



OOP COMPLEX ENGINEERING ACTIVITY.

PROJECT REPORT.



SUBMITTED TO: SIR UZAIR

GROUP MEMBERS:

Name	Roll Number
AHMED KHALID	200148
BISMA SHAKEEL	200214
AQIB ALI	200210

SEMSTER/ DEPARTMENT: BEET-III-A

DATE: 26/12/2021

AIRTEX AIRLINE BOOKING SYSTEM

Contents

Abstract:	2
Introduction:	3
Implementation and Design:	3
C++ Program Results:	5
C++ DEV CODE:	5
CODE OUTPUT:	30
EXCEL FILES:	33
DRIVE LINK WITH ALL WORKING FILES AND REPORT:	34
Program Outputs:	34
Learning Outputs:	34

Abstract:

You are to develop a C++ code for AIRTEX airline's online ticket booking system. The code should be user-friendly to allow the customer to easily initiate a new booking, view his existing booking, or cancel it. On the other hand, the code also allows the airline managers to add or update information on available flights, and perform other clerical tasks (like checking the number of bookings, passenger information, etc.). For simplicity, the code does not need to implement the functionality of a payment (just entering the PAY string is enough).

The problem is very open-ended, meaning you can use any of the concepts learnt in the course to implement different additional functionalities to your code (and earn bonus points!).

Use concepts like class inheritance, friend classes, constructors, private data, arrays, etc to add better structure to your code. For example, your definitions of Manager and Customer class can inherit from a Person class.

The code should fulfill the following functionalities:

- Have 2 different levels of authentications; Manager and Customer.
- Both Manager and Customer should have the following attributes: name, user id, and password.
- Details of all registered users should be saved in a Users excel file.
- All "excel files" can be implemented as excel files of mentioned names.

Manager:

- Manager can sign in and logout only. No sign up required for managers (these users should already exist in the users list). A sample manager's name can be "Ahmed Khalid" and the password is "123"
- New flights are added to the Flights database by the Managers. The flight information should contain enough attributes to distinctly identify it.
- A manager can access information about all flights listed in the system.

Customer:

- Needs to sign up with a username and password. o Uses the credentials saved at sign up to log in and start a new session.
- The session starts with the display of the customer console view, where he can view existing bookings, cancel a booking, or make a new booking. o A new booking is made on the available list of flights. Seats can be booked for multiple passengers in a single booking. All details on a new booking should be saved in the Booking file.
- Customer should be able to cancel an existing booking from his console view.
- Make additional classes, as you deem necessary, to facilitate a proper functioning of the ticket reservation software.
- Additional points will be awarded to more interactive console screen designs.

Introduction:

In this CEA we are required to develop a C++ code for an airline's online booking system which should allow the customers to do new booking, view their existing booking or cancel it. In addition to that the code should also allow the airline managers to add or update information on available flights.

Features:

Talking about the features of the airline's online booking system it has 2 different levels of authentications; Manager and Customer. The customers need to sign up with a username and password. Through which they can login and the session starts with the display of the customer console where they can make new booking and view or cancel the existing bookings. On the other hand the managers do not require the sign up. They can sign in and log out only to access information about the scheduled flights in the system.

The code was designed keeping the abstract and the oop theory in mind we used the following header files starting:

- `#include<iostream>` : It is used as a stream of Input and Output using cin and cout.
- `#include<sstream>` : Which can be useful for employing stream-style manipulation on them example carrier functions.
- `#include<stdlib.h>` : Defines four variable types, several macros, and various functions for performing general functions for example `size_t`: This is the unsigned integral type and is the result of the `sizeof` keyword.
- `#include<conio.h>` : Used to `getch()` and `clrscr()` etc
- `#include<vector>` : Used to implement template classes for example `push_back()` function.
- `#include<windows.h>` : Used to sleep which means a general decrease in the system `sleep(1100)`; where 1100 is time in milliseconds
- `#include<iomanip>` : input, output manipulator for ex used to set width etc etc
- `#include<fstream>` : It is used to control the data to read from a file as an input and data to write into the file as an output. Basically used it to read and store data from excel files.

Header files tell a lot about a code meaning how it was designed and implemented well it took us some time to get started but once we started it was pretty easy to get going.

Implementation and Design:

The whole program is executed based on the concepts of functions, file handling and classes. The program consisted mainly of three classes having their own data variables, member functions. The classes as a whole execute a working system of an airline.

Base class:

It is an abstract class with two pure virtual functions; Login and Signup. The other classes manager and customer are publicly inherited by this class.

Manager class:

This class is a sub class of the base class having the following member functions. Each of these functions perform a specific functionality.

- Login
- Show booking
- Update booking

1. Login:

It is a virtual function if the base class and has different outcome in different inherited classes. The function read the manager data file then It takes input from the user and matches the data read from the file. If the entered data exists in the database, the login is successful and access is granted.

2. Show booking:

This function reads the booking data file from the data base and consoles it out.

3. Update booking:

It a function with access from only the manager. This function enables the manager to add a new flight schedule into the schedule details file using ofstream. The format un which the new entry is added is predefined in the order: booking number, from, to, time, flight code.

Customer class:

The functions include:

- Signup
- Login
- New booking
- Show booking
- Cancel booking

1. Signup:

Signup function is only limited to the customer class. It takes the info of the new user and writes it to the user data file.

2. Login:

This function in the customer class performs a similar functionality that of the manager class. It reads the user data file, and takes an input from the user. If the users unfo matches the user record, the customer is granted access.

3. New booking:

It asks the user about the type of flight they want to book, it can either be International or Domestic and as selected by the user, available flight list is consoled out to the user. Then it's up to the user to select from the list. After selecting the flight, the user is asked about a few more booking details. After that the booking detail of the user us writted in the specific file lf the user.

4. Show booking:

This function reads and displays all the bookings the customer made.

5. Cancel booking:

It displays all the bookings and asks the customer to choose which specific booking they want to delete. And using the logic of renaming and removing a file, the respective booking is removed from the file.

C++ Program Results:

C++ DEV CODE:

```
#include<iostream>
#include<sstream>
#include<stdlib.h>
#include<conio.h>
#include<vector>
#include<windows.h>
#include<iomanip>
#include<fstream>
using namespace std;

vector <string> temp;
short spacing(short,short);
```



```
class manager;
void UBwrite();
void details();
short TF();
void customerdetails();

void declare(){
    for (short i=0;i<4;i++)
        temp.push_back("0");
}

string shold,shold2,shold3;
short ihold,ihold2;
short bookey[2];
bool NEW=false;
short S[4]={16,21,14,14};

bool spacecheck(string k){
    for (short i=0;k[i];i++){
        if (k[i]==' ')
            return 1;
    }
    return 0;
}

bool commacheck(string k){
    for (short i=0;k[i];i++){
        if (k[i]==',')
            return 1;
    }
    return 0;
}
```

```
bool asteriskcheck(string k){
    for (short i=0;k[i];i++){
        if (k[i]=='*')
            return 1;
    }
    return 0;
}
```

```
short convert(string x){
    stringstream carrier(x);
    short hold;carrier>>hold;
    return hold;
}
```

```
string chconvert(char x){
    stringstream carrier;
    carrier<<x;
    string hold;
    carrier>>hold;
    return hold;
}
```

```
string iconvert(short t){
    stringstream carrier;
    carrier<<t;
    string hold=carrier.str();
    return hold;
}
```

```
void decode(string q){
    ihold=0;
```



```

        shold.erase();
        for (short i=0;q[i];i++){
            if (q[i]!='/')
                shold+=chconvert(q[i]);
            else{
                bookey[ihold]=convert(shold);
                ihold++;
                shold.erase();
            }
        }
        if (ihold==1){
            bookey[ihold]=convert(shold);
        }
    }

short MAIN(){
repeat:
    cout<<"*****\n";
    cout<<"\t MAIN MENU \n";
    cout<<"*****\n\n";
    cout<<"1) Customer Sign Up \n";
    cout<<"2) Customer Login \n";
    cout<<"3) Manager Login \n";
    cout<<"4) Exit Airtex\n\n";
    cout<<"Choose Appropriate Operation Number (1-4) to Proceed : ";
    cin>>shold;

    ihold2=convert(shold);
    if (ihold2>=1&& ihold2<=3)
        return ihold2;

    if (ihold2==4){
        system("CLS");
        cout << "\n\t\t" << "See you later!\n\t\t AIRTEX.\n\n";
    }
}

```

```

        Sleep(1000);
        exit (0);
        return 0;
    }
    else{
        cout<<"INVALID INPUT ! TRY AGAIN \n";
        goto repeat;
    }
}

class airline {
protected :
    static vector<vector<string> > flights;
public:
};

vector<vector<string> > airline :: flights;

class customer : public airline{
protected:
    string name,id,password;
    short Tbookings,cancellations;
    vector <string> booking ;
public:
    customer(){
        cancellations=0;
    }
    string idaccess(){
        return id;
    }
    void signup(){
        bool point=false;

```

```
rep0:
cout<<"Enter Full Name : ";

if(point==false)
cin.ignore();
getline(cin,shold);

if(commacheck(shold)==1||asteriskcheck(shold)==1){
    cout<<"INVALID FORMAT ! TRY AGAIN \n";
    point=true;
    goto rep0;
}
name=shold;
rep1:
cout<<"Enter LOGIN ID : ";
getline(cin,shold);

if
(spacecheck(shold)==1||shold=="RETURN"||commacheck(shold)==1||asteriskcheck(s
hold)==1){

    cout<<"INVALID INPUT ! TRY AGAIN \n";
    goto rep1;
}
id=shold;
rep2:
cout<<"Enter Password : ";
getline(cin,shold);

if (commacheck(shold)==1||asteriskcheck(shold)==1){
    cout<<"INVALID INPUT ! TRY AGAIN \n";
    goto rep2;
}
password=shold;
```

```

        NEW=true;
        UBwrite();
        cout<<"Account Creation Successful ! \n";
        throwback0();
    }

    void authentication(){
        ihold2=0;
        rep:
        cout<<"Enter Password : ";

        if(ihold2==0)
            cin.ignore();
            getline(cin,shold);

            if(shold==password){
                cout<<"LOGIN Successful ! \n";
                throwback0();
            }
            else {
                cout<<"Incorrect Password ! TRY AGAIN \n";
                ihold2++;
                goto rep;
            }
        }

        void throwback0(){
            cout<<"PRESS any Key to Enter Customer Menu ";
            getch();
            system("CLS");
            customermenu();
        }

        void throwback(){
            cout<<"PRESS any Key to Return to Main Menu ";

```

```

        getch();
        system("CLS");
        customermenu();
    }
    void customermenu(){
        mainmenu:
            cout<<"Welcome "<<name<<" ! \n \n";
            cout<<"***** \n";
            cout<<"\t CUSTOMER MENU \n";
            cout<<"***** \n \n";
            cout<<"1) List of Available Flights \n";
            cout<<"2) Make a New Booking \n";
            cout<<"3) Information about Existing Booking \n";
            cout<<"4) Cancel a Booking \n";
            cout<<"5) Logout \n \n";
            cout<<"Choose Appropriate Operation Number (1-4) to
Proceed : ";
            cin>>shold;
            short choice;
            choice=convert(shold);

            if (choice==1){
                if (TF()==0)
                    cout<<"NO FLIGHTS AVAILABLE AT THE MOMENT !
\n";
                else {
                    cout<<"    ORIGIN        DESTINATION
DATE            TIME        \n \n";
                    for (short i=0;i<TF();i++){
                        cout<<i+1<<" ";
                        for (short j=0;j<4;j++){
                            cout<<flights[i][j];

```

```

                                for (short
t=0;t<spacing(flights[i][j].size(),S[j]);t++)

                                cout<<" ";

                                }

                                cout<<"\n";

                                }

                                }

                                throwback();

                                }

                                else if (choice==2){

                                cout<<"    ORIGIN        DESTINATION
DATE        TIME        \n \n";

                                for (short i=0;i<TF();i++){

                                cout<<i+1<<" " ;

                                for (short j=0;j<4;j++){

                                cout<<flights[i][j];

                                for (short
k=0;k<spacing(flights[i][j].size(),S[j]);k++)

                                cout<<" ";

                                }

                                cout<<"\n";

                                }

                                cout<<"\n";

                                repl:

                                cout<<"Enter the Corresponding Flight Number to
Book : ";

                                cin>>shold;

                                if (shold=="RETURN"){

                                system("CLS");

                                goto mainmenu;

                                }

                                ihold=convert(shold);

                                if (ihold>=1&&ihold<=TF()){

```

```

        ihold--;
        ihold2=0;
        for (;ihold2<Tbookings;ihold2++){

            if (iconvert(ihold)==chconvert(booking[ihold2][0])){

                cout<<"You Have Already
Booked this Flight ! \n";

                throwback();

            }
        }
        shold=iconvert(ihold);
        shold2=shold;
        rep2:
        cout<<"Enter Number of Seats to Book : ";
        cin>>shold;
        if (shold=="RETURN"){
            system("CLS");
            goto mainmenu;
        }
        ihold=convert(shold);
        if (ihold>0){
            shold3=shold;
            shold=shold2+"/"+shold3;
            if (cancellations==0)
                booking.push_back(shold);
            else {
                booking[Tbookings]=shold;
                cancellations--;
            }
            Tbookings++;
            cout<<"Booking was Successful ! \n
\n";

            UBwrite();

```



```

        throwback();
    }
    else {
        cout<<"INVALID INPUT ! TRY AGAIN

\n";

        goto rep2;
    }
}

else {
    cout<<"INVALID INPUT ! TRY AGAIN

\n";

    goto rep1;
}

}
else if (choice==3){
    if(Tbookings!=0){
        cout<<"      ORIGIN      DESTINATION
DATE      TIME      No. of Seats      \n \n";
        for (short i=0,k=1;i<Tbookings;i++){
            shold=booking[i];
            decode(shold);
            cout<<k<<" ";
            k++;
            for (short j=0;j<4;j++){
                cout<<flights[bookey[0]][j];
                for (short
k=0;k<spacing(flights[bookey[0]][j].size(),S[j]);k++)
                    cout<<" ";
                if (j==3)
                    cout<<bookey[1];
            }
            cout<<"\n";
        }
    }
}

```

```

    }

    else

        cout<<"You Have No Bookings ! \n";

        throwback();

    }

    else if (choice==4){

        if(Tbookings!=0){

            cout<<"    ORIGIN        DESTINATION

DATE            TIME            \n \n";

            for (short i=0;i<Tbookings;i++){

                shold=booking[i];

                decode(shold);

                cout<<bookey[0]+1<<" ) ";

                for (short j=0;j<4;j++){

                    cout<<flights[bookey[0]][j];

                    for (short

k=0;k<spacing(flights[bookey[0]][j]).size(),S[j]);k++)

                        cout<<" ";

                    if (j==3)

                        cout<<bookey[1];}

                    cout<<"\n";

                }

                rep3:

                cout<<"Enter the Corresponding Booking

Number to Cancel : ";

                cin>>shold;

                if (shold=="RETURN") {

                    system("CLS");

                    goto mainmenu;}

                ihold=convert(shold);

                if (ihold>=1&&ihold<=TF()){

                    ihold2=ihold;

                    ihold2--;

```

```

        short i=0;
        bool P=false;
        for (;i<Tbookings;i++){
            shold=booking[i];
            decode(shold);
            if (bookey[0]==ihold2){
                booking[i].erase();
                P=true;
                break;
            }
        }
        if (P!=true){
            cout<<"INVALID INPUT ! TRY AGAIN
\n";

            goto rep3;
        }
        Tbookings--;
        cancellations++;
        for (;i<Tbookings;i++)
            swap(booking[i],booking[i+1]);
        UBwrite();
        cout<<"Cancellation was Successful ! \n";
    }
    else {
        cout<<"INVALID INPUT ! TRY
AGAIN \n";

        goto rep3;
    }
}

else
cout<<"You Have No Bookings ! \n";
throwback();
}

```

```

        else if (choice==5){
            system("CLS");
            cout << "\n\t\t" << "Thankyou for choosing
AIRTEX.\n\n\n";

            Sleep(1000);
            system("CLS");}

        else {
            cout<<"INVALID SELECTION TRY AGAIN ! \n";
            throwback();
        }
    }

};

class manager : public customer {
    static string id,name,password;
    static short Tflights,Tcustomers;
    friend void UBwrite();
    friend void retrieve();
    friend void details();
    friend void customerdetails();
    friend void redirect();
public:
    static string idcheck(){
        return id;
    }
    static string passwordcheck(){
        return password;
    }
    static void throwback(){
        cout<<"PRESS any Key to Return to Main Menu ";
        getch();
    }
};

```

```

        system("CLS");

        managemenu();}

static void customerInc(){
    Tcustomers++;
}

static short giveTflights(){
    return Tflights;
}

static short giveTcustomers(){
    return Tcustomers;
}

static void managemenu(){
    mainmenu:
    cout<<"Welcome "<<name<<"\n";
    cout<<"***** \n";
    cout<<"\t MANAGER MENU \n";
    cout<<"***** \n\n";
    cout<<"1) Add New Flight Information \n";
    cout<<"2) View All Flights & Booking Details \n";
    cout<<"3) View Customer Details \n";
    cout<<"4) LOGOUT \n";
    cout<<"Choose Appropriate Operation Number (1-4) to Proceed
: ";

    cin>>shold;
    short ch;
    ch=convert(shold);
    if (ch==1){
        cout<<"Enter Origin : ";
        cin.ignore();
        getline(cin,shold);
        if (shold=="RETURN") {
            system("CLS");

```

```

        goto mainmenu;}

temp[0]=shold;
cout<<"Enter Destination : ";
getline(cin,shold);
if (shold=="RETURN") {
    system("CLS");
    goto mainmenu;}

temp[1]=shold;
cout<<"Enter Flight Date : ";
getline(cin,shold);
if (shold=="RETURN") {
    system("CLS");
    goto mainmenu;}

temp[2]=shold;
cout<<"Enter Flight Time : ";
getline(cin,shold);
if (shold=="RETURN") {
    system("CLS");
    goto mainmenu;}

temp[3]=shold;
flights.push_back(temp);
Tflights++;
Fwrite();
cout<<"Flight Information Uploaded Successfully !
\n";

    throwback();

}

else if (ch==2){
    details();
    throwback();}

else if (ch==3){

```

```

        customerdetails();
        throwback();}

    else if (ch==4) {
        system("CLS");
        cout << "\n\t\t" << "Thankyou for choosing
AIRTEX.\n\n\n";

        Sleep(1000);
        system("CLS");
    }

}

static void Fwrite(){
    fstream flight;
    flight.open("FLIGHTS.csv",ios::out);
    for (short i=0;i<Tflights;i++){

        for (short j=0;j<4;j++){
            flight<<flights[i][j];
            if(j<=2)
                flight<<',';
            flight<<'*';
            flight<<"\n";
        }
        flight.close();
    }

};

string manager::id="ADMIN";
string manager::name="Ahmed, Bisma, Aqib";
string manager::password="123";
short manager::Tcustomers;
short manager::Tflights;

```



```

manager *ptr;
vector<manager> object;

short TF() {
    return manager::giveTflights();
}

void UBwrite() {
    fstream booking;
    booking.open("User & Booking Data.csv",ios::out);
    ihold=0;
    short limit;
    if (NEW==true) {
        object[manager::Tcustomers].customer::Tbookings=0;
        manager::Tcustomers++;
    }
    NEW=false;
    for (;ihold<manager::Tcustomers;ihold++){
        booking<<object[ihold].customer::name;
        booking<<',';
        booking<<object[ihold].customer::id;
        booking<<',';
        booking<<object[ihold].customer::password;
        if (object[ihold].customer::Tbookings!=0)
            booking<<',';
        for (short i=0;i<object[ihold].customer::Tbookings;i++){
            booking<<object[ihold].customer::booking[i];
            if (i<=(object[ihold].customer::Tbookings-2))
                booking<<',';
            booking<<'\n';
        }
        booking.close();
    }
}

```

```

}

void retrieve(){
    fstream flight;
    fstream booking;
    flight.open("FLIGHTS.csv",ios::in);
    ihold=0;
    while(flight){
        getline(flight,shold);
        ihold++;
    }
    flight.close();
    manager::Tflights=ihold-1;
    string hold[manager::Tflights];
    ihold=0;
    flight.open("FLIGHTS.csv",ios::in);
    while(ihold<manager::Tflights){
        getline(flight,hold[ihold]);
        ihold++;
    }
    flight.close();
    ihold=0;
    shold.erase();
    for (short i=0;i<manager::Tflights;i++){
        for (short j=0;hold[i][j];j++){
            if(hold[i][j]!=','){
                if(hold[i][j]!='*')
                    shold+=hold[i][j];
                else goto land1; }

            else {
                land1:

```

```

        temp[ihold]=shold;
        ihold++;
        shold.erase();
    }
}
ihold=0;
customer::flights.push_back(temp);
}
booking.open("User & Booking Data.csv",ios::in);
ihold=0;
while(booking){
    getline(booking,shold);
        ihold++;
    }
booking.close();
manager::Tcustomers=ihold-1;
ihold=0;
for (;ihold<manager::Tcustomers;ihold++){
    manager *ptr= new manager;
    object.push_back(*ptr);
}
string hold2[manager::Tcustomers];
ihold=0;
booking.open("User & Booking Data.csv",ios::in);
while(ihold<manager::Tcustomers){
    getline(booking,hold2[ihold]);
        ihold++;
    }
booking.close();
shold.erase();
for (short i=0;i<manager::Tcustomers;i++){
    ihold=0;

```

```

        ihold2=0;
    for (;hold2[i][ihold2];ihold2++){
        if(hold2[i][ihold2]==' '){
            ihold++;}
        object[i].customer::Tbookings=ihold-2;
        ihold=0;
    for (short j=0;hold2[i][j];j++){

        if(hold2[i][j]!=' '){
            if(hold2[i][j]!='*')
                shold+=hold2[i][j];
            else goto land2; }

        else {
            land2:
            if (ihold==0)
                object[i].customer::name=shold;
            else if (ihold==1)
                object[i].customer::id=shold;
            else if (ihold==2)
                object[i].customer::password=shold;
            else
                object[i].customer::booking.push_back(shold);
            ihold++;
            shold.erase();}

        }

        ihold=0;
    }

}

short spacing (short x,short y){
    return (y-x);
}

```

```

}

void details(){

    cout<<"Total Number of Flights Available are :
"<<manager::Tflights<<"\n \n";

    ihold=0;

    cout<<"    ORIGIN            DESTINATION            DATE            TIME
Total Bookings \n \n";

    for (;ihold<manager::Tflights;ihold++){

        short i=0,j=0;

        while(i<manager::Tcustomers){

            for (short k=0;k<object[i].customer::Tbookings;k++){

                short
p=convert(chconvert(object[i].customer::booking[k][0]));

                if(p==ihold)

                    j++;

            }

            i++;

        }

        cout<<ihold+1<<" ) ";

        ihold2=0;

        for (;ihold2<4;ihold2++){

            cout<<manager::flights[ihold][ihold2];

            for (short
n=0;n<spacing(manager::flights[ihold][ihold2].size(),S[ihold2]);n++)

                cout<<" ";

            cout<<j;

            cout<<"\n";

        }

    }

}

void customerdetails(){

    short S=24;

    cout<<"Total Customers = "<<manager::Tcustomers<<"\n \n";

```

```

        if(manager::Tcustomers!=0){
            cout<<"    NAME                TOTAL BOOKINGS \n";
            ihold=0;
            for (;ihold<manager::Tcustomers;ihold++){
                cout<<ihold+1<<" ) ";
                cout<<object[ihold].customer::name;
                for
                (ihold2=0;ihold2<spacing(object[ihold].customer::name.size(),S);ihold2++)
                    cout<<" ";
                cout<<object[ihold].customer::Tbookings<<"\n";}}
        }

void intro() {

    cout << "\n\n\n\t\t *****" << endl;
    cout << "\t\t\t WELCOME TO AIRTEX" << endl;
    cout << "\t\t\t *****" << endl;
    Sleep(1500);
    cout << "\n\n\n\t\t AIRTEX Welcome you on behalf of:" << endl;
    Sleep(220);
    cout << "\n\t\t\t MANAGER: Ahmed Khalid" << endl;
    Sleep(220);
    cout << "\n\t\t\t MANAGER: Bisma Shakeel" << endl;
    Sleep(220);
    cout << "\n\t\t\t MANAGER: Aqib Ali" << endl;
    Sleep(1600);
    system("cls");
}

int main(){
    intro();
    declare();
    retrieve();
}

```

```

start:
short choice;
choice=MAIN();
if (choice==1){
    ptr= new manager;
    object.push_back(*ptr);
    object[manager::giveTcustomers()].signup();
}
else if (choice==2){
    rep1:
    ihold=0;
    bool flag=false;
        cout<<"Enter ID : ";
        cin>>shold;
        if (shold=="RETURN")
            goto start;
    for (;ihold<manager::giveTcustomers();ihold++){
        if (object[ihold].idaccess()==shold){
            flag=true;
            break;
        }}
        if (flag==true){
            object[ihold].authentication();
        }
    else {
        cout<<"NO ACCOUNT EXISTS AGAINST THIS ID ! TRY AGAIN ! \n";
        cout<<"(*TO RETURN TO MAIN MENU ENTER \"RETURN\") \n";
        goto rep1;
    }
}
else if (choice==3){
    rep2:

```



```
        cout<<"Enter ID : ";
        cin>>shold;
        if (shold=="RETURN")
            goto start;
        if (shold==manager::idcheck()){
            rep3:
            cout<<"Enter Password : ";
            cin>>shold;
            if (shold=="RETURN")
                goto start;
            if (shold==manager::passwordcheck()){
                system("CLS");
                manager::managemenu();}
            else {
                cout<<"Incorrect Password ! TRY AGAIN \n";
                goto rep3;}}
            else {
                cout<<"Incorrect ID ! TRY AGAIN \n";
                goto rep2;}
        }
        if (choice!=4)
            goto start;
        else {
            delete ptr;
        }
    return 0;
}
```

CODE OUTPUT:

Introduction:

```
C:\Users\Ahmed\Desktop\Third Semester\OOP LAB\PROJECT\PROJECTCOMPLETE.exe

*****
      WELCOME TO AIRTEX
*****

AIRTEX Welcome you on behalf of:

      MANAGER: Ahmed Khalid

      MANAGER: Bisma Shakeel

      MANAGER: Aqib Ali
```

Main menu:

```
C:\Users\Ahmed\Desktop\Third Semester\OOP LAB\PROJECT\PROJECTCOMPLETE.exe

*****
      MAIN MENU
*****

1) Customer Sign Up
2) Customer Login
3) Manager Login
4) Exit Airtex

Choose Appropriate Operation Number (1-4) to Proceed :
```

Exit page:

```
C:\Users\Ahmed\Desktop\Third Semester\OOP LAB\PROJECT\PROJECTCOMPLETE.exe

      See you later!
      ~AIRTEX.

-----
Process exited after 100.9 seconds with return value 0
Press any key to continue . . .
```

Customer Login:

```
C:\Users\Ahmed\Desktop\Third Semester\OOP LAB\PROJECT\PROJECTCOMPLETE.exe
*****
MAIN MENU
*****
1) Customer Sign Up
2) Customer Login
3) Manager Login
4) Exit Airtex

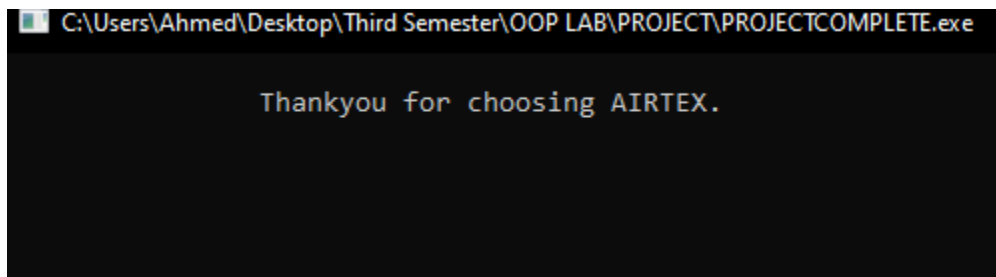
Choose Appropriate Operation Number (1-4) to Proceed : 2
Enter ID : Customer2
Enter Password : 123
LOGIN Successful !
PRESS any Key to Enter Customer Menu
```

```
C:\Users\Ahmed\Desktop\Third Semester\OOP LAB\PROJECT\PROJECTCOMPLETE.exe
Welcome Christopher Novel !

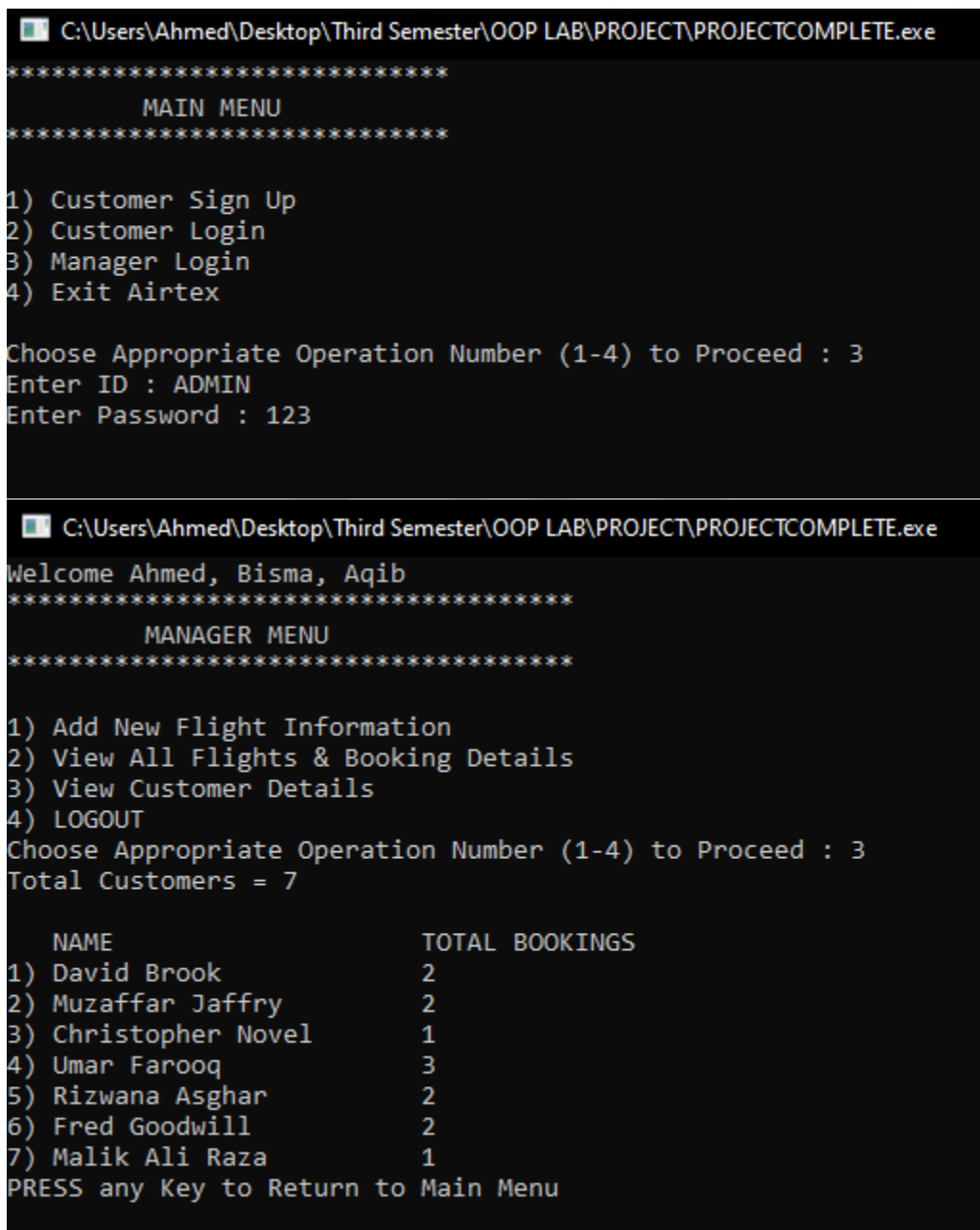
*****
CUSTOMER MENU
*****
1) List of Available Flights
2) Make a New Booking
3) Information about Existing Booking
4) Cancel a Booking
5) Logout

Choose Appropriate Operation Number (1-4) to Proceed : 1
  ORIGIN      DESTINATION      DATE      TIME
1) Karachi    Islamabad        15-12-2020  07:35 AM
2) Islamabad  Abu Dhabhi       2/1/2021    02:50 AM
3) Dubai      Los Angeles      14-01-2021  05:25 PM
4) Los Angeles Doha            05-02-2021  05:55 PM
5) Doha       Malibu           17-02-2021  12:00 AM
6) Malibu     Jeddah           25-02-2021  07:00 AM
7) Jeddah     Abu Dhabhi       15-03-2021  05:55 PM
PRESS any Key to Return to Main Menu
```

When logged out is selected:



Manager login:



You can also add flights

```
C:\Users\Ahmed\Desktop\Third Semester\OOP LAB\PROJECT\PROJECTCOMPLETE.exe
Welcome Ahmed, Bisma, Aqib
*****
MANAGER MENU
*****
1) Add New Flight Information
2) View All Flights & Booking Details
3) View Customer Details
4) LOGOUT
Choose Appropriate Operation Number (1-4) to Proceed : 2
Total Number of Flights Available are : 7

  ORIGIN      DESTINATION      DATE      TIME      Total Bookings
1) Karachi    Islamabad        15-12-2020 07:35 AM    1
2) Islamabad  Abu Dhabhi       2/1/2021   02:50 AM    6
3) Dubai      Los Angeles      14-01-2021 05:25 PM    1
4) Los Angeles Doha            05-02-2021 05:55 PM    2
5) Doha        Malibu           17-02-2021 12:00 AM    1
6) Malibu      Jeddah           25-02-2021 07:00 AM    1
7) Jeddah      Abu Dhabhi       15-03-2021 05:55 PM    1
PRESS any Key to Return to Main Menu

C:\Users\Ahmed\Desktop\Third Semester\OOP LAB\PROJECT\PROJECTCOMPLETE.exe

Thankyou for choosing AIRTEX.
```

Now goes back to main menu and repeats its course

Other functions include booking customer flights un booking them then creating new customer accounts then in the manager area creating new flights by adding destination etc etc

EXCEL FILES:

- This code is to be run with to files saved with it in the same folder one is FLIGHTS other one is User & Booking Data both are .csv then the code reads and writes data into the csv files

DRIVE LINK WITH ALL WORKING FILES AND REPORT:

https://drive.google.com/drive/folders/1fgES5AN9e4ZS_2A5-YkQ3wpLI2Ab-yR3?usp=sharing

Program Outputs:

- User Friendly
- Easy to run
- Simple Outcomes to Excel
- Case Sensitive code
- Flight System for manager and customer
- Attractive Display

Learning Outputs:

- By using the knowledge of classes, inheritance and file handling we made a working project of airline's online booking system. The program covers basic online booking system functions including booking of flights viewing of existing flights and its cancelation by the customers and adding and updating of flights by the manager.
- To conclude this program enabled us to see our critical thinking and improve our logic building skills. We improve and make best of the concepts of the topics including classes, inheritance and file handling.
- Now we can use this knowledge to write simple as well as complex codes for particular functions.
- And we took a lot of help from the internet where ever we got stuck.