UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

BACHELOR OF COMPUTER SCIENCE (HONOURS)

UCCD1004 PROGRAMMING CONCEPTS AND PRATICES

Name	Lee Jian Zhen
Student ID	19ACB02281
Email Address	jianzhen@1utar.my

No.	Content.	Page.
1	Objective	2
2	Pseudocode	2
3	Flowchart	3 - 4
4	Test Case	5 - 10
5	Source Code	11 - 29

Objective

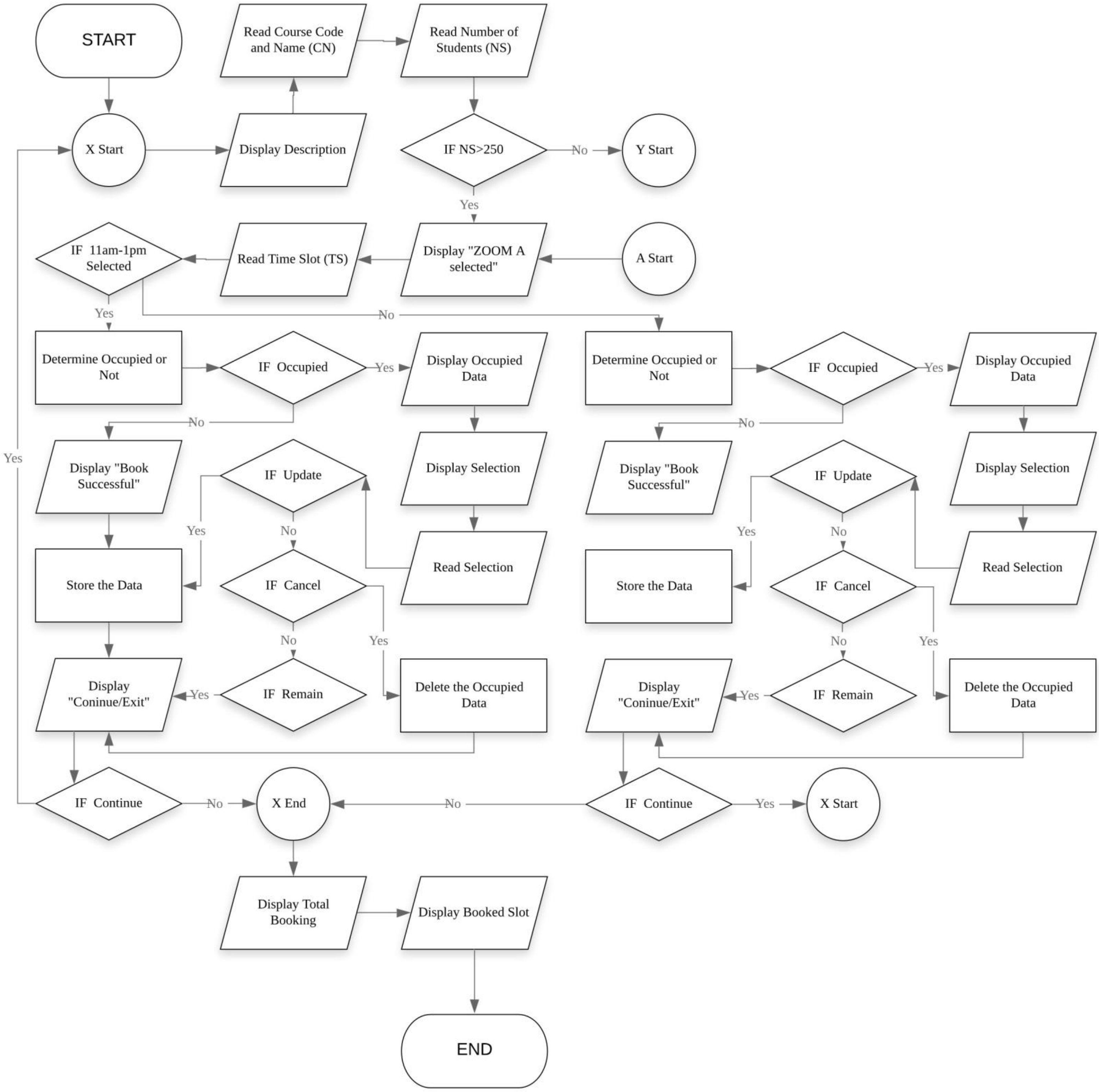
Write a C++ program (without using any library functions other than iostream, string and iomanip) which enable an admin to book a ZOOM account (on behalf of lecturers) for a particular course to conduct online classes.

Pseudocode

Declare all variable with suitable data types

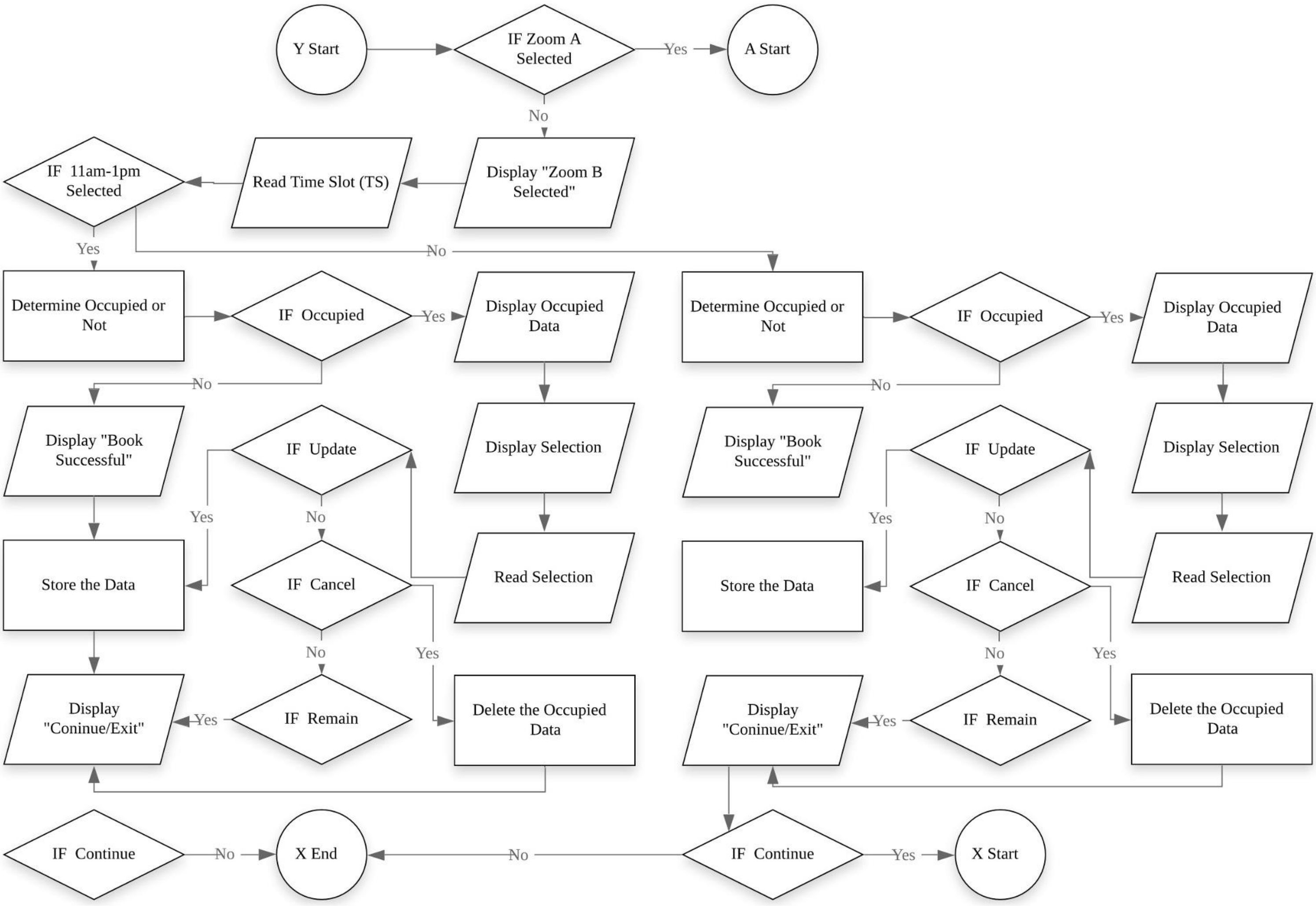
```
REPEAT
_INPUT course code and name, number of students.
___IF number of students more than 250.
_____INPUT time slot.
_____IF particular time selected.
_______IF occupied.
_________Display course code and name.
_________Ask for update/cancel/remain.
_________IF update.
___________Replace the occupied data with new data.
_________ELSE IF cancel.
___________Delete the occupied data.
_________ELSE IF remain.
___________Remain the data.
_______ELSE book successful.
___ELSE number of students less than or equal to 250.
_____INPUT zoom slot.
_____IF particular zoom selected.
_______INPUT time slot.
_______IF particular time selected.
_________IF occupied.
___________Display course code and name.
___________Ask for update/cancel/remain.
___________IF update.
_____________Replace the occupied data with new data.
___________ELSE IF cancel.
_____________Delete the occupied data.
___________ELSE IF remain.
_____________Remain the data.
_________ELSE book successful.
Display Total Booking and Booked slot.
```

Flowchart



START
X Start
Display Description
Read Course Code and Name (CN)
Read Number of Students (NS)
IF NS>250
No
Y Start
Yes
A Start
Display "ZOOM A selected"
Read Time Slot (TS)
IF 11am-1pm Selected
Yes
No
Determine Occupied or Not
IF Occupied
Yes
Display Occupied Data
No
Display "Book Successful"
IF Update
Display Selection
Yes
No
Read Selection
Store the Data
IF Cancel
No
Yes
Display "Coninue/Exit"
Yes
IF Remain
Delete the Occupied Data
IF Continue
No
X End
Yes
Determine Occupied or Not
IF Occupied
Yes
Display Occupied Data
No
Display "Book Successful"
IF Update
Display Selection
Yes
No
Store the Data
IF Cancel
Read Selection
No
Yes
Display "Coninue/Exit"
Yes
IF Remain
Delete the Occupied Data
No
IF Continue
Yes
X Start
Display Total Booking
Display Booked Slot
END

Flowchart continues



Test Case

Course code and name:

1. UCCD1004 Programming Concepts and Practices
2. UCCD1143 Probability and Statistics for Computing
3. UCCN1004 Data Communications and Networking
4. UCCM1153 Introduction to Calculus and Applications

Zoom account:

1. Zoom A
2. Zoom B

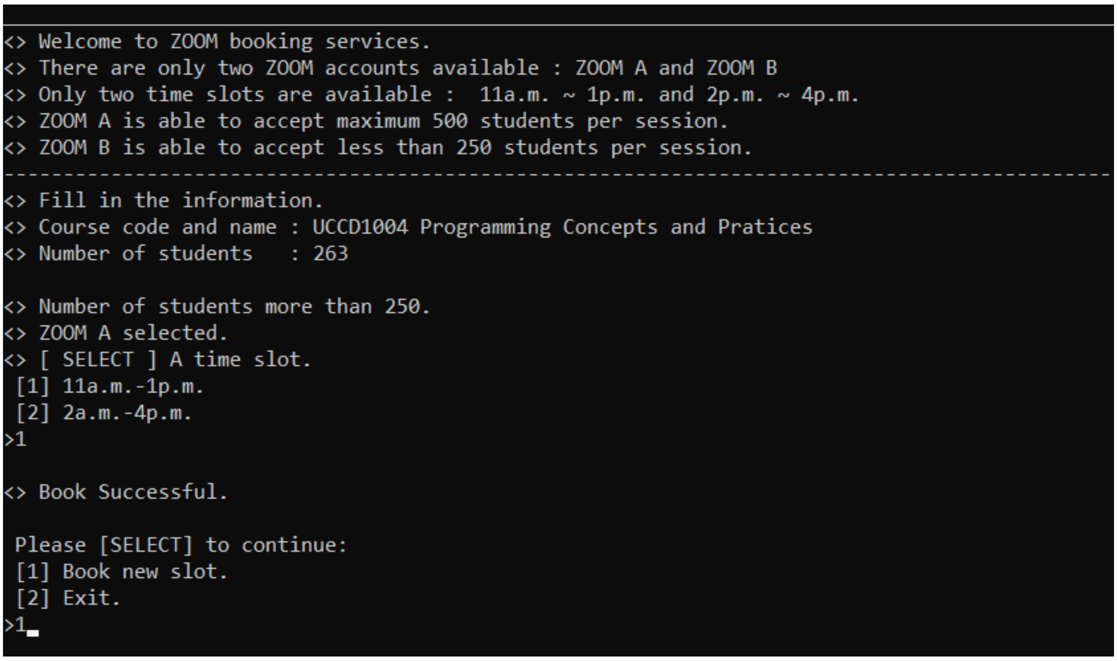
Time slot:

1. 11a.m. ~ 1p.m.
2. 2a.m. ~ 4p.m.

No.	Test Case	Page.
1	Identify booked account	6 - 7
2	Update and Cancel the Zoom account	8
3	Total bookings	10
4	Input validation	10

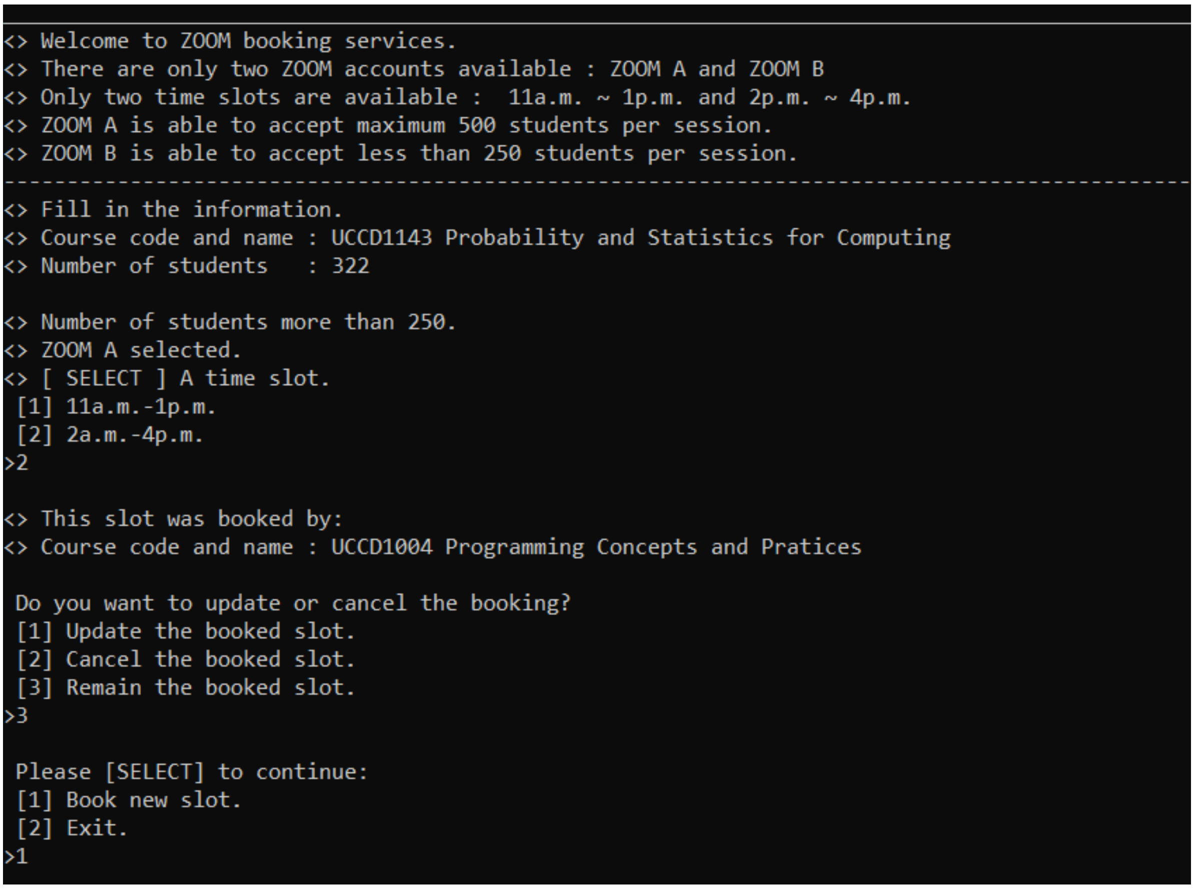
<u>Test Case 1</u>: Identify the booked Zoom account

First, I insert the course code and name **(UCCD1004 Programming Concepts and Practices)** and input the **number of students (263)**. After that, the system will **select Zoom A automatically** because of number of students more than 250 and we select the time **(11a.m. ~ 1p.m.)** and it show success.



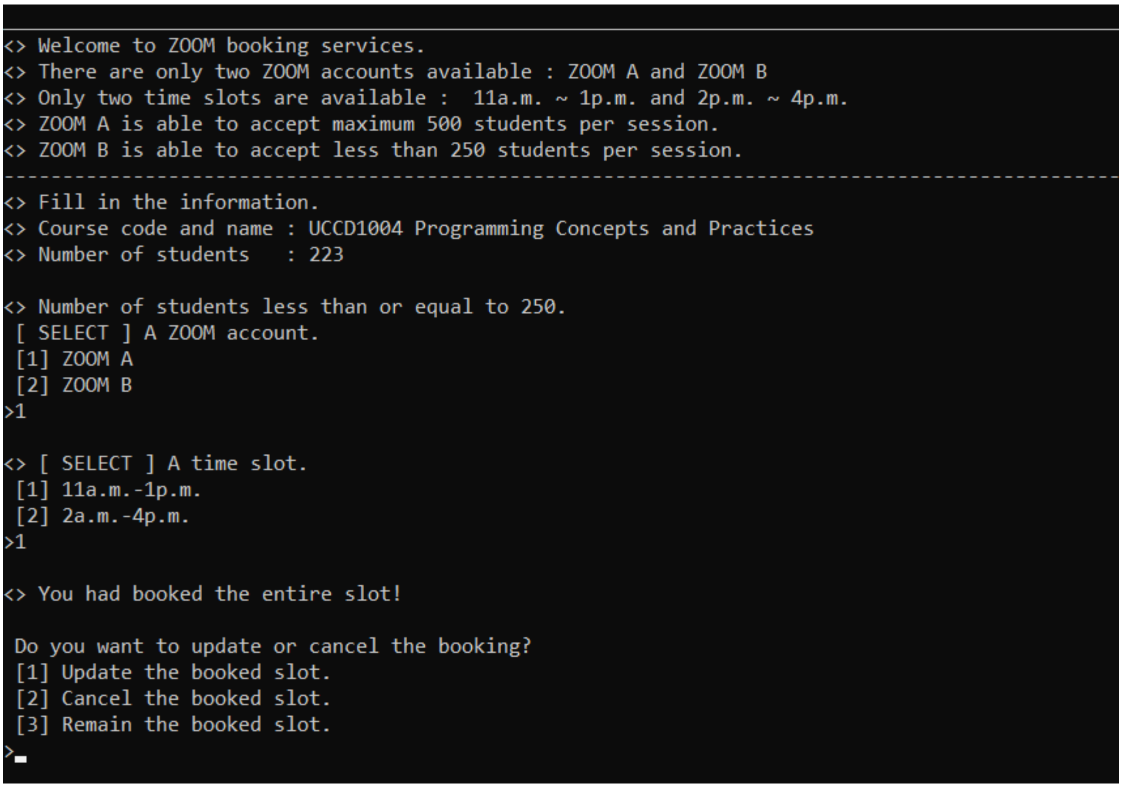
```
<> Welcome to ZOOM booking services.
<> There are only two ZOOM accounts available : ZOOM A and ZOOM B
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
-----------------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCD1004 Programming Concepts and Pratices
<> Number of students   : 263

<> Number of students more than 250.
<> ZOOM A selected.
<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>1

<> Book Successful.

 Please [SELECT] to continue:
 [1] Book new slot.
 [2] Exit.
>1
```

I select Book new slot for continue booking, then I insert the second course code and name **(UCCD1143 Probability and Stastistics for Computing)**, number of **students (322)** which number of students more than 250 **(Zoom A selected)** and select the **time (2p.m. ~ 4p.m.)**. As we know that we had booked an account for <u>UCCD1004 Programming Concepts and Practices which was Zoom A and 11a.m. ~ 1p.m.</u> As a result, the Zoom account was occupied and the system will show the occupied account details.



```
<> Welcome to ZOOM booking services.
<> There are only two ZOOM accounts available : ZOOM A and ZOOM B
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
-----------------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCD1143 Probability and Statistics for Computing
<> Number of students   : 322

<> Number of students more than 250.
<> ZOOM A selected.
<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>2

<> This slot was booked by:
<> Course code and name : UCCD1004 Programming Concepts and Pratices

 Do you want to update or cancel the booking?
 [1] Update the booked slot.
 [2] Cancel the booked slot.
 [3] Remain the booked slot.
>3

 Please [SELECT] to continue:
 [1] Book new slot.
 [2] Exit.
>1
```

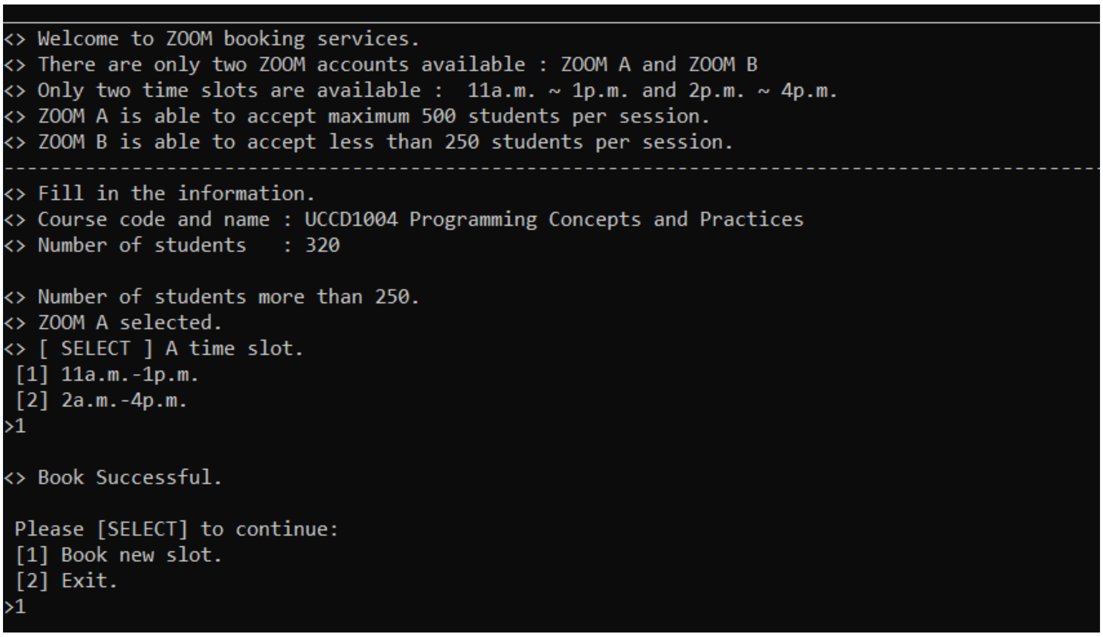
Afterward, I insert again the course code and name **(UCCD1004 Programming Concepts and Practices)** and number of **students (233**). Since the number of students less than 250, we can select both of the Zoom account so I select **Zoom A and 11a.m. ~ 1p.m**. Just now, I had booked an account which was the same course code and name so that the system now shows us that we had book the entire slot.



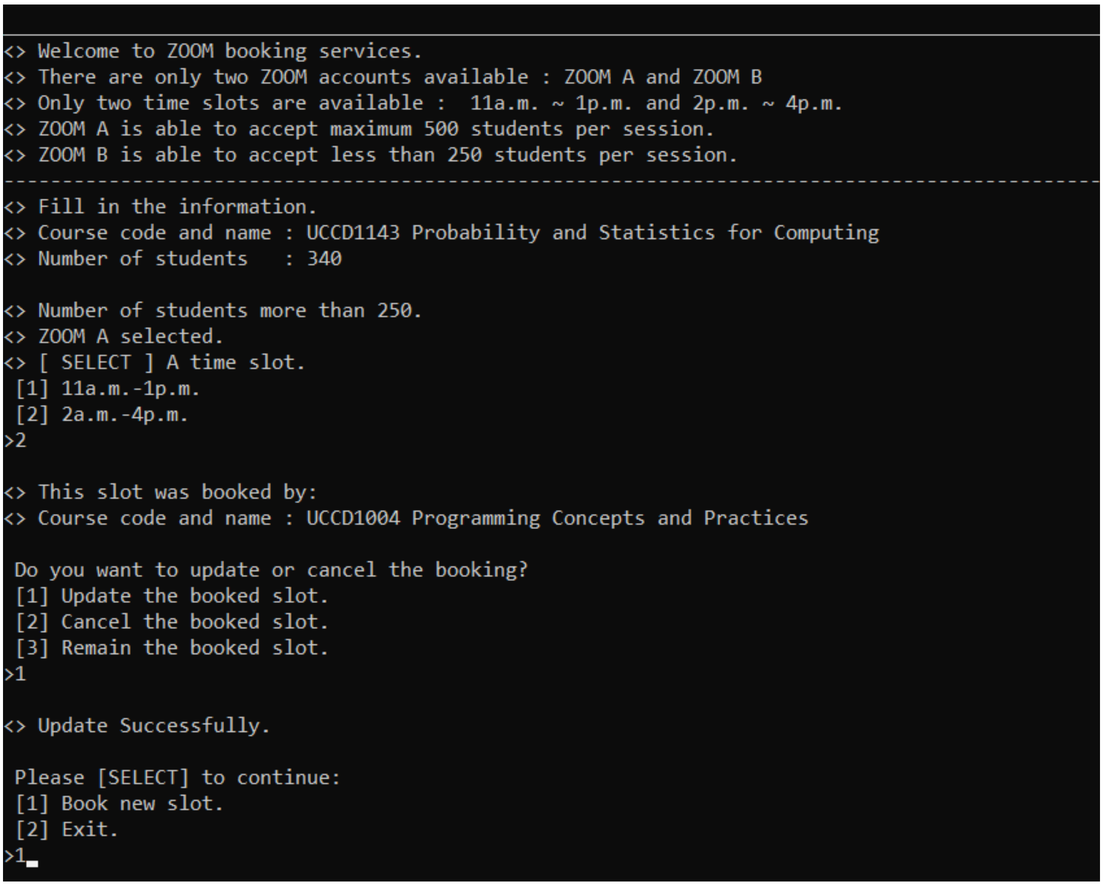
```
<> Welcome to ZOOM booking services.
<> There are only two ZOOM accounts available : ZOOM A and ZOOM B
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
-----------------------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCD1004 Programming Concepts and Practices
<> Number of students   : 223

<> Number of students less than or equal to 250.
 [ SELECT ] A ZOOM account.
 [1] ZOOM A
 [2] ZOOM B
>1

<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>1

<> You had booked the entire slot!

 Do you want to update or cancel the booking?
 [1] Update the booked slot.
 [2] Cancel the booked slot.
 [3] Remain the booked slot.
>
```

Test Case 2: Update and Cancel the Zoom account

First, we book a Zoom account with course code and name **(UCCD1004 Programming Concepts and Practices), number of students (320), Zoom A, and time 11a.m. ~ 1p.m**. Then, we book another account with course code and name **(UCCD1143 Probability and Statistics for Computing), total students (340), Zoom A, and time 2p.m. ~ 4p.m**. It will show occupied and we try to update the first account to second account.

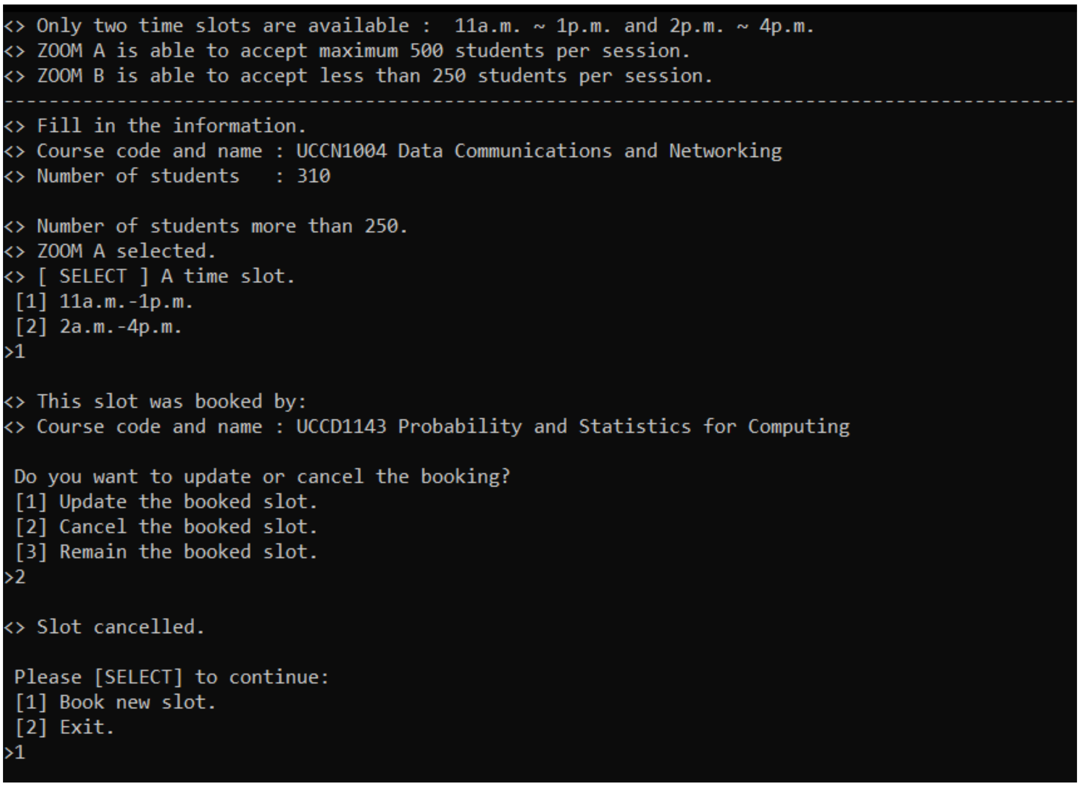


```
<> Welcome to ZOOM booking services.
<> There are only two ZOOM accounts available : ZOOM A and ZOOM B
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
----------------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCD1004 Programming Concepts and Practices
<> Number of students   : 320

<> Number of students more than 250.
<> ZOOM A selected.
<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>1

<> Book Successful.

 Please [SELECT] to continue:
 [1] Book new slot.
 [2] Exit.
>1
```

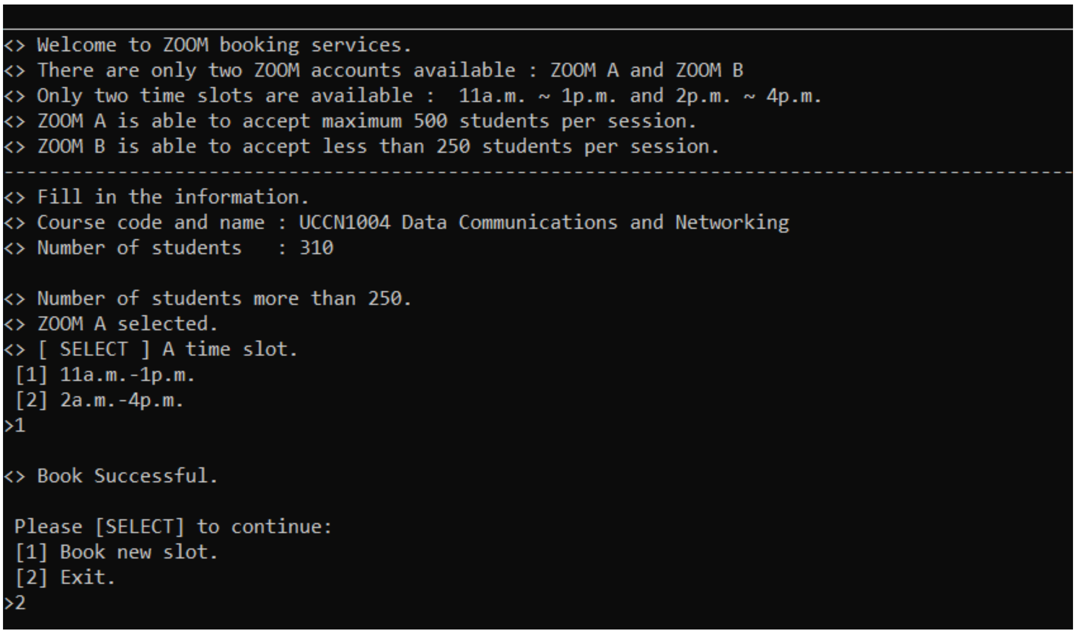


```
<> Welcome to ZOOM booking services.
<> There are only two ZOOM accounts available : ZOOM A and ZOOM B
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
----------------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCD1143 Probability and Statistics for Computing
<> Number of students   : 340

<> Number of students more than 250.
<> ZOOM A selected.
<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>2

<> This slot was booked by:
<> Course code and name : UCCD1004 Programming Concepts and Practices

 Do you want to update or cancel the booking?
 [1] Update the booked slot.
 [2] Cancel the booked slot.
 [3] Remain the booked slot.
>1

<> Update Successfully.

 Please [SELECT] to continue:
 [1] Book new slot.
 [2] Exit.
>1
```

Next, we book an zoom account with course code and name **(UCCN1004 Data Communications and Networking), total stutends (310), Zoom A, and time 11a.m. ~ 1p.m.** Then the system will show the occupied details which is the second zoom account we booked. We try to cancel the occupied account and test whether we can book the same account or not.



```
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
-----------------------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCN1004 Data Communications and Networking
<> Number of students   : 310

<> Number of students more than 250.
<> ZOOM A selected.
<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>1

<> This slot was booked by:
<> Course code and name : UCCD1143 Probability and Statistics for Computing

 Do you want to update or cancel the booking?
 [1] Update the booked slot.
 [2] Cancel the booked slot.
 [3] Remain the booked slot.
>2

<> Slot cancelled.

 Please [SELECT] to continue:
 [1] Book new slot.
 [2] Exit.
>1
```

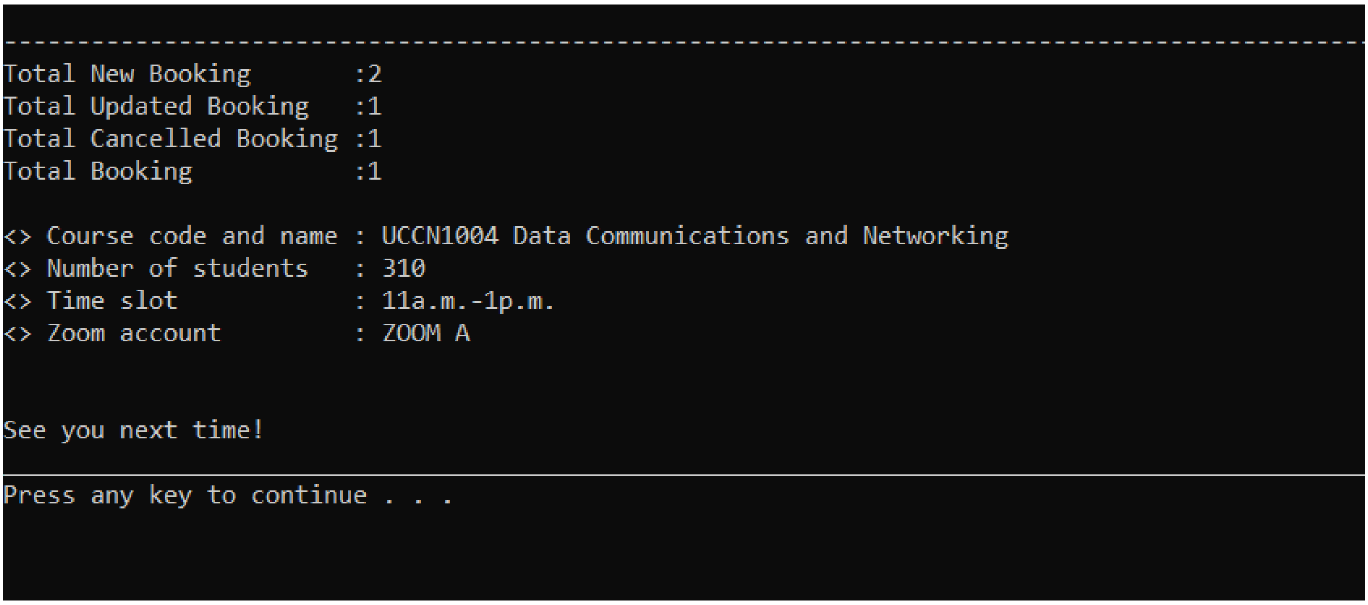


```
<> Welcome to ZOOM booking services.
<> There are only two ZOOM accounts available : ZOOM A and ZOOM B
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
-----------------------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCN1004 Data Communications and Networking
<> Number of students   : 310

<> Number of students more than 250.
<> ZOOM A selected.
<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>1

<> Book Successful.

 Please [SELECT] to continue:
 [1] Book new slot.
 [2] Exit.
>2
```

At the end, we are able to book the account that we cancelled just now.

Test Case 3: Total bookings

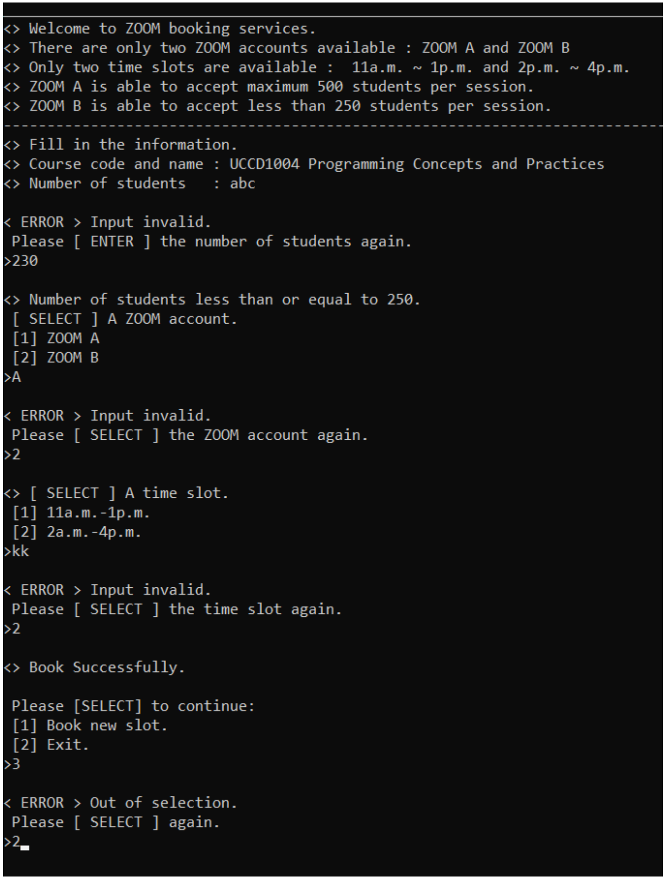
In test case 2, we had book 2 empty zoom account successfully. At the same time, we also test the update and cancel function each once. At the end, we had just booked one account since we cancelled one account just now.



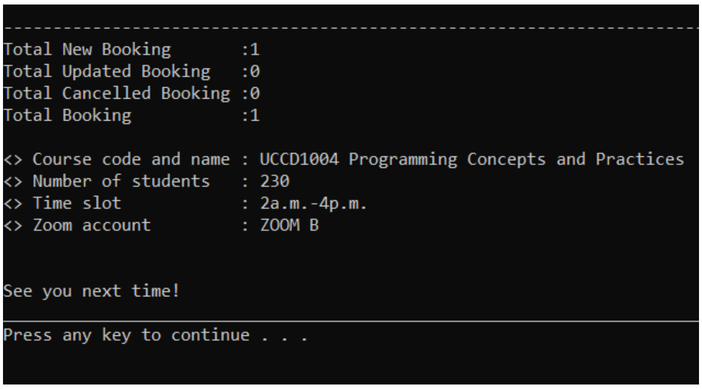
```
--------------------------------------------------------------------------------
Total New Booking       :2
Total Updated Booking   :1
Total Cancelled Booking :1
Total Booking           :1

<> Course code and name : UCCN1004 Data Communications and Networking
<> Number of students   : 310
<> Time slot            : 11a.m.-1p.m.
<> Zoom account         : ZOOM A


See you next time!
Press any key to continue . . .
```

Test Case 4: Input validations

All of the input will only accept integers instead of character or string.



```
<> Welcome to ZOOM booking services.
<> There are only two ZOOM accounts available : ZOOM A and ZOOM B
<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.
<> ZOOM A is able to accept maximum 500 students per session.
<> ZOOM B is able to accept less than 250 students per session.
--------------------------------------------------------------------------------
<> Fill in the information.
<> Course code and name : UCCD1004 Programming Concepts and Practices
<> Number of students   : abc

< ERROR > Input invalid.
 Please [ ENTER ] the number of students again.
>230

<> Number of students less than or equal to 250.
 [ SELECT ] A ZOOM account.
 [1] ZOOM A
 [2] ZOOM B
>A

< ERROR > Input invalid.
 Please [ SELECT ] the ZOOM account again.
>2

<> [ SELECT ] A time slot.
 [1] 11a.m.-1p.m.
 [2] 2a.m.-4p.m.
>kk

< ERROR > Input invalid.
 Please [ SELECT ] the time slot again.
>2

<> Book Successfully.

 Please [SELECT] to continue:
 [1] Book new slot.
 [2] Exit.
>3

< ERROR > Out of selection.
 Please [ SELECT ] again.
>2
```



```
--------------------------------------------------------------------------------
Total New Booking       :1
Total Updated Booking   :0
Total Cancelled Booking :0
Total Booking           :1

<> Course code and name : UCCD1004 Programming Concepts and Practices
<> Number of students   : 230
<> Time slot            : 2a.m.-4p.m.
<> Zoom account         : ZOOM B


See you next time!
Press any key to continue . . .
```

Source Code

```
#include<iostream>
#include<iomanip>
#include<string>
using namespace std;
int main(void) {
        //Course Code and Name.
        string CN, CN1, CN2, CN3, CN4;
        CN = "", CN1 = "", CN2 = "", CN3 = "", CN4 = "";
        //Number of Students.
        int NS, NS1, NS2, NS3, NS4;
        NS = 0, NS1 = 0, NS2 = 0, NS3 = 0, NS4 = 0;
        //Time Slots.
        int TS = 0;
        string TSC, TSD;
        TSC = "11a.m.-1p.m.";
        TSD = "2a.m.-4p.m.";
        int C = 0, D = 0;
        //ZOOM Accounts.
        int ZA = 0;
        string ZAA = "ZOOM A";
        string ZAB = "ZOOM B";
        int A = 0, B = 0;
        //Booked account.
        int AC, AD, BC, BD;
        AC = 0, AD = 0, BC = 0, BD = 0;
        //Display.
        string Dcn = "<> Course code and name";
        string Dns = "<> Number of students";
        string Dts = "<> Time slot";
        string Dza = "<> Zoom account";
        //Count.
        int NB = 0, UB = 0, CB = 0, TB = 0;
        //Decisions.
        int select;
        bool valid = false, end_valid = false;
        do {
                cout << "\n______________________________________________________________________________________________________\n";
                cout << "<> Welcome to ZOOM booking services.\n";
                cout << "<> There are only two ZOOM accounts available : ZOOM A and ZOOM B\n";
                cout << "<> Only two time slots are available :  11a.m. ~ 1p.m. and 2p.m. ~ 4p.m.\n";
                cout << "<> ZOOM A is able to accept maximum 500 students per session.\n";
                cout << "<> ZOOM B is able to accept less than 250 students per session.\n";
                cout << "-----------------------------------------------------------------------------------------------------------\n";
                cout << "<> Fill in the information.\n";
                cout << setw(24) << left << Dcn << ": ";
                getline(cin, CN);
                cout << setw(24) << left << Dns << ": ";
                do {
                    if (cin >> NS) {
                        cin.ignore();
                        if (NS > 0 && NS < 501) {
                            valid = true;
                        }
                        else {
                            cout << " \n< ERROR > Number is out of limit.\n" << " Please [ ENTER ] the number of students again.\n" << ">";
                            continue;
                        }
                    }
                    else {
                        cin.clear();
                        cin.ignore(10000, '\n');
                        cout << " \n< ERROR > Input invalid.\n" << " Please [ ENTER ] the number of students again.\n" << ">";
                    }
                } while (!valid);
                valid = false;
                if (NS > 250) {
                    cout << "\n<> Number of students more than 250.\n" << "<> ZOOM A selected.\n";
                    cout << "<> [ SELECT ] A time slot.\n" << " [1] " << TSC << "\n [2] " << TSD << "\n>";
                    do {
                        if (cin >> TS) {
                            cin.ignore();
                            if (TS > 0 && TS < 3) {
                                valid = true;
                            }
                            else {
                                cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] the time slot again.\n" << ">";
                                continue;
                            }
                        }
                        else {
                            cin.clear();
                            cin.ignore(10000, '\n');
                            cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] the time slot again.\n" << ">";
                        }
                    } while (!valid);
                    valid = false;
                    if (TS == 1) {
                        if (A == 1 || C == 1) {
                            if (CN1 == CN) {
                                cout << "\n<> You had booked the entire slot! \n";
                            }
                            else if (AD == 1 && BC == 1) {
                                cout << "\n<> This slot was booked by: \n";
                                cout << setw(24) << left << Dcn << ": " << CN2 << endl;
                                cout << setw(24) << left << Dcn << ": " << CN3 << endl;
                            }
                            else if (AC == 1) {
                                cout << "\n<> This slot was booked by: \n";
                                cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                            }
                            else if (BC == 0) {
                                cout << "\n<> This slot was booked by: \n";
                                cout << setw(24) << left << Dcn << ": " << CN2 << endl;
                            }
                            else if (AD == 0) {
                                cout << "\n<> This slot was booked by: \n";
                                cout << setw(24) << left << Dcn << ": " << CN3 << endl;
                            }
                            cout << " \n Do you want to update or cancel the booking?\n" << " [1] Update the booked slot.\n" << " [2]
        Cancel the booked slot.\n"<<" [3] Remain the booked slot.\n" << ">";
                            do {
                                if (cin >> select) {
                                    cin.ignore();
                                    if (select > 0 && select < 4) {
                                        valid = true;
                                    }
                                    else {
                                        cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ]
        again.\n" << ">";
                                        continue;
                                    }
                                }
                                else {
                                    cin.clear();
                                    cin.ignore(10000, '\n');
                                    cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
                                }
                            } while (!valid);
```

```
        valid = false;
        if (select == 1) {
                if (AD == 1 && BC == 1) {
                        AD = 0, BC = 0, A = 0, B = 0, C = 0, D = 0;
                }
                else if (BC == 0) {
                        AD = 0, A = 0, D = 0;
                }
                else if (AD == 0) {
                }
                AC = 1, A = 1, C = 1;
                UB++;
                cout << "\n<> Update Successfully.\n";
                CN1 = CN, NS1 = NS;
        }
        else if (select == 2) {
                if (AD == 1 && BC == 1) {
                        AD = 0, BC = 0, A = 0, B = 0, C = 0, D = 0;
                        CN2 = "", CN3 = "";
                }
                else if (AC == 1) {
                        AC = 0, A = 0, C = 0;
                        CN1 = "";
                }
                else if (BC == 0) {
                        AD = 0, A = 0, D = 0;
                        CN2 = "";
                }
                else if (AD == 0) {
                        BC = 0, B = 0, C = 0;
                        CN3 = "";
                }
                CB++, TB--;
                cout << "\n<> Slot cancelled.\n";
        }
}
else {
        NB++, TB++;
        cout << "\n<> Book Successful.\n";
        AC = 1, A = 1, C = 1;
        CN1 = CN, NS1 = NS;
}
cout << "\n Please [SELECT] to continue:\n" << " [1] Book new slot.\n" << " [2] Exit.\n" << ">";
```

```
                do {
                    if (cin >> select) {
                        cin.ignore();
                        if (select > 0 && select < 3) {
                            valid = true;
                        }
                        else {
                            cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" <<
">";
                            continue;
                        }
                    }
                    else {
                        cin.clear();
                        cin.ignore(10000, '\n');
                        cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
                    }
                    if (select == 2) {
                        end_valid = true;
                    }
                } while (!valid);
                valid = false;
                system("cls");
            }
            else {
                if (A == 1 || D == 1) {
                    if (CN2 == CN) {
                        cout << "\n<> You had booked the entire slot! \n";
                    }
                    else if (AC == 1 && BD == 1) {
                        cout << "\n<> This slot was booked by: \n";
                        cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                        cout << setw(24) << left << Dcn << ": " << CN4 << endl;
                    }
                    else if (AD == 1) {
                        cout << "\n<> This slot was booked by: \n";
                        cout << setw(24) << left << Dcn << ": " << CN2 << endl;
                    }
                    else if (BD == 0) {
                        cout << "\n<> This slot was booked by: \n";
                        cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                    }
                    else if (AC == 0) {
                        cout << "\n<> This slot was booked by: \n";
                        cout << setw(24) << left << Dcn << ": " << CN4 << endl;
                    }
                    cout << " \n Do you want to update or cancel the booking?\n" << " [1] Update the booked slot.\n" << " [2] Cancel the booked slot.\n" << " [3] Remain the booked slot.\n" << ">";
                    do {
                        if (cin >> select) {
                            cin.ignore();
                            if (select > 0 && select < 4) {
                                valid = true;
                            }
                            else {
                                cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" << ">";
                                continue;
                            }
                        }
                        else {
                            cin.clear();
                            cin.ignore(10000, '\n');
                            cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
                        }
                    } while (!valid);
                    valid = false;
                    if (select == 1) {
                        if (AC == 1 && BD == 1) {
                            AC = 0, BD = 0, A = 0, C = 0, B = 0, D = 0;
                        }
                        else if (BD == 0) {
                            AC = 0, A = 0, C = 0;
                        }
                        else if (AC == 0) {
                            BD = 0, B = 0, D = 0;
                        }
                        AD = 1, A = 1, D = 1;
                        UB++;
                        cout << "\n<> Update Successfully.\n";
                        CN2 = CN, NS2 = NS;
                    }
                    else if (select == 2) {
                        if (AC == 1 && BD == 1) {
                            AC = 0, BD = 0, A = 0, C = 0, B = 0, D = 0;
                            CN1 = "", CN4 = "";
                        }
                        else if (AD == 1) {
                                AD = 0, A = 0, D = 0;
                                CN2 = "";
                            }
                            else if (BD == 0) {
                                AC = 0, A = 0, C = 0;
                                CN1 = "";
                            }
                            else if (AC == 0) {
                                BD = 0, B = 0, D = 0;
                                CN4 = "";
                            }
                            CB++, TB--;
                            cout << "\n<> Slot cancelled.\n";
                        }
                    }
                    else {
                        NB++, TB++;
                        cout << "\n<> Book Successfully.\n";
                        AD = 1, A = 1, D = 1;
                        CN2 = CN, NS2 = NS;
                    }
                    cout << "\n Please [SELECT] to continue:\n" << " [1] Book new slot.\n" << " [2] Exit.\n" << ">";
                    do {
                        if (cin >> select) {
                            cin.ignore();
                            if (select > 0 && select < 3) {
                                valid = true;
                            }
                            else {
                                cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" <<
">";
                                continue;
                            }
                        }
                        else {
                            cin.clear();
                            cin.ignore(10000, '\n');
                            cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
                        }
                        if (select == 2) {
                            end_valid = true;
                        }
                    } while (!valid);
                    valid = false;
```

```
                system("cls");
            }
        }
        else {
            cout << "\n<> Number of students less than or equal to 250.";
            cout << "\n [ SELECT ] A ZOOM account." << "\n [1] ZOOM A" << "\n [2] ZOOM B" << "\n>";
            do {
                if (cin >> ZA) {
                    cin.ignore();
                    if (ZA > 0 && ZA < 3) {
                        valid = true;
                    }
                    else {
                        cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] the ZOOM account
again.\n" << ">";
                        continue;
                    }
                }
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] the ZOOM account again.\n" << ">";
                }
            } while (!valid);
            valid = false;
            cout << "\n<> [ SELECT ] A time slot.\n" << " [1] " << TSC << "\n [2] " << TSD << "\n>";
            do {
                if (cin >> TS) {
                    cin.ignore();
                    if (TS > 0 && TS < 3) {
                        valid = true;
                    }
                    else {
                        cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] the time slot again.\n" <<
">";
                        continue;
                    }
                }
                else {
                    cin.clear();
                    cin.ignore(10000, '\n');
                    cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] the time slot again.\n" << ">";
                }
            } while (!valid);
            valid = false;
            if (ZA == 1) {
                if (TS == 1) {
                    if (A == 1 || C == 1) {
                        if (CN1 == CN) {
                            cout << "\n<> You had booked the entire slot! \n";
                        }
                        else if (AD == 1 && BC == 1) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN2 << endl;
                            cout << setw(24) << left << Dcn << ": " << CN3 << endl;
                        }
                        else if (AC == 1) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                        }
                        else if (BC == 0) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN2 << endl;
                        }
                        else if (AD == 0) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN3 << endl;
                        }
                        cout << " \n Do you want to update or cancel the booking?\n" << " [1] Update the booked
slot.\n" << " [2] Cancel the booked slot.\n" << " [3] Remain the booked slot.\n" << ">";
                        do {
                            if (cin >> select) {
                                cin.ignore();
                                if (select > 0 && select < 4) {
                                    valid = true;
                                }
                                else {
                                    cout << " \n< ERROR > Out of selection.\n" << " Please
[ SELECT ] again.\n" << ">";
                                    continue;
                                }
                            }
                            else {
                                cin.clear();
                                cin.ignore(10000, '\n');
                                cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ]
again.\n" << ">";
                            }
                        } while (!valid);
```

```
		valid = false;
		if (select == 1) {
			if (AD == 1 && BC == 1) {
				AD = 0, BC = 0, A = 0, B = 0, C = 0, D = 0;
			}
			else if (BC == 0) {
				AD = 0, A = 0, D = 0;
			}
			else if (AD == 0) {
				BC = 0, B = 0, C = 0;
			}
			AC = 1, A = 1, C = 1;
			UB++;
			cout << "\n<> Update Successfully.\n";
			CN1 = CN, NS1 = NS;
		}
		else if (select == 2) {
			if (AD == 1 && BC == 1) {
				AD = 0, BC = 0, A = 0, B = 0, C = 0, D = 0;
				CN2 = "", CN3 = "";
			}
			else if (AC == 1) {
				AC = 0, A = 0, C = 0;
				CN1 = "";
			}
			else if (BC == 0) {
				AD = 0, A = 0, D = 0;
				CN2 = "";
			}
			else if (AD == 0) {
				BC = 0, B = 0, C = 0;
				CN3 = "";
			}
			CB++, TB--;
			cout << "\n<> Slot cancelled.\n";
		}
	}
	else {
		NB++, TB++;
		cout << "\n<> Book Successful.\n";
		AC = 1, A = 1, C = 1;
		CN1 = CN, NS1 = NS;
	}
```

```
                    cout << "\n Please [SELECT] to continue:\n" << " [1] Book new slot.\n" << " [2] Exit.\n" << ">";
                    do {
                        if (cin >> select) {
                            cin.ignore();
                            if (select > 0 && select < 3) {
                                valid = true;
                            }
                            else {
                                cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ]
again.\n" << ">";
                                continue;
                            }
                        }
                        else {
                            cin.clear();
                            cin.ignore(10000, '\n');
                            cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
                        }
                        if (select == 2) {
                            end_valid = true;
                        }
                    } while (!valid);
                    valid = false;
                    system("cls");
                }
                else {
                    if (A == 1 || D == 1) {
                        if (CN2 == CN) {
                            cout << "\n<> You had booked the entire slot! \n";
                        }
                        else if (AC == 1 && BD == 1) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                            cout << setw(24) << left << Dcn << ": " << CN4 << endl;
                        }
                        else if (AD == 1) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN2 << endl;
                        }
                        else if (BD == 0) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                        }
                        else if (AC == 0) {
```

```
        cout << "\n<> This slot was booked by: \n";
        cout << setw(24) << left << Dcn << ": " << CN4 << endl;
    }
    cout << " \n Do you want to update or cancel the booking?\n" << " [1] Update the booked slot.\n" << " [2] Cancel the booked slot.\n" << " [3] Remain the booked slot.\n" << ">";
    do {
        if (cin >> select) {
            cin.ignore();
            if (select > 0 && select < 4) {
                valid = true;
            }
            else {
                cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" << ">";
                continue;
            }
        }
        else {
            cin.clear();
            cin.ignore(10000, '\n');
            cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
        }
    } while (!valid);
    valid = false;
    if (select == 1) {
        if (AC == 1 && BD == 1) {
            AC = 0, BD = 0, A = 0, C = 0, B = 0, D = 0;
        }
        else if (BD == 0) {
            AC = 0, A = 0, C = 0;
        }
        else if (AC == 0) {
            BD = 0, B = 0, D = 0;
        }
        AD = 1, A = 1, D = 1;
        UB++;
        cout << "\n<> Update Successfully.\n";
        CN2 = CN, NS2 = NS;
    }
    else if (select == 2) {
        if (AC == 1 && BD == 1) {
            AC = 0, BD = 0, A = 0, C = 0, B = 0, D = 0;
            CN1 = "", CN4 = "";
        }
        else if (AD == 1) {
            AD = 0, A = 0, D = 0;
            CN2 = "";
        }
        else if (BD == 0) {
            AC = 0, A = 0, C = 0;
            CN1 = "";
        }
        else if (AC == 0) {
            BD = 0, B = 0, D = 0;
            CN4 = "";
        }
        CB++, TB--;
        cout << "\n<> Slot cancelled.\n";
    }
}
else {
    NB++, TB++;
    cout << "\n<> Book Successfully.\n";
    AD = 1, A = 1, D = 1;
    CN2 = CN, NS2 = NS;
}
cout << "\n Please [SELECT] to continue:\n" << " [1] Book new slot.\n" << " [2] Exit.\n" << ">";
do {
    if (cin >> select) {
        cin.ignore();
        if (select > 0 && select < 3) {
            valid = true;
        }
        else {
            cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" << ">";
            continue;
        }
    }
    else {
        cin.clear();
        cin.ignore(10000, '\n');
        cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
    }
    if (select == 2) {
        end_valid = true;
    }
} while (!valid);
```

```
                    valid = false;
                    system("cls");
                }
            }
            else {
                if (TS == 1) {
                    if (B == 1 || C == 1) {
                        if (CN3 == CN) {
                            cout << "\n<> You had booked the entire slot! \n";
                        }
                        else if (BD == 1 && AC == 1) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN4 << endl;
                            cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                        }
                        else if (BC == 1) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN3 << endl;
                        }
                        else if (AC == 0) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN4 << endl;
                        }
                        else if (BD == 0) {
                            cout << "\n<> This slot was booked by: \n";
                            cout << setw(24) << left << Dcn << ": " << CN1 << endl;
                        }
                        cout << " \n Do you want to update or cancel the booking?\n" << " [1] Update the booked slot.\n" << " [2] Cancel the booked slot.\n" << " [3] Remain the booked slot.\n" << ">";
                        do {
                            if (cin >> select) {
                                cin.ignore();
                                if (select > 0 && select < 4) {
                                    valid = true;
                                }
                                else {
                                    cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" << ">";
                                    continue;
                                }
                            }
                            else {
                                cin.clear();
                                cin.ignore(10000, '\n');
                                cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ]
again.\n" << ">";
                            }
                        } while (!valid);
                        valid = false;
                        if (select == 1) {
                            if (BD == 1 && AC == 1) {
                                BD = 0, AC = 0, A = 0, B = 0, C = 0, D = 0;
                            }
                            else if (BD == 0) {
                                AC = 0, A = 0, C = 0;
                            }
                            else if (AC == 0) {
                                BD = 0, B = 0, D = 0;
                            }
                            BC = 1, A = 1, C = 1;
                            UB++;
                            cout << "\n<> Update Successfully.\n";
                            CN3 = CN, NS3 = NS;
                        }
                        else if (select == 2) {
                            if (BD == 1 && AC == 1) {
                                BD = 0, AC = 0, A = 0, B = 0, C = 0, D = 0;
                                CN4 = "", CN1 = "";
                            }
                            else if (BC == 1) {
                                BC = 0, B = 0, C = 0;
                                CN3 = "";
                            }
                            else if (BD == 0) {
                                AC = 0, A = 0, C = 0;
                                CN1 = "";
                            }
                            else if (AC == 0) {
                                BD = 0, B = 0, D = 0;
                                CN4 = "";
                            }
                            CB++, TB--;
                            cout << "\n<> Slot cancelled.\n";
                        }
                    }
                    else {
                        NB++, TB++;
                        cout << "\n<> Book Successful.\n";
```

```
        BC = 1, B = 1, C = 1;
        CN3 = CN, NS3 = NS;
    }
    cout << "\n Please [SELECT] to continue:\n" << " [1] Book new slot.\n" << " [2] Exit.\n" << ">";
    do {
        if (cin >> select) {
            cin.ignore();
            if (select > 0 && select < 3) {
                valid = true;
            }
            else {
                cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" << ">";
                continue;
            }
        }
        else {
            cin.clear();
            cin.ignore(10000, '\n');
            cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
        }
        if (select == 2) {
            end_valid = true;
        }
    } while (!valid);
    valid = false;
    system("cls");
}
else {
    if (B == 1 || D == 1) {
        if (CN4 == CN) {
            cout << "\n<> You had booked the entire slot! \n";
        }
        else if (AD == 1 && BC == 1) {
            cout << "\n<> This slot was booked by: \n";
            cout << setw(24) << left << Dcn << ": " << CN2 << endl;
            cout << setw(24) << left << Dcn << ": " << CN3 << endl;
        }
        else if (BD == 1) {
            cout << "\n<> This slot was booked by: \n";
            cout << setw(24) << left << Dcn << ": " << CN4 << endl;
        }
        else if (BC == 0) {
            cout << "\n<> This slot was booked by: \n";
            cout << setw(24) << left << Dcn << ": " << CN2 << endl;
        }
        else if (AD == 0) {
            cout << "\n<> This slot was booked by: \n";
            cout << setw(24) << left << Dcn << ": " << CN3 << endl;
        }
        cout << " \n Do you want to update or cancel the booking?\n" << " [1] Update the booked slot.\n" << " [2] Cancel the booked slot.\n" << " [3] Remain the booked slot.\n" << ">";
        do {
            if (cin >> select) {
                cin.ignore();
                if (select > 0 && select < 4) {
                    valid = true;
                }
                else {
                    cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" << ">";
                    continue;
                }
            }
            else {
                cin.clear();
                cin.ignore(10000, '\n');
                cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
            }
        } while (!valid);
        valid = false;
        if (select == 1) {
            if (AD == 1 && BC == 1) {
                AD = 0, BC = 0, A = 0, C = 0, B = 0, D = 0;
            }
            else if (AD == 0) {
                BC = 0, B = 0, C = 0;
            }
            else if (BC == 0) {
                AD = 0, A = 0, D = 0;
            }
            AD = 1, A = 1, D = 1;
            UB++;
            cout << "\n<> Update Successfully.\n";
            CN4 = CN, NS4 = NS;
        }
        else if (select == 2) {
            if (AD == 1 && BC == 1) {
```

```
			AD = 0, BC = 0, A = 0, C = 0, B = 0, D = 0;
			CN2 = "", CN3 = "";
		}
		else if (BD == 1) {
			BD = 0, B = 0, D = 0;
			CN4 = "";
		}
		else if (BC == 0) {
			BC = 0, B = 0, C = 0;
			CN3 = "";
		}
		else if (AD == 0) {
			AD = 0, A = 0, D = 0;
			CN2 = "";
		}
		CB++, TB--;
		cout << "\n<> Slot cancelled.\n";
	}
}
else {
	NB++, TB++;
	cout << "\n<> Book Successfully.\n";
	BD = 1, B = 1, D = 1;
	CN4 = CN, NS4 = NS;
}
cout << "\n Please [SELECT] to continue:\n" << " [1] Book new slot.\n" << " [2] Exit.\n" << ">";
do {
	if (cin >> select) {
		cin.ignore();
		if (select > 0 && select < 3) {
			valid = true;
		}
		else {
			cout << " \n< ERROR > Out of selection.\n" << " Please [ SELECT ] again.\n" << ">";
			continue;
		}
	}
	else {
		cin.clear();
		cin.ignore(10000, '\n');
		cout << " \n< ERROR > Input invalid.\n" << " Please [ SELECT ] again.\n" << ">";
	}
	if (select == 2) {
```

```
                            end_valid = true;
                        }
                    } while (!valid);
                    valid = false;
                    system("cls");
                }
            }
        }
    } while (!(end_valid));
    cout << "\n\n----------------------------------------------------------------------------------------------------\n";
    cout << "Total New Booking       :" << NB << endl;
    cout << "Total Updated Booking   :" << UB << endl;
    cout << "Total Cancelled Booking :" << CB << endl;
    cout << "Total Booking           :" << TB << endl << endl;
    if (AC == 1) {
        cout << setw(24) << left << Dcn << ": " << CN1 << endl;
        cout << setw(24) << left << Dns << ": " << NS1 << endl;
        cout << setw(24) << left << Dts << ": " << TSC << endl;
        cout << setw(24) << left << Dza << ": " << ZAA << endl << endl;
    }
    if (AD == 1) {
        cout << setw(24) << left << Dcn << ": " << CN2 << endl;
        cout << setw(24) << left << Dns << ": " << NS2 << endl;
        cout << setw(24) << left << Dts << ": " << TSD << endl;
        cout << setw(24) << left << Dza << ": " << ZAA << endl << endl;
    }
    if (BC == 1) {
        cout << setw(24) << left << Dcn << ": " << CN3 << endl;
        cout << setw(24) << left << Dns << ": " << NS3 << endl;
        cout << setw(24) << left << Dts << ": " << TSC << endl;
        cout << setw(24) << left << Dza << ": " << ZAB << endl << endl;
    }
    if (BD == 1) {
        cout << setw(24) << left << Dcn << ": " << CN4 << endl;
        cout << setw(24) << left << Dns << ": " << NS4 << endl;
        cout << setw(24) << left << Dts << ": " << TSD << endl;
        cout << setw(24) << left << Dza << ": " << ZAB << endl << endl;
    }
    cout << "\nSee you next time!";
    cout << "\n______________________________________________________________________________________________\n";
    std::system("pause");
    return 0;
}
```