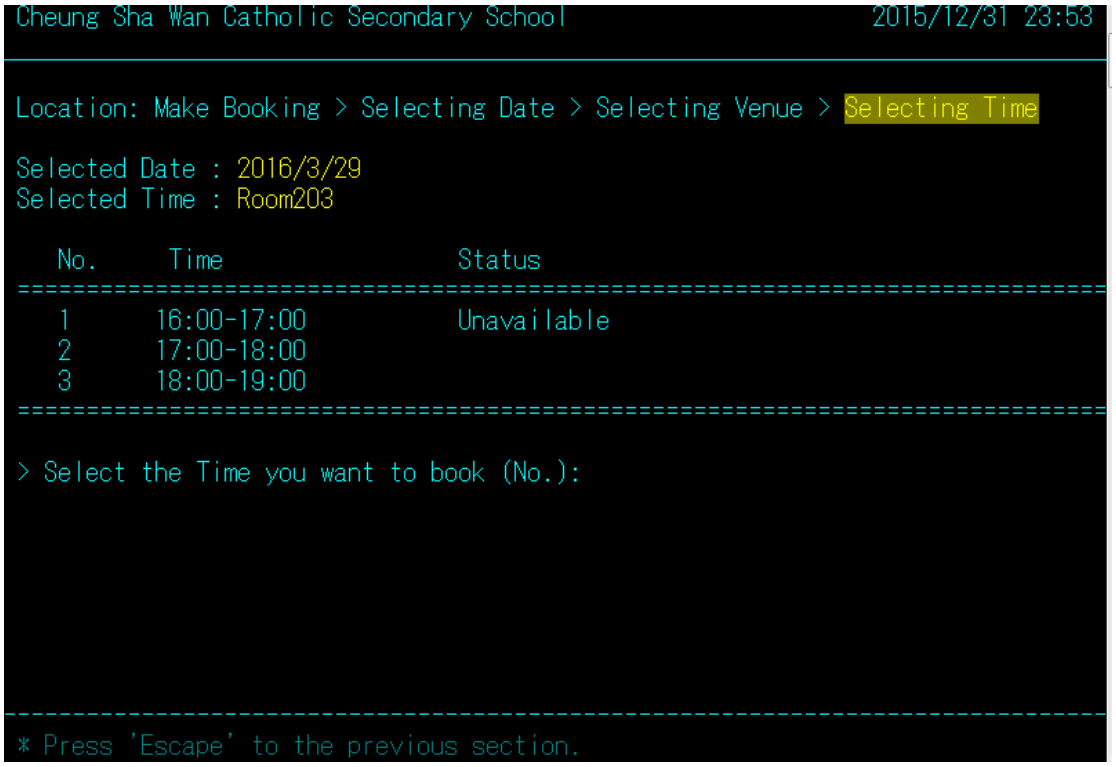
```
Cheung Sha Wan Catholic Secondary School                        2015/12/31 23:46

Location: Make Booking > Selecting Date > Selecting Venue

Selected Date : 2016/3/29

   No.     VENUE
===============================================================================
   1      Room201
   2      Room202
   3      Room203
   4      Basketball Court
   5      Science Lab
===============================================================================

> Select the Venue you want to book (No.):



-------------------------------------------------------------------------------
* Press 'Escape' to the previous section.
```

[3.3f] – Selecting time

Feature:

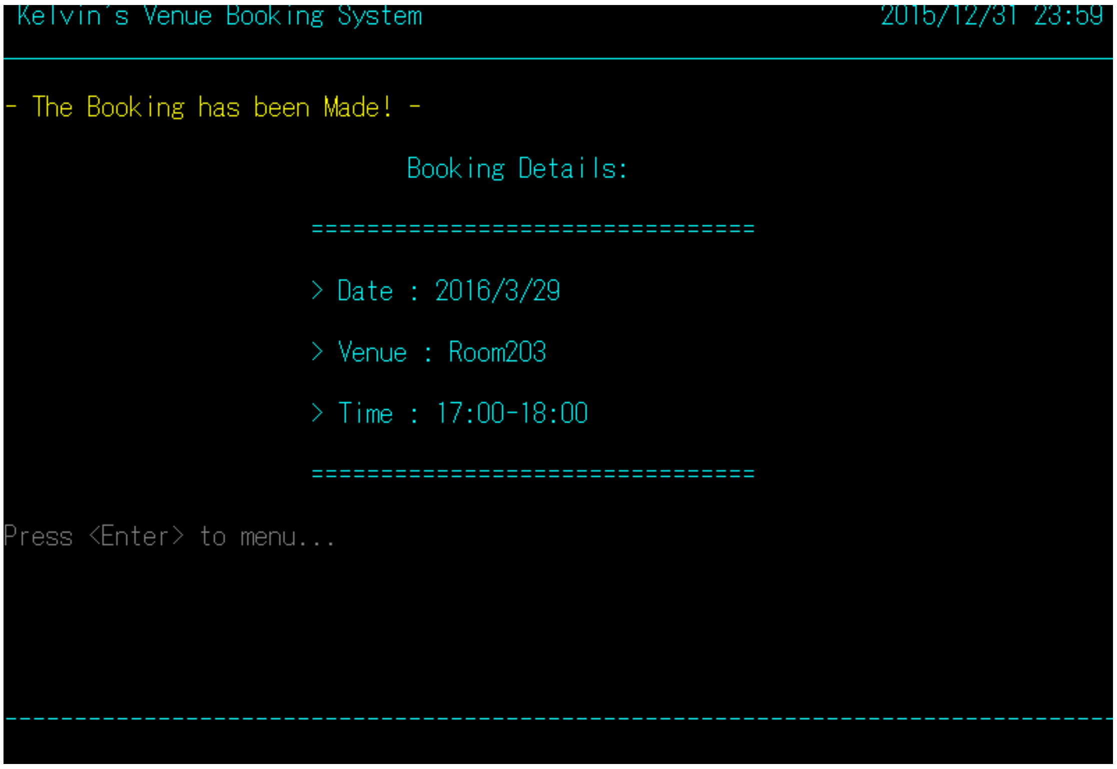
The list of three available time would be displayed: 16:00 – 17:00, 17:00 – 18:00 and 18:00 – 19:00 with a No. and status aside. VBS would check the bookings record (date, venue and time). If VBS finds the existence of a booking record with the same date, venue and time, the text “Unavailable” would be displayed under status. Otherwise, the slots would be empty.

Then, The users would be asked to enter a valid No. to select the time they want to book. If the input in within the range of No. and the selected time is available, the user would proceed. Otherwise, error messages would be displayed and the page would be refreshed.



```
Cheung Sha Wan Catholic Secondary School                          2015/12/31 23:53

Location: Make Booking > Selecting Date > Selecting Venue > Selecting Time

Selected Date : 2016/3/29
Selected Time : Room203

   No.      Time                    Status
==============================================================================
   1       16:00-17:00              Unavailable
   2       17:00-18:00
   3       18:00-19:00
==============================================================================

> Select the Time you want to book (No.):






------------------------------------------------------------------------------
* Press 'Escape' to the previous section.
```

Successful Booking:
All the details of booking record would be displayed on screen. Meanwhile, the booking would be stored into bookings record and be updated into the database.



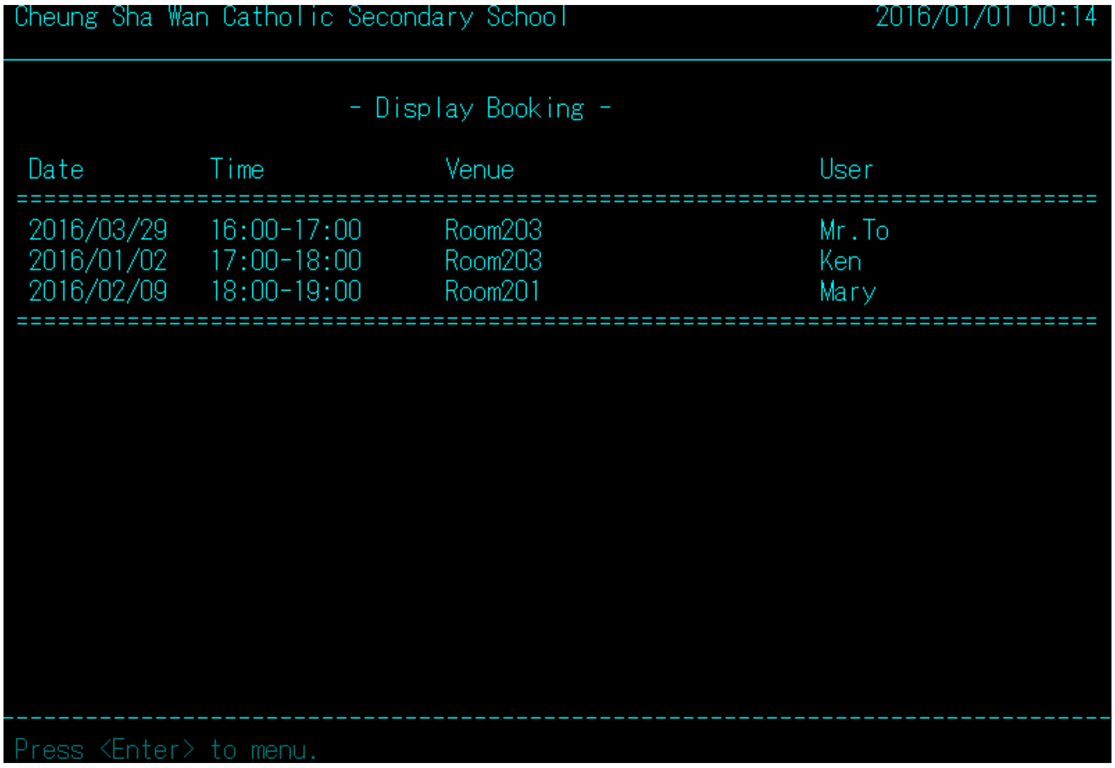
```
 Kelvin's Venue Booking System                                         2015/12/31 23:59
--------------------------------------------------------------------------------------

- The Booking has been Made! -

                                    Booking Details:

                              ================================

                              > Date : 2016/3/29

                              > Venue : Room203

                              > Time : 17:00-18:00

                              ================================

Press <Enter> to menu...
```

[3g] – Cancel Booking
A full list of the booking records of the user would be displayed on screen and the No. of the reocrd by a for loop from 1 to total. Admin are then requested to enter the No. of the ID to cancel the unwanted booking. If the input is within the user's total number of records and is a integer, starting from unwanted booking record, VBS would replace the record with the next record until the end of non-empty record with a while loop and the loop would end once the loop counter equal to (total_userbooking + 1). The bookings record would be updated to the database finally.

[3h] – Display Booking (only for admin)

Feature:

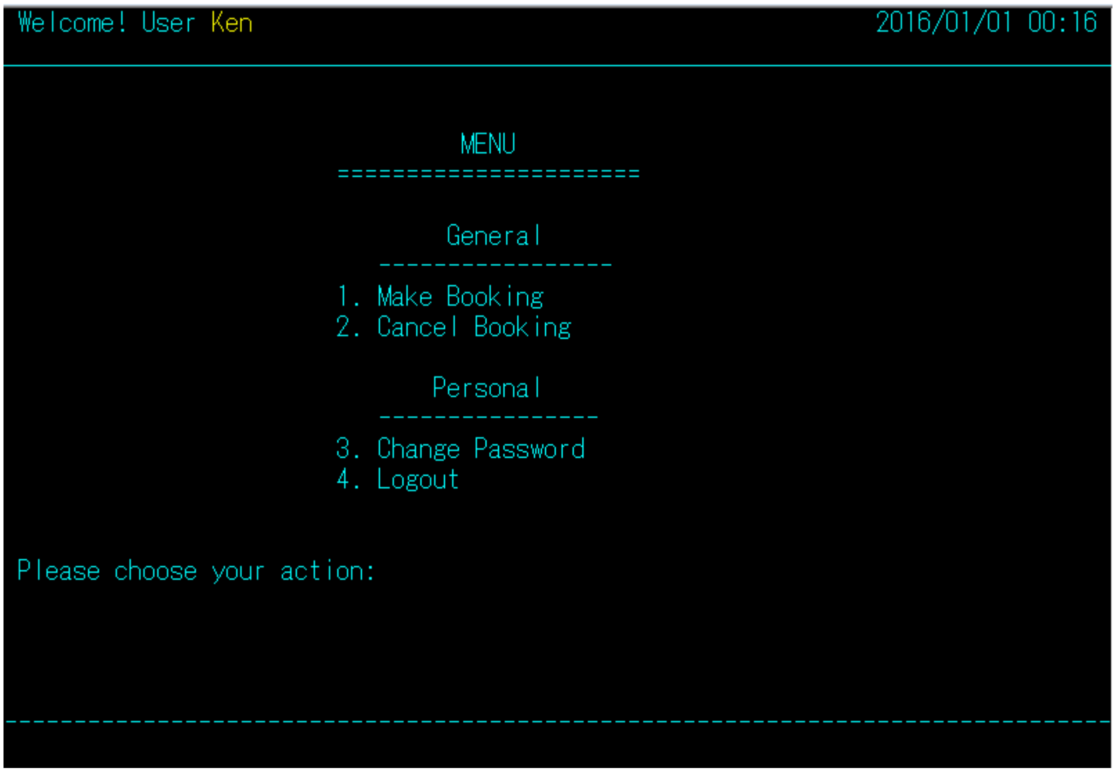
Display all the booking records (date, venue, time, user)



```
Cheung Sha Wan Catholic Secondary School                      2016/01/01 00:14

                         - Display Booking -

 Date          Time              Venue                      User
==============================================================================
 2016/03/29    16:00-17:00       Room203                    Mr.To
 2016/01/02    17:00-18:00       Room203                    Ken
 2016/02/09    18:00-19:00       Room201                    Mary
==============================================================================

------------------------------------------------------------------------------
Press <Enter> to menu.
```

[4] – User_Menu

The main menu displayed to users logged in the system, listing all the functions available and requesting users' choice on using which function.



```
Welcome! User Ken                                         2016/01/01 00:16
---------------------------------------------------------------------------

                                 MENU
                         =====================

                                General
                            ----------------
                         1. Make Booking
                         2. Cancel Booking

                                Personal
                            ---------------
                         3. Change Password
                         4. Logout


Please choose your action:



---------------------------------------------------------------------------
```

[4a] – Make Booking

Read [3f].

[4b] – Cancel Booking

Read [3g].

[4c] – Change Password

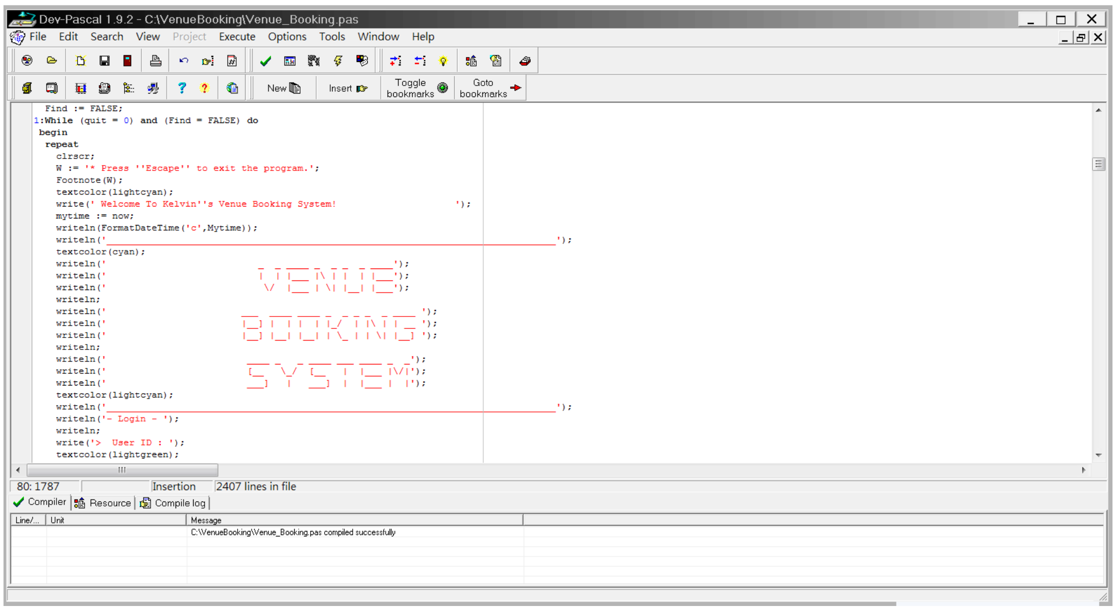
Read [3d].

[4d] – Logout

Read [3e].

3.5 Program Coding

The VBS program is written and complied by Dev-Pascal. The Source program is made of w13 parts, which contains 1 main program and 12 procedures/Functions. Meanwhile, there are 8 sub-programs and 4 sub-programs under Menu_Admin and Menu_User respectively .


Dev-Pascal 1.9.2 - C:\VenueBooking\Venue_Booking.pas
File Edit Search View Project Execute Options Tools Window Help
New Insert Toggle bookmarks Goto bookmarks
Compiler Resource Compile log

```
 Find := FALSE;
1:While (quit = 0) and (Find = FALSE) do
 begin
  repeat
    clrscr;
    W := '* Press ''Escape'' to exit the program.';
    Footnote(W);
    textcolor(lightcyan);
    write(' Welcome To Kelvin''s Venue Booking System!                ');
    mytime := now;
    writeln(FormatDateTime('c',Mytime));
    writeln('________________________________________________________________');
    textcolor(cyan);
    writeln('                    _   _ ____ _   _ _   _ ____');
    writeln('                    | | | |___ |\ | |   | |___');
    writeln('                     \/  |___ | \| |___| |___');
    writeln;
    writeln('                  ___  ____ ____ _  _ _ _   _ ____ ');
    writeln('                  |__] |  | |  | |_/  | |\  | | __ ');
    writeln('                  |__] |__| |__| | \_ | | \| |__] ');
    writeln;
    writeln('                   ____ _  _ ____ ___ ____ _  _');
    writeln('                   [__   \_/  [__   |  |___ |\/|');
    writeln('                   ___]   |   ___]  |  |___ |  |');
    textcolor(lightcyan);
    writeln('________________________________________________________________');
    writeln('- Login - ');
    writeln;
    write('>  User ID : ');
    textcolor(lightgreen);
```

```
80: 1787        Insertion    2407 lines in file
```

Line/...	Unit	Message
		C:\VenueBooking\Venue_Booking.pas compiled successfully

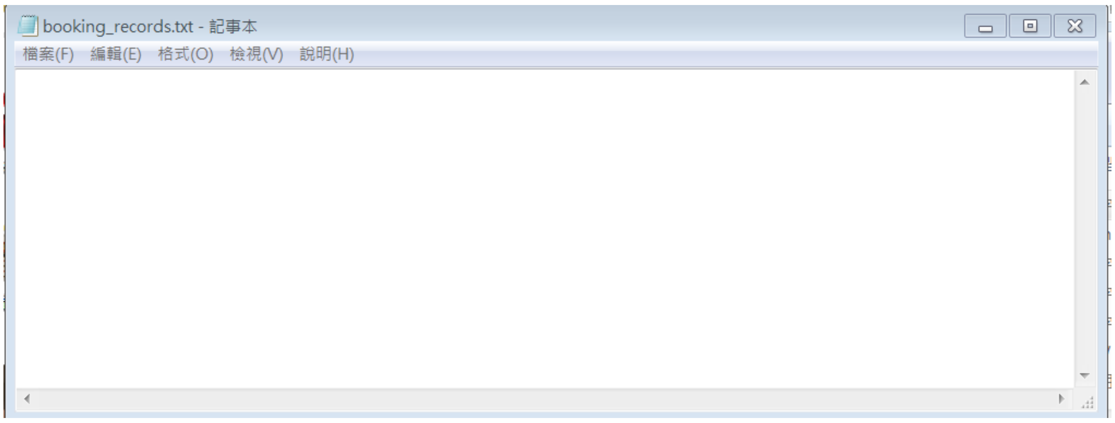
3.6 Program Execution

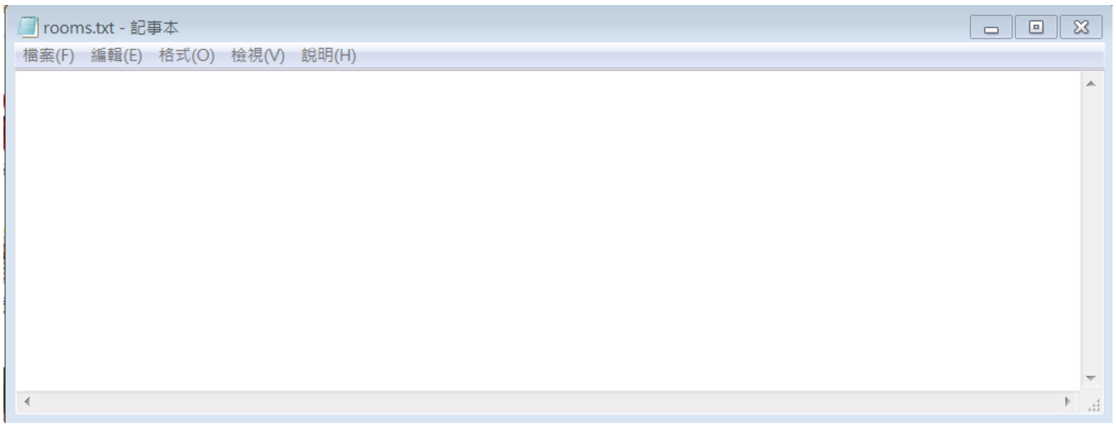
To execute the program VBS, first put the text files, user.txt, rooms.txt, booking_records.txt, valid_dates.txt with the program Venue_Booking.exe, then the program is ready to start.

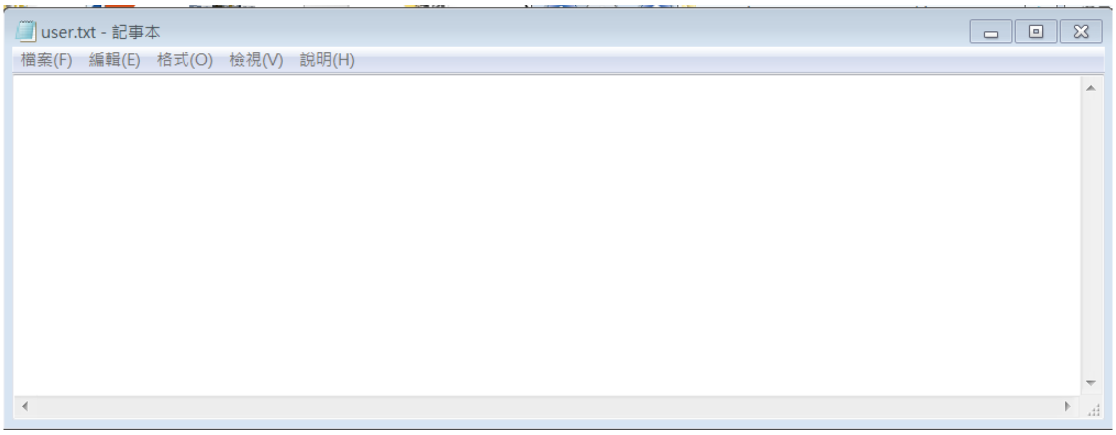
1. Program file: Venue_Booking.exe Venue_Booking.exe

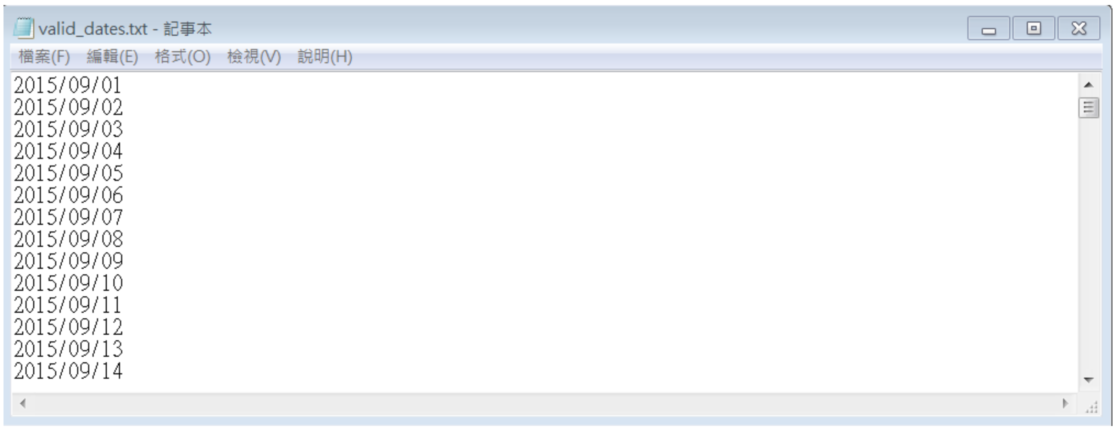
2. Data file to be prepared:

- User Info file: user.txt user.txt
- Venues available for booking: rooms.txt rooms.txt
- Booking records: booking_records.txt booking_records.txt
- Valid Dates for booking: valid_dates.txt valid_dates.txt


booking_records.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

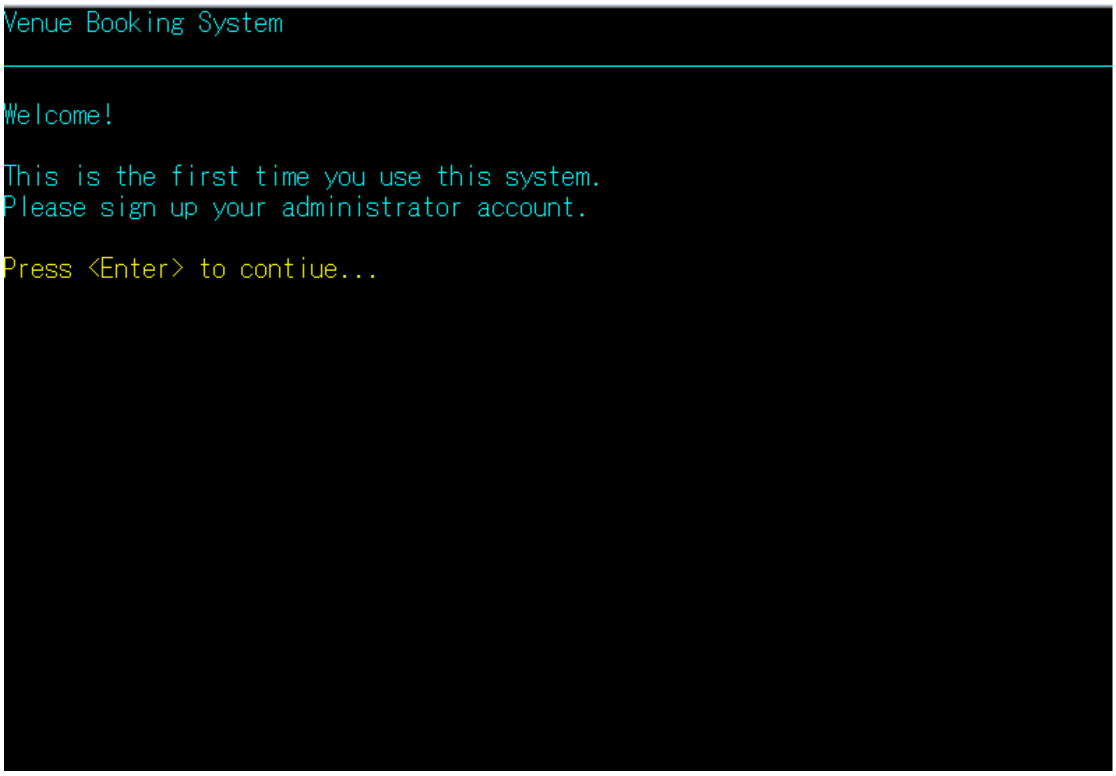






```
2015/09/01
2015/09/02
2015/09/03
2015/09/04
2015/09/05
2015/09/06
2015/09/07
2015/09/08
2015/09/09
2015/09/10
2015/09/11
2015/09/12
2015/09/13
2015/09/14
```

3. User-interface of the program
   {First Launch}



```
Venue Booking System
________________________________________________

Welcome!

This is the first time you use this system.
Please sign up your administrator account.

Press <Enter> to contiue...
```

{Signup}

Shown in [2a].

{Login}

Shown in [2.1a].

{Main Menu (Admin)}

Shown in [3].

{Main Menu (User)}

Shown in [4]

4.1 Description

In this chapter, a set of testing is done to find outthe bugs in the program and to check whether theprogram can achieve its purposes, thus to debug andimprove the program based on the testing results.

4.2 Testing and Evaluation Plan

The program will be tested and evaluated according to the following plan:

1. The program will be tested by me, the programmer, several test cases will be set to test the program. The main purpose of this test is to check how the program handle invalid input or data reasonably.

2. The program will be evaluated by some of my classmates and me according to its level of user-friendly, performance, flexibility for future development and reusability of program codes.

4.3 Internal testing

Table of test cases:

No.	Functions
1.	Missing database simulation
2.	Normal booking process
3.	Database update simulation
4.	Removing user/venue simulation

Test case 1

Purpose	To check how the program reacts with missing database/text files.
Input	Launch the program without correct text files in the same file of the program
Expected output	A file error message would be displayed on screen, asking user to check the files
Actual output	All actual results are the same as the expected results
Test Result	Pass, no bugs found
Follow-up Action	Nil

Test case 2

Purpose	To check how the program reacts with different combination of booking.
Input	Different combination of booking

Expected Output	All possible combination are accepted, storing the booking records into the database
Actual Output	All actual results are the same as the expected results
Test Result	Pass, no bugs found
Follow-up Action	Nil

Test case 3

Purpose	To check how the program update the database according to the inputs of users
Input	Possible combinations for different functions
Expected output	All possible combinations can be accepted, replacing the outdated records with the possible combinations
Actual output	All actual results are the same as the expected results
Test Result	Pass, no bugs found
Follow-up Action	Nil

Test case 4

Purpose	To check how the program reacts with different combination of user and venue management in an Admin account
Input	Unique venue names and user names, as well as passwords. Select inputted venue and user accounts to remove
Expected output	All unique names should be accepted and selected names should be removed
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

The program has additional functions, such as hiding password, input validation, ‘Escape’ function, heading function and footnote function to provide users with comfortable and clear UI. All the operations are held inside two lines which visually looks better and tidier. Users are guided by clear and short instruction to smoothen the Venue Booking operations and system management. Also, input validation allows the system to work reliably and steadily. Other than that, the hiding password enhances the security of the system.

Although the system has many pages and functions, the ‘Escape’ function eliminates the problem of inefficiency when warping through pages which allows VBS to provide comprehensive and convenient service. Besides, free access to user and venue management opens up many possibilities. The admin can customize the number of users and venues, as well as the username and venue name which greatly enlarge the range of application.

On the other hand, the system has limited venue booking service on weekends as it would not change the available booking time of weekends into the whole daytime, constraining the comprehensiveness of the system.

5.1 Pros and Cons of my program

Pros	Cons
Comfortable interface	Unable to provide special booking time for weekends
Variety of Functions	Complicated Permission system may confuse IT beginners
Instant automatic problem handling	Fixed Outlook
Very Flexible booking service	Lack of file validation

5.2 Future improvement

After a step of improvement, there are still imperfect places to be improved. Here is the future improvement of the program:

- Feature customizations for time available for booking
- Better make use of procedures
- Include File Validation
- Algorithms should be more precise

5.3 Self-Reflection

In this project, I acquired a precious experience on creating a large scale computer system. I never did something this big before because of my limited programming knowledge and time. However, computer systems to be large scale and comprehensive are the cornerstones to success nowadays so this is really a valuable experience.

Besides, my programming skills and logic skills were refined in doing this project as it is very complicated and huge in scale, involving a lot of calculations and complex logical thinking. Now, I can write more precise algorithms and solve problems more effectively.

I am truly grateful to have this opportunity, hoping that the skills acquired can be made good use of in my future career.

Chapter 6 Reference and Acknowledgement

Internet

1. http://www.freepascal.org/
2. http://pascal.programming.info/

Books

1. NSS ICT Elective D1 Software Development

Acknowledgement

1. ICT teacher Mr. Chu – For his programming advice and lessons
2. Schoolfreeware (https://www.youtube.com/playlist?list=PLB24C56953A79987A)
   – For his compact programming tutorials

Appendix 1: Program Code

```
program venue;

uses
      sysutils, crt, dos;

label 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20;

const
    time : array[1..3] of string = ('16:00-17:00', '17:00-18:00', '18:00-19:00');

type
      UserData = record
                            id : string;
                            pw : string;
                            permission : string;
                      end;

    BookingData = record
                            Year : word;
                            Month : word;
                            Day : word;
                            user : string;
                            venue : integer;
                            time : integer;
                  end;

var
        booking_num : array[1..200] of integer;
        status : array[1..3] of boolean;
        find_location, count : integer;
        input_yyyy, input_mm, input_dd : word;
        input_venue, input_time : integer;
        Year, Month, Day, WDay : word;
        yyyy, mm, dd : array[1..400] of word;
        flag : array [1..3] of boolean;
        W, A :string;
        Pressed, done,logout, menu, find : boolean;
        quit, user_num, error, total, total_room, total_date, total_booking,
total_userbooking : integer;
```

```
        user_id, password, pw1, pw2, permission, schoolname : string;
        UserR: array[1..500] of UserData;
        activitate : string;
        user, Rooms, valid_date, booking_record : text;
        MyTime : TDateTime;
        room : array[1..500] of string;
        booking : array[1..500] of BookingData;

procedure Available_date;
var
      i :integer;
      Find : boolean;
begin

    find := FALSE;

    GetDate(Year,Month,Day,WDay);
    Day := day + 1;
    for i := 1 to total_date do
    begin
        if year = yyyy[i] then
        begin
            if Month = mm[i] then
                if Day = dd[i] then
                begin
                    find := TRUE;
                    find_location := i;
                end;
        end;
    end;

    if find = FALSE then
    begin
        Day := 1;
        Month := Month + 1;
        if month > 12 then
        begin
            month := 1;
            year := year + 1;
        end;
    end;

end;


function Input(hide_pw : boolean; Escape : integer): string;
var
    S, W : string;
    C : Char;
begin
    Pressed := FALSE;
    S := '';
    repeat
        C := ReadKey;
        if (C <> #10) and (C <> #13) and (C <> #8) and (C <> #27) then
            begin
                S := S + C;
                if hide_pw = FALSE then
                    write(c)
                else write('*');
            end
        else if C = #8 then
            begin
                S[0] := Chr(Length(S) - 1);
                GotoXY(WhereX - 1, WhereY);
                write(' ');
                GotoXY(WhereX - 1, WhereY);
            end
        else if c = #27 then
        begin
          Pressed := TRUE;
          if Escape = 1 then
          begin
              repeat
                  clrscr;
                  gotoxy(1,9);
                  textcolor(cyan);
                  writeln('
========================================= ');
                  textcolor(lightred);
                  gotoxy(1,2);
                  writeln('                                              ! Warning !');
                  writeln;
                  textcolor(cyan);
                  writeln('
======================================== ');
                  textcolor(lightcyan);
                  writeln('                                  Are you sure you want to
Exit?');
                  writeln;
                  write('                                                (y/n): ');
                  readln(W);
                  if (W = 'Y') or (W = 'y') then
                     quit := 1
                  else if (W = 'N') or (W = 'n') then
                     quit := 0;
              until (W = 'y') or (W = 'Y') or (W = 'n') or (W = 'N');
        end
        else if Escape = 2 then
        begin
              repeat
                  clrscr;
                  gotoxy(1,9);
                  textcolor(cyan);
                  writeln('
============================================= ');
                  textcolor(lightred);
                  gotoxy(1,2);
                  writeln('                                              ! Warning !');
                  writeln;
                  textcolor(cyan);
                  writeln('
=============================================');
                  textcolor(lightcyan);
                  writeln('                            Are you sure you want to go to Main
Menu?');
                  writeln('                               (Any Unsaved data will not be
saved)');
                  writeln;
                  write('                                                (y/n): ');
                  readln(W);
                  if (W = 'Y') or (W = 'y') then
                     done := TRUE
                  else if (W = 'N') or (W = 'n') then
                     done := false;
              until (W = 'y') or (W = 'Y') or (W = 'n') or (W = 'N');
        end
        else
        begin
           repeat
                  clrscr;
                  gotoxy(1,9);
                  textcolor(cyan);
                  writeln('
============================================== ');
                  textcolor(lightred);
                  gotoxy(1,2);
                  writeln('                                                  ! Warning !');
                  writeln;
                  textcolor(cyan);
                  writeln('
==============================================');
                  textcolor(lightcyan);
                  writeln('                        Are you sure you want to go to the
previous page?');
                  writeln('                                 (Any Unsaved data will not be
saved)');
                  writeln;
                  write('                                                    (y/n): ');
                  readln(W);
                  if (W = 'Y') or (W = 'y') then
                     done := TRUE
                  else if (W = 'N') or (W = 'n') then
                     done := false;
              until (W = 'y') or (W = 'Y') or (W = 'n') or (W = 'N');
        end;
     end;
     until (C = #10) or (C = #13) or (C = #27);
     Input := S;
     writeLn;
```

```
end;


procedure Heading;
begin
      gotoxy(1,1);
      textcolor(lightcyan);
      write(' Kelvin''s Venue Booking System
');
      mytime := now;
      writeln(FormatDateTime('c',Mytime));

writeln('____________________________________________________________
_____________________');
end;

procedure Heading_Booking(mess : string);
begin
    gotoxy(1,1);
    textcolor(lightcyan);
    write(' ', mess);
    mytime := now;
    writeln(FormatDateTime('c',Mytime):(78 - length(mess)));

writeln('____________________________________________________________
_____________________');
end;

procedure Heading_Login(mess : string; permiss : integer);
begin
    gotoxy(1,1);
    textcolor(lightcyan);
    gotoxy(1,1);

    if permiss = 2 then
    begin
        write(' Welcome! Admin ');
        textcolor(yellow);
        write(mess);
        textcolor(lightcyan);
    mytime := now;
    writeln(FormatDateTime('c',Mytime):(63 - length(mess)));

writeln('____________________________________________________________
_____________________');
  end
  else
  begin
    write(' Welcome! User ');
    textcolor(yellow);
    write(mess);
    textcolor(lightcyan);
    mytime := now;
    writeln(FormatDateTime('c',Mytime):(64 - length(mess)));

writeln('____________________________________________________________
_____________________');
  end;
end;


procedure Footnote(mess : string);
begin
   textcolor(lightcyan);
   gotoxy(1,24);
   write('------------------------------------------------------------------------------');
   textcolor(cyan);
   writeln(' ', mess);
   gotoxy(1,1);
end;


procedure WriteText;
var
   i : integer;
begin
  rewrite(user);
  for i := 1 to total do
  begin
    with userR[i] do
```

```
        begin
            writeln(user, id);
            writeln(user, pw);
            writeln(user, permission);
        end;
    end;
    close(user);


    rewrite(Rooms);
    writeln(Rooms, schoolname);

    for i := 1 to total_room do
    begin
        writeln(Rooms, Room[i]);
    end;
    close(Rooms);


    rewrite(booking_record);
    for i := 1 to total_booking do
    begin
        with booking[i] do
        begin
            writeln(booking_record, Year, '/', Month, '/', Day);
            writeln(booking_record, User);
            writeln(booking_record, Venue);
            writeln(booking_record, Time);
        end;
    end;
    close(booking_record);
end;


procedure login;
var
    i : integer;
    find : boolean;
begin
    Find := FALSE;
```

```
1:While (quit = 0) and (Find = FALSE) do
  begin
    repeat
        clrscr;
        W := '* Press "Escape" to exit the program.';
        Footnote(W);
        textcolor(lightcyan);
        write(' Welcome To Kelvin''s Venue Booking System!
');
        mytime := now;
        writeln(FormatDateTime('c',Mytime));

writeln('____________________________________________________________
____________________');
        textcolor(cyan);
        writeln('                                    _    _ _____ _    _ _    _ _____');
        writeln('                                   |    | |____ |\ | |    | |____');
        writeln('                                     \/    |____ | \| |___| |____');
        writeln;
        writeln('                               ____    _____ _____ _    _ _ _    _ _____ ');
        writeln('                              |___] |    | |    | |_/    | |\ | | ___ ');
        writeln('                              |___] |___| |___| | \_ | | \| |___] ');
        writeln;
        writeln('                                _____ _      _ _____ ____ _____ _    _');
        writeln('                                [___      \_/    [___      |    |____ |\/|');
        writeln('                                ____]      |      ____]    |    |____ |    |');
        textcolor(lightcyan);

writeln('____________________________________________________________
____________________');
        writeln('- Login - ');
        writeln;
        write('>    User ID : ');
        textcolor(lightgreen);
        user_id := input(FALSE,1);
        if Pressed = TRUE then
           goto 1;
        writeln;
        textcolor(lightcyan);
        write('> Password : ');
        textcolor(lightgreen);
        password := input(TRUE,1);
        if Pressed = TRUE then
           goto 1;

        for i:= 1 to total do
             if (user_id = userR[i].id) and (password = userR[i].pw) then
             begin
                find := TRUE;
                permission := userR[i].permission;
                user_num := i;
             end;
        if find = TRUE then
        begin
           textcolor(green);
           writeln('                                             - Login Sucessful! -');
           delay(2100);
        end
        else
        begin
           textcolor(lightred);
           writeln('                                  - Invalid User_ID or Password! -');
           delay(2100);
        end;
     until find = TRUE;
  end;
end;



procedure Menu_Admin;
var
     remove_id : integer;
     new_id, new_pw1, new_pw2 : string;
     action, i, j, choice : integer;
begin
  3:repeat
        logout := FALSE;
        clrscr;
        W := '';
        footnote(W);
        textcolor(lightcyan);
        heading_login(user_id, 2);
        writeln;
        writeln('                                   MENU');
        writeln('                          =====================');
        writeln;
        writeln('                              Administration ');
        writeln('                            ---------------- ');
        writeln('                        1. Add User Account');
        writeln('                        2. Remove User Account');
        writeln('                        3. Customize your School''s info');
        writeln;
        writeln('                   Personal                     General');
        writeln('                 ------------                 -----------');
        writeln('             4. Change Password            6. Make
Booking');
        writeln('             5. Logout                      7. Cancel
Booking');
        writeln('                                              8. Dsiplay All
Bookings');
        writeln;
        writeln;
        write('> Please choose your action: ');
        textcolor(yellow);
        readln(W);
        val(W, action, error);
        if (error <> 0) or (action <= 0) or (action > 8) then
        begin
            writeln;
            textcolor(lightred);
            writeln('- Invalid Input! - ');
            writeln;
            writeln('Press <Enter> to retry...');
            readln;
        end
        else
        begin
          if action = 1 then
          begin
```

```
clrscr;
find := FALSE;
flag[1] := FALSE;
flag[2] := FALSE;
done := FALSE;

5:While (flag[2] = FALSE) and (done = FALSE) do
begin
    repeat
    clrscr;
    textcolor(lightcyan);
    W := '* Press <Escape> to menu';
    footnote(W);
    heading;
    writeln('- Create User Accounts -');
    writeln;
    write('> User ID (Unique ID not shorter than 2): ');
    textcolor(lightgreen);
    new_id := input(FALSE,2);
    if done = TRUE then
        goto 3
    else if pressed then
        goto 5;

    for i := 1 to total do
        if (new_id = userR[i].id) and (length(new_id) =
length(userR[i].id)) then
            Find := TRUE;

    if (length(new_id) > 2) and (Find = FALSE)    then
        flag[1] := TRUE
    else if find = TRUE then
    begin
        writeln;
        textcolor(lightred);
        writeln('- Duplicated User ID! -');
        writeln;
        write('Press <Enter> to retry...');
        readln;
```

```
      clrscr;
   end
   else
   begin
      writeln;
      textcolor(lightred);
      writeln('- Your Input is too short! -');
      writeln;
      write('Press <Enter> to retry...');
      readln;
      clrscr;
 end;
 until flag[1] = TRUE;

 writeln;
 textcolor(lightcyan);
 write('> Password (PW not shorter than 3): ');
 textcolor(lightgreen);
 new_pw1 := Input(TRUE,2);
 if done = TRUE then
    goto 3
 else if pressed then
    goto 5;

 writeln;
 textcolor(lightcyan);
 write('> Confirm Password : ');
 textcolor(lightgreen);
 new_pw2 := Input(True,2);
 if done = TRUE then
    goto 3
 else if pressed then
    goto 5;

 if (new_pw1 = new_pw2) and (length(new_pw1) > 2) then
 begin
    delay(1000);
    clrscr;
    W := '';
    footnote(W);
```

```
                    heading;
                    textcolor(yellow);
                    writeln('- A New User Account has been Created! -');
                    textcolor(darkgray);
                    delay(1400);
                    writeln;

                    textcolor(lightcyan);
                    writeln('Please give the following information to your designated
user:');
                    writeln;
                    textcolor(lightcyan);
                    writeln('
================================');
                    writeln;
                    writeln('                              > User ID : ', new_id);
                    writeln;
                    writeln('                              > Password : ', new_pw1);
                    writeln;
                    writeln('
================================');
                    writeln;
                    textcolor(darkgray);
                    write('Press <Enter> to menu...');
                    readln;

                    total := total + 1;
                    userR[total].id := new_id;
                    userR[total].pw := new_pw1;
                    userR[total].permission := '1';

                    flag[2] := TRUE;
                    done := TRUE;

                    WriteText;
                 end
                 else
                 begin
                    writeln;
                    textcolor(lightred);
```

```
                if new_pw1 <> new_pw2 then
                begin
                   writeln('- Confirm password doesn''t match! -');
                   writeln;
                   writeln('Press <Enter> to retry...');
                   readln;
                   clrscr;
                end
                else
                begin
                   textcolor(lightred);
                   writeln('- Your Input is too short! -');
                   writeln;
                   write('Press <Enter> to retry...');
                   readln;
                   clrscr;
                end;
             end;
          end;

     end

     else if action = 2 then
     begin
       if total > 1 then

         6:repeat
               done := FALSE;
               remove_id := 0;
               clrscr;
               W := '* Press "Escape" to menu.';
               if total > 13 then
               begin
                  textcolor(lightcyan);
                  gotoxy(1,(24 + total - 13));

write('--------------------------------------------------------------------------------');
                  textcolor(cyan);
                  writeln(W);
                  gotoxy(1,1);
```

```
        end
        else
          footnote(W);

        heading;
        W := '';
        writeln('- Remove User Account -');
        writeln;
        writeln(' No.          UserID');
        write('
============================================================
======================');
        for i:= 1 to (total-1) do
        begin
          if i < 10 then
            writeln(' ',i,'            ',userR[i+1].id)
          else if i < 100 then
            writeln(' ',i,'          ',userR[i+1].id)
          else
            writeln(' ',i,'        ',userR[i+1].id);
        end;
        write('
============================================================
======================');
        writeln;

        write('> Select the User you want to remove (No.) : ');
        W := input(FALSE, 2);
        val(W, remove_id, error);

        if done = TRUE then
          goto 3
        else if pressed then
          goto 6;

        if (error = 0) and (remove_id > 0) and (remove_id <= total) then
        begin
          i := 1;
          repeat
            if userR[remove_id + 1].id = booking[i].user then
                find := TRUE;
            i := i + 1;
          until (i > total_booking) or (find = TRUE);

          if find = TRUE then
          begin
            i := i - 1;
            booking[i].Year := booking[i + 1].Year;
            booking[i].Month := booking[i + 1].Month;
            booking[i].Day := booking[i + 1].Day;
            booking[i].User := booking[i + 1].User;
            booking[i].Venue := booking[i + 1].Venue;
            booking[i].Time := booking[i + 1].Time;

            i := i + 1;

            While i <> (total_booking + 1) do
            begin
                booking[i].Year := booking[i + 1].Year;
                booking[i].Month := booking[i + 1].Month;
                booking[i].Day := booking[i + 1].Day;
                booking[i].User := booking[i + 1].User;
                booking[i].Venue := booking[i + 1].Venue;
                booking[i].Time := booking[i + 1].Time;
                i := i + 1;
            end;

            total_booking := total_booking - 1;
          end;



          remove_id := remove_id + 1;
          userR[remove_id].id := userR[remove_id + 1].id;
          userR[remove_id].pw := userR[remove_id + 1].pw;
          userR[remove_id].permission := userR[remove_id +
1].permission;
          i := remove_id + 1;
          While i <> (total + 1) do
          begin
            userR[i].id := userR[i+1].id;
```

```
                    userR[i].pw := userR[i+1].pw;
                    userR[i].permission := userR[i+1].permission;
                    i := i + 1;
                  end;

                  done := TRUE;
                  total := total - 1;
                  WriteText;
                end
                else if (error <> 0) and (W <> '') or (remove_id < 0) or
(remove_id > total) then
                begin
                  textcolor(lightred);
                  writeln('- Invalid Input! -');
                  writeln;
                  write('Press <Enter> to retry...');
                  readln;
                  goto 6;
                  clrscr;
                end;

              until (done = TRUE)
            else
            begin
              clrscr;
              textcolor(lightcyan);
              W := '';
              footnote(W);
              heading;
              textcolor(lightred);
              writeln('- No User to remove! -');
              writeln;
              write('Press <Enter> to menu...');
              readln;
              goto 3;
            end;

          end
```

```
        else if action = 3 then
        begin
          menu := FALSE;
        9:repeat
          clrscr;
          choice := 0;
          done := FALSE;
          W := '* Press "Escape" to menu.';
          Footnote(W);
          heading;
          writeln('- Customize your school''s info -');
          writeln;
          textcolor(cyan);
          writeln('========================');
          textcolor(lightcyan);
          writeln('1. School Name');
          writeln('2. Add Venue');
          writeln('3. Remove Venue');
          textcolor(cyan);
          writeln('========================');
          textcolor(lightcyan);

          reset(rooms);
          readln(rooms, schoolname);
          close(rooms);
          if schoolname = '#' then
          begin
            textcolor(lightred);
            writeln('>> The Booking Service wouldn''t be Avilable before a
schoolname is Entered.');
            textcolor(lightcyan);
          end;

          writeln;
          write('> Please choose your action: ');
          textcolor(yellow);
          W := input(FALSE, 2);
          val(W, choice, error);
          if done = TRUE then
            goto 3
```

```
else if (pressed) and (done = FALSE) then
   goto 9;


if (error <> 0) or (choice <= 0) or (choice > 3) then
begin
   writeln;
   textcolor(lightred);
   writeln('- Invalid Input! - ');
   writeln;
   writeln('Press <Enter> to retry...');
   readln;
end

else if choice = 1 then
begin
   10:clrscr;
   done := FALSE;
   W := '* Press "Escape" to the previous page.';
   Footnote(W);
   heading;
   writeln;
   writeln('                              ','    Enter your school''s name: ');
   writeln('                     (will be displayed when users make
booking)');
   writeln;
   write('                       ');
   textcolor(yellow);
   W := input(FALSE, 3);

   textcolor(lightcyan);
   if done = TRUE then
      goto 9
   else if (pressed) and (done = FALSE) THEN
      goto 10;

   schoolname := W;
   WriteText;

end
```

```
          else if choice = 2 then
          begin
            11:clrscr;
            W := '* Press "Escape" to the previous page.';
            Footnote(W);
            heading;
            count := 1;
            done := FALSE;
            writeln(' Add Venues for your school: (Enter 1 - 8 number of Venue
names)');
            writeln(' ----------------------------------------------------------------');
            repeat
              textcolor(lightcyan);
              write('> ');
              textcolor(lightgreen);
              W := input(FALSE, 3);

              if done = TRUE then
                goto 9
              else if pressed then
                goto 11;

              for i := 1 to total_room do
                if (W = Room[i]) and (length(w) = length(room[i])) then
                  Find := TRUE;

              if find = FALSE then
              begin

                textcolor(yellow);
                writeln('(Venue added!)');
                textcolor(lightcyan);
                count := count + 1;
                total_room := total_room + 1;
                Room[total_room] := W;
                WriteText;
              end
              else
              begin
```

```
                clrscr;
                textcolor(lightred);
                writeln('- Duplicated Venue! -');
                writeln;
                write('Press <Enter> to retry...');
                readln;
                goto 11;
                clrscr;
            end;
        until (done = TRUE) or (count = 9);
        if i = 9 then
        begin
            textcolor(yellow);
            writeln(' - 10 Venues Added! -)');
            textcolor(lightgray);
            write('Press <Enter> to the previous page...');
            textcolor(lightcyan);
            readln;
        end;
    end

    else if choice = 3 then
    begin
      if total_room >0 then
      begin
        12:clrscr;
        W := '* Press "Escape" to the previous page.';
        if total_room > 13 then
            begin
                textcolor(lightcyan);
                gotoxy(1,(24 + total_room - 13));

write('-------------------------------------------------------------------------------');
                textcolor(cyan);
                writeln(W);
                gotoxy(1,1);
            end
            else
                footnote(W);
            heading;
        W := '';
        writeln('- Remove School''s Venues -');
        writeln;
        writeln(' No.        Venue');
        write('
=======================================================
====================');
        for i:= 1 to total_room do
        begin
          if i < 10 then
            writeln(' ',i,'      ',Room[i])
          else if i < 100 then
            writeln(' ',i,'     ',Room[i])
          else
            writeln(' ',i,'    ',Room[i]);
        end;
        write('
=======================================================
====================');
        writeln;

        write('> Select the Venue you want to remove (No.) : ');
        W := input(FALSE,3);
        val(W, remove_id, error);
        if (pressed) and (done = FALSE) then
          goto 12;


        if (error = 0) and (remove_id > 0) and (remove_id <= total_room)
then
        begin
          i := 1;
          repeat
            if Room[remove_id] = Room[booking[i].venue] then
              find := TRUE;
            i := i + 1;
          until (i > total_booking) or (find = TRUE);

          if find = TRUE then
```

```
                    begin
                        i := i - 1;
                        booking[i].Year := booking[i + 1].Year;
                        booking[i].Month := booking[i + 1].Month;
                        booking[i].Day := booking[i + 1].Day;
                        booking[i].User := booking[i + 1].User;
                        booking[i].Venue := booking[i + 1].Venue;
                        booking[i].Time := booking[i + 1].Time;

                        i := i + 1;

                        While i <> (total_booking + 1) do
                        begin
                            booking[i].Year := booking[i + 1].Year;
                            booking[i].Month := booking[i + 1].Month;
                            booking[i].Day := booking[i + 1].Day;
                            booking[i].User := booking[i + 1].User;
                            booking[i].Venue := booking[i + 1].Venue;
                            booking[i].Time := booking[i + 1].Time;
                            i := i + 1;
                        end;

                        total_booking := total_booking - 1;
                    end;



                    Room[remove_id] := Room[remove_id + 1];
                    i := remove_id + 1;
                    While i <> (total_room + 1) do
                    begin
                        Room[i] := Room[i+1];
                        i := i + 1;
                    end;
                    total_room := total_room - 1;
                    WriteText;
                end
                else if (error <> 0) and (W <> '') or (remove_id < 0) or
(remove_id > total_room) then
                begin
                    textcolor(lightred);
```

```
            writeln('- Invalid Input! -');
            writeln;
            write('Press <Enter> to retry...');
            readln;
            goto 12;
            clrscr;
          end;
        end
        else
      begin
        clrscr;
        textcolor(lightcyan);
        W := '';
        footnote(W);
        heading;
        textcolor(lightred);
        writeln('- No Venue to remove! -');
        writeln;
        write('Press <Enter> to the previous page...');
        readln;
      end;
      end;
    until (error = 0) and (choice > 0) and (choice < 3) and (menu = TRUE);



    end

    else if action = 4 then
    begin

      4:repeat
          done := FALSE;
          clrscr;
          W := '* Press "Escape" to menu.';
          Footnote(W);
          heading;
          writeln('- Change Password -');
          writeln;
          write('> Old Password: ');
      textcolor(green);
      W := Input(TRUE,2);
      if done = TRUE then
         goto 3
      else if pressed then
         goto 4;

      if W <> password then
      begin
         textcolor(lightred);
         writeln('                                    - Invalid Password! -');
         delay(2100);
      end;
   until (W = password);

   writeln;
   textcolor(lightcyan);
   write('> New Password (PW not shorter than 3): ');
   textcolor(lightgreen);
   pw1 := Input(TRUE,2);
 if done = TRUE then
    goto 3
 else if pressed then
    goto 4;
 writeln;

 textcolor(lightcyan);
 write('> Confirm New Password : ');
 textcolor(lightgreen);
 pw2 := Input(True,2);
 if done = TRUE then
    goto 3
 else if pressed then
    goto 4;

 if (pw1 = pw2) and (length(pw1) > 3) then
 begin
    done := TRUE;
    delay(1000);
    writeln;
```

```
          textcolor(yellow);
          writeln('- Password Changed! -');
          delay(2100);
          userR[user_num].pw := pw1;
          WriteText;
      end
      else
      begin
          writeln;
          textcolor(lightred);
          if pw1 <> pw2 then
          begin
              writeln('- Confirm password doesn''t match! -');
              writeln;
              writeln('Press <Enter> to retry...');
              readln;
              goto 4;
              clrscr;
          end
          else
          begin
              textcolor(lightred);
              writeln('- Your Input is too short! -');
              writeln;
              write('Press <Enter> to retry...');
            readln;
            goto 4;
            clrscr;
        end;
    end;

end

else if action = 5 then
begin

    repeat
        clrscr;
        writeln;
        gotoxy(1,9);
```

```
                textcolor(cyan);
                writeln('
=======================================');
                textcolor(lightred);
                gotoxy(1,2);
                writeln('                              ! Warning !');
                writeln;
                textcolor(cyan);
                writeln('
=======================================');
                textcolor(lightcyan);
                writeln('                      Are you sure you want to
logout?');
                writeln;
                write('                              (y/n): ');
                readln(W);
                if (W = 'Y') or (W = 'y') then
                begin
                  logout := TRUE;
                end
                else if (W = 'N') or (W = 'n') then
                  logout := FALSE;
              until (W = 'y') or (W = 'Y') or (W = 'n') or (W = 'N');

          end

          else if action = 7 then
          begin
        19:if schoolname = '#' then
            begin
              clrscr;
              W := '* Press "Escape" to menu.';
              Footnote(W);
              Heading;
              textcolor(lightred);
              writeln('You hasn''t inputted a school name yet!');
              writeln;
              textcolor(lightgray);
              write('Press <Enter> to main menu...');
              readln;
            end
            else
            begin


              total_userbooking := 0;
              i := 1;

              for j := 1 to total_booking do
              begin
                if booking[j].user = user_id then
                begin
                  total_userbooking := total_userbooking + 1;
                  booking_num[i] := j;
                  i := i + 1;
                end;
              end;


              if total_userbooking > 0 then
              begin
                clrscr;
                W := '* Press "Escape" to the previous page.';
                if total_userbooking > 13 then
                  begin
                    textcolor(lightcyan);
                    gotoxy(1,(24 + total_userbooking - 13));

write('-------------------------------------------------------------------------------');
                    textcolor(cyan);
                    writeln(W);
                    gotoxy(1,1);
                  end
                  else
                    footnote(W);
                  Heading_booking(schoolname);

                  W := '';
                  writeln(                      '- Remove Booking Records -');
                  writeln;
```

```
                writeln('    No.              Date                 Time
Venue                                 ');
                write('
=============================================================
===================');
                writeln;

                for i:= 1 to total_userbooking do
                begin
                  if i < 10 then
                     write('    ', i, '                    ',booking[booking_num[i]].Year,
'/')
                  else if i < 100 then
                    writeln('    ', i, '                  ',booking[booking_num[i]].Year,
'/')
                  else
                    writeln('    ', i, '                ',booking[booking_num[i]].Year,
'/');



                  if booking[booking_num[i]].Month < 10 then
                    write('0',booking[booking_num[i]].Month,'/')
                  else
                    write(booking[booking_num[i]].Month,'/');
                  if booking[booking_num[i]].Day < 10 then
                    write('0',booking[booking_num[i]].Day)
                  else
                    write(booking[booking_num[i]].Day);

                  write('      ',time[booking[booking_num[i]].time],'            ',
copy(room[booking[booking_num[i]].venue],1,18));
                  writeln;

                end;

                write('
=============================================================
===================');
                writeln;
```

```
                    writeln;

                    write('> Select the record you want to remove (No.) : ');
                    textcolor(lightgreen);
                    W := input(FALSE,3);
                    textcolor(lightcyan);
                    val(W, remove_id, error);
                    if done = true then
                       goto 3
                    else if (pressed) and (done = FALSE) then
                       goto 19;



                    if (error = 0) and (remove_id > 0) and (remove_id <=
total_userbooking) then
                    begin

                       booking[booking_num[remove_id]].Year :=
booking[booking_num[remove_id] + 1].Year;
                       booking[booking_num[remove_id]].Month :=
booking[booking_num[remove_id] + 1].Month;
                       booking[booking_num[remove_id]].Day :=
booking[booking_num[remove_id] + 1].Day;
                       booking[booking_num[remove_id]].User :=
booking[booking_num[remove_id] + 1].User;
                       booking[booking_num[remove_id]].Venue :=
booking[booking_num[remove_id] + 1].Venue;
                       booking[booking_num[remove_id]].Time :=
booking[booking_num[remove_id] + 1].Time;

                       i := remove_id + 1;

                       While i <> (total_userbooking + 1) do
                       begin
                          booking[booking_num[i]].Year := booking[booking_num[i]
+ 1].Year;
                          booking[booking_num[i]].Month := booking[booking_num[i]
+ 1].Month;
                          booking[booking_num[i]].Day := booking[booking_num[i] +
1].Day;
```

```
                 booking[booking_num[i]].User := booking[booking_num[i]
+ 1].User;
                 booking[booking_num[i]].Venue := booking[booking_num[i]
+ 1].Venue;
                 booking[booking_num[i]].Time := booking[booking_num[i]
+ 1].Time;
                 i := i + 1;
               end;

               total_userbooking := total_userbooking - 1;
               total_booking := total_booking - 1;

               WriteText;
             end

             else if (error <> 0) and (W <> '') or (remove_id < 0) or
(remove_id > total_userbooking) then
             begin
               clrscr;
               textcolor(lightred);
               writeln('- Invalid Input! -');
               writeln;
               write('Press <Enter> to retry...');
               readln;
               goto 19;
               clrscr;
             end;
           end

           else
           begin
             clrscr;
             textcolor(lightcyan);
             W := '';
             footnote(W);
             heading;
             textcolor(lightred);
             writeln('- No record to cancel! -');
             writeln;
             write('Press <Enter> to the previous page...');
```

```
                readln;
            end;
          end;
        end

        else if action = 8 then
        begin
            done := FALSE;
            flag[1] := FALSE;
            Available_date;
            clrscr;
            W := 'Press <Enter> to menu.';
            if total_booking > 13 then
            begin
                textcolor(lightcyan);
                gotoxy(1,(24 + total_booking - 13));

write('-------------------------------------------------------------------------------');
                textcolor(cyan);
                writeln(W);
                gotoxy(1,1);
            end
            else
                footnote(W);
            Heading_booking(schoolname);
            writeln('                                  - Display Booking -');
            writeln;
            writeln('    Date              Time                    Venue
User');
            write('
=================================================================
====================');
            writeln;
            for i:= 1 to total_booking do
            begin
                write('    ',booking[i].Year, '/');
                if booking[i].Month < 10 then
                    write('0',booking[i].Month,'/')
                else
                    write(booking[i].Month,'/');
                if booking[i].Day < 10 then
                    write('0',booking[i].Day)
                else
                    write(booking[i].Day);

                write('      ',time[booking[i].time],'           ',
copy(room[booking[i].venue],1,18));

writeln(copy(booking[i].user,1,18):(27+length(copy(booking[i].user,1,18))-length(
copy(room[booking[i].venue],1,18))));
            end;
            write('
=================================================================
===================');
            readln;

        end



        else if action = 6 then
        begin

            16:if schoolname = '#' then
            begin
                clrscr;
                W := '* Press "Escape" to menu.';
                Footnote(W);
                Heading;
                textcolor(lightred);
                writeln('You hasn''t inputted a school name yet!');
                writeln;
                textcolor(lightgray);
                write('Press <Enter> to main menu...');
                readln;
            end
            else
            begin
                repeat
                    done := FALSE;
                    flag[1] := FALSE;
```

```
Available_date;
clrscr;
W := '* Press "Escape" to menu.';
Footnote(W);
Heading_booking(schoolname);
write(' Location: Make Booking > ');
textcolor(yellow);
textbackground(yellow);
writeln('Selecting Date');
textbackground(black);
textcolor(lightcyan);
writeln;
write('    Booking Date Available : ');
textcolor(yellow);
writeln(Year, '/', month, '/', Day, ' - ', yyyy[total_date], '/',
mm[total_date], '/', dd[total_date]);
textcolor(lightcyan);
writeln;
write(' > Please enter the date you want to book: ');
textcolor(lightgreen);
W := input(FALSE, 2);
textcolor(lightcyan);

if done = TRUE then
   goto 3
else if (pressed) and (done = FALSE) then
   goto 16;

A := copy(W, 1, 4);
val(A, input_yyyy, error);
delete(W, 1, 5);
A := copy(W, 1, (pos('/', W) - 1));
val(A, input_mm, error);
delete(W, 1, pos('/', W));
A := copy(W, 1, length(W));
val(A, input_dd, error);

if error = 0 then
begin
```

```
            for i := find_location to total_date do
                if input_yyyy = yyyy[i] then
                    if input_mm = mm[i] then
                        if input_dd = dd[i] then
                            flag[1] := TRUE;

            if flag[1] = FALSE then
            begin
                writeln;
                textcolor(lightred);
                writeln('- Invalid Date! - ');
                writeln;
                textcolor(lightgray);
                writeln('Press <Enter> to retry...');
                textcolor(lightcyan);
                readln;
            end

        end
        else if error <> 0 then
        begin
            writeln;
            textcolor(lightred);
            writeln('- Invalid Input! - ');
            writeln;
            writeln('Press <Enter> to retry...');
            readln;
        end;

    until (error = 0) and (flag[1] = TRUE);


17:repeat
        done := FALSE;
        flag[2] := FALSE;
        Available_date;
        clrscr;

        W := '* Press "Escape" to the previous section.';
        if total_room > 10 then
```

```
                    begin
                       textcolor(lightcyan);
                       gotoxy(1,(24 + total_room - 10));

write('-------------------------------------------------------------------------------');
                       textcolor(cyan);
                       writeln(W);
                       gotoxy(1,1);
                    end
                    else
                       footnote(W);

                    Heading_booking(schoolname);
                    write(' Location: Make Booking > Selecting Date > ');
                    textcolor(yellow);
                    textbackground(yellow);
                    writeln('Selecting Venue');
                    textbackground(black);
                    textcolor(lightcyan);
                    writeln;
                    write(' Selected Date : ');
                    textcolor(yellow);
                    writeln(input_yyyy,'/', input_mm, '/', input_dd);
                    textcolor(lightcyan);
                    writeln;
                    writeln('       No.         VENUE');
                    write('
==============================================================
====================');
                    for i:= 1 to total_room do
                       if i < 10 then
                          writeln('       ',i,'            ',Room[i])
                       else if i < 100 then
                          writeln('       ',i,'          ',Room[i])
                       else
                          writeln('       ',i,'       ',Room[i]);
                    write('
==============================================================
====================');
                    writeln;
```

```
                write(' > Select the Venue you want to book (No.): ');
                textcolor(lightgreen);
                W := input(FALSE, 3);
                textcolor(lightcyan);
                if done = TRUE then
                   goto 16
                else if (pressed) and (done = FALSE) then
                   goto 17;

                val(W, input_venue, error);
                if (error = 0) and (input_venue > 0) and (input_venue <=
total_room) then
                     flag[2] := TRUE
                else
                begin
                   clrscr;
                   writeln;
                   textcolor(lightred);
                   writeln('- Invalid Input! - ');
                   writeln;
                   textcolor(lightgray);
                   writeln('Press <Enter> to retry...');
                   textcolor(lightcyan);
                   readln;
                end;

             until (flag[2] = TRUE) and (error = 0);



         18:repeat
                for i := 1 to 3 do
                   status[i] := FALSE;
                done := FALSE;
                flag[3] := FALSE;
                Available_date;
                clrscr;
                W := '* Press "Escape" to the previous section.';
                Footnote(W);
                Heading_booking(schoolname);
                write(' Location: Make Booking > Selecting Date > Selecting
Venue > ');
                textcolor(yellow);
                textbackground(yellow);
                writeln('Selecting Time');
                textbackground(black);
                textcolor(lightcyan);
                writeln;
                write(' Selected Date : ');
                textcolor(yellow);
                writeln(input_yyyy,'/', input_mm, '/', input_dd);
                textcolor(lightcyan);
                write(' Selected Time : ');
                textcolor(yellow);
                writeln(Room[input_venue]);
                textcolor(lightcyan);
                writeln;

                for i := 1 to total_booking do
                  if (input_yyyy = booking[i].year) and (input_mm =
booking[i].month) and (input_dd = booking[i].day) and (input_venue =
booking[i].venue) then
                    status[booking[i].time] := TRUE;



                writeln('     No.      Time                    Status');
                write('
=================================================================
====================');

                for i:= 1 to 3 do
                begin
                  write('     ',i,'       ',Time[i]);
                  if status[i] = TRUE then
                    writeln('              Unavailable')
                  else
                    writeln;
                end;

                write('
=====================================================================
====================');
                writeln;
                write(' > Select the Time you want to book (No.): ');
                textcolor(lightgreen);
                W := input(FALSE, 3);
                textcolor(lightcyan);

                if done = TRUE then
                  goto 17
                else if (pressed) and (done = FALSE) then
                  goto 18;

                val(W, input_time, error);
                if (error = 0) and (input_time > 0) and (input_time <= 3) and
(status[input_time] =    FALSE) then
                  flag[3] := TRUE
                else if status[input_time] = TRUE then
                begin
                  clrscr;
                  writeln;
                  textcolor(lightred);
                  writeln('- Time Unavailable! - ');
                  writeln;
                  textcolor(lightgray);
                  writeln('Press <Enter> to retry...');
                  textcolor(lightcyan);
                  readln;
                end
                else
                begin
                  clrscr;
                  writeln;
                  textcolor(lightred);
                  writeln('- Invalid Input! - ');
                  writeln;
                  textcolor(lightgray);
                  writeln('Press <Enter> to retry...');
                  textcolor(lightcyan);
                  readln;
```

```
                  end;

                until (flag[1] = TRUE) and (flag[2] = TRUE) and (flag[3] = TRUE);

                delay(1000);
                clrscr;
                W := '';
                footnote(W);
                heading;
                textcolor(yellow);
                writeln('- The Booking has been Made! -');
                textcolor(darkgray);
                delay(1400);
                writeln;

                textcolor(lightcyan);
                writeln('                                        Booking Details:');
                writeln;
                textcolor(lightcyan);
                writeln('
=================================');
                writeln;
                writeln('                              > Date : ', input_yyyy, '/',
input_mm, '/', input_dd);
                writeln;
                writeln('                              > Venue : ',
Room[input_venue]);
                writeln;
                writeln('                              > Time : ', time[input_time]);
                writeln;
                writeln('
=================================');
                writeln;
                textcolor(darkgray);
                write('Press <Enter> to menu...');
                readln;

                total_booking := total_booking + 1;
                With booking[total_booking] do
                begin
```

```
            Year := input_yyyy;
            Month := input_mm;
            Day := input_dd;
            Venue := input_venue;
            Time := input_time;
          end;
          booking[total_booking].user := userR[user_num].id;

          WriteText;


        end;
      end;
    end;
  until (error = 0) and (logout = TRUE);
end;



procedure Menu_User;
var
  action, i, j, remove_id : integer;
begin
  7:repeat
    logout := FALSE;
    clrscr;
    W := '';
    footnote(W);
    textcolor(lightcyan);
    heading_login(user_id, 1);
    writeln;
    writeln('                                MENU');
    writeln('                        ======================');
    writeln;
    writeln('                               General ');
    writeln('                           ----------------- ');
    writeln('                        1. Make Booking');
    writeln('                        2. Cancel Booking');
    writeln;
    writeln('                              Personal');
    writeln('                                    ----------------');
    writeln('                                 3. Change Password');
    writeln('                                 4. Logout');
    writeln;
    writeln;
    write(' Please choose your action: ');
    textcolor(yellow);
    readln(W);
    val(W, action, error);
    if (error <> 0) or (action <= 0) or (action > 4) then
    begin
        writeln;
        textcolor(lightred);
        writeln('- Invalid Input! - ');
        writeln;
        writeln('Press <Enter> to retry...');
        readln;
    end
    else
    begin
      if action = 1 then
      begin
    13:if schoolname = '#' then
        begin
          clrscr;
          W := '* Press "Escape" to menu.';
          Footnote(W);
          Heading;
          textcolor(lightred);
          writeln('The Admin hasn"t start the booking service!');
          writeln;
          textcolor(lightgray);
          write('Press <Enter> to main menu...');
          readln;
        end
        else
        begin
          repeat
            pressed := FALSE;
            done := FALSE;
```

```
                    flag[1] := FALSE;
                    Available_date;
                    clrscr;
                    W := '* Press "Escape" to menu.';
                    Footnote(W);
                    Heading_booking(schoolname);
                    write(' Location: Make Booking > ');
                    textcolor(yellow);
                    textbackground(yellow);
                    writeln('Selecting Date');
                    textbackground(black);
                    textcolor(lightcyan);
                    writeln;
                    write('    Booking Date Available : ');
                    textcolor(yellow);
                    writeln(Year, '/', month, '/', Day, ' - ', yyyy[total_date], '/',
mm[total_date], '/', dd[total_date]);
                    textcolor(lightcyan);
                    writeln;
                    write(' > Please enter the date you want to book: ');
                    textcolor(lightgreen);
                    W := input(FALSE, 2);
                    textcolor(lightcyan);

                    if done = TRUE then
                       goto 7
                    else if (pressed) and (done = FALSE) then
                       goto 13;

                    A := copy(W, 1, 4);
                    val(A, input_yyyy, error);
                    delete(W, 1, 5);
                    A := copy(W, 1, (pos('/', W) - 1));
                    val(A, input_mm, error);
                    delete(W, 1, pos('/', W));
                    A := copy(W, 1, length(W));
                    val(A, input_dd, error);

                    if error = 0 then
                    begin
```

```
            for i := find_location to total_date do
                if input_yyyy = yyyy[i] then
                    if input_mm = mm[i] then
                        if input_dd = dd[i] then
                            flag[1] := TRUE;

            if flag[1] = FALSE then
            begin
                writeln;
                textcolor(lightred);
                writeln('- Invalid Date! - ');
                writeln;
                textcolor(lightgray);
                writeln('Press <Enter> to retry...');
                textcolor(lightcyan);
                readln;
            end

        end
        else if error <> 0 then
        begin
            writeln;
            textcolor(lightred);
            writeln('- Invalid Input! - ');
            writeln;
            writeln('Press <Enter> to retry...');
            readln;
        end;

    until (error = 0) and (flag[1] = TRUE);


14:repeat
        done := FALSE;
        flag[2] := FALSE;
        Available_date;
        clrscr;
        W := '* Press "Escape" to the previous section.';
        Footnote(W);
```

```
                    Heading_booking(schoolname);
                    write(' Location: Make Booking > Selecting Date > ');
                    textcolor(yellow);
                    textbackground(yellow);
                    writeln('Selecting Venue');
                    textbackground(black);
                    textcolor(lightcyan);
                    writeln;
                    write(' Selected Date : ');
                    textcolor(yellow);
                    writeln(input_yyyy,'/', input_mm, '/', input_dd);
                    textcolor(lightcyan);
                    writeln;
                    writeln('       No.         VENUE');
                    write('
============================================================
======================');
                    for i:= 1 to total_room do
                       if i < 10 then
                          writeln('       ',i,'            ',Room[i])
                       else if i < 100 then
                          writeln('       ',i,'          ',Room[i])
                       else
                          writeln('       ',i,'       ',Room[i]);
                    write('
============================================================
======================');
                    writeln;
                    write(' > Select the Venue you want to book (No.): ');
                    textcolor(lightgreen);
                    W := input(FALSE, 3);
                    textcolor(lightcyan);
                    if done = TRUE then
                       goto 13
                    else if (pressed) and (done = FALSE) then
                       goto 14;

                    val(W, input_venue, error);
                    if (error = 0) and (input_venue > 0) and (input_venue <=
total_room) then
```

```
                flag[2] := TRUE
            else
            begin
                clrscr;
                writeln;
                textcolor(lightred);
                writeln('- Invalid Input! - ');
                writeln;
                textcolor(lightgray);
                writeln('Press <Enter> to retry...');
                textcolor(lightcyan);
                readln;
            end;

        until (flag[2] = TRUE) and (error = 0);



    15:repeat
            for i := 1 to 3 do
                status[i] := FALSE;
            done := FALSE;
            flag[3] := FALSE;
            Available_date;
            clrscr;
            W := '* Press "Escape" to the previous section.';
            Footnote(W);
            Heading_booking(schoolname);
            write(' Location: Make Booking > Selecting Date > Selecting
Venue > ');
            textcolor(yellow);
            textbackground(yellow);
            writeln('Selecting Time');
            textbackground(black);
            textcolor(lightcyan);
            writeln;
            write(' Selected Date : ');
            textcolor(yellow);
            writeln(input_yyyy,'/', input_mm, '/', input_dd);
            textcolor(lightcyan);
            write(' Selected Time : ');
            textcolor(yellow);
            writeln(Room[input_venue]);
            textcolor(lightcyan);
            writeln;

            for i := 1 to total_booking do
              if (input_yyyy = booking[i].year) and (input_mm =
booking[i].month) and (input_dd = booking[i].day) and (input_venue =
booking[i].venue) then
                status[booking[i].time] := TRUE;



            writeln('      No.       Time                         Status');
            write('
=================================================================
=====================');

            for i:= 1 to 3 do
            begin
              write('      ',i,'         ',Time[i]);
              if status[i] = TRUE then
                writeln('                  Unavailable')
              else
                writeln;
            end;

            write('
=================================================================
=====================');
            writeln;
            write(' > Select the Tie you want to book (No.): ');
            textcolor(lightgreen);
            W := input(FALSE, 3);
            textcolor(lightcyan);

            if done = TRUE then
              goto 14
            else if (pressed) and (done = FALSE) then
              goto 15;
```

```
          val(W, input_time, error);
          if (error = 0) and (input_time > 0) and (input_time <= 3) and
(status[input_time] =    FALSE) then
            flag[3] := TRUE
          else if status[input_time] = TRUE then
          begin
            clrscr;
            writeln;
            textcolor(lightred);
            writeln('- Time Unavailable! - ');
            writeln;
            textcolor(lightgray);
            writeln('Press <Enter> to retry...');
            textcolor(lightcyan);
            readln;
          end
          else
          begin
            clrscr;
            writeln;
            textcolor(lightred);
            writeln('- Invalid Input! - ');
            writeln;
            textcolor(lightgray);
            writeln('Press <Enter> to retry...');
            textcolor(lightcyan);
            readln;
          end;

        until (flag[1] = TRUE) and (flag[2] = TRUE) and (flag[3] = TRUE);

        delay(1000);
        clrscr;
        W := '';
        footnote(W);
        heading;
        textcolor(yellow);
        writeln('- The Booking has been Made! -');
        textcolor(darkgray);
```

```
          delay(1400);
          writeln;

          textcolor(lightcyan);
          writeln('                         Booking Details:');
          writeln;
          textcolor(lightcyan);
          writeln('
=================================');
          writeln;
          writeln('                   > Date : ', input_yyyy, '/',
input_mm, '/', input_dd);
          writeln;
          writeln('                   > Venue : ',
Room[input_venue]);
          writeln;
          writeln('                   > Time : ', time[input_time]);
          writeln;
          writeln('
=================================');
          writeln;
          textcolor(darkgray);
          write('Press <Enter> to menu...');
          readln;

          total_booking := total_booking + 1;
          With booking[total_booking] do
          begin
            Year := input_yyyy;
            Month := input_mm;
            Day := input_dd;
            Venue := input_venue;
            Time := input_time;
          end;
          booking[total_booking].user := userR[user_num].id;

          WriteText;


        end;
```

```
end

else if action = 2 then
begin
    20:if schoolname = '#' then
  begin
      clrscr;
      W := '* Press "Escape" to menu.';
      Footnote(W);
      Heading;
      textcolor(lightred);
      writeln('The Admin hasn''t start the booking service!');
      writeln;
      textcolor(lightgray);
      write('Press <Enter> to main menu...');
      readln;
  end
  else
  begin


    total_userbooking := 0;
    i := 1;

    for j := 1 to total_booking do
    begin
        if booking[j].user = user_id then
        begin
            total_userbooking := total_userbooking + 1;
            booking_num[i] := j;
            i := i + 1;
        end;
    end;



    if total_userbooking > 0 then
      begin
        clrscr;
        W := '* Press "Escape" to the previous page.';
```

```
          if total_userbooking > 13 then
            begin
              textcolor(lightcyan);
              gotoxy(1,(24 + total_userbooking - 13));

write('--------------------------------------------------------------------------------');
              textcolor(cyan);
              writeln(W);
              gotoxy(1,1);
            end
            else
              footnote(W);
            Heading_booking(schoolname);

            W := '';
            writeln(                              '- Remove Booking Records -');
            writeln;
            writeln('    No.              Date                 Time
Venue                                  ');
            write('
==============================================================
===================');
            writeln;

            for i:= 1 to total_userbooking do
            begin
              if i < 10 then
                  write('    ', i, '                   ',booking[booking_num[i]].Year,
'/')
              else if i < 100 then
                writeln('    ', i, '                  ',booking[booking_num[i]].Year,
'/')
              else
                writeln('    ', i, '                ',booking[booking_num[i]].Year,
'/');



              if booking[booking_num[i]].Month < 10 then
                write('0',booking[booking_num[i]].Month,'/')
```

```
                        else
                           write(booking[booking_num[i]].Month,'/');
                        if booking[booking_num[i]].Day < 10 then
                           write('0',booking[booking_num[i]].Day)
                        else
                           write(booking[booking_num[i]].Day);

                        write('      ',time[booking[booking_num[i]].time],'            ',
copy(room[booking[booking_num[i]].venue],1,18));
                        writeln;

                     end;

                     write('
==========================================================
===================');
                     writeln;
                     writeln;

                     write('> Select the record you want to remove (No.) : ');
                     textcolor(lightgreen);
                     W := input(FALSE,3);
                     textcolor(lightcyan);
                     val(W, remove_id, error);
                     if done = true then
                        goto 7
                     else if (pressed) and (done = FALSE) then
                        goto 20;


                     if (error = 0) and (remove_id > 0) and (remove_id <=
total_userbooking) then
                     begin

                        booking[booking_num[remove_id]].Year :=
booking[booking_num[remove_id] + 1].Year;
                        booking[booking_num[remove_id]].Month :=
booking[booking_num[remove_id] + 1].Month;
                        booking[booking_num[remove_id]].Day :=
booking[booking_num[remove_id] + 1].Day;
```

```
                booking[booking_num[remove_id]].User :=
booking[booking_num[remove_id] + 1].User;
                booking[booking_num[remove_id]].Venue :=
booking[booking_num[remove_id] + 1].Venue;
                booking[booking_num[remove_id]].Time :=
booking[booking_num[remove_id] + 1].Time;

                i := remove_id + 1;

                While i <> (total_userbooking + 1) do
                begin
                  booking[booking_num[i]].Year := booking[booking_num[i]
+ 1].Year;
                  booking[booking_num[i]].Month := booking[booking_num[i]
+ 1].Month;
                  booking[booking_num[i]].Day := booking[booking_num[i] +
1].Day;
                  booking[booking_num[i]].User := booking[booking_num[i]
+ 1].User;
                  booking[booking_num[i]].Venue := booking[booking_num[i]
+ 1].Venue;
                  booking[booking_num[i]].Time := booking[booking_num[i]
+ 1].Time;
                  i := i + 1;
                end;

                total_userbooking := total_userbooking - 1;
                total_booking := total_booking - 1;

                WriteText;
              end

              else if (error <> 0) and (W <> '') or (remove_id < 0) or
(remove_id > total_userbooking) then
              begin
                clrscr;
                textcolor(lightred);
                writeln('- Invalid Input! -');
                writeln;
                write('Press <Enter> to retry...');
```

```
            readln;
            goto 20;
            clrscr;
          end;
        end

        else
        begin
          clrscr;
          textcolor(lightcyan);
          W := '';
          footnote(W);
          heading;
          textcolor(lightred);
          writeln('- No record to cancel! -');
          writeln;
          write('Press <Enter> to the previous page...');
          readln;
      end;
    end;
end

else if action = 3 then
begin
  8:repeat
          done := FALSE;
          clrscr;
          W := '* Press "Escape" to menu.';
          Footnote(W);
          heading;
          writeln('- Change Password -');
          writeln;
          write('> Old Password: ');
          textcolor(green);
          W := Input(TRUE,3);
          if done = TRUE then
            goto 7
          else if pressed then
            goto 8;
          if W <> password then
          begin
            textcolor(lightred);
            writeln('                              - Invalid Password! -');
            delay(2100);
          end;
        until (W = password);

        writeln;
        textcolor(lightcyan);
        write('> New Password (PW not shorter than 3): ');
        textcolor(lightgreen);
        pw1 := Input(TRUE,2);
      if done = TRUE then
        goto 7
      else if pressed then
        goto 8;
      writeln;

      textcolor(lightcyan);
      write('> Confirm New Password : ');
      textcolor(lightgreen);
      pw2 := Input(True,2);
      if done = TRUE then
        goto 7
      else if pressed then
        goto 8;

      if (pw1 = pw2) and (length(pw1) > 3) then
      begin
        done := TRUE;
        delay(1000);
        writeln;
        textcolor(yellow);
        writeln('- Password Changed! -');
        delay(2100);
        userR[user_num].pw := pw1;
        WriteText;
      end
      else
```

```
          begin
            writeln;
            textcolor(lightred);
            if pw1 <> pw2 then
            begin
              writeln('- Confirm password doesn"t match! -');
              writeln;
              writeln('Press <Enter> to retry...');
              readln;
              goto 8;
              clrscr;
            end
            else
            begin
              textcolor(lightred);
              writeln('- Your Input is too short! -');
              writeln;
              write('Press <Enter> to retry...');
             readln;
             goto 8;
             clrscr;
           end;
         end;

       end

       else if action = 4 then
       begin

         repeat
           clrscr;
           writeln;
           gotoxy(1,9);
           textcolor(cyan);
           writeln('
====================================== ');
           textcolor(lightred);
           gotoxy(1,2);
           writeln('                                                             ! Warning !');
           writeln;
```

```
                  textcolor(cyan);
                  writeln('
======================================== ');
                  textcolor(lightcyan);
                  writeln('                                        Are you sure you want to
logout?');
                  writeln;
                  write('                                                       (y/n): ');
                  readln(W);
                  if (W = 'Y') or (W = 'y') then
                  begin
                     logout := TRUE;
                  end
                  else if (W = 'N') or (W = 'n') then
                     logout := FALSE;
               until (W = 'y') or (W = 'Y') or (W = 'n') or (W = 'N');

            end;
         end;
     until (error = 0) and (logout = TRUE);


end;



procedure signup;
var
     user_id:string;
begin
     textcolor(lightcyan);
     writeln('Venue Booking System');

writeln('_______________________________________________________________
_____________________');
     writeln('Welcome!');
     writeln;
     writeln('This is the first time you use this system.');
     writeln('Please sign up your administrator account.');
     delay(1500);
     writeln;
```

```
      textcolor(yellow);
      write('Press <Enter> to contiue...');
      readln;
      clrscr;
      flag[1] := FALSE;
      flag[2] := FALSE;

  2:While (flag[2] = FALSE) and (quit = 0) do
      begin
         repeat
            clrscr;
            textcolor(lightcyan);
            writeln('Venue Booking System');

writeln('______________________________________________________________
____________________');
            writeln('- Signup -');
            writeln;
            write('> User ID (ID not shorter than 3): ');
            textcolor(lightgreen);
            user_id := input(FALSE,1);
            if Pressed = TRUE then
               goto 2;
            if length(user_id) > 3 then
               flag[1] := TRUE
            else
            begin
               writeln;
               textcolor(lightred);
               writeln('- Your Input is too short! -');
               writeln;
               write('Press <Enter> to retry...');
               readln;
               clrscr;
            end;
         until flag[1] = TRUE;

         writeln;
         textcolor(lightcyan);
         write('> Password (PW not shorter than 3): ');
```

```
textcolor(lightgreen);
pw1 := Input(TRUE,1);
if Pressed = TRUE then
   goto 2;
writeln;
textcolor(lightcyan);
write('> Confirm Password : ');
textcolor(lightgreen);
pw2 := Input(True,1);
if Pressed = TRUE then
   goto 2;
if (pw1 = pw2) and (length(pw1) > 3) then
begin
   delay(1000);
   writeln;
   textcolor(yellow);
   writeln('- Signup Successful! -');
   textcolor(darkgray);
   writeln;
   write('Press <Enter> to login...');
   readln;

   flag[2] := TRUE;
   rewrite(user);
   writeln(user,user_id);
   writeln(user,pw1);
   writeln(user,'2');
   close(user);

   With UserR[1] do
   begin
        id := user_id;
        pw := pw1;
        permission := '2';
   end;
end
else
begin
     writeln;
     textcolor(lightred);
```

```
                if pw1 <> pw2 then
                begin
                    writeln('- Confirm password doesn''t match! -');
                    writeln;
                    writeln('Press <Enter> to retry...');
                    readln;
                    clrscr;
                end
                else
                begin
                    textcolor(lightred);
                    writeln('- Your Input is too short! -');
                    writeln;
                    write('Press <Enter> to retry...');
                    readln;
                    clrscr;
                end;
          end;
      end;
end;


procedure InsertRecord;
var
      i : integer;
begin
    total_booking := 0;
    total_room := 0;
    total := 0;
    total_date := 0;

    i := 0;
    reset(user);
    while not eof(user) do
    begin
          i := i + 1;
          with UserR[i] do
          begin
                readln(user, id);
                readln(user, pw);
```

```
            readln(user, permission);
      end;
      total := total + 1;
end;
close(user);

reset(Rooms);
readln(Rooms, schoolname);

i := 1;
while not eof(Rooms) do
begin
    readln(Rooms, Room[i]);
    i := i + 1;
    total_room := total_room + 1;
end;
close(Rooms);


i := 1;
reset(valid_date);
While not eof(valid_date) do
begin
    readln(valid_date, W);

    A := copy(W, 1, 4);
    val(A, yyyy[i], error);
    delete(W, 1, 5);
    A := copy(W, 1, (pos('/', W) - 1));
    val(A, mm[i], error);
    delete(W, 1, pos('/', W));
    A := copy(W, 1, length(W));
    val(A, dd[i], error);

    i := i + 1;

    total_date := total_date + 1;
end;
close(valid_date);
```

```
  i := 1;
  reset(booking_record);
  While not eof(booking_record) do
  begin
    readln(booking_record, W);

    A := copy(W, 1, 4);
    val(A, booking[i].Year, error);
    delete(W, 1, 5);
    A := copy(W, 1, (pos('/', W) - 1));
    val(A, booking[i].Month, error);
    delete(W, 1, pos('/', W));
    A := copy(W, 1, length(W));
    val(A, booking[i].Day, error);

    readln(booking_record, booking[i].user);

    readln(booking_record, W);
    val(W, booking[i].venue, error);
    readln(booking_record, W);
    val(W, booking[i].time, error);

    i := i + 1;

    total_booking := total_booking + 1;
  end;


end;




begin
    quit := 0;
    assign(booking_record, 'booking_records.txt');
    assign(valid_date, 'valid_dates.txt');
    assign(rooms, 'rooms.txt');
    assign(user,'user.txt');
    try
      reset(user);
```

```
   readln(user, activitate);
   close(user);

   reset(rooms);
   readln(Rooms, schoolname);
   close(rooms);
   if schoolname = '' then
   begin
      rewrite(rooms);
      writeln(Rooms,'#');
   end;

   reset(valid_date);
   close(valid_date);

   reset(booking_record);
   close(booking_record);
except
   textcolor(lightred);
   writeln('File Error - Please Check Your File: ');
   writeln('user.txt, rooms.txt, booking_records.txt, valid_dates.txt');
   quit := 1;
   activitate := 'E';
   writeln;
   writeln;
   textcolor(darkgray);
   write('Press <Enter> to leave...');
   readln;
end;
if activitate = '' then
   signup;
InsertRecord;
While quit = 0 do
begin
   login;
   if quit = 0 then
      if permission = '2' then
            Menu_Admin
      else if permission = '1' then
            Menu_user;
```

```
    end;
end.
```

Appendix 2: Working schedule

Date	**Event**
Mar-2015	Choice of Topic
Apr-2015	Background research
May-2015	Define the objectives
Jun-2015	Design of Solution
Summer-2015	Design + Implementation
Sept-Nov -2015	Testing + Evaluation
Dec-2015	Conclusion + Discussion