Hong Kong Diploma of Secondary Education

Examination 2016

Information and Communication Technology

(Coursework)

Option D: Software Development

Title: Hostel Booking System

Contents

Chapter 1 Introduction

1.1 Problem & Situation

The Wakenial Hostel, a medium-sized motel, finds it uneasy to organize the traditional booking records of the rooms. It can be attributed to the inevitable blunders, for example, the malorganized clients' data and booking records. It especially happens after the cancel of clients' booking. In a bid to address the problem, the Wakenial Hostel wants to set up a program so that the clients can book the rooms by the program and the data can be organized systematically.

1.2 Objectives

As a friend of the owner's son, I was asked to help develop a program which can fulfil the basic requirements of the hostel booking system. He hopes that the program can help minimize the risk of mistaken and mismatched booking records.

Chapter 2 Design of Solution

2.1 Brief Description

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

I will design:

1. The overall structure of the program
   I. Homepage
   II. User page
2. Basic registration
4. User identification
5. Basic functions of hostel booking system

I assume that:

1. The clients are information literate and should be able to utilize the program.
2. The clients have acquired a certain degree of understanding
3. The clients' phone numbers are in 8 digits.
4. The available booking period is within 1 year
   i.e. When the clients are making the booking on January, the available booking period is up to December.
5. There are no more than 10000 clients in total.

2.2 Intended users and their expectations

The intended users are basically the clients of the Wakenial Hotel who would like to book rooms.

The intended users can expect that the program would be able to help them create an account, change their personal information, book the rooms and cancel the booking, which involves lower risk of blunders. The program they expect should be user-friendly by showing clear instructions and directions as to show how to operate. They may also expect the program to present clean structures so that the users would not be easily confused by the weary interface.

Below are some of the features which may be able to meet the users' expectations:

Features	Function of the feature	How to meet users' expectations
Register	Allow the user to create their accounts	Allow the users to use their own accounts to have further information
Amend personal information	Allow the users to update their personal information	Ensure that the motel staff can keep contact with the clients
Make booking	Users can book rooms and know more information of the rooms	Reduce the risk of blunders due to carelessness or repeated booking
Cancel booking	Users can cancel the appointments they have made	Reduce the risk of blunders due to carelessness or repeated cancelling
Display menu	Allow the users to choose the service they would like to have	Make the program more user-friendly

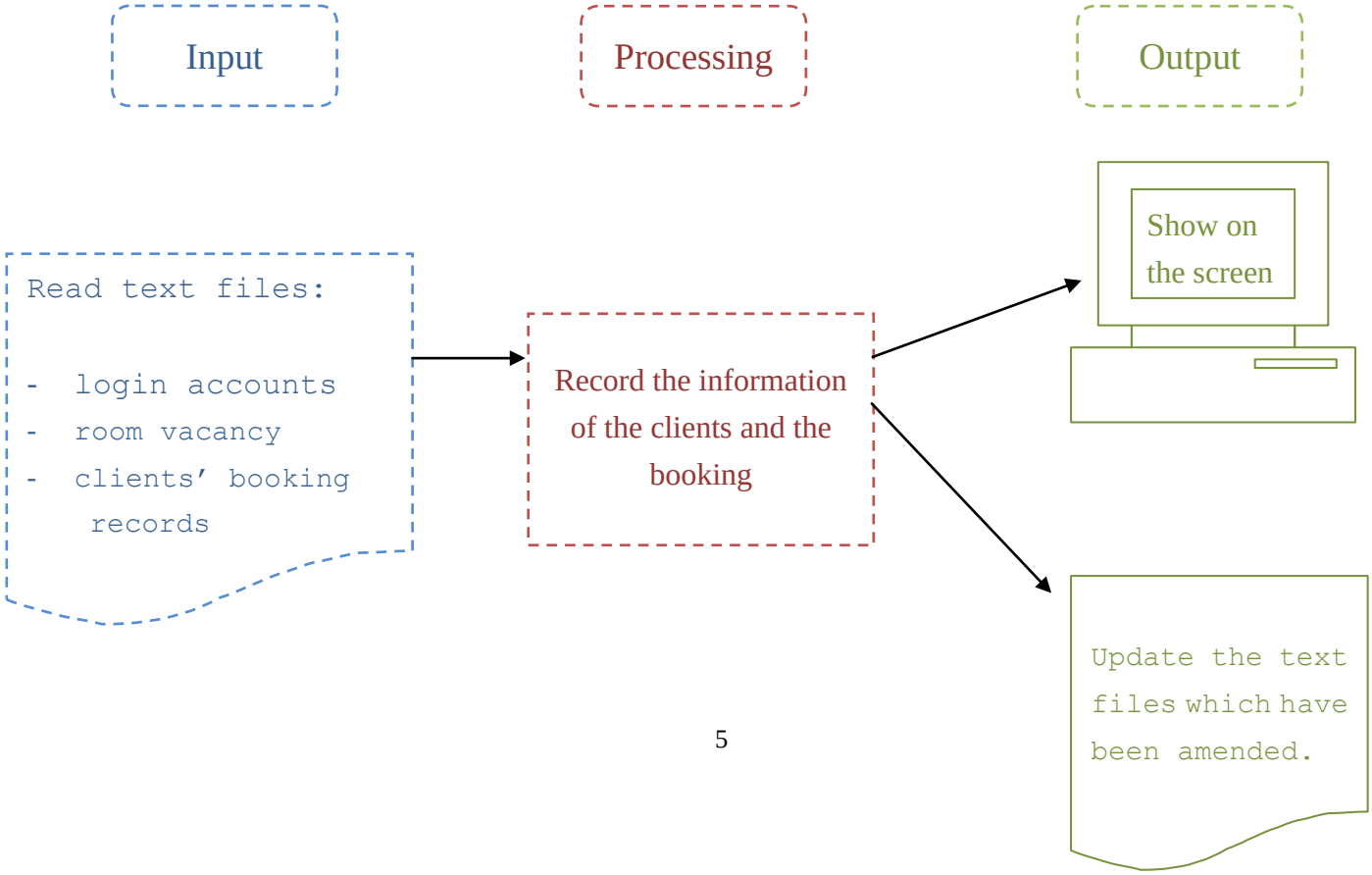
2.3 Refinement of Problem

Handling different kinds of data will result in a mess. In order to avoid such plight, two text files containing clients and the booking records are needed.

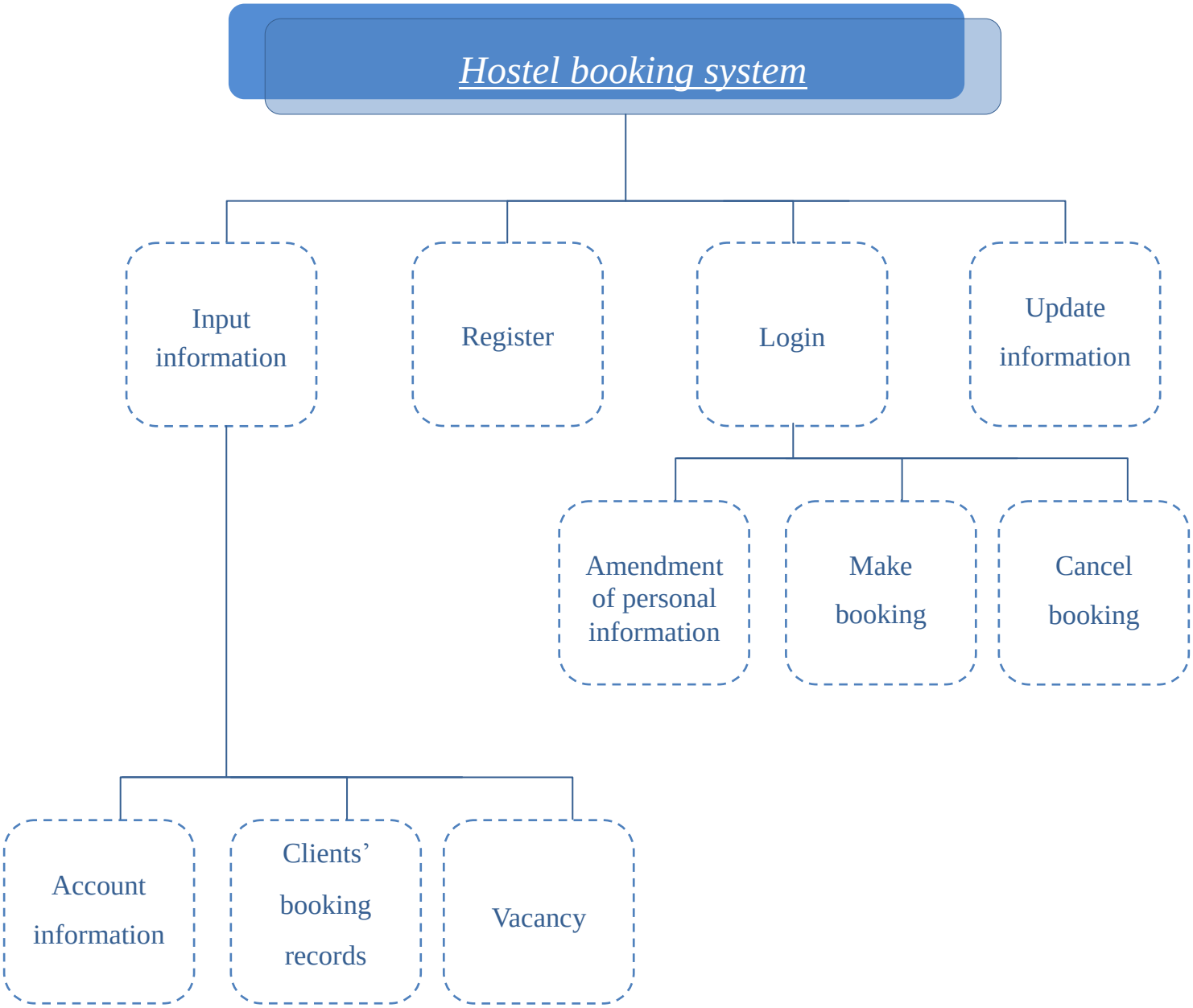
On one hand, similar to the traditional system, the program requires personal information so that the staff can make contact with the clients. Therefore, it is necessary for the clients to register before booking the rooms. One text file will be needed to store the personal information of the clients. On the other hand, another text file will be needed to store clients' booking records. Only when the clients book or cancel their appointments will the second text file be amended. The two files will be hence independent, reducing the risk of blunders and enhancing the efficiency.

However, it is not yet enough. The motel is well-known among all local motels that there will be especially a good many appointments in vacations like summer holiday and Easter Holiday. It is possible for insufficient vacancies during the peak periods. Thus, text files are needed to record which rooms have been booked. Twelve text files are hence used for storing the vacancy of each month.

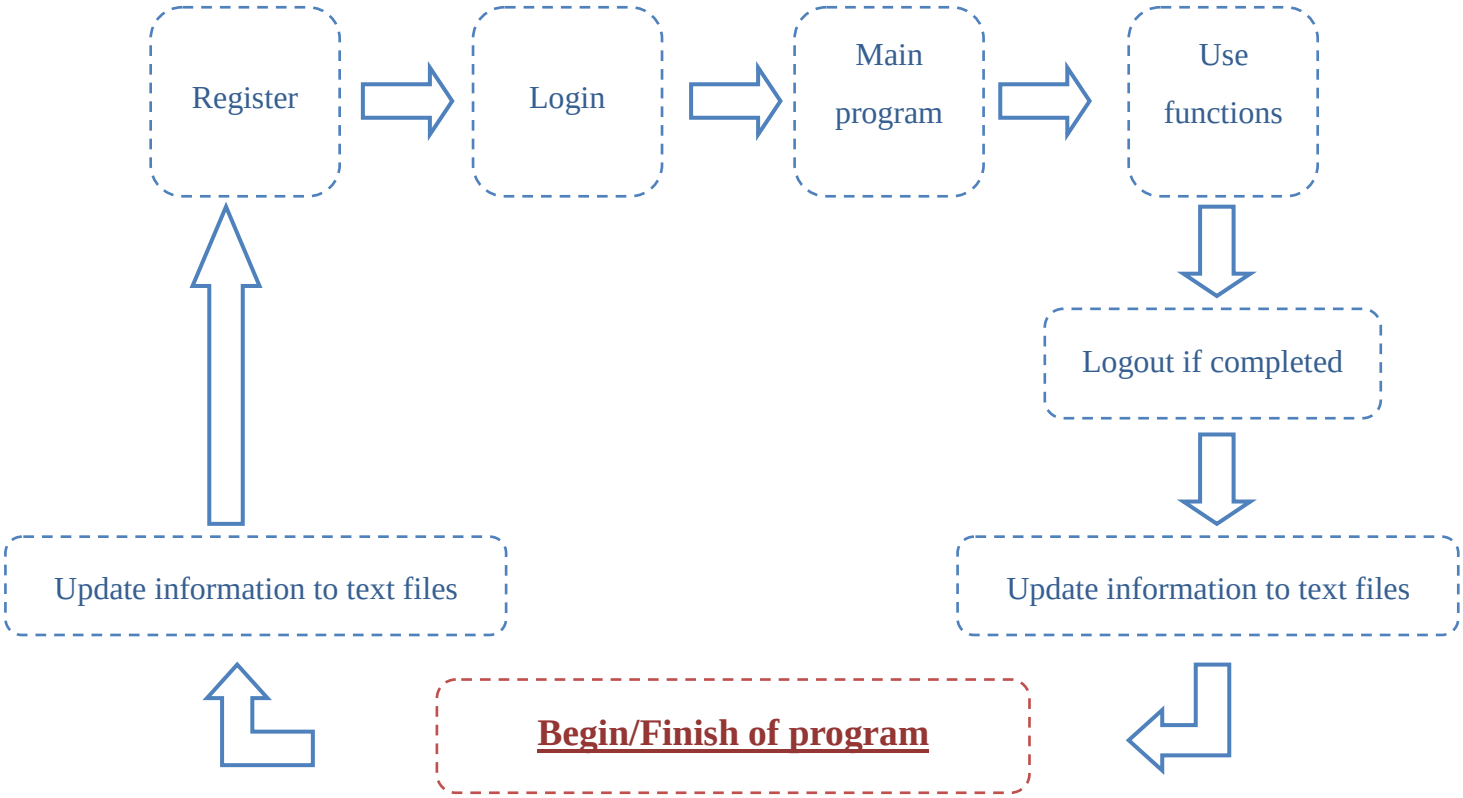
Data flow



Design of the main program



Design of processing



Characteristics of the program

The program will be built by command line interface using Dev-Pascal.

The program is made up of many small-divided procedures, which reassures the reusability of the procedures and minimizes the number of logic statements or algorithms needed. In other words, the file size will be smaller.

Also, it allows the development of the program for a further use as the program can be edited by modifying the procedures or adding extra procedures into the main program.

2.3 Input Data File Formats

Inputting information from text files

It would be good to assign the program to 14 database files:

1. 12 files concerning the vacancy from January to December
2. a file concerning the details of the clients' accounts
3. a file concerning the booking records of all clients

It helps improve database management as the independent text files facilitate information updating while the complexity is lower. For example, the booking records and personal information of the clients are separated to avoid unnecessary amendments. In other words, this can enhance the efficiency of the program.

File1: records.txt

This text file is used to record the login accounts and personal information of the clients. Once a new account has been created, the text file will immediately update the information so that the clients can log in their accounts right after the registration. The text file will be read at the very beginning automatically.

Information Catalogue	Data types
Username	12 strings
Password	20 strings
Sex	1 character
Surname	10 characters
First name	22 characters
Telephone number	1 integer

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

e.g.

Username	Password	Sex	Surname	First name	Telephone number
Wakenial	Hostel66	F	Wak	Chan	6204 6698

Sample file:

Password Surname First name

```
s10000  a12345678  M Cheung  Chi Chung          93399210
s10002  a12345678  M Cheung  Koo Ho             92656221
s10003  a12345678  F Cheung  Yee Hung, Teresa   91913232
s10004  a12345678  F Chui    Tse Lung           91170243
s10005  a12345678  F Chung   Kei Man, Mandy     90427254
s10006  a12345678  F Fung    Siu Wan, Melanie   52065563
s10007  a12345678  M Ho      Kam Shui, Tom      62064552
s10008  a12345678  F Ho      Li, Lily           68554273
s10009  a12345678  M Hui     Chung Hong         62905816
s10010  a12345678  F Hung    Yen Yen            31154226
s10011  a12345678  M Kam     Yung Kai           60153321
s10012  a12345678  M Kwan    Wan Cheong         94678530
s10013  a12345678  F Kwan    Yuen Ching, Cindy  92605817
s10014  a12345678  M Kwok    Foo Ho             98543210
s10015  a12345678  M Lam     Tik Man            57366320
s10016  a12345678  M Lam     Ting Cheong        58275980
s10017  a12345678  M Lau     Yiu Wing           54011256
s10018  a12345678  F Lau     Yum Meow           92435581
s10019  a12345678  M Tang    Tsz Kit            95405810
s10020  a12345678  M Li      Man Hon            54891130
s10021  a12345678  M Tse     Wai Tak            53215321
s10022  a12345678  M Tang    Ho Ming            67868213
s10023  a12345678  F Leung   Chun Ying          68182020
s10024  a12345678  M Ngai    Yat To             59773256
s10025  a12345678  M Tan     Chi Chung          60152765
s10026  a12345678  M Tsang   King Fung          60138912
s10027  a12345678  M Tang    Chi Fung           58666214
s10028  a12345678  M Leung   Wai Fung           59744202
s10029  a12345678  F Tsang   Kwong Wing         68758765
s10030  a12345678  F Tang    Kwun Leong         69135846
s10031  a12345678  F Tang    Pui Yip            69953258
s10032  a12345678  M Tong    Chun Wai           63204795
s10033  a12345678  M Tam     Wai Fai            60153574
s10034  a12345678  F Pong    Ying Sze           63248954
s10035  a12345678  F Pang    Ling Yee           64358721
s10036  a12345678  F Tam     King Kwun          32548965
s10037  a12345678  M Ng      Yu Kit             66453356
s10038  a12345678  F Po      Pak Hong           66412333
s10039  a12345678  F Siu     Ting Yee           63386338
s10040  a12345678  F Tam     Chung Lai          36891541
```

Username Sex Telephone number

Remark:

Other possible structures of **records.txt**:

```
Wakenial
Hostel66
F
Wak
Chan
62046698

Carson_66
19980606
M
Carson
Chan
68769626
.
.
.
```

Record of first client

Record of second client

File 2 - File13: rm1.txt, rm2.txt, rm3.txt, … , rm12.txt

Thess text file are used to store the vacancy of the hostel in different months, from January to December. Once the clients booked the rooms, or, once the clients cancelled the booking, the text files will immediately update the information so that the risk of blunders will reduce. The text files will be read at the very beginning automatically.

Information Catalogue	Data types
Room number	1 integer
Type of room	1 integer
Day & Status	4 strings

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

e.g.1 In rm1.txt,

Room number	Type of room	Day & Status	Day & Status	…	Day & Status
101	1	1	2x		31

e.g.2 In rm1.txt,

Room number	Type of room	Day & Status	Day & Status	…	Day & Status
101	1	1x	2x		31x

In rm2.txt,

Room number	Type of room	Day & Status	Day & Status	…	Day & Status
101	1	1x	2x		31

For 'Type of room': 1 stands for single room, 2 stands for twin room, 3 stands for family room and 4 stand for suite room.

For 'Day & Status': if there is 'x' after the day, it represents that the room is booked for that day. Take e.g.1 as an example, the '2x' implies that Room 101 is not available on 2 January as it is reserved. That there are 31 days every month is not true, but it helps keep the format of the 12 text files consistent without imposing grave effects on the functionality of the program.

If the duration of the booking lasts for more than a month, the records in more than 1 text file. Take e.g.2 as an example, a client has booked Room 101 from 1st January. to 2nd February. Thus, it will be recorded in both rm2.txt and rm1.txt.

Sample file:

101	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
102	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
103	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
104	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
105	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
106	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
201	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
202	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
203	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
204	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
205	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
206	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
301	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
302	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
303	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
304	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
305	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
306	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
401	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
402	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
403	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
404	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
405	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
406	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
501	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
502	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
503	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
504	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
505	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
506	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
601	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
602	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
603	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
604	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
605	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
606	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
701	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
702	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
703	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
704	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
705	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
706	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
107	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Type of room

108	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
109	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
110	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
111	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
112	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
207	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
208	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
209	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
210	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
211	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
212	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
307	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
308	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
309	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
310	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
311	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
312	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
407	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
408	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
409	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
410	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
411	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
412	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
507	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
508	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
509	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
510	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
511	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
512	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
607	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
608	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
609	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
610	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
611	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
612	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
707	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
708	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
709	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
710	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
711	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
712	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
801	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
802	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
803	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
804	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
805	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
806	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
901	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
902	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
903	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
904	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
905	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
906	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1001	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1002	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1003	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1004	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1005	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1006	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1100	4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Day and Status

Room number

File 14: records_booked.txt

The text file is used to store the booking records of each client. Once the clients booked the rooms, or, once the clients cancelled the booking, the text file will immediately update the information so that the risk of blunders will reduce. The text file will be read at the very beginning automatically after the clients log in their accounts.

Information Catalogue	Data types
Username	12 strings
Number of booking	1 integer
Room	1 integer
Check-in month	1 integer
Check-in day	1 integer
Check-out month	1 integer
Check-out day	1 integer

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

e.g.1

Username	**Number of booking**	**Room**	**Check-in month**	**Check-in day**	**Check-out month**	**Check-out day**
Wakenial	0					

e.g.2

Username	**Number of booking**	**Room**	**Check-in month**	**Check-in day**	**Check-out month**	**Check-out day**
Hostel	1	3	2	10	2	10

e.g.3

User-name	Number of booking	Room	Check-in month	Check-in day	Check-out month	Check-out day	…	Room	Check-in month	Check-in day	Check-out month	Check-out day
Samuel	4	1	2	10	2	12		21	3	2	3	5

If the clients did not make any booking before, the ‘number of booking’ will be zero. Take e.g.1 as an example, the client, ‘Wakenial’, did not make any booking, thus the ‘number of booking’ is 0. Also there will not be the records of ‘Room’, ‘Check-in month’, ‘Check-in day’, ‘Check-out month’, ‘Check-out day’ written.

If the clients have made more than 1 booking, the records of each booking, will be written in the same line. Take e.g.3 as an example, the client, ‘Samuel’ have 4 booking. Thus, the information of his first booking, including ‘Room’, ‘Check-in month’, ‘Check-in day’, ‘Check-out month’, ‘Check-out day’, will be written first, followed by the second booking, so and so on.

For ‘Room’, the number represents which line the room number (recorded in **File 2 - File13**) is located. Take e.g.2 as an example, the ‘Room’ is 3, implying that Room 103 (located in 3rd line in **File 2 - File13**) is reserved for the client, ’Hostel’.

Sample file:

```
s10001      2 1 1 1 1 1 1 2 2 2 2
s10002      0
s10003      0
s10004      0
s10005      0
s10006      0
s10007      0
s10008      0
s10009      0
s10010      0
s10011      0
s10012      0
s10013      0
s10014      0
s10015      0
s10016      0
```

2.4 Output Report Format

Output results on screen

1. Personal information

As clients can change their personal information afterwards, their personal information will be shown for the amendments.

```
Username: s10001
Password: a12345678
Sex: M
Name: Chi Chung Cheung
Phone_no: 93399210
--------------------------------------------------------------------------------

Which one do you want to amend?(Please only enter the number)

1.Username

2.Password

3.Sex

4.Name

5.Phone number
--------------------------------------------------------------------------------

Enter your choice(0 to leave): _
```

2. Booking records

The booking records will be shown so that the clients will know the booking they have made and which to cancel.

Sample layout:

```
You have booked
--------------------------------------------------------------------------------

                1. Room    101
                   The check-in date:1/1
                   The check-out date:1/1

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

                2. Room    101
                   The check-in date:10/2
                   The check-out date:13/2

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

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

Which would you like to cancel (Enter the 0 for exit) ?
```

Output results on text files

File1: records.txt

New clients' information will be added in the last line of the text file after the registration has been done.

Sample layout:

Before client 's10006' has registered:

```
s10000      a12345678      M Cheung    Chi Chung            93399210
s10002      a12345678      M Cheung    Koo Ho               92656221
s10003      a12345678      F Cheung    Yee Hung, Teresa     91913232
s10004      a12345678      F Chui      Tse Lung             91170243
s10005      a12345678      F Chung     Kei Man, Mandy       90427254
```

After client 's10006' has registered:

```
s10000      a12345678      M Cheung    Chi Chung            93399210
s10002      a12345678      M Cheung    Koo Ho               92656221
s10003      a12345678      F Cheung    Yee Hung, Teresa     91913232
s10004      a12345678      F Chui      Tse Lung             91170243
s10005      a12345678      F Chung     Kei Man, Mandy       90427254
s10006      a12345678      F Fung      Sin Wan, Melanie     52065563
```

File 2 - File13: rm1.txt, rm2.txt, rm3.txt, … , rm12.txt

'x' will be added in the 'Day & Status' after the booking has been made.

Sample layout:

A client wants to book a single room from 1st February to 28th February. As the Room 101 is already reserved from 10th February to 13th February, Room 101 is not available. As a result, Room 102 is reserved for the client.

In rm2.txt,

Before the booking of client:

101	1	1	2	3	4	5	6	7	8	9	10x	11x	12x	13x	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
102	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
103	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
104	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
105	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
106	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
201	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

After the booking of client:

101	1	1	2	3	4	5	6	7	8	9	10x	11x	12x	13x	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
102	1	1x	2x	3x	4x	5x	6x	7x	8x	9x	10x	11x	12x	13x	14x	15x	16x	17x	18x	19x	20x	21x	22x	23x	24x	25x	26x	27x	28x	29	30	31
103	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
104	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
105	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
106	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
201	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

File 14: records_booked.txt

New clients' username will be added in the last line of the text file after the registration has been done. As this is a new account which does not have any records of booking, the 'number of booking' will be zero and there will not be any records of 'Room', 'Check-in month', 'Check-in day', 'Check-out month', 'Check-out day' written.

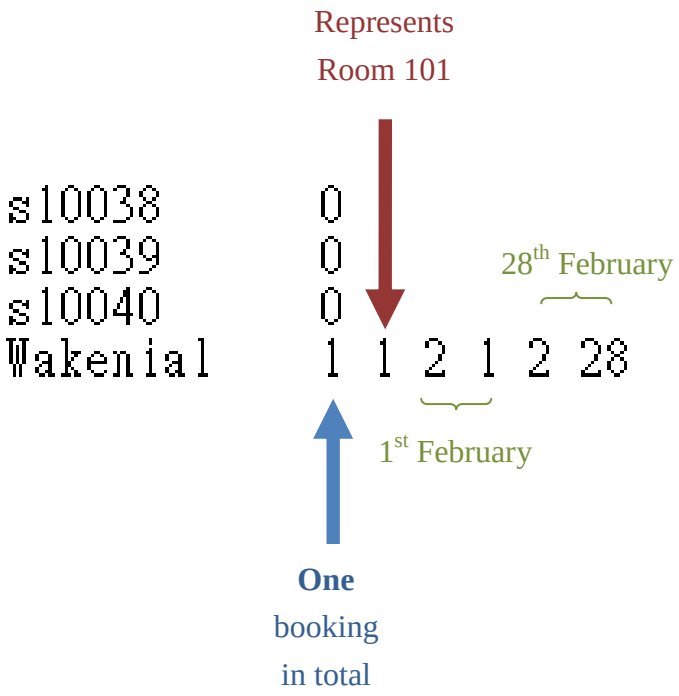
The records of the booking will be updated whenever the clients book rooms or cancel the booking.

Sample layout:

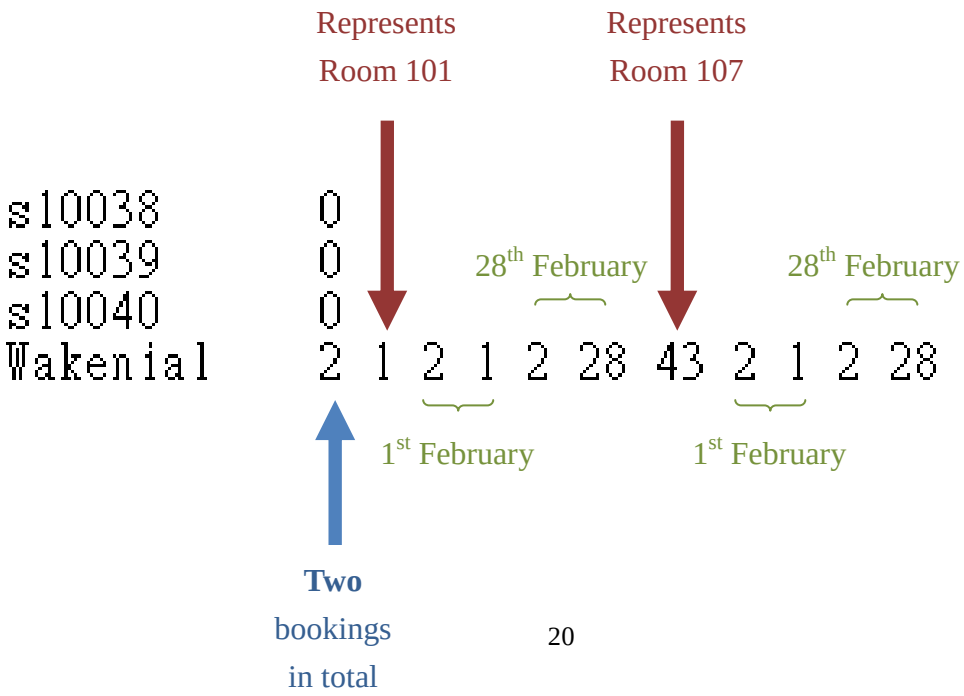
The client, ‘Wakenial’, has already a booking of Room 101 from 1st February to 28th February.

He now wants to have one more booking of a twin room from 1st February to 28th February. And Room 107 is available. Thus, the new record is written into the text file.

Before the booking of client:



After the booking of client:



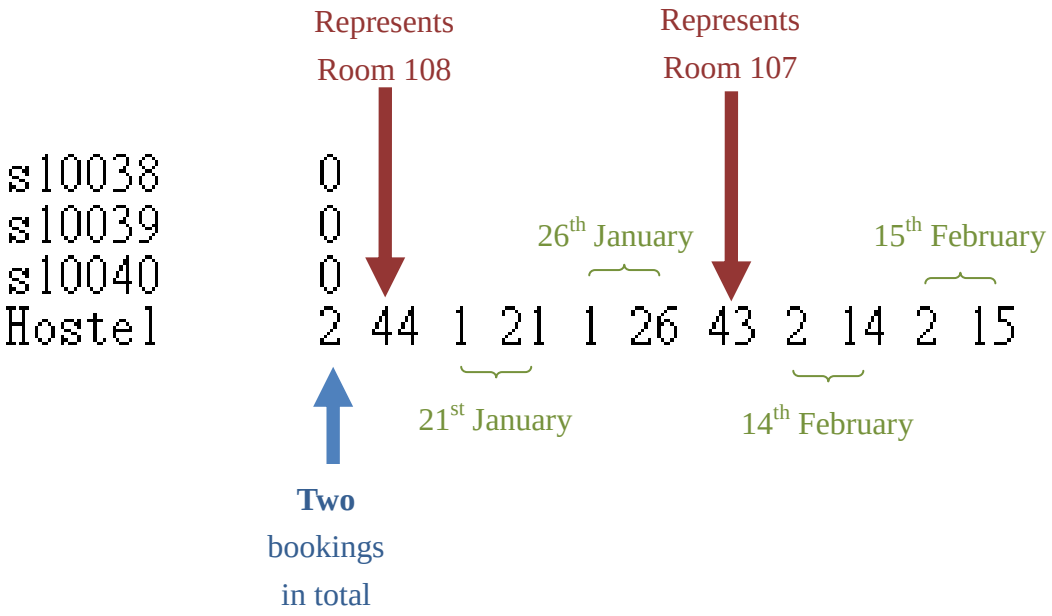
Vice versa, if a client cancels the booking, the records of the cancelled booking will be deleted.

Sample layout:

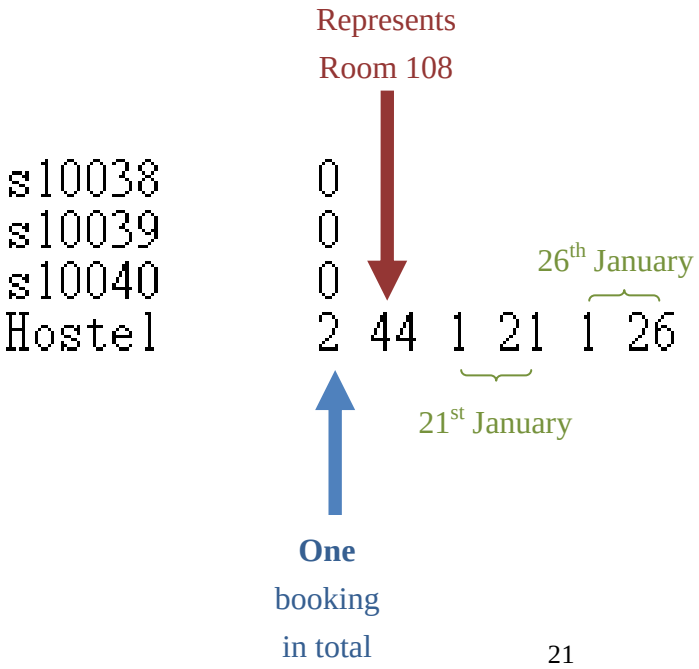
The client, 'Hostel', has two bookings. They are: a twin room, Room 108, from 21st January to 26th January and a twin room, Room 107, from 14th February to 15th February.

He now wants to cancel the booking of Room 107. Thus, the record of Room 107's booking will be deleted

Before the booking of client:



After the booking of client:



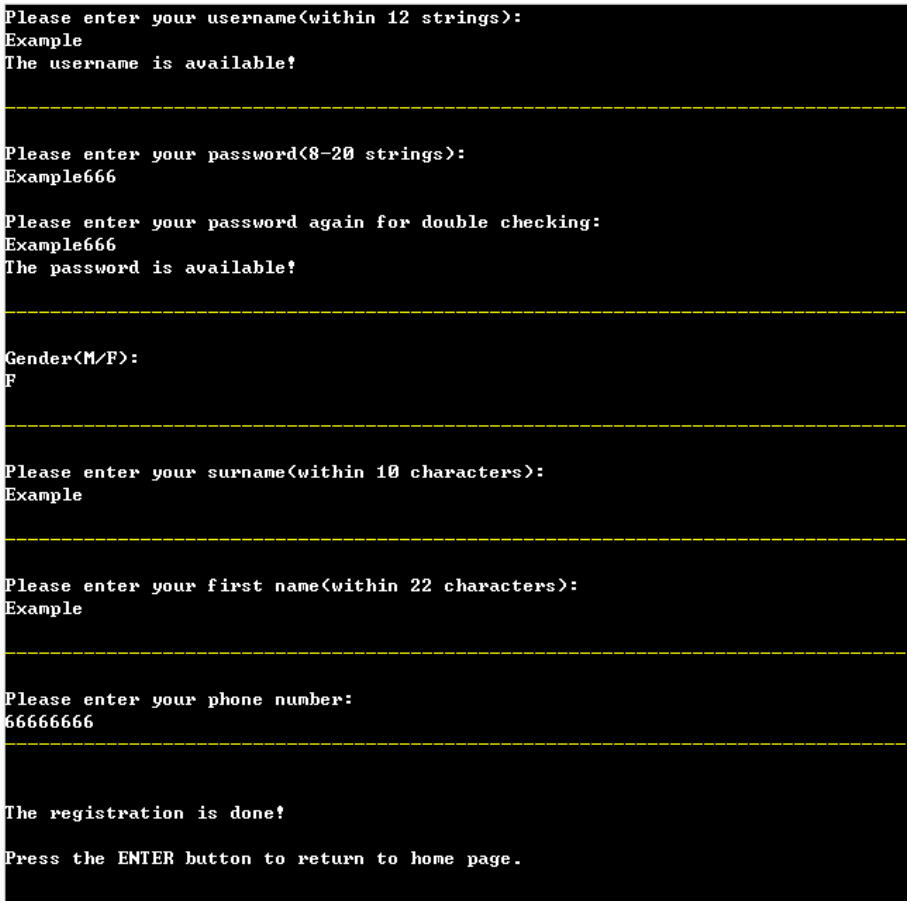
2.5 Functionality

Front page

Promote the hostel and show the icon of the hostel.

Register account

The clients are requested to have their personal account before they can make booking. Some personal information and basic registration is requested.



Input valid checking for registration

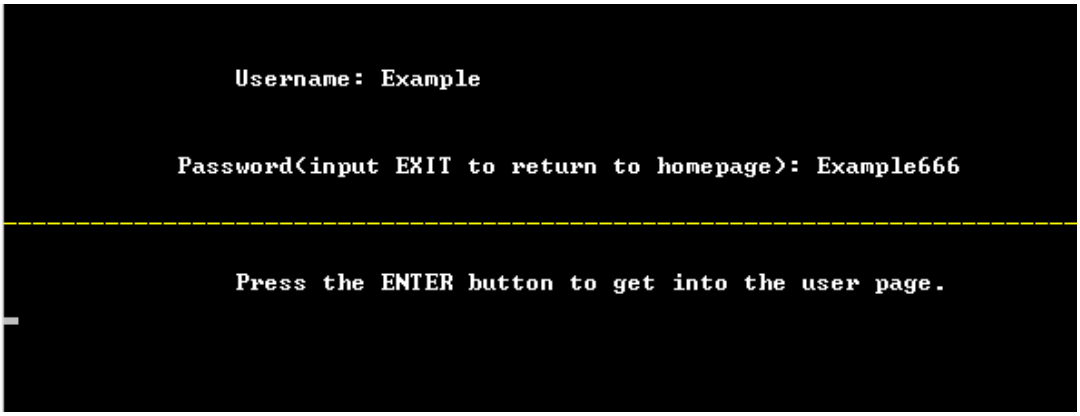
Multi-checking is made to ensure that there is no conflict occurs when the account is created. For instance:

- Field length: the length of username, password should be reasonable
- Fixed value check: the gender of the client should be either ‘M’ of ‘F’
- Type check: phone number should not contain anything else except numbers
- Filed presence check: the password should not be empty
- Check if the username is already registered
- Input data twice: clients are requested to input the password twice

All these helps minimize garbage-in-garbage-out situation.

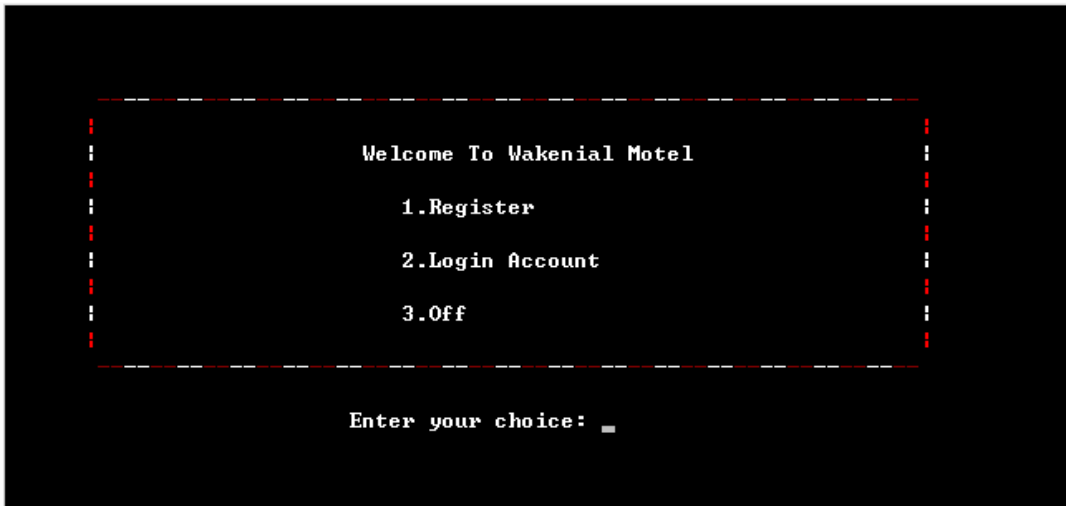
Login system

Having inputted the username and password, clients could login the system. Each of them should have an account for identifying and authenticating the clients.



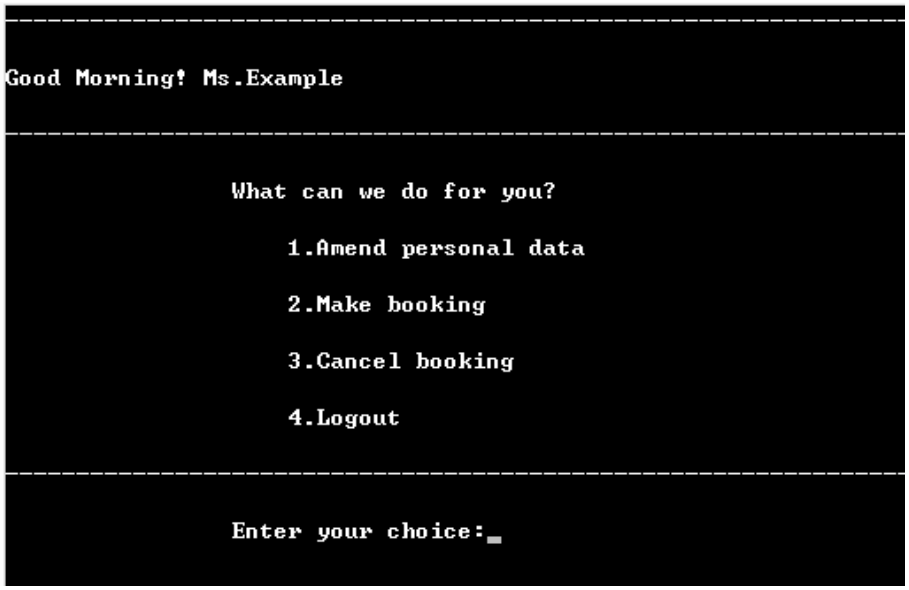
Main menu

If the clients have already had an account, they can choose ‘Login account’ directly, or they have to create an account through ‘Register’.



User’s page

After login the account, the clients can choose different functions by inputting the options.



Functions of user interface

Amend personal information

The clients can amend their personal information stored and the information will be updated immediately.

```
Username: Example
Password: Example666
Sex: F
Name: Example Example
Phone_no: 66666666
--------------------------------------------------------------------------------

Which one do you want to amend?(Please only enter the number)

1.Username

2.Password

3.Sex

4.Name

5.Phone number
--------------------------------------------------------------------------------

Enter your choice(0 to leave): _
```

Make booking

The clients can make booking with the restriction shown in the first line. The available room will be reserved for the clients and reminder will be shown in case of no vacancies. The booking records will be updated immediately.

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

The check-out date is up to Decemeber 2016

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

Check-in(month):1

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

Check-in(day):1

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

Check-out(month):1

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

Check-out(day):1

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

What kind of room(Single room(1)/Twin room(2)/Family room(3)/Suite Room(4),Pleas
e enter the number):1

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

Room    101      is reserved for you.
_
```

Cancel booking

The clients can cancel their bookings. The reminder will be shown in case of no booking records made.

```
You have booked
--------------------------------------------------------------------------------

                1. Room    101
                   The check-in date:1/1
                   The check-out date:1/1

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

                2. Room    101
                   The check-in date:1/2
                   The check-out date:3/2

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

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

Which would you like to cancel (Enter the 0 for exit) ?
```

```
You have not booked any rooms!
```

Chapter 3 Implementation

3.1 Brief Description and program coding

In this part, I am going to discuss how I implement the hostel booking system by Dev-pascal.

In the following part, I am going to showcase the data structures of my program. Besides, I would also describe the functions in this program by displaying the procedure coding in the program.

3.2 Data Structure

Records

I have made use of the characteristic of ‘record’ in pascal since it helps me hold a set of data which belongs to the same class.

This record is used to store the information of the registered clients from ‘records.txt’. Information of each client’s booking records from ‘records_booked.txt’ is also stored in this record.

```
{ usertype = record
                  SurName:string[10];
                  FirstName:string[22];
```

```
                Sex:string[1];
                Phone_no:string[8];
                Username:string[12];
                Password:string[20];
                Rooms:integer;
                Rooms_no:array[1..103] of integer;
                checkin_m:array[1..103] of integer;
                checkout_m:array[1..103] of integer;
                checkin_d:array[1..103] of integer;
                checkout_d:array[1..103] of integer;
            end;}
```

This record is used to store the information of the rooms from the record, ‘Droom_type’.

```
{ roomtype = record
                rm_no:array[1..103] of string;
                rm_cap:array[1..103] of string;
                date:array[1..103,1..31] of string[4];
            end;}
```

*‘Date’ is a two-dimensional array used to store 31-day status of 103 rooms.

This record is used to store all the information of each room from ‘rm1’to ‘rm12’, which is then used to split the information into the record ‘roomtype’.

```
{ Droom_type = record
                manyrms:array[1..103] of string;
            end;}
```

Global variables

Here are the global variables used in the program so that the processes of calling in or out variables and procedures will not affect the vital information.

Texts

The text variables are used to input data to the program or output data to the actual file. In other words, they represent the actual text files in the program.

Inf_client

This text variable is responsible for ‘records.txt’

Book

This text variable is responsible for ‘records_booked.txt’

Although the information of room is also store in text files, it is in the form of array as there are 12 similar text files representing information of room vacancy in different months. Thus, it is better to group them into an array.

Arrays

These store the records mentioned before into arrays. These arrays could be considered as multi-dimensional arrays.

```
{ Inf_room:array[1..12]of text;
  Client:array[1..10000] of usertype;
  Room:array [1..12]of roomtype;
  client_detail:array [1..10000]of string;
  Room_detail:array [1..12]of droom_type;}
```

*The array of ‘Inf_room’ represents that there are 12 files storing the information of 12 months’ room vacancy.

Integer

Integers usually are used for storing indexes.

```
{ choice,u_choice:integer;}
```

Words

Integers usually are used for storing indexes.

```
{ h,min,s,ms,y,m,d,td:word;}
```

*These words are all used to store the time of the date

3.3 Procedures in the Program

Booleans, characters and other types of variables can be found in the procedures.

According to the functions and design illustrated in Chapter 2, the following procedures will be constructed in the program:

*The instruction statements may only be the major parts of the procedures

Input information

Input clients information

These procedures are aimed to read all the information of clients from ‘records.txt’, the booking records of all clients, as well as the information of room status of the whole year.

These details will be used for the functions described in chapter 2, like login,

cancel booking, etc.…

Procedure of inputting clients' information:

```
{procedure Read_ClientRecord;
var count,Counting:integer;
begin
    assign(Inf_client,'records.txt');
    reset(Inf_client);
    count:=0;
    counting:=0;
    while not eof(inf_client) do
    begin
         count:=count+1;
         readln(Inf_client,client_detail[count]);
    end;
    close(Inf_client);
    counting:=count;
    for count :=1 to counting do
    with client[count] do
        begin
            username:=copy(client_detail[count],1,12);
            password:=copy(client_detail[count],13,32);
            sex:=copy(client_detail[count],33,33);
            SurName:=copy(client_detail[count],35,44);
            FirstName:=copy(client_detail[count],45,66);
            phone_no:=copy(client_detail[count],67,74);
        end;
end;}
```

Procedure of inputting booking records of clients:

```
{var useless:string[12];
   h,i:integer;
begin
    h:=0;
    assign(book,'records_booked.txt');
    reset(book);
    while not eof(book) do
    begin
        h:=h+1;
        read(book,useless,client[h].rooms);
        for i:=1 to client[h].rooms do
           with client[h] do

read(book,rooms_no[i],checkin_m[i],checkin_d[i],checkout_
m[i],checkout_d[i]);
        readln(book);
    end;
    close(book);
end;}
```

Procedure of inputting rooms' status:

```
{ var i,j,k,l:integer;
begin
    l:=1;
    for i:=1 to 12 do
      begin
          case i of

1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));
          10:assign(Inf_room[i],'rm10.txt');
          11:assign(Inf_room[i],'rm11.txt');
          12:assign(Inf_room[i],'rm12.txt');
          end;
          reset(Inf_room[i]);
          {readln;}
      for j := 1 to 103  do
          begin

readln(Inf_room[i],room_detail[i].manyrms[j]);
              {writeln(room_detail[i] [j]);}

room[i].rm_no[j]:=copy(room_detail[i].manyrms[j],1,9);

room[i].rm_cap[j]:=copy(room_detail[i].manyrms[j],10,7);
              l:=1;
              for k :=1 to 31 do
                 begin

room[i].date[j,k]:=copy(room_detail[i].manyrms[j],16+l,4)
;
                      {WRITELN(room[i].date[k]);}
                      l:=l+4;
                 end;
          end;
          close(Inf_room[i])
      end;
```

```
end;}
```

As to optimize the management of data in the program and in the text files, all usernames and passwords are limited in 12 strings and 20 strings. Also, days and status are restricted in 4 digits for the same purpose.

User interface

Main menu

To start with, the client has to create an account if he/she does not have one.

Procedure of main menu:

```
{begin
    clrscr;
    Read_ClientRecord;
    writeln(' ':6,'|                Welcome To Wakenial
Motel              |');
    writeln(' ':6,'|                   1.Register
|');
    writeln(' ':6,'|                   2.Login Account
|');
    writeln(' ':6,'|                   3.Off
|');
    write('                      Enter your choice: ');
    readln(choice);}
```

Procedure of register :

The clients are requested to input some personal information including username, password, sex, first name and surname and phone number. Several loops are used to ensure valid input of personal information.

Loop of username:

```
{ while ((length(R_username))>12) or
((length(R_username))<=0) or avail  do
         begin
             writeln('Please enter your username(within 12
strings):');
             readln(R_username);
             if ((length(R_username))>12) or
(length(R_username)<=0) then
             begin
                 writeln('Your username is not available.
Your username should neither be empty nor exceed 12
strings.');
             end
             else begin{}
                 for j:=1 to (12-length(R_username)) do
                    R_username:=R_username+' ';
                avail:=false;
                 for i := 1 to 100 do
                    if R_username=client[i].username
                      then avail:=true;
                 if avail=true then
                   begin
                       writeln('The username is already
occupied, please enter another username.');
```

```
                        writeln;
                        end else
                        writeln('The username is
available!');
               end
         end;}
```

Loop of password:

```
{while (length(R_password)>20) or (length(R_password)<8)or
dbcheck do
          begin
               writeln('Please enter your password(8-20
strings):');
               readln(R_password);
               if (length(R_password)>20) or
(length(R_password)<8) then
                 begin
                      writeln('Your password is not available.
Your password should obtain at least 8 strings and within 20
strings.');
                      writeln;
                 end
                 else begin
                      writeln('Please enter your password
again for double checking:');
                      readln(R_password1);
                      if R_password1<>R_password
                        then begin
                             writeln('Not consistent! Please
enter again!');
                             end
                        else     dbcheck:=false;
```

```
                        end;
          end;
     for j:=1 to (20-length(R_password)) do
     R_password:=R_password+' ';
     writeln('The password is available!');
```

Loop of sex:

```
{while (R_sex<>'M') and (R_sex<>'F') do
          begin
               writeln('Gender(M/F):');
               readln(R_sex);
               If (R_sex<>'M') and (R_sex<>'F')
                  then     begin
                              writeln('Not available!
Please enter again!');
                              writeln();
                          end;
          end;
     R_sex:=R_sex+' ';
     writeln();}
```

Loop of surname:

```
{while (length(R_SurName)>10) or (length(R_SurName)<=0) do
          begin
               writeln('Please enter your surname(within 10
characters):');
               readln(R_SurName);
               if (length(R_SurName)>10) or
(length(R_SurName)<=0) then
               begin
                    writeln('Your surname is not available.
Your surname should neither be empty nor exceed 10
```

```
characters.');
```

```
                end;
            end;
      for j:=1 to (10-length(R_surname)) do
         R_surname:=R_surname+' ';
      writeln();}
```

Loop of first name:

```
{while (length(R_FirstName)>22) or (length(R_FirstName)<=0)
do
            begin
                writeln('Please enter your first name(within
22 characters):');
                readln(R_FirstName);
                if (length(R_FirstName)>22) or
(length(R_FirstName)<=0) then
                begin
                    writeln('Your first name is not available.
Your first name should neither be empty nor exceed 22
characters.');
                end;
            end;
      for j:=1 to (22-length(R_FirstName)) do
         R_FirstName:=R_FirstName+' ';
      writeln();}
```

Loop of phone number:

```
{while (length(R_Phone_no)<>8) or (real=0) do
            begin
                writeln('Please enter your phone number:');
```

```
                readln(R_Phone_no);
                val(R_Phone_no,real,code);
                if real=0
                   then begin
                            writeln('This is not a phone
number!Please enter again!');
                       end
                   else if (length(R_Phone_no)<>8) then
                       begin
                           writeln('Your phone number is not
available. Your phone number must be in 8 numbers');
                       end;
           end;
```

After the registration has been done, the information of new established account and client's information will be updated to the text file.

Output the information of client:

```
{assign(Inf_client,'records.txt');
     append(Inf_client);

write(Inf_client,R_username,R_Password,R_Sex,R_SurName,R_
FirstName,R_Phone_no);
     writeln(Inf_client);
     close(Inf_client);
     assign(f,'records_booked.txt');
     append(f);
     write(f,R_username,'0');
     writeln(f);
     close(f);
     writeln('The registration is done!');
     writeln('Press the ENTER button to return to home
```

```
page.');
     readln();
     clrscr;
end;}
```

Procedure of login:

The clients have to login before have further functions like booking a room. After the identification and authentication, it will turn to user's page.

```
{begin
while (corr=false) and (exit1=false) do
         begin
             clrscr;
             write('Username: ':26);
             readln(log_id);
             write('Password(input EXIT to return to
homepage): ':56);
             readln(log_pd);
             if log_pd='EXIT'
                then
                   begin
                       clrscr;
                       exit1:=true;
                   end
                else begin
                        for j:=1 to (12-length(log_id)) do
                                   log_id:=log_id+' ';
                        for j:=1 to (20-length(log_pd)) do
                                   log_pd:=log_pd+' ';
                        for i:=1 to 10000 do
                           if log_id=client[i].username
                              then if
```

```
log_pd=client[i].password
                                    then begin
                                        corr:= true;
                                        keyword:=i;
                                        end;
                        if corr=false
                           then begin
                               writeln('
Invalid username or password!');
                               end;
                    end;
         END;
     if exit1=false then begin
  Press the ENTER button to get into the user page.');
     readln();
     clrscr;
     user_page(keyword);      end;
end;}
```

User's page

There are some basic functions of the booking system and account management.

Procedure of user's page:

```
{ begin
    Read_ClientRecord;
    Read_RoomRecord;
    Read_Booked;
```

```
    if (h<12) then mae:='Good Morning!';
    if (h<18) and (h>=12) then mae:='Good Afternoon!';
    if (h>=18) then mae:='Good Evening!';
    if client[users].sex='M'
       then writeln(mae,' Mr.',client[users].surname)
       else writeln(mae,' Ms.',client[users].surname);
    writeln('':16,'What can we do for you?');
    writeln('':20,'1.Amend personal data');
    writeln('':20,'2.Make booking');
    writeln('':20,'3.Cancel booking');
    writeln('':20,'4.Logout');
    write('':16,'Enter your choice:');
    readln(u_choice);
    case u_choice of
        1:amend_UserRecord(users);
        2:make_booking(users);
        3:cancel_booking(users);
    end;
    if u_choice<>4 then begin clrscr;user_page(users);end;}
```

There are total three functions, including amend clients' personal information, make booking and cancel booking.

Procedure of amending personal information:

It is simply normal account management, clients can make amendments on personal information so that the manager of the hostel can keep in contact with clients.

First and foremost, clients should have a preview of the original information.

Original information:

```
{begin    clrscr;
    Read_ClientRecord;
    with client[K_word] do
        begin
            R_username:=username;
            R_Password:=Password;
            R_Sex:=sex;
            R_SurName:=surname;
            R_FirstName:=firstname;
            R_Phone_no:=Phone_no;
            writeln('Username: ',username);
            writeln('Password: ',Password);
            writeln('Sex: ',sex);
            S_name:=firstname;
            repeat
                 if copy(S_name,length(S_name),1)=' '
                    then
S_name:=copy(S_name,1,length(S_name)-1);
            until copy(S_name,length(S_name),1)<>' ';
            writeln('Name: ',S_name,' ',surname);
            writeln('Phone_no: ',Phone_no);
        end;}
```

Choose one item:

Then, clients can choose which of the personal information to change.

```
{ writeln('Which one do you want to amend?(Please only enter
the number)');
    writeln('1.Username');
    writeln('2.Password');
```

```
    writeln('3.Sex');
    writeln('4.Name');
    writeln('5.Phone number');
    write('Enter your choice(0 to leave): ');
    readln(z);}
```

Similar to the procedure of registration, there are several loops for data validation.

Loop of username:

```
{ If z =1
then begin
while ((length(R_username))>12) or ((length(R_username))<=0)
or avail  do
begin
writeln('Please enter your username(within 12 strings):');
readln(R_username);
if ((length(R_username))>12) or (length(R_username)<=0) then
begin
writeln('Your username is not available. Your username should
neither be empty nor exceed 12 strings.');
end
else
begin
for j:=1 to (12-length(R_username)) do
R_username:=R_username+' ';
{writeln(length(copy(R_username,1,12))); can test   fill in
the blanks}
avail:=false;
for i := 1 to 100 do
if R_username=client[i].username
then avail:=true;
if avail=true then
begin
```

```
writeln('The username is already occupied, please enter
another username.');
end else
begin
writeln('The username is available!');
end
end
end;
avail:=true;
end;}
```

Loop of password:

```
{ If z =2
then begin
while (length(R_password)>20) or (length(R_password)<8)or
vail do
begin
writeln('Please enter your password(8-20 strings):');
readln(R_password);
if (length(R_password)>20) or (length(R_password)<8) then
begin
writeln('Your password is not available. Your password should
obtain at least 8 strings and within 20 strings.');
writeln;
end
else begin
writeln('Please enter your password again for double
checking:');
readln(R_password1);
if R_password1<>R_password
then begin
writeln('Not consistent! Please enter again!');
```

```
end
else     dbcheck:=false;
end;
avail:=false;
end;
for j:=1 to (20-length(R_password)) do
R_password:=R_password+' ';
writeln('The password is available!');
avail:=true;
end;}
```

Loop of sex:

```
{ If z =3
then begin
while (R_sex<>'M') and (R_sex<>'F') or avail do
begin
writeln('Gender(M/F):');
readln(R_sex);
If (R_sex<>'M') and (R_sex<>'F')
then     begin
writeln('Not available! Please enter again!');
end;
avail:=false;
end;
R_sex:=R_sex+' ';
avail:=true;
end;}
```

Loop of name:

```
{ If z =4
then begin
```

```
while (length(R_SurName)>10) or (length(R_SurName)<=0) or
avail do
begin
writeln('Please enter your surname(within 10 characters):');
readln(R_SurName);
if (length(R_SurName)>10) or (length(R_SurName)<=0) then
begin
writeln('Your surname is not available. Your surname should
neither be empty nor exceed 10 characters.');
end;
avail:=false;
end;
avail:=true;
for j:=1 to (10-length(R_surname)) do
R_surname:=R_surname+' ';
while (length(R_FirstName)>22) or (length(R_FirstName)<=0)
or avail do
begin
writeln('Please enter your first name(within 22
characters):');
readln(R_FirstName);
if (length(R_FirstName)>22) or (length(R_FirstName)<=0) then
begin
writeln('Your first name is not available. Your first name
should neither be empty nor exceed 22 characters.');
end;
avail:=false;
end;
avail:=true;
for j:=1 to (22-length(R_FirstName)) do
R_FirstName:=R_FirstName+' ';
end;}
```

Loop of phone number:

```
{ If z =5
then begin
while (length(R_Phone_no)<>8) or (real=0) do
begin
writeln('Please enter your phone number:');
readln(R_Phone_no);
val(R_Phone_no,real,code);
if real=0
then begin
writeln('This is not a phone number!Please enter again!');
end
else if (length(R_Phone_no)<>8) then
begin
writeln('Your phone number is not available. Your phone number
must be in 8 numbers');
end;
end;
end;}
```

More amendments:

The clients can choose to amend more than one item of personal information afterwards.

```
{ writeln('Anymore?(Y/N)');
    readln(more);
    if more='N' then anymore:=true;
    writeln();
    until anymore;}
```

Update the amendment:

```
{assign(Inf_client,'records.txt');
rewrite(Inf_client);
for haha:=1 to (K_word-1) do
writeln(Inf_client,client_detail[haha]);
writeln(Inf_client,R_username,R_Password,R_Sex,'
',R_SurName,R_FirstName,R_Phone_no);
for haha:=k_word+1 to 10000 do
begin
if client_detail[haha]<>''
then write(Inf_client,client_detail[haha]);
if client_detail[haha+1]<>''
then writeln(Inf_client);
end;
close(Inf_client);
end;}
```

Procedure of making booking:

Restrictions:

As mentioned in Chapter 2, it is assumed that the availability of the booking is within 1 year. Thus, it should be clearly mentioned in the layout.

```
{ write('The check-out date is up to ');
    case m of
    1:write('Decemeber ');
    2:write('January ');
    3:write('February ');
    4:write('March ');
```

```
    5:write('April ');
    6:write('May ');
    7:write('June ');
    8:write('July ');
    9:write('August ');
    10:write('September ');
    11:write('October ');
    12:write('November ');end;
    if m <>1
      then    writeln(y+1)
      else    writeln(y);}
```

The booking will be looped if the providing information is invalid.

Loop of check-in date:

```
{ repeat
         write('Check-in(month):');
         readln(month_i);
         If (month_i>12) or (month_i<1)
           then begin writeln('Not available! Please enter
again!'); writeln; end
           else really:=true;
    until really;
    really:=false;
repeat
         write('Check-in(day):');
         readln(day_i);
         case (month_i) of
         1,3,5,7,8,10,12:  if (day_i>=1) and (day_i<=31)
                             then really:=true;
         4,6,9,11:        if (day_i>=1) and (day_i<=30)
```

```
                                then really:=true;
          2:              begin
                              day1:=28;
                              if (y mod 4)= 0  then day1:=29;
                              if (day_i>=1) and (day_i<=day1)
                                then really:=true;
                          end;
          end;
          If not really
            then writeln('Not available! Please enter
again!');
    until really;
    really:=false;}
```

Loop of check-out date:

```
{ repeat
          write('Check-out(month):');
          readln(month_o);
          If (month_o>12) or (month_o<1)
            then writeln('Not available! Please enter
again!')
            else really:=true;
    until really;
    really:=false;
repeat
          write('Check-out(day):');
          readln(day_o);
          case (month_o) of
          1,3,5,7,8,10,12:  if (day_o>=1) and (day_o<=31)
                                then really:=true;
          4,6,9,11:        if (day_o>=1) and (day_o<=30)
                                then really:=true;
```

```
        2:              begin
                            day1:=28;
                            if m>month_o
                               then y:=y+1;
                            if (y mod 4)= 0  then day1:=29;
                            if (day_o>=1) and (day_o<=day1)
                               then really:=true;
                            y:=y-1;
                        end;
        end;
        If not really
           then writeln('Not available! Please enter
again!');
    until really;
    really:=false;}
```

Loop of room type:

```
{ repeat
        write('What kind of room(Single room(1)/Twin
room(2)/Family room(3)/Suite Room(4),Please enter the
number):');
        readln(cap);
        If (cap=1) or (cap=2) or (cap=3) or (cap=4)
           then really:=true;
    until really;
    really:=false;}
```

Update the text files:

Both the booking records and the room status will be updated.

```
{ assign(book,'records_booked.txt');
                    rewrite(book);
```

```
                    for i := 1 to (who-1) do
                    begin

write(book,client[i].username,client[i].rooms);
                        for j:= 1 to client[i].rooms do
                           with client[i] do
                               write(book,'
',rooms_no[j],' ',checkin_m[j],' ',checkin_m[j],'
',checkout_m[j],' ',checkout_m[j]);
                               writeln(book);
                    end;


client[who].rooms:=client[who].rooms+1;

write(book,client[who].username,client[who].rooms);
                    for i:=1 to(client[who].rooms-1) do
                       with client[who] do
                       write(book,' ',rooms_no[i],'
',checkin_m[i],' ',checkin_d[i],' ',checkout_m[i],'
',checkout_d[i]);
                    write(book,' ',final,' ',month_i,'
',day_i,' ',month_o,' ',day_o);
                    writeln(book);
                    for i := (who+1) to 10000 do
                       if client[i].username<>''  then
                           begin

write(book,client[i].username,client[i].rooms);
                               for j:= 1 to client[i].rooms
do
                               with client[i] do
```

```
write(book,rooms_no[j],checkin_m[j],checkin_m[j],checkout
_m[j],checkout_m[j]);
                                        writeln(book);
                                end;
                        close(book);
                    end;
        readln;
end;}
```

Procedure of cancelling booking:

Read and display records:

The booking records should be input to determine whether there are any bookings and what the bookings are as to display them to the clients.

```
{ begin
    clrscr;
    If client[who].rooms=0
    then
        writeln('':25,'You have not booked any rooms!');
    If client[who].rooms<>0
    then
       begin
           writeln('You have booked ');
           for i:=1 to client[who].rooms do
              begin
                  writeln(' ':15,i,'. ','Room
',room[1].rm_no[client[who].rooms_no[i]]);
                  writeln(' ':18,'The check-in
date:',client[who].checkin_d[i],'/',client[who].checkin_m
```

```
[i]);
                    writeln(' ':18,'The check-out
date:',client[who].checkout_d[i],'/',client[who].checkout
_m[i]);
                end;}
```

Choose one booking:

```
{ writeln('Which would you like to cancel (Enter the 0 for
exit) ?');
            readln(option);
            if (option>i) or (option<0)
               then writeln('Not available! Please enter
again!');
            until (option<=i) and (option>=0);}
```

Update the text files:

Both the booking records and the room status will be updated.

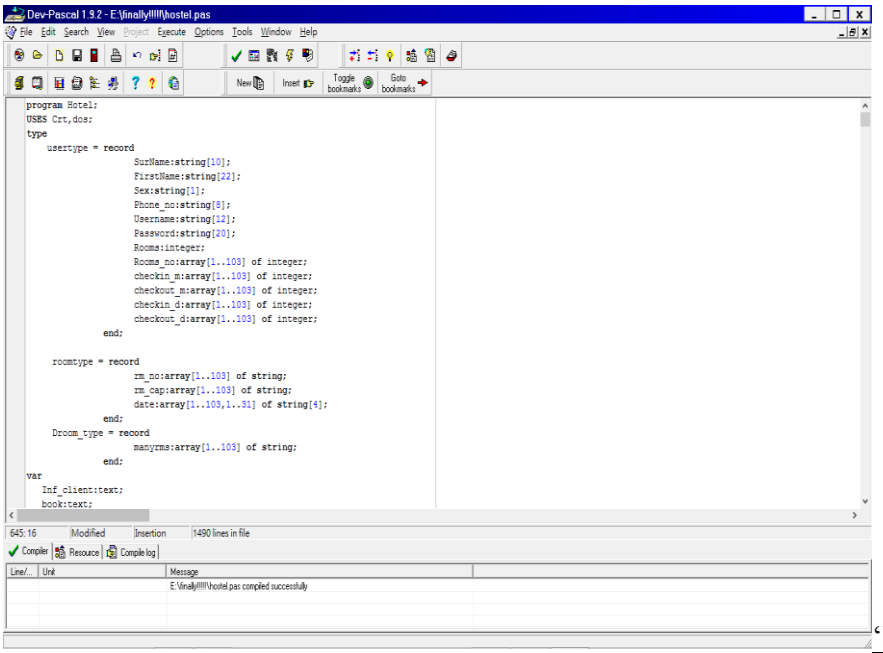
```
{ assign(book,'records_booked.txt');
                    rewrite(book);
                    for i := 1 to (who-1) do
                    begin
write(book,client[i].username,client[i].rooms);
                        for j:= 1 to client[i].rooms do
                            with client[i] do
write(book,' ',rooms_no[j],' ',checkin_m[j],'
',checkin_m[j],' ',checkout_m[j],' ',checkout_m[j]);
writeln(book);
                    end;
client[who].rooms:=client[who].rooms-1;
write(book,client[who].username,client[who].rooms);
                    for i:=1 to(option-1) do
                        with client[who] do
```

```
write(book,' ',rooms_no[i],' ',checkin_m[i],'
',checkin_d[i],' ',checkout_m[i],' ',checkout_d[i]);
                 for i:=option+1 to(client[who].rooms+1) do
                        with client[who] do
write(book,' ',rooms_no[i],' ',checkin_m[i],'
',checkin_d[i],' ',checkout_m[i],' ',checkout_d[i]);
                    writeln(book);
                    for i := (who+1) to 10000 do
                       if client[i].username<>'' then
                           begin
write(book,client[i].username,client[i].rooms);
                            for j:= 1 to client[i].rooms do
                               with client[i] do
write(book,rooms_no[j],checkin_m[j],checkin_d[j],checkout
_m[j],checkout_d[j]);
                                     writeln(book);
                           end;
                    close(book);
                end;
   readln;
   end;
end;}
```

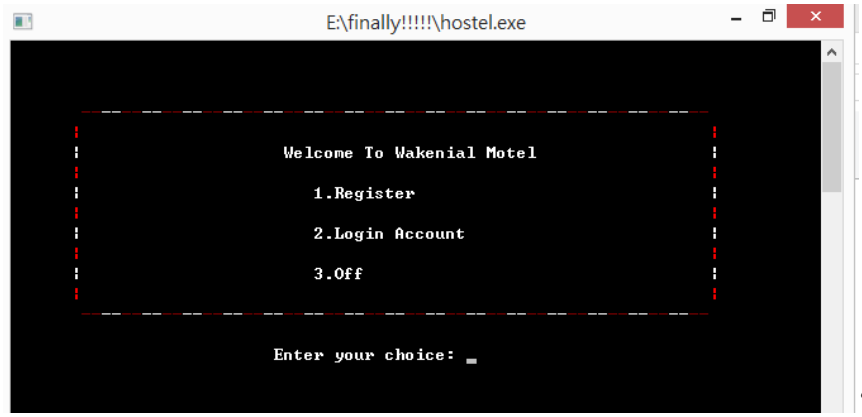
3.4 Program Coding

As mentioned in Chapter 3.1, I choose Dev-pascal to develop the program. It's because I am rather experienced in using it. It is believed that it helps me complete the work efficiently. In addition, Dev-Pascal offers a compiler to convert the source program into object program automatically. It saves a lot of time from coding. I have used dos. and crt., two units of dev-pascal to help make the interface look more user-friendly.

The filename of the source program is 'hostel.pas' while the filename of the object program is 'hostel.exe'.



'hostel.pas'



'hostel.exe'

Chapter 4 Testing & Evaluation

4.1 Brief Description

Generally speaking, there might be three types of errors occur in the program, including syntax, run-time and logic errors.

1. Syntax error:

When developing the program, syntax errors always are present. Transposition errors, very likely the typo, are a kind of syntax error in Pascal. Insertion error may also occur if the inappropriate strings are used. For instance, it may happen when I turn to find some sources in Chinese and hence type Chinese in the program. Moreover, the complier in Pascal would interrupt the translating of statements due to incomplete content. However, it is quite easy to debug the above syntax errors. While it is unable to run due to syntax errors, the user-friendly debugger in Dev Pascal would locate the errors. As a result, all these errors in the program could be corrected step-by-step manually with the help of Dev-pascal's debugger.

2. Run-time error:

Indeed, this kind of errors requires a rather long period of time for debugging as Pascal does not give any warning if there are any run-time errors. Only when the malfunction occurs will I know that there are some run-time errors.

Actually, run-time errors occur usually because of the incorrect route of the file, and input a data which does not match with the type of the variable.

Furthermore, after assigning a file, a run-time error may occur when it is forgotten to close the file after reading or rewriting it.

As a result, run-time errors occur frequently. Little carelessness can lead to

Discover the cause of problem would somehow slow the progress of the program.

3. Logic error:

It should be the most difficult part in debugging. Neither Pascal nor the program will show warnings when logic errors occur. It requires a clear mindset to discover and plug the loopholes.

I always find out the errors by splitting and separating the procedures in different parts and run these parts one by one.

4.2 Testing and Evaluation Plan

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

Internal testing and evaluation / Tested and evaluated by me (the programmer):

- The program will be tested intensively by me – the programmer
- I will prepare different test cases to test the program thoroughly
- The test cases include some correct input data (from files & keyboard) with known results for checking the correctness of the program, some incorrect input data to see whether the program can handle invalid input reasonably, etc.
- I will also evaluate the programs according to its user-friendless, performance, flexibility for future development, reusability of program codes, etc.

Tested and evaluated by users:

I will invite some targeted users/my classmates/friends to test and evaluate the program.

I will give the object program (the .exe file) and some sample data files onto a share folder to the friend who asked me to develop. Thus, I can know whether the program fulfil his requirements. Also, I will upload the corresponding files to social network like Facebook and ask my friends to try. Then, the users are invited to report the bugs they find and give their comments and suggestions on my program

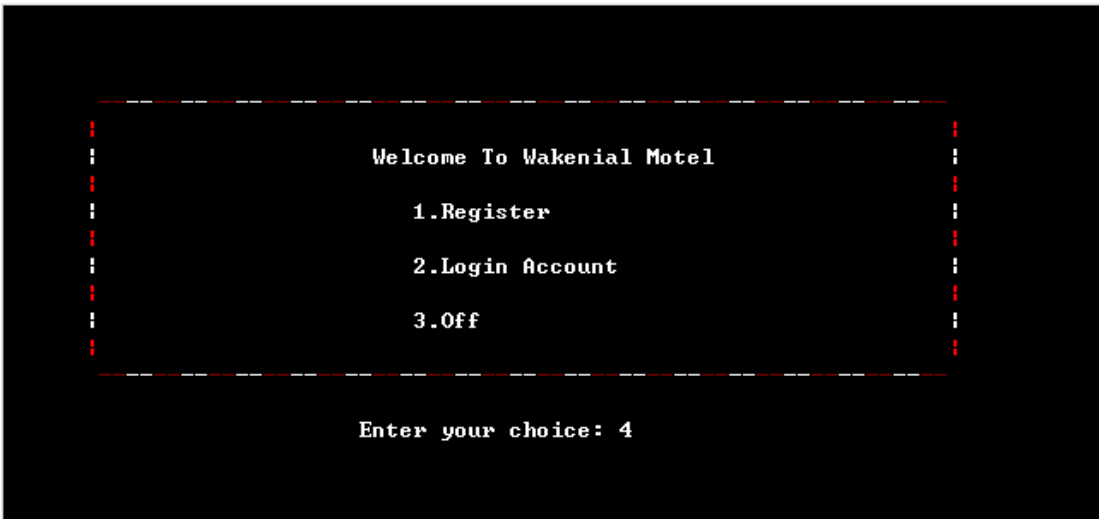
I will try to modify the program according to the reported bugs and suggestions. The new version of program after making modifications will be shown in Appendix 1.

4.3 Internal Testing

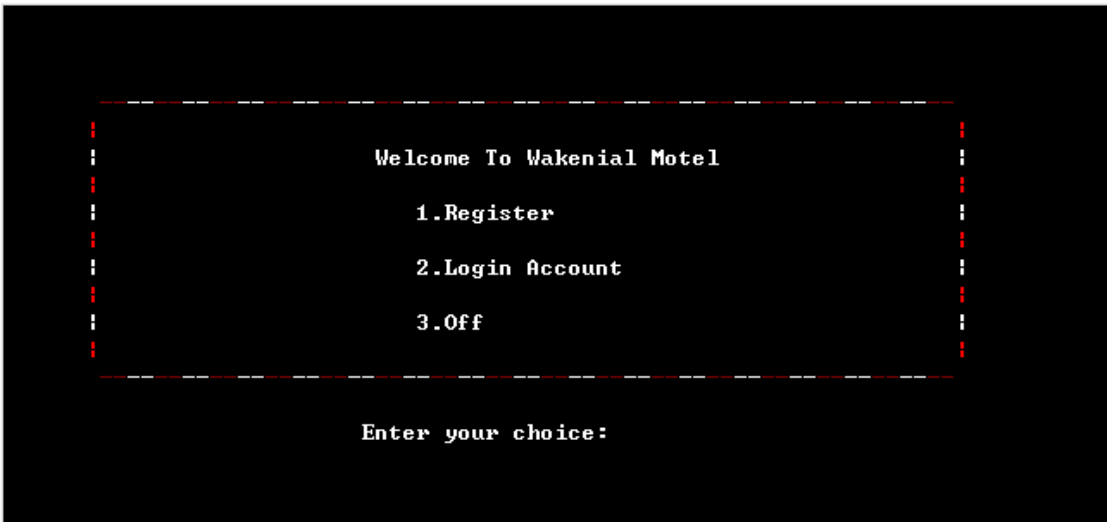
Test case of main menu

Test case 1

Purpose:	To test if it will refresh the page if the invalid option is input
Input:	Invalid option
Expected Output:	Refresh the page
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

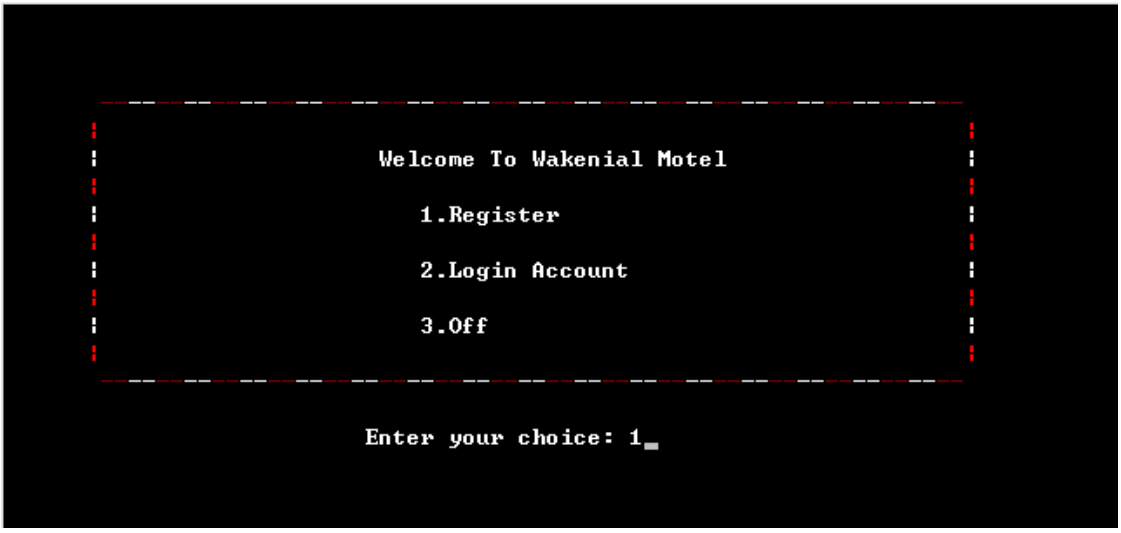


then refreshes

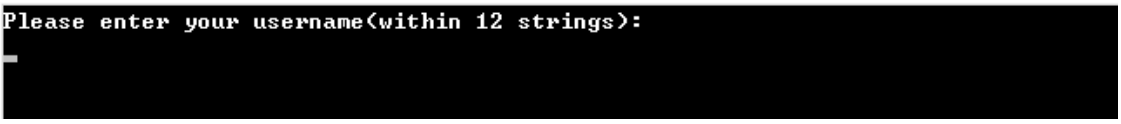


Test case 2

Purpose:	To test if it goes to corresponding pages with the valid input
Input:	Valid option
Expected Output:	Goes to corresponding page
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



then goes to the stage of registration



```
Welcome To Wakenial Motel

1.Register

2.Login Account

3.Off

Enter your choice: 2
```

then goes to the stage of login

```
Welcome To Wakenial Motel

1.Register

2.Login Account

3.Off

Enter your choice: 3
```

then the whole program ends

Test case of registration

Test case 1

Purpose:	To test if the invalid information can be input
Input:	Invalid information which does not follow the instructions given
Expected Output:	Warning will be given and the client should enter once again
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

```
Please enter your username(within 12 strings):
1234567890123
Your username is not available. Your username should neither be empty nor exceed
 12 strings.

Please enter your username(within 12 strings):

Your username is not available. Your username should neither be empty nor exceed
 12 strings.

Please enter your username(within 12 strings):
123456
The username is available!

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

*The length of username is longer than 12 in the1st trial.

*Username is empty in the 2nd trial.

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

Please enter your password(8-20 strings):
123456
Your password is not available. Your password should obtain at least 8 strings a
nd within 20 strings.

Please enter your password(8-20 strings):
12345678901234567890 1
Your password is not available. Your password should obtain at least 8 strings a
nd within 20 strings.

Please enter your password(8-20 strings):
```

*The length of password is shorter than 8 in the 1st trial.

*The length of password is longer than 20 in the 2nd trial.

```
Please enter your password(8-20 strings):
12345678

Please enter your password again for double checking:
1234568
Not consistent! Please enter again!

Please enter your password(8-20 strings):
12345678

Please enter your password again for double checking:
12345678
The password is available!

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

Gender(M/F):
```

*The passwords are not consistent in the 1st trial.

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

Gender(M/F):
A
Not available! Please enter again!

Gender(M/F):
F

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

Please enter your surname(within 10 characters):
```

*The gender is not valid in the 1st trial.

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

Please enter your phone number:
abcdefgh
This is not a phone number!Please enter again!

Please enter your phone number:
1234567
Your phone number is not available. Your phone number must be in 8 numbers

Please enter your phone number:
```

*The phone number contains characters in the 1st trial.

*The length of the phone number is too short in the 2nd trial.

Test case 2

Purpose:	To test if the program can detect the existed accounts
Input:	Existed username
Expected Output:	Warning will be given and the client should enter once again
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

In ‘records.txt’, there is a record of client with the username, ‘Wakenial’.

```
Please enter your username(within 12 strings):
Wakenial
The username is already occupied, please enter another username.


Please enter your username(within 12 strings):
```

Test case of login

Test case 1

Purpose:	To check if the program can detect invalid input
Input:	Invalid username and password
Expected Output:	Warning will be given and the client should enter once again
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

In ‘records.txt’, there is a record of client with the username, ‘Wakenial’ and the password is ‘hostel666’. There isn’t any record of client with username, ‘Hostel_Wak’.

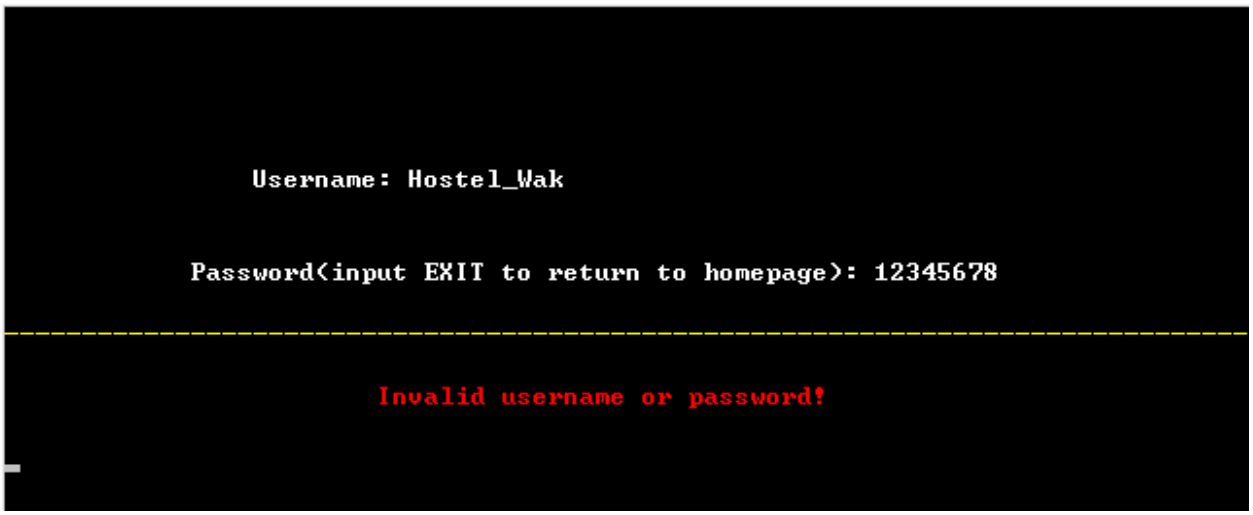
```
Username: Wakenial

Password(input EXIT to return to homepage): hahawrong
--------------------------------------------------------------------------------
                    Invalid username or password!
```

```
Username: Wakenial

Password(input EXIT to return to homepage): Hostel666
--------------------------------------------------------------------------------
                    Invalid username or password!
```

*Invalid password



*Invalid username

Test case 2

Purpose:	To check if the program can detect valid input and go to user page appropriately
Input:	Valid username and password
Expected Output:	Reminder of going to user page and then move to the stage of user page
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

In ‘records.txt’, there is a record of client, Ms. Lee, with the username, ‘Wakenial’ and the password is ‘hostel666’.

```
Username: Wakenial

Password(input EXIT to return to homepage): hostel666
--------------------------------------------------------------------------------
Press the ENTER button to get into the user page.
```

then goes to user page

```
--------------------------------------------------------------------------------
Good Morning! Ms.Lee
--------------------------------------------------------------------------------
What can we do for you?

    1.Amend personal data

    2.Make booking

    3.Cancel booking

    4.Logout
--------------------------------------------------------------------------------
Enter your choice:
```

Test case of user page

Test case 1

Purpose:	To test if it will refresh the page if the invalid option is input
Input:	Invalid option
Expected Output:	Refresh the page
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 test if it goes to corresponding pages with the valid input
Input:	Valid option
Expected Output:	Goes to corresponding page
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

As the results and the rationale are similar to the test cases of main menu, the screenshots can be skipped.

Test case of clients' information's amendments

Test case 1

Purpose:	To check if the program can amend clients' personal information
Input:	Corresponding option and corresponding data
Expected Output:	The related text file should be changed.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

Record of client, 'Wakenial':

Username	Password	Sex	Surname	First name	Telephone number
Wakenial	hostel66	F	Wak	Lee	12345678

```
Username: Wakenial
Password: hostel666
Sex: F
Name: Wak Lee
Phone_no: 12345678
--------------------------------------------------------------------------------

Which one do you want to amend?(Please only enter the number)

1.Username

2.Password

3.Sex

4.Name

5.Phone number
--------------------------------------------------------------------------------

Enter your choice(0 to leave):
```

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

Enter your choice(0 to leave): 1
Please enter your username(within 12 strings):
Wakenial_1
The username is available!
--------------------------------------------------------------------------
```

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

Enter your choice(0 to leave): 2
Please enter your password(8-20 strings):
Hostel666
Please enter your password again for double checking:
Hostel666
The password is available!
--------------------------------------------------------------------------
```

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

Enter your choice(0 to leave): 3
Gender(M/F):
M
--------------------------------------------------------------------------
```

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

Enter your choice(0 to leave): 4
Please enter your surname(within 10 characters):
Chan

Please enter your first name(within 22 characters):
Kaw
--------------------------------------------------------------------------
```

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

Enter your choice(0 to leave): 5
Please enter your phone number:
87654321
--------------------------------------------------------------------------
```

New record of the client':

Username	Password	Sex	Surname	First name	Telephone number
Wakenial_1	Hostel66	M	Kaw	Chan	87654321

Test case 2

Purpose:	To check if the client can return to the user page
Input:	‘0’
Expected Output:	Goes back to user page
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

```
Username: Wakenial
Password: hostel666
Sex: M
Name: Wak Lee
Phone_no: 12345678
--------------------------------------------------------------------------------

Which one do you want to amend?(Please only enter the number)

1.Username

2.Password

3.Sex

4.Name

5.Phone number
--------------------------------------------------------------------------------

Enter your choice(0 to leave): 0
Anymore?(Y/N)
N

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

The amendment has been done!
Press the ENTER button to return to the previous page.
```

then goes back to user page

Test case of booking rooms

Test case 1

Purpose:	To check if the program can facilitate the booking
Input:	Valid and available booking
Expected Output:	The related text file should be changed.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

Ms. Lee wants to book a twin room from 14th February to 15th February celebrate the Valentine's Day with her boyfriend. Meanwhile, Room 107 is available. Ms. Lee has no any booking records, with the username, 'Wakenial'

In rm2.txt, for the records of Room 107:

Room number	Type of room	Day & Status	Day & Status	…	Day & Status
107	2	1	2		31

In records_booked.txt, for Ms.Lee:

Username	Number of booking	Room	Check-in month	Check-in day	Check-out month	Check-out day
Wakenial	0					

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

The check-out date is up to Decemeber 2015

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

Check-in(month):2

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

Check-in(day):14

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

Check-out(month):2

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

Check-out(day):15

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

What kind of room(Single room(1)/Twin room(2)/Family room(3)/Suite Room(4),Pleas
e enter the number):2

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

Room    107      is reserved for you.
```

In rm2.txt, new records of Room 107: (In the same line,)

Room number	Type of room	Day & Status
107	2	1

...

Day & Status	Day & status
14x	15x

...

Day & Status
31

In records_booked.txt, for Ms.Lee:

Username	Number of booking	Room	Check-in month	Check-in day	Check-out month	Check-out day
Wakenial	1	43	2	14	2	15

Test case 2

Purpose:	To check if the program can detect unavailable booking
Input:	Valid but unavailable booking
Expected Output:	Reminder of no vacancy should be displayed
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

Ms. Lee also wants to book a twin room from 25th February to 26th February to celebrate Christmas with family. Unfortunately, no room is available. Ms. Lee has no any booking records, with the username, ‘Wakenial’

```
The check-out date is up to Decemeber 2013

Check-in(month):12

Check-in(day):25

Check-out(month):12

Check-out(day):26

What kind of room(Single room(1)/Twin room(2)/Family room(3)/Suite Room(4),P
e enter the number):4
There are no vacancies!
```

Test case 3

Purpose:	To check if the program can detect invalid booking
Input:	Invalid booking
Expected Output:	Reminder of invalid booking should be displayed
Actual Output:	Some invalid bookings are seemly finished.
Test Result:	The program did not response as expected
Follow-up Action:	One more condition is added into the program to ensure that the program can detect the invalid booking

```
The check-out date is up to Decemeber 2016

Check-in(month):1

Check-in(day):1

Check-out(month):1

Check-out(day):1

What kind of room(Single room(1)/Twin room(2)/Family room(3)/Suite Room(4),Pleas
e enter the number):1

Room    101      is reserved for you.
```

*Despite the restriction, the client can seemly book 1st January 2017. (Given the computer time is 2nd January)

After the follow-up action:

```
The available booking period is up to Decemeber 2016

Check-in(month):1

Check-in(day):1

Check-out(month):1

Check-out(day):1

The booking exceeds the available period! Please enter again!

Check-in(month):
```

Test case of cancelling rooms

Test case 1

Purpose:	To check if the program can facilitate canceling the booking
Input:	Valid option
Expected Output:	The related text file should be changed.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

Ms. Lee has booked Room 107, a twin room, from 14th February to 15th February to celebrate the Valentine's Day with her boyfriend. Her boyfriend has special arrangements on the day, so she decides to cancel the booking. Ms. Lee has no other booking records, with the username, 'Wakenial'.

In rm2.txt, new records of Room 107: (In the same line,)

Room number	Type of room	Day & Status
107	2	1

...

Day & Status	Day & status
14x	15x

...

Day & Status
31

In records_booked.txt, for Ms.Lee:

Username	Number of booking	Room	Check-in month	Check-in day	Check-out month	Check-out day
Wakenial	1	43	2	14	2	15

```
You have booked
--------------------------------------------------------------------------------

                1. Room    107
                   The check-in date:14/2
                   The check-out date:15/2

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

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

Which would you like to cancel (Enter the 0 for exit) ?
1
Cancelled!
```

After cancelling the booking,

In rm2.txt, for the records of Room 107:

Room number	Type of room	Day & Status	Day & Status	...	Day & Status
107	2	1	2		31

In records_booked.txt, for Ms.Lee:

Username	Number of booking	Room	Check-in month	Check-in day	Check-out month	Check-out day
Wakenial	0					

Test case 1

Purpose:	To check if the program can detect the existed booking
Input:	/
Expected Output:	Reminder of no booking should be displayed
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

After cancelling the booking, Ms. Lee check once again to confirm that the booking is cancelled. Ms. Lee should have no booking records now, with the username, ‘Wakenial’.

4.4 Self-Evaluation

Pros of the program

The program has a clear and clean structure and interface. The real nitty-gritty of the program is making booking and cancelling the booking, but not unnecessary and gaudy decorations. I reckon the inter-face is user-friendly with the simplicity.

From the perspective of my friend, who entrusted me to develop the program, the program has a different meaning. It is to some extent annoying to check and re-check all these trivial information and organize them from clutter to eutaxy. It is especially true when it is busy season. With this program, it saves time from manually processed booking system. To be precise, it saves time from checking if there are any blunders.

As a little programmer, I would say it is easier for future development of the program owing to the reusable procedures.

Shortcomings of the program

However, there are also various shortcomings in this program. For instance, the types of functions are not rich enough. To be frank, it is a large order for me to transfer all the brainstormed ideas into codes for the program with a limited time.

Also, this program does nothing with the details of the hostel. It fails to promote the hostel. In other words, it requires the clients to do date research and have a comprehensive understanding of the hostel, which somehow deters their motivation to book the rooms in this hostel.

Future development

As my friend ask me to develop the program as soon as possible to alleviate their workload, I am foredoomed fall short of their expectation. What I can do is to at least work out the basic functions of a booking system.

If I will be given a chance to further develop the program, I will try to apply a more attractive interface and transform the cancelling function into changing the booking system. Besides, I am wondering if it is possible to set an administer account so that the hostel can have a deeper investigation on customers. This would help my friend think of strategies to attract customers. It is also possible to add more information of the hostel.

4.5 External Testing and Evaluation

As mentioned in Chapter4.2, in order to have external testing and reflections from other people, I have sent my program to my friends for testing and evaluation.

The following is the questionnaire that I distributed to the testers. Testers may report bugs, give ratting and suggestions through the “Testing and Evaluation Form”.

Testing and Evaluation Form

Please help to evaluate the program: Hostel booking system. Thanks!

Instruction:

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

Report on Bugs:

No.	Description of errors

Program Evaluation:

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

		Rating				
		Agree		Average		Disagree
1.	The design of the program	5	4	3	2	1
2.	Smoothness of the operation (whether there are loopholes)	5	4	3	2	1
3.	Comprehensibility of the program	5	4	3	2	1
4.	Readability of the output	5	4	3	2	1
5	Suitability of the program for customers					
6	Functionality of the program					

Which features you like most?

_____

What other functions should be provided by the program?

_____

_____

_____

Other Suggestions

_____

_____

_____

Summary of the evaluation

		Average score
1.	The design of the program	3.5
2.	Smoothness of the operation (whether there are loopholes)	4
3.	Comprehensibility of the program	4
4.	Readability of the output	4
5	Suitability of the program for customers	3.5
6	Functionality of the program	3

Summary of testers' comments

With reference to the collected questionnaire, most of the testers think that the program is user-friendly as they found the program gives clear and enough instructions to operate. Though comprehensive functions are involved, they agreed to the readability of the program. In addition, they found the front page interesting when the program was executed. They gave positive comment on it. The tester also responded with room to improve, in terms of the functionality. They thought that more functions are preferred.

To conclude, the testers perceived that the system designed is suitable to be used for the booking of a hostel.

Specific comments

There are some positive comments with advice given and I made some amendments afterwards.

Some testers remind me that clients cannot exit the registration. It is quite unreasonable if the clients simply enter the wrong option and they are forced to create a new account.

After amendments:

Also, the clients should have a preview of what kinds of rooms they have booked which is not displayed in my program.

Before amendments:

```
You have booked
--------------------------------------------------------------------------------

                1. Room    101
                   The check-in date:1/1
                   The check-out date:1/1

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

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

Which would you like to cancel (Enter the 0 for exit) ?
```

After amendments:

```
You have booked
--------------------------------------------------------------------------------

                1. A single room of
                   Room    101
                   The check-in date:1/2
                   The check-out date:1/2

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

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

Which would you like to cancel (Enter the 0 for exit) ?
```

Chapter 5 Conclusion & Discussion

5.1 Pros and cons of my Program

After the amendments of the program, the program is more appealing than before. It is believed that clearer instructions and user-friendly design can comfort the customers. It will impress the customers and boost the reputation of the hostel.

Of course, there are also various shortcomings in this program. For instance, the types of functions are not rich enough. To be frank, it is a large order for me to transfer all the brainstormed ideas into codes for the program with a limited time.

5.2 Future Improvement

I really wish to see the final product with various functions. But sadly it turns out kind of simple with insufficient time.

As mentioned, although I have done as much as I can in the functionality, I did not put much adherence to the appearance. Thus, I will try to apply a more attractive interface and transform the cancelling function into changing the booking system if possible. Besides, I am wondering if it is decent to set an administer account so that the hostel can have a deeper investigation on customers. This would help my friend think of strategies to attract customers. It is also a good idea to add more information of the hostel. Not every tourist grasps all the information of the hostel. If the hostel can provide up-to-date information like the discount and the pay for the room, it will be definitely user friendly to customers.

I have also thought of letting the customers to know the vacancy of the hostel as well. Yet, the function is seemingly repeated to the booking function as the clients will know whether the room is able. If I am going to develop the program further, I may make some changes on the design and transform it into the design like the online cinema ticket buying system. The clients can first have a brief preview of the vacancy before they log in.

5.3 Self-Reflection

First and foremost, time management is of the utmost importance. It is absolutely awful to strike a balance between academics and program design and coding, not to mention the leisure time. A tight timetable should be devised to settle down all the stuff while I should try my best to follow the timetable. There are always some inevitable accidents that I have to catch up with the ideal progress. But the truth is, the seemingly little, minor trivialities turns out form an enormous monster. Time flies, silently, with no evidence. It is no use crying over milk. Prepare ourselves for every new start. Fortunately, it is better than never. As long as we adopt a passionate attitude towards everything, and then persevere, the results are foreseeable. It may not be success, but it may be something meaningful, something paving the success of the rest of our life.

In addition to time management, I find myself somehow careless. Yet, thanks to the carelessness, I learn something that the others cannot experience. For instance, I typed an invalid string mistakenly in the program that the entire program failed to run but no one knew what happened to it. It is really a good example to illustrate that, failure does not matter. What really matters is that whether we accept the shortcoming or the failure in our life, find it out and then resolve. And we will really learn a lesson from the failure. It does not belong to anyone else, but only we can grasp it in heart. That is why introspection always outweigh others' criticism.

Chapter 6 Reference and Acknowledgement

From Internet websites

1. http://zhidao.baidu.com/question/230159219.html
2. http://tieba.baidu.com/p/3359139887
3. http://baike.baidu.com/item/pascal
4. http://baike.baidu.com/subview/1190/14530306.htm
5. http://www.bloodshed.net/devpascal.html
6. http://www.miniforum.net/showpost.fcgi?tempid=2&MGID=2345096&page=

From books

1. NSS Information and Communication Technology D1
2. NSS Information and Communication Technology D2

Acknowledgement

1. ICT Teacher Mr. Chu
2. Friends and family members who have given feedbacks to me about the flaws of my program.

Appendices

Appendix 1 – Program Code (after Testing & Evaluation)

program MCAnalysis;

……..

```
{ program Hotel;
USES Crt,dos;
type
   usertype = record
                 SurName:string[10];
                 FirstName:string[22];
                 Sex:string[1];
                 Phone_no:string[8];
                 Username:string[12];
                 Password:string[20];
                 Rooms:integer;
                 Rooms_no:array[1..103] of integer;
                 checkin_m:array[1..103] of integer;
                 checkout_m:array[1..103] of integer;
                 checkin_d:array[1..103] of integer;
                 checkout_d:array[1..103] of integer;
            end;

    roomtype = record
                 rm_no:array[1..103] of string;
                 rm_cap:array[1..103] of string;
                 date:array[1..103,1..31] of string[4];
            end;
    Droom_type = record
                 manyrms:array[1..103] of string;
            end;
var
   Inf_client:text;
   book:text;
   Inf_room:array[1..12]of text;
   Client:array[1..10000] of usertype;
```

```
   Room:array [1..12]of roomtype;
   client_detail:array [1..10000]of string;
   Room_detail:array [1..12]of droom_type;
   choice,u_choice:integer;
   h,min,s,ms,y,m,d,td:word;
   {keyword:integer;}

procedure moment;

begin
  gettime(h,min,s,ms);
  getdate(y,m,d,td);
  {writeln(y,'-',m,'-',d);
  writeln(h,':',min,':',s,'.',ms);}
  {readkey();??}
end;

procedure Read_Booked;
var useless:string[12];
    h,i:integer;
begin
     h:=0;
     assign(book,'records_booked.txt');
     reset(book);
     while not eof(book) do
     begin
         h:=h+1;
         read(book,useless,client[h].rooms);
         for i:=1 to client[h].rooms do
            with client[h] do

read(book,rooms_no[i],checkin_m[i],checkin_d[i],checkout_m[i],
checkout_d[i]);
         readln(book);
     end;
     close(book);
end;
```

```
procedure Read_ClientRecord;
var count,Counting:integer;
begin
    assign(Inf_client,'records.txt');
    reset(Inf_client);
    count:=0;
    counting:=0;
    while not eof(inf_client) do
    begin
         count:=count+1;
         readln(Inf_client,client_detail[count]);
    end;
    close(Inf_client);
    counting:=count;
    for count :=1 to counting do
    with client[count] do
        begin
            username:=copy(client_detail[count],1,12);
            password:=copy(client_detail[count],13,32);
            sex:=copy(client_detail[count],33,33);
            SurName:=copy(client_detail[count],35,44);
            FirstName:=copy(client_detail[count],45,66);
            phone_no:=copy(client_detail[count],67,74);
            {writeln(username,password,sex,'
',surname,firstname,phone_no); can test}
        end;
end;
{Readclient_record procedure done}

procedure amend_UserRecord(K_word:integer);
var z,haha:integer;
   S_name:string;

   i,j:integer;                          {same as the
registration one}
```

```
    R_SurName:string;
    R_FirstName:string;
    R_Sex:string;
    R_Phone_no:string;
    R_Username:string;
    R_Password:string;
    R_Password1:string;
    avail,dbcheck,anymore:boolean;
    more:string;
    code,real:integer;
begin    clrscr;

     Read_ClientRecord;
     real:=0;
     avail:=true;
     anymore:=false;
     dbcheck:=true;
     with client[K_word] do
         begin
             R_username:=username;
             R_Password:=Password;
             R_Sex:=sex;
             R_SurName:=surname;
             R_FirstName:=firstname;
             R_Phone_no:=Phone_no;
             writeln('Username: ',username);
             writeln('Password: ',Password);
             writeln('Sex: ',sex);
             S_name:=firstname;
             repeat
                  if copy(S_name,length(S_name),1)=' '
                     then
S_name:=copy(S_name,1,length(S_name)-1);
             until copy(S_name,length(S_name),1)<>' ';
             writeln('Name: ',S_name,' ',surname);
             writeln('Phone_no: ',Phone_no);
         end;
     repeat
```

```
    textcolor(yellow);

writeln('-------------------------------------------------------
----------------------------');
    textcolor(white);
    writeln('Which one do you want to amend?(Please only enter the
number)');
    writeln;
    writeln('1.Username');
    writeln;
    writeln('2.Password');
    writeln;
    writeln('3.Sex');
    writeln;
    writeln('4.Name');
    writeln;
    writeln('5.Phone number');
    textcolor(yellow);

writeln('-------------------------------------------------------
----------------------------');
    textcolor(white);
    write('Enter your choice(0 to leave): ');
    readln(z);
    If z =1
{amend of username}
    then                  begin
                              while ((length(R_username))>12) or
((length(R_username))<=0) or avail  do
                                   begin
                                       writeln('Please enter
your username(within 12 strings):');
                                       readln(R_username);
                                       if
((length(R_username))>12) or (length(R_username)<=0) then
                                          begin
                                              writeln('Your
username is not available. Your username should neither be empty
```

```
nor exceed 12 strings.');
                                        writeln;
                                    end
                                    else
                                    begin
                                        for j:=1 to
(12-length(R_username)) do

R_username:=R_username+' ';

{writeln(length(copy(R_username,1,12))); can test  fill in the
blanks}
                                        avail:=false;
                                        for i := 1 to 100 do
                                        if
R_username=client[i].username
                                        then avail:=true;
                                        if avail=true then
                                        begin
                                            writeln('The
username is already occupied, please enter another username.');

textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');

textcolor(white);
                                        end else
                                           begin
                                           writeln('The
username is available!');

textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
```

```
textcolor(white);
                                                    end
                                            end
                                        end;
                                avail:=true;
                            end;
    If z =2
{amend of password}
    then                    begin
                                while (length(R_password)>20) or
(length(R_password)<8)or dbcheck or avail do
                                        begin
                                            writeln('Please enter
your password(8-20 strings):');
                                            readln(R_password);
                                            if
(length(R_password)>20) or (length(R_password)<8) then
                                                begin
                                                    writeln('Your
password is not available. Your password should obtain at least
8 strings and within 20 strings.');
                                                    writeln;
                                                end
                                                else begin

writeln('Please enter your password again for double checking:');

readln(R_password1);
                                                        if
R_password1<>R_password
                                                            then begin

writeln('Not consistent! Please enter again!');

writeln();
                                                                end
                                                            else
dbcheck:=false;
```

```
                                                end;
                                      avail:=false;
                                      end;
                                for j:=1 to (20-length(R_password))
do
                                R_password:=R_password+' ';
                                writeln('The password is
available!');
                                textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
                                textcolor(white);
                                avail:=true;
                            end;
    If z =3
{amend of sex}
    then                     begin
                                while (R_sex<>'M') and (R_sex<>'F')
or avail do
                                     begin

writeln('Gender(M/F):');
                                         readln(R_sex);
                                         If (R_sex<>'M') and
(R_sex<>'F')
                                         then     begin

writeln('Not available! Please enter again!');
                                                        writeln();
                                                  end;
                                         avail:=false;
                                     end;
                                     R_sex:=R_sex+' ';
                                     textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
```

```
                                    textcolor(white);
                                avail:=true;
                            end;
    If z =4
{amend of name}
    then                    begin
                                while (length(R_SurName)>10) or
(length(R_SurName)<=0) or avail do
                                    begin
                                        writeln('Please enter
your surname(within 10 characters):');
                                        readln(R_SurName);
                                        if (length(R_SurName)>10)
or (length(R_SurName)<=0) then
                                        begin
                                            writeln('Your
surname is not available. Your surname should neither be empty nor
exceed 10 characters.');
                                            writeln;
                                        end;
                                        avail:=false;
                                    end;
                                avail:=true;
                                for j:=1 to (10-length(R_surname))
do
                                R_surname:=R_surname+' ';
                                writeln();
                                while (length(R_FirstName)>22) or
(length(R_FirstName)<=0) or avail do
                                    begin
                                        writeln('Please enter
your first name(within 22 characters):');
                                        readln(R_FirstName);
                                        if
(length(R_FirstName)>22) or (length(R_FirstName)<=0) then
                                          begin
                                          writeln('Your first
name is not available. Your first name should neither be empty nor
```

```
exceed 22 characters.');
                                        writeln;
                                        end;
                                    avail:=false;
                                end;
                            avail:=true;
                            for j:=1 to
(22-length(R_FirstName)) do
                            R_FirstName:=R_FirstName+' ';
                            textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
                            textcolor(white);
                        end;
    If z =5
{amend of phone no}
    then                 begin
                            while (length(R_Phone_no)<>8) or
(real=0) do
                                begin
                                    writeln('Please enter
your phone number:');
                                    readln(R_Phone_no);

val(R_Phone_no,real,code);
                                    if real=0
                                    then begin
                                            writeln('This is
not a phone number!Please enter again!');
                                            writeln();
                                        end
                                    else if
(length(R_Phone_no)<>8) then
                                        begin
                                            writeln('Your
phone number is not available. Your phone number must be in 8
numbers');
```

```
                                                    writeln;
                                                end;
                                            end;
                                        textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
                                        textcolor(white);
                                    end;
    writeln('Anymore?(Y/N)');
    readln(more);
    if more='N' then anymore:=true;
    writeln();
    until anymore;
    assign(Inf_client,'records.txt');
    rewrite(Inf_client);
    for haha:=1 to (K_word-1) do
       writeln(Inf_client,client_detail[haha]);
    writeln(Inf_client,R_username,R_Password,R_Sex,'
',R_SurName,R_FirstName,R_Phone_no);

    for haha:=k_word+1 to 10000 do
       begin
       if client_detail[haha]<>''
       then write(Inf_client,client_detail[haha]);
       if client_detail[haha+1]<>''
       then writeln(Inf_client);
       end;
    close(Inf_client);
    textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
    textcolor(white);
    writeln('The amendment has been done!');
    writeln('Press the ENTER button to return to the previous
page.');
    readln();
```

```
end;
{amenduser_record procedure done}


procedure  Read_RoomRecord;
var i,j,k,l:integer;
begin
    l:=1;
    for i:=1 to 12 do
      begin
          case i of

1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));
          10:assign(Inf_room[i],'rm10.txt');
          11:assign(Inf_room[i],'rm11.txt');
          12:assign(Inf_room[i],'rm12.txt');
          end;
          reset(Inf_room[i]);
          {readln;}
      for j := 1 to 103  do
          begin
              readln(Inf_room[i],room_detail[i].manyrms[j]);
              {writeln(room_detail[i] [j]);}

room[i].rm_no[j]:=copy(room_detail[i].manyrms[j],1,9);

room[i].rm_cap[j]:=copy(room_detail[i].manyrms[j],10,7);
              l:=1;
              for k :=1 to 31 do
                 begin

room[i].date[j,k]:=copy(room_detail[i].manyrms[j],16+l,4);
                      {WRITELN(room[i].date[k]);}
```

```
                        l:=l+4;
                    end;
            end;
            close(Inf_room[i])
        end;
end;
{Readroom_record procedure done}

Procedure Cancel_booking(who:integer);
var i,option:integer;
   month_i,day_i,month_o,day_o,month_record:integer;
   longterm:string;
   final,j,long,update,the_rm:integer;
begin
    clrscr;
    If client[who].rooms=0
    then begin

writeln;writeln;writeln;writeln;writeln;writeln;writeln;writel
n;writeln;writeln;writeln;
        textcolor(11);
        writeln('':25,'You have not booked any rooms!');
        textcolor(white);
        sound(1);
        readln();
        end;
    If client[who].rooms<>0
    then
       begin
           writeln;
           writeln('You have booked ');
           writeln;
           textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
           textcolor(white);
           for i:=1 to client[who].rooms do
```

```
                begin
                    write(' ':15,i,'. ');
                    case client[who].rooms_no[i] of
                    1..42:writeln('A single room of');
                    43..84:writeln('A twin room of');
                    85..102:writeln('A family room of');
                    103:writeln('A suite room of   ');
                    end;
                    writeln(' ':18,'Room
',room[1].rm_no[client[who].rooms_no[i]]);
                    writeln(' ':18,'The check-in
date:',client[who].checkin_d[i],'/',client[who].checkin_m[i]);
                    writeln(' ':18,'The check-out
date:',client[who].checkout_d[i],'/',client[who].checkout_m[i]
);
                    writeln;
                    textcolor(lightblue);

writeln('-------------------------------------------------------
---------------------------');
                    textcolor(white);
                end;
            textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
            textcolor(white);
            repeat
            writeln('Which would you like to cancel (Enter the 0
for exit) ?');
            readln(option);
            if (option>i) or (option<0)
               then writeln('Not available! Please enter again!');
            until (option<=i) and (option>=0);
        if option<>0 then
        begin
            month_i:=client[who].checkin_m[option];
            month_o:=client[who].checkout_m[option];
```

```
            day_i:=client[who].checkin_d[option];
            day_o:=client[who].checkout_d[option];
            final:= client[who].rooms_no[option];

                    if (month_i<=month_o)then
                    for i:=month_i to month_o do
                       begin
                           case i of

1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));

10:assign(Inf_room[i],'rm10.txt');

11:assign(Inf_room[i],'rm11.txt');

12:assign(Inf_room[i],'rm12.txt');
                           end;
                           rewrite(Inf_room[i]);
                       end;
                    if (month_i>month_o)then
                    begin
                        for i:=month_i to 12 do
                                begin
                                    case i of

1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));

10:assign(Inf_room[i],'rm10.txt');

11:assign(Inf_room[i],'rm11.txt');

12:assign(Inf_room[i],'rm12.txt');
                                    end;
                                    rewrite(Inf_room[i]);
                                end;
                        for i:=1 to month_o do
                                begin
                                    case i of
```

```
1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));

10:assign(Inf_room[i],'rm10.txt');

11:assign(Inf_room[i],'rm11.txt');

12:assign(Inf_room[i],'rm12.txt');
                                    end;
                                    rewrite(Inf_room[i]);
                                end;
                    end;

                    if (month_i=month_o)then
                                        begin
                                            for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);

write(Inf_room[month_i],room[month_i].rm_no[final],room[month_
i].rm_cap[final]);
                                            for update:=1 to
(day_i-1) do

write(Inf_room[month_i],room[month_i].date[final,update]);
                                            for update:=day_i
to day_o do
                                               begin

str(update,longterm);
                                                   for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_i],longterm);
```

```
                                                end;
                                            for update:=
day_o+1 to 31 do

write(Inf_room[month_i],room[month_i].date[final,update]);

writeln(Inf_room[month_i]);
                                            for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);

close(Inf_room[month_i]);
                                        end;




                    if (month_i<month_o)then
                                        begin
                                            for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);


write(Inf_room[month_i],room[month_i].rm_no[final],room[month_
i].rm_cap[final]);
                                            for update:=1 to
(day_i-1) do

write(Inf_room[month_i],room[month_i].date[final,update]);
                                            for update:=day_i
to 31 do
                                                begin
```

```
str(update,longterm);
                                                    for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_i],longterm);
                                                 end;


writeln(Inf_room[month_i]);
                                             for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);

close(Inf_room[month_i]);


                                             for
month_record:=month_i+1 to month_o-1 do
                                                 begin
                                                     for
the_rm:=1 to (final-1) do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

write(Inf_room[month_record],room[month_record].rm_no[final],r
oom[month_record].rm_cap[final]);

                                                     for
update:=1 to 31 do
                                                        begin

str(update,longterm);
                                                            for
```

```
long:=1 to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_record],longterm);
                                                        end;

writeln(Inf_room[month_record]);

                                            for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

close(Inf_room[month_record]);
                                               end;

                                            for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);


write(Inf_room[month_o],room[month_o].rm_no[final],room[month_
o].rm_cap[final]);
                                            for update:=1 to
day_o do
                                               begin

str(update,longterm);
                                                   for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_o],longterm);
```

```
                                                end;
                                            for
update:=(day_o+1) to 31 do

write(Inf_room[month_o],room[month_o].date[final,update]);



writeln(Inf_room[month_o]);
                                            for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);

close(Inf_room[month_o]);
                                        end;
                    if (month_i>month_o)then
                                        begin
                                            for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);


write(Inf_room[month_i],room[month_i].rm_no[final],room[month_
i].rm_cap[final]);
                                            for update:=1 to
(day_i-1) do

write(Inf_room[month_i],room[month_i].date[final,update]);
                                            for update:=day_i
to 31 do
                                                begin

str(update,longterm);
                                                    for long:=1
```

```
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_i],longterm);
                                                end;


writeln(Inf_room[month_i]);
                                            for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);

close(Inf_room[month_i]);


                                            for
month_record:=month_i+1 to 12 do
                                                begin
                                                    for
the_rm:=1 to (final-1) do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

write(Inf_room[month_record],room[month_record].rm_no[final],r
oom[month_record].rm_cap[final]);

                                                    for
update:=1 to 31 do
                                                       begin

str(update,longterm);
                                                           for
long:=1 to (4-length(longterm)) do
```

```
longterm:=longterm+' ';

write(Inf_room[month_record],longterm);
                                                        end;

writeln(Inf_room[month_record]);

                                                      for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

close(Inf_room[month_record]);
                                                  end;


                                              for
month_record:=1 to month_o-1 do
                                                  begin
                                                      for
the_rm:=1 to (final-1) do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

write(Inf_room[month_record],room[month_record].rm_no[final],r
oom[month_record].rm_cap[final]);

                                                      for
update:=1 to 31 do
                                                         begin

str(update,longterm);
                                                             for
long:=1 to (4-length(longterm)) do

longterm:=longterm+' ';
```

```
write(Inf_room[month_record],longterm);
                                                        end;

writeln(Inf_room[month_record]);

                                                    for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

close(Inf_room[month_record]);
                                                   end;

                                              for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);


write(Inf_room[month_o],room[month_o].rm_no[final],room[month_
o].rm_cap[final]);
                                              for update:=1 to
day_o do
                                                   begin

str(update,longterm);
                                                    for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_o],longterm);
                                                   end;
                                              for
update:=(day_o+1) to 31 do
```

```
write(Inf_room[month_o],room[month_o].date[final,update]);



writeln(Inf_room[month_o]);
                                            for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);

close(Inf_room[month_o]);
                                        end;
                    writeln('Cancelled!');

                    assign(book,'records_booked.txt');
                    rewrite(book);
                    for i := 1 to (who-1) do
                    begin

write(book,client[i].username,client[i].rooms);
                        for j:= 1 to client[i].rooms do
                           with client[i] do
                               write(book,' ',rooms_no[j],'
',checkin_m[j],' ',checkin_m[j],' ',checkout_m[j],'
',checkout_m[j]);
                               writeln(book);
                    end;
                    client[who].rooms:=client[who].rooms-1;

write(book,client[who].username,client[who].rooms);
                    for i:=1 to(option-1) do
                       with client[who] do
                       write(book,' ',rooms_no[i],'
',checkin_m[i],' ',checkin_d[i],' ',checkout_m[i],'
',checkout_d[i]);
                    for i:=option+1 to(client[who].rooms+1) do
```

```
                        with client[who] do
                        write(book,' ',rooms_no[i],'
',checkin_m[i],' ',checkin_d[i],' ',checkout_m[i],'
',checkout_d[i]);



                    writeln(book);
                    for i := (who+1) to 10000 do
                        if client[i].username<>'' then
                            begin

write(book,client[i].username,client[i].rooms);
                                for j:= 1 to client[i].rooms do
                                with client[i] do
                                    write(book,'
',rooms_no[j],' ',checkin_m[j],' ',checkin_d[j],'
',checkout_m[j],' ',checkout_d[j]);
                                    writeln(book);
                            end;
                    close(book);
                end;
   readln;
   end;
end;




procedure  Make_booking(who:integer);
var month_i,day_i,month_o,day_o,day1,cap,month_record:integer;
   shortterm,booked_rm,longterm:string;

final,i,j,rmstart,rmend,l,long,update,the_rm,day,ini:integer;
   really,ok,restriction:boolean;
   availability: array[1..103] of boolean;
begin
    clrscr;
```

```
    moment;
    ok:=false;
    restriction:=false;
    for ini:=1 to 103 do
       availability[ini]:=true;
    booked_rm:='';
    really:=false;
    textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
    textcolor(white);
    write('The available booking period is up to ');
    case m of
    1:write('Decemeber ');
    2:write('January ');
    3:write('February ');
    4:write('March ');
    5:write('April ');
    6:write('May ');
    7:write('June ');
    8:write('July ');
    9:write('August ');
    10:write('September ');
    11:write('October ');
    12:write('November ');end;
    if m <>1
      then    writeln(y+1)
      else    writeln(y);
    writeln;
    repeat

    textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
    textcolor(white);
    repeat
```

```
         write('Check-in(month):');
         readln(month_i);
         If (month_i>12) or (month_i<1)
            then begin textcolor(11);
                    sound(1);writeln('Not available! Please
enter again!');writeln; textcolor(white); end
            else really:=true;
    until really;
    really:=false;
    writeln;
    textcolor(lightblue);

writeln('-----------------------------------------------------
---------------------------');
    textcolor(white);
    repeat
         write('Check-in(day):');
         readln(day_i);
         case (month_i) of
         1,3,5,7,8,10,12:  if (day_i>=1) and (day_i<=31)
                              then really:=true;
         4,6,9,11:        if (day_i>=1) and (day_i<=30)
                              then really:=true;
         2:               begin
                              day1:=28;
                              if (y mod 4)= 0  then day1:=29;
                              if (day_i>=1) and (day_i<=day1)
                                 then really:=true;
                          end;
         end;
         If not really
            then begin
                    textcolor(11);
                    sound(1);
                    writeln('Not available! Please enter
again!');
                    writeln;
                    textcolor(white);
```

```
                end;
    until really;
    really:=false;
    writeln;
    textcolor(lightblue);

writeln('-------------------------------------------------------
---------------------------');
    textcolor(white);
    repeat
         write('Check-out(month):');
         readln(month_o);
         If (month_o>12) or (month_o<1)
           then begin
                   textcolor(11);
                   sound(1);
                   writeln('Not available! Please enter
again!');
                   writeln;
                   textcolor(white);
                end
           else really:=true;
    until really;
    really:=false;
    writeln;
    textcolor(lightblue);

writeln('-------------------------------------------------------
---------------------------');
    textcolor(white);
    repeat
         write('Check-out(day):');
         readln(day_o);
         case (month_o) of
         1,3,5,7,8,10,12:  if (day_o>=1) and (day_o<=31)
                              then really:=true;
         4,6,9,11:        if (day_o>=1) and (day_o<=30)
                              then really:=true;
```

```
         2:              begin
                             day1:=28;
                             if m>month_o
                                then y:=y+1;
                             if (y mod 4)= 0  then day1:=29;
                             if (day_o>=1) and (day_o<=day1)
                                then really:=true;
                             y:=y-1;
                          end;
         end;
         If not really
            then begin
                     textcolor(11);
                     sound(1);
                     writeln('Not available! Please enter
again!');
                     writeln;
                     textcolor(white);
                  end;
    until really;
    really:=false;
    writeln;
    textcolor(lightblue);

writeln('-------------------------------------------------------
---------------------------');
    textcolor(white);
    if ((month_i= m) and (day_i<d)) or  ((month_o= m) and
(day_o<d))
    then  begin
         textcolor(11);
         sound(1);
         writeln('The booking exceeds the available period! Please
enter again!');
         textcolor(lightblue);

writeln('-------------------------------------------------------
---------------------------');
```

```
          textcolor(white);
          restriction:=false;
          end
     else restriction:=true;
     until  restriction;

     repeat
          write('What kind of room(Single room(1)/Twin
room(2)/Family room(3)/Suite Room(4),Please enter the number):');
          readln(cap);
          If (cap=1) or (cap=2) or (cap=3) or (cap=4)
            then really:=true
            else begin
                    textcolor(11);
                    sound(1);
                    writeln('Not available! Please enter
again!');
                    textcolor(white);
                end;
     until really;
     really:=false;

     case cap of
     1:   begin
              rmstart:=1;rmend:=42;
         end;
     2:   begin
              rmstart:=43;rmend:=84;
         end;
     3:   begin
              rmstart:=85;rmend:=102;
         end;
     4:   begin
              rmstart:=103;rmend:=103;
         end;
     end;

     if month_i=month_o then
```

```
      for the_rm:=rmstart to rmend do
         for  day:=day_i to day_o do
             begin
                 str(day,shortterm);
                 for l:= 1 to (4-length(shortterm)) do
                    shortterm:=shortterm+' ';
                 if room[month_i].date[the_rm,day]<>shortterm
                   then availability[the_rm]:=false;
             end;

    if month_i<month_o then
      begin
          for the_rm:=rmstart to rmend do
             for  day:=day_i to 31 do
                 begin
                     str(day,shortterm);
                     for l:= 1 to (4-length(shortterm)) do
                        shortterm:=shortterm+' ';
                     if
room[month_i].date[the_rm,day]<>shortterm
                        then availability[the_rm]:=false;
                 end;

          for month_record:=month_i to month_o-1 do
             for the_rm:=rmstart to rmend do
                for  day:=day_i to day_o do
                    begin
                        str(day,shortterm);
                        for l:= 1 to (4-length(shortterm)) do
                           shortterm:=shortterm+' ';
                        if
room[month_record].date[the_rm,day]<>shortterm
                           then availability[the_rm]:=false;
                    end;
          for the_rm:=rmstart to rmend do
             for  day:=1 to day_o do
                 begin
                     str(day,shortterm);
```

```
                        for l:= 1 to (4-length(shortterm)) do
                            shortterm:=shortterm+' ';
                        if
room[month_o].date[the_rm,day]<>shortterm
                           then availability[the_rm]:=false;
                    end;
       end;

       if month_i>month_o then
       begin
           for month_record:=month_i to 12 do
               for the_rm:=rmstart to rmend do
                   for  day:=day_i to 31 do
                       begin
                           str(day,shortterm);
                           for l:= 1 to (4-length(shortterm)) do
                               shortterm:=shortterm+' ';
                           if
room[month_record].date[the_rm,day]<>shortterm
                              then availability[the_rm]:=false;
                       end;

           for month_record:=1 to month_o do
               for the_rm:=rmstart to rmend do
                   for  day:=day_i to 31 do
                       begin
                           str(day,shortterm);
                           for l:= 1 to (4-length(shortterm)) do
                               shortterm:=shortterm+' ';
                           if
room[month_record].date[the_rm,day]<>shortterm
                              then availability[the_rm]:=false;
                       end;
       end;
       for the_rm:=rmend downto rmstart do
           if availability[the_rm]=true
             then
                begin
```

```
                    final:=the_rm;
                    booked_rm:=room[1].rm_no[final];
                    {writeln(booked_rm);}
                    ok:=true;
                end;
      if (not ok) then writeln('There are no vacancies!')
           else begin
                    if (month_i<=month_o)then
                    for i:=month_i to month_o do
                       begin
                           case i of

1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));

10:assign(Inf_room[i],'rm10.txt');

11:assign(Inf_room[i],'rm11.txt');

12:assign(Inf_room[i],'rm12.txt');
                           end;
                           rewrite(Inf_room[i]);
                       end;
                    if (month_i>month_o)then
                    begin
                        for i:=month_i to 12 do
                                begin
                                    case i of

1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));

10:assign(Inf_room[i],'rm10.txt');

11:assign(Inf_room[i],'rm11.txt');

12:assign(Inf_room[i],'rm12.txt');
                                    end;
                                    rewrite(Inf_room[i]);
                                end;
```

```
                        for i:=1 to month_o do
                                begin
                                    case i of

1..9:assign(Inf_room[i],concat('rm',char(48+i),'.txt'));

10:assign(Inf_room[i],'rm10.txt');

11:assign(Inf_room[i],'rm11.txt');

12:assign(Inf_room[i],'rm12.txt');
                                    end;
                                    rewrite(Inf_room[i]);
                                end;
                    end;

                    if (month_i=month_o)then
                                        begin
                                            for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);
                                            {writeln('##@@
',day_i);                                 used to check what
happened

writeln(room[month_i].rm_no[final],room[month_i].rm_cap[final]
);}

write(Inf_room[month_i],room[month_i].rm_no[final],room[month_
i].rm_cap[final]);
                                            for update:=1 to
(day_i-1) do

write(Inf_room[month_i],room[month_i].date[final,update]);
                                            for update:=day_i
to day_o do
```

```
                                                    begin
str(update,longterm);
longterm:=longterm+'x';
                                                        for long:=1
to (4-length(longterm)) do
longterm:=longterm+' ';
write(Inf_room[month_i],longterm);
                                                    end;
                                                for update:=
day_o+1 to 31 do
write(Inf_room[month_i],room[month_i].date[final,update]);
writeln(Inf_room[month_i]);
                                                for
the_rm:=(final+1) to 103 do
writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);
close(Inf_room[month_i]);
                                            end;



                    if (month_i<month_o)then
                                            begin
                                                for the_rm:=1 to
(final-1) do
writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);
```

```
write(Inf_room[month_i],room[month_i].rm_no[final],room[month_
i].rm_cap[final]);
                                        for update:=1 to
(day_i-1) do

write(Inf_room[month_i],room[month_i].date[final,update]);
                                        for update:=day_i
to 31 do
                                           begin

str(update,longterm);

longterm:=longterm+'x';
                                               for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_i],longterm);
                                           end;


writeln(Inf_room[month_i]);
                                        for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);

close(Inf_room[month_i]);
                                        for
month_record:=month_i+1 to month_o-1 do
                                           begin
                                               for
the_rm:=1 to (final-1) do

writeln(Inf_room[month_record],room_detail[month_record].manyr
```

```
ms[the_rm]);

write(Inf_room[month_record],room[month_record].rm_no[final],r
oom[month_record].rm_cap[final]);
                                                    for
update:=1 to 31 do
                                                       begin

str(update,longterm);

longterm:=longterm+'x';
                                                          for
long:=1 to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_record],longterm);
                                                       end;

writeln(Inf_room[month_record]);

                                                    for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);


close(Inf_room[month_record]);
                                                 end;

                                              for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);
```

```
write(Inf_room[month_o],room[month_o].rm_no[final],room[month_
o].rm_cap[final]);
                                            for update:=1 to
day_o do
                                               begin

str(update,longterm);

longterm:=longterm+'x';
                                                   for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_o],longterm);
                                               end;
                                            for
update:=(day_o+1) to 31 do

write(Inf_room[month_o],room[month_o].date[final,update]);



writeln(Inf_room[month_o]);
                                            for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);

close(Inf_room[month_o]);
                                        end;
                    if (month_i>month_o)then
                                        begin
                                            for the_rm:=1 to
(final-1) do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
```

```
);

write(Inf_room[month_i],room[month_i].rm_no[final],room[month_
i].rm_cap[final]);
                                        for update:=1 to
(day_i-1) do

write(Inf_room[month_i],room[month_i].date[final,update]);
                                        for update:=day_i
to 31 do
                                           begin

str(update,longterm);

longterm:=longterm+'x';
                                               for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_i],longterm);
                                           end;


writeln(Inf_room[month_i]);
                                        for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_i],room_detail[month_i].manyrms[the_rm]
);

close(Inf_room[month_i]);


                                        for
month_record:=month_i+1 to 12 do
                                           begin
```

```
                                                    for
the_rm:=1 to (final-1) do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

write(Inf_room[month_record],room[month_record].rm_no[final],r
oom[month_record].rm_cap[final]);

                                                    for
update:=1 to 31 do
                                                       begin

str(update,longterm);

longterm:=longterm+'x';
                                                           for
long:=1 to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_record],longterm);
                                                       end;

writeln(Inf_room[month_record]);

                                                    for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

close(Inf_room[month_record]);
                                                 end;


                                              for
month_record:=1 to month_o-1 do
```

```
                                                    begin
                                                        for
the_rm:=1 to (final-1) do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

write(Inf_room[month_record],room[month_record].rm_no[final],r
oom[month_record].rm_cap[final]);

                                                        for
update:=1 to 31 do
                                                           begin

str(update,longterm);

longterm:=longterm+'x';
                                                               for
long:=1 to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_record],longterm);
                                                           end;

writeln(Inf_room[month_record]);

                                                        for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_record],room_detail[month_record].manyr
ms[the_rm]);

close(Inf_room[month_record]);
                                                    end;

                                                for the_rm:=1 to
(final-1) do
```

```
writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);


write(Inf_room[month_o],room[month_o].rm_no[final],room[month_
o].rm_cap[final]);
                                        for update:=1 to
day_o do
                                           begin

str(update,longterm);

longterm:=longterm+'x';
                                                for long:=1
to (4-length(longterm)) do

longterm:=longterm+' ';

write(Inf_room[month_o],longterm);
                                           end;
                                        for
update:=(day_o+1) to 31 do

write(Inf_room[month_o],room[month_o].date[final,update]);



writeln(Inf_room[month_o]);
                                        for
the_rm:=(final+1) to 103 do

writeln(Inf_room[month_o],room_detail[month_o].manyrms[the_rm]
);

close(Inf_room[month_o]);
                                     end;
                    writeln;
```

```
                    textcolor(yellow);

writeln('-------------------------------------------------------
----------------------------');
                    textcolor(white);
                    write('Room
');textcolor(lightred);write(booked_rm);textcolor(white);write
ln('is reserved for you.');

                    assign(book,'records_booked.txt');
                    rewrite(book);

                    for i := 1 to (who-1) do
                    begin

write(book,client[i].username,client[i].rooms);
                        for j:= 1 to client[i].rooms do
                           with client[i] do
                               write(book,' ',rooms_no[j],'
',checkin_m[j],' ',checkin_m[j],' ',checkout_m[j],'
',checkout_m[j]);
                               writeln(book);
                    end;

                    client[who].rooms:=client[who].rooms+1;

write(book,client[who].username,client[who].rooms);
                    for i:=1 to(client[who].rooms-1) do
                       with client[who] do
                       write(book,' ',rooms_no[i],'
',checkin_m[i],' ',checkin_d[i],' ',checkout_m[i],'
',checkout_d[i]);
                    write(book,' ',final,' ',month_i,' ',day_i,'
',month_o,' ',day_o);
                    writeln(book);
                    for i := (who+1) to 10000 do
                       if client[i].username<>''  then
                           begin
```

```
write(book,client[i].username,client[i].rooms);
                                for j:= 1 to client[i].rooms do
                                with client[i] do
                                    write(book,'
',rooms_no[j],' ',checkin_m[j],' ',checkin_d[j],'
',checkout_m[j],' ',checkout_d[j]);
                                    writeln(book);
                            end;
                    close(book);
                end;
       readln;
end;




procedure Homepage;
begin
    clrscr;
    Read_ClientRecord;
    writeln();writeln();writeln();
    textcolor(red);write(' ':6,'
--');textcolor(white);write('--');textcolor(red);write('--');t
extcolor(white);write('--');textcolor(red);write('--');textcol
or(white);write('--');textcolor(red);write('--');textcolor(whi
te);write('--');textcolor(red);write('--');textcolor(white);wr
ite('--');textcolor(red);write('--');textcolor(white);write('-
-');textcolor(red);write('--');textcolor(white);write('--');te
xtcolor(red);write('--');textcolor(white);write('--');textcolo
r(red);write('--');textcolor(white);write('--');textcolor(red)
;write('--');textcolor(white);write('--');textcolor(red);write
('--');textcolor(white);write('--');textcolor(red);write('--')
;textcolor(white);write('--');textcolor(red);write('--');textc
olor(white);write('--');textcolor(red);write('--');textcolor(w
hite);write('--');textcolor(red);write('--');textcolor(white);
write('--');textcolor(red);writeln('--');
    textcolor(lightred);writeln(' ':6,'|
```

```
|');
    textcolor(white);writeln(' ':6,'|                    Welcome To
Wakenial Motel              |');
    textcolor(lightred);writeln(' ':6,'|
|');
    textcolor(white);writeln(' ':6,'|
1.Register                          |');
    textcolor(lightred);writeln(' ':6,'|
|');
    textcolor(white);writeln(' ':6,'|                      2.Login
Account                     |');
    textcolor(lightred);writeln(' ':6,'|
|');
    textcolor(white);writeln(' ':6,'|                       3.Off
|');
    textcolor(lightred);writeln(' ':6,'|
|');
    textcolor(red);write(' ':6,'
--');textcolor(white);write('--');textcolor(red);write('--');t
extcolor(white);write('--');textcolor(red);write('--');textcol
or(white);write('--');textcolor(red);write('--');textcolor(whi
te);write('--');textcolor(red);write('--');textcolor(white);wr
ite('--');textcolor(red);write('--');textcolor(white);write('-
-');textcolor(red);write('--');textcolor(white);write('--');te
xtcolor(red);write('--');textcolor(white);write('--');textcolo
r(red);write('--');textcolor(white);write('--');textcolor(red)
;write('--');textcolor(white);write('--');textcolor(red);write
('--');textcolor(white);write('--');textcolor(red);write('--')
;textcolor(white);write('--');textcolor(red);write('--');textc
olor(white);write('--');textcolor(red);write('--');textcolor(w
hite);write('--');textcolor(red);write('--');textcolor(white);
write('--');textcolor(red);writeln('--');
    writeln();
    textcolor(white); write('                        Enter your
choice: ');
    readln(choice);
end;
{homepage procedure done}
```

```
procedure register_client;
var i,j:integer;
   R_SurName:string;
   R_FirstName:string;
   R_Sex:string;
   R_Phone_no:string;
   R_Username:string;
   R_Password:string;
   R_Password1:string;
   avail,dbcheck:boolean;
   code,real:integer;
   f:text;
begin
    clrscr;
    dbcheck:=true;
    R_username:='';
    R_Password:='';
    R_Sex:='';
    R_SurName:='';
    R_FirstName:='';
    R_Phone_no:='';
    real:=0;
    avail:=true;
    while ((length(R_username))>12) or ((length(R_username))<=0)
or avail  do
         begin
             writeln('Please enter your username(within 12
strings):');
             readln(R_username);
             if ((length(R_username))>12) or
(length(R_username)<=0) then
             begin
                 textcolor(lightred);
                 writeln('Your username is not available. Your
username should neither be empty nor exceed 12 strings.');
```

```
                sound(1);
                writeln;
                textcolor(white);
            end
            else begin{}
                for j:=1 to (12-length(R_username)) do
                   R_username:=R_username+' ';

{writeln(length(copy(R_username,1,12))); can test   fill in the
blanks}
                avail:=false;
                for i := 1 to 100 do
                   if R_username=client[i].username
                     then avail:=true;
                if avail=true then
                  begin
                      textcolor(lightred);
                      writeln('The username is already occupied,
please enter another username.');
                      sound(1);
                      writeln;
                      textcolor(white);
                  end else
                      writeln('The username is available!');
                writeln();
            end
        end;
{R_username done}
     textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
     textcolor(white);
     while (length(R_password)>20) or (length(R_password)<8)or
dbcheck do
         begin
             writeln('Please enter your password(8-20
strings):');
```

```
              readln(R_password);
              if (length(R_password)>20) or
(length(R_password)<8) then
                begin
                    textcolor(lightred);
                    writeln('Your password is not available. Your
password should obtain at least 8 strings and within 20 strings.');
                    sound(1);
                    writeln;
                    textcolor(white);
                end
                else begin
                    writeln();
                    writeln('Please enter your password again for
double checking:');
                    readln(R_password1);
                    if R_password1<>R_password
                      then begin
                          textcolor(lightred);
                          writeln('Not consistent! Please enter
again!');
                          sound(1);
                          writeln();
                          textcolor(white);
                          end
                      else     dbcheck:=false;
                    end;
          end;
     for j:=1 to (20-length(R_password)) do
     R_password:=R_password+' ';
     writeln('The password is available!');
     writeln();
{R_password done}
     textcolor(yellow);

writeln('-------------------------------------------------------
----------------------------');
     textcolor(white);
```

```
     while (R_sex<>'M') and (R_sex<>'F') do
          begin
               writeln('Gender(M/F):');
               readln(R_sex);
               If (R_sex<>'M') and (R_sex<>'F')
                  then     begin
                              textcolor(lightred);
                              writeln('Not available! Please
enter again!');
                              sound(1);
                              writeln();
                              textcolor(white);
                          end;
          end;
     R_sex:=R_sex+' ';
     writeln();
{R_sex done}
     textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
     textcolor(white);
     while (length(R_SurName)>10) or (length(R_SurName)<=0) do
          begin
               writeln('Please enter your surname(within 10
characters):');
               readln(R_SurName);
               if (length(R_SurName)>10) or (length(R_SurName)<=0)
then
               begin
                    textcolor(lightred);
                    writeln('Your surname is not available. Your
surname should neither be empty nor exceed 10 characters.');
                    sound(1);
                    writeln;
                    textcolor(white);
               end;
          end;
```

```
     for j:=1 to (10-length(R_surname)) do
        R_surname:=R_surname+' ';
     writeln();
{R_surname done}
     textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
     textcolor(white);
     while (length(R_FirstName)>22) or (length(R_FirstName)<=0)
do
          begin
              writeln('Please enter your first name(within 22
characters):');
              readln(R_FirstName);
              if (length(R_FirstName)>22) or
(length(R_FirstName)<=0) then
              begin
                  textcolor(lightred);
                  writeln('Your first name is not available. Your
first name should neither be empty nor exceed 22 characters.');
                  sound(1);
                  writeln;
                  textcolor(white);
              end;
          end;
     for j:=1 to (22-length(R_FirstName)) do
        R_FirstName:=R_FirstName+' ';
     writeln();
{R_FirstName done}
     textcolor(yellow);

writeln('-----------------------------------------------------
---------------------------');
     textcolor(white);
     while (length(R_Phone_no)<>8) or (real=0) do
          begin
              writeln('Please enter your phone number:');
```

```
               readln(R_Phone_no);
               val(R_Phone_no,real,code);
               if real=0
                  then begin
                          textcolor(lightred);
                          writeln('This is not a phone
number!Please enter again!');
                          sound(1);
                          writeln();
                          textcolor(white);
                      end
                  else if (length(R_Phone_no)<>8) then
                      begin
                          textcolor(lightred);
                          writeln('Your phone number is not
available. Your phone number must be in 8 numbers');
                          sound(1);
                          writeln;
                          textcolor(white);
                      end;
          end;
     textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
     textcolor(white);
     writeln();
{R_Phone_no done}
     assign(Inf_client,'records.txt');
     append(Inf_client);

write(Inf_client,R_username,R_Password,R_Sex,R_SurName,R_First
Name,R_Phone_no);
     writeln(Inf_client);
     close(Inf_client);
     assign(f,'records_booked.txt');
     append(f);
     write(f,R_username,'0');
```

```
     writeln(f);
     close(f);
     writeln('The registration is done!');
     writeln();
     writeln('Press the ENTER button to return to home page.');
     readln();
     clrscr;
end;{procedure done}

procedure user_page(users:integer);
var mae:string;
begin
    Read_ClientRecord;
    Read_RoomRecord;
    Read_Booked;

writeln('-------------------------------------------------------
----------------------------');
    if (h<12) then mae:='Good Morning!';
    if (h<18) and (h>=12) then mae:='Good Afternoon!';
    if (h>=18) then mae:='Good Evening!';
    if client[users].sex='M'
       then writeln(mae,' Mr.',client[users].surname)
       else writeln(mae,' Ms.',client[users].surname);
    writeln();

writeln('-------------------------------------------------------
----------------------------');
    writeln('':16,'What can we do for you?');
    writeln;
    writeln('':20,'1.Amend personal data');
    writeln;
    writeln('':20,'2.Make booking');
    writeln;
    writeln('':20,'3.Cancel booking');
    writeln;
    writeln('':20,'4.Logout');
    writeln;
```

```
writeln('-------------------------------------------------------
---------------------------');
    textcolor(white);
    write('':16,'Enter your choice:');
    readln(u_choice);
    case u_choice of
        1:amend_UserRecord(users);
        2:make_booking(users);
        3:cancel_booking(users);
    end;
    if u_choice<>4 then begin clrscr;user_page(users);end;
    {if u_choice=4
      then begin
          clrscr;
          homepage;
      end;}
end;
{procedure user_page done}

procedure return_or_not;
var escape:char;
begin
    clrscr;
    textcolor(11);
    writeln;writeln;writeln;writeln;writeln;writeln;writeln;
    writeln('You are entering the page of registration...');
    writeln();writeln;
    writeln('Press R to return to main menu, press other buttons
to continue.');
    delay(1000);
    textcolor(white);
    escape:=Readkey;
    if escape <> 'r'
    then register_client;
end;

procedure login;
```

```
var  log_id,log_pd:string;
    corr,exit1:boolean;
    i,j,keyword:integer;
begin
    clrscr;
    corr:=false;
    exit1:=false;
    while (corr=false) and (exit1=false) do
         begin
             clrscr;

writeln();writeln();writeln();writeln();writeln();
             write('Username: ':26);
             readln(log_id);
             writeln();
             writeln();
             write('Password(input EXIT to return to homepage):
':56);
             readln(log_pd);
             writeln();
             if log_pd='EXIT'
               then
                  begin
                      clrscr;
                      exit1:=true;
                  end
               else begin
                       for j:=1 to (12-length(log_id)) do
                                  log_id:=log_id+' ';
                       for j:=1 to (20-length(log_pd)) do
                                  log_pd:=log_pd+' ';
                       for i:=1 to 10000 do
                          if log_id=client[i].username
                            then if log_pd=client[i].password
                                 then begin
                                      corr:= true;
                                      keyword:=i;
                                      end;
```

```
                        if corr=false
                           then begin
                               textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
                               textcolor(lightred);
                               writeln('
Invalid username or password!');
                               sound(1);
                               writeln();
                               textcolor(white);
                               readln();
                               end;
                    end;
         END;
     if exit1=false then begin
     textcolor(yellow);

writeln('-------------------------------------------------------
---------------------------');
     textcolor(white);
     writeln('              Press the ENTER button to get into the
user page.');
     readln();
     clrscr;
     user_page(keyword);      end;
end;{procedure done}


begin
    moment;
    Read_Booked;
    Read_RoomRecord;
```

```
    Read_ClientRecord;
    textcolor(yellow);
    writeln();writeln();writeln();

writeln('#######################################################
##########################');delay(20);

writeln('#######################################################
##########################');delay(20);

writeln('#######################################################
##########################');delay(20);
    writeln('##############  ##################  ######
########  ######  ###############');delay(20);
    writeln('##############  #################  ######  ##
#######  #####  ################');delay(20);
    writeln('##############  #################  #####  ####
######  ####  #################');delay(20);
    writeln('###############  ######  ######  ######  ####
######  ##  ##################');delay(20);
    writeln('###############  ######  ######  #####  ######
#####   ######################');delay(20);
    writeln('###############  ######  ######  #####  ######
#####   ######################');delay(20);
    writeln('################  ####  #  ####  #####
####  #  ####################');delay(20);
    writeln('################  ###  ###  ###  #####  ########
####  ##  ###################');delay(20);
    writeln('################  ##  #####  ##  ####  ##########
###  ####  #################');delay(20);
    writeln('#################  #########  #####  ##########
###  ######  ###  #########');delay(20);

writeln('#######################################################
##########################');delay(20);

writeln('#######################################################
##########################');delay(20);
```

```
writeln('#####################################################
##########################');delay(100);
    writeln();writeln();delay(600);sound(1);

textcolor(white);write('-');delay(5);write('-');delay(5);write
('-');delay(5);write('-');delay(5);write('-');delay(5);write('
-');delay(5);write('-');delay(5);write('-');delay(5);write('-'
);delay(5);write('-');delay(5);write('-');delay(5);write('-');
delay(5);write('-');delay(5);write('-');delay(5);write('-');de
lay(5);write('-');delay(5);write('-');delay(5);write('-');dela
y(5);write('-');delay(5);write('-');delay(5);write('-');delay(
5);write('-');delay(5);write('-');delay(5);write('-');delay(5)
;write('-');delay(5);write('-');delay(5);
    write(' Press enter to continue ');delay(10);

write('-');delay(5);write('-');delay(5);write('-');delay(5);wr
ite('-');delay(5);write('-');delay(5);write('-');delay(5);writ
e('-');delay(5);write('-');delay(5);write('-');delay(5);write(
'-');delay(5);write('-');delay(5);write('-');delay(5);write('-
');delay(5);write('-');delay(5);write('-');delay(5);write('-')
;delay(5);write('-');delay(5);write('-');delay(5);write('-');d
elay(5);write('-');delay(5);write('-');delay(5);write('-');del
ay(5);write('-');delay(5);write('-');delay(5);write('-');delay
(5);write('-');delay(5);write('-');delay(5);writeln('>');delay
(5);
    readln();
    repeat
    homepage;
    case choice of
    1:return_or_not;
    2:login;
    end;
    until choice=3;
    readln;
end.}
```

Appendix 2 - Working schedule

Date	Event
April-2015	Choice of Topic
May-2015	Background research + Define the objectives + Propose functions
June-2015	Design of solution
Oct-2015	Implementation
Nov-2015	Testing & Evaluation
Dec-2015	Conclusion & Discussion + Final Report