OBJECT ORIENTED PROGRAMMING

FINAL PROJECT

SUBMITTED TO: MADAM SOBIA

SUBMITTED BY: SANA AKBAR

UBAIDA WAHEED

TAYYABA SALAMAT

GROUP: BSE-II A

FLIGHT RESERVATION SYSTEM

REPORT:

We have made flight reservation system as our final project of Object Oriented Programming.

In our project we have majorly used the concept of inheritance, polymorphism, aggregation and dynamic memory allocation.

In our code we have made two base classes while 6 derived classes.

First base class is of **Seat** from which two classes are derived: **economyClassSeat & businessClassSeat.**

Second base class is of **Aircraft** from which 4 classes are derived: **airbus, ATR, PIA, Boeing.**

Base class **Seat** and base class **Aircraft** have **has a** relationship. Aircraft class Seat class pointer objects are made which are further used in the derived classes to access the data members and functions.

In main pointer array of size four of class **Aircraft** are made and using dynamic memory allocation **airbus, ATR, PIA, Boeing** objects are pointed.

On run time all the values are initially zero. So on the top all the values will be displayed that will be keep on updating. Then there will be the main menu from where first the manager who knows the password will update the flight details. After that user can reserve the ticket, can see the flight status, can see his ticket details, can cancel his ticket and can eventually exit the program.

CODE

// exam lab.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

#include <string.h>

using namespace std;

int aircraftID = 0;

int RESERVATION = 0;

class seat {

protected:

string passenger\_name;

int passenger\_age;

int seat\_number;

bool booked; //whether seat is booked or not

int reservation\_number; //reservation number of the seat

public:

seat() {

passenger\_name = "NULL";

passenger\_age = 0;

seat\_number = 0;

booked = false;

reservation\_number = 0;

}

virtual void seatDetails() = 0;

bool getBookedStatus() {

return booked;

}

int getReservationNumber() {

return reservation\_number;

}

void setSeatNumber(int i) {

seat\_number = i;

}

void setReservationNumber(int r) {

reservation\_number = r;

}

void bookSeat() {

cout << "Enter Passenger's Name: ";

cin >> passenger\_name;

cout << "Enter Passenger's age: ";

cin >> passenger\_age;

booked = true;

}

void unBookSeat() {

passenger\_name = "NULL";

passenger\_age = 0;

booked = false;

}

};

class economy\_seat :public seat {

public:

void seatDetails() {

cout << endl;

cout << "Passenger Name: " << passenger\_name << endl;

cout << "Passenger Age: " << passenger\_age << endl;

cout << "Seat Number: " << seat\_number << endl;

cout << "Seat Type: Economy" << endl;

cout << "Booked Status: ";

if (booked == true) {

cout << "Booked" << endl;

}

if (booked == false) {

cout << "Vacant" << endl;

}

cout << "Reservation Number: " << reservation\_number << endl;

}

};

class business\_seat :public seat {

void seatDetails() {

cout << endl;

cout << "Passenger Name: " << passenger\_name << endl;

cout << "Passenger Age: " << passenger\_age << endl;

cout << "Seat Number: " << seat\_number << endl;

cout << "Seat Type: Business" << endl;

cout << "Booked Status: ";

if (booked == true) {

cout << "Booked" << endl;

}

if (booked == false) {

cout << "Vacant" << endl;

}

cout << "Reservation Number: " << reservation\_number << endl;

}

};

class aircraft {

protected:

int ID;

int totalSeats, businessSeats, economySeats, available\_economySeats, available\_businessSeats;

string startingPoint, destination;

seat\* b, \* e;

public:

aircraft() {

ID = ++aircraftID;

totalSeats = 0;

businessSeats = 0;

economySeats = 0;

available\_economySeats = 0;

available\_businessSeats = 0;

startingPoint = "NULL";

destination = "NULL";

}

void setFlighId(int i) {

ID = i;

}

virtual void flightDetails() = 0;

virtual void updateFlightDetails() = 0;

virtual void reserveTicket() = 0;

virtual void cancelTicket(int r) = 0;

virtual void showTicketStatus(int r) = 0;

virtual void flightSchedule() = 0;

};

class airbus :public aircraft {

string p;

public:

void flightDetails() {

cout << endl << "\t\t<<<<Flight Information>>>>\n\n";

cout << "Flight Name: Airbus-" << ID << endl;

cout << "Starting Point is: " << startingPoint << endl;

cout << "Destination is: " << destination << endl;

cout << "Available Business Seats: " << available\_businessSeats << endl;

cout << "Available Economy Seats: " << available\_economySeats << endl << endl << endl;

}

void updateFlightDetails() {

cout<<"Are you MANAGER"<<endl<<"Enter Pasword to prove your ID"<<endl;

cin>>p;

if(p=="UST")

{cout << "Enter Starting Point: ";

cin >> startingPoint;

cout << "Enter Destination: ";

cin >> destination;

do {

cout << "Total Number of Seats should be between 300 and 500\n";

cout << "Enter the number of business seats: ";

cin >> businessSeats;

cout << "Enter the number of economy seats: ";

cin >> economySeats;

totalSeats = businessSeats + economySeats;

} while (totalSeats <= 0 || totalSeats >= 500);

available\_businessSeats = businessSeats;

available\_economySeats = economySeats;

b = new business\_seat[businessSeats];

e = new economy\_seat[economySeats];

for (int i = 0; i < businessSeats; i++) {

b[i].setSeatNumber(i + 1);

b[i].setReservationNumber(++RESERVATION);

}

for (int i = 0; i < economySeats; i++) {

e[i].setSeatNumber(i + 1);

e[i].setReservationNumber(++RESERVATION);

}}

else

{cout<<"YOU HAVE ENTERED INVALID PASWORD"<<endl;

}

}

void showTicketStatus(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r) {

cout << "\nFlight Name: Airbus-" << ID;

b[i].seatDetails();

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r) {

cout << "\nFlight Name: Airbus-" << ID;

e[i].seatDetails();

break;

}

}

}

void flightSchedule() {

cout << "Airbus-" << ID << "\t" << ID << "\t\t" << startingPoint << "\t\t\t" << destination << "\t\t" << available\_businessSeats << "\t\t" << available\_economySeats << endl;

}

void cancelTicket(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r && b[i].getBookedStatus() == true) {

b[i].unBookSeat();

available\_businessSeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r && e[i].getBookedStatus() == true) {

e[i].unBookSeat();

available\_economySeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

}

void reserveTicket() {

if (available\_businessSeats > 0 || available\_economySeats > 0) {

int i = 0;

bool m = false;

do {

cout << "\nEnter 1 for Business Class\n";

cout << "Enter 2 for Economy Class\n";

cout << "Enter 3 to exit\n";

cin >> i;

if (i == 1 && available\_businessSeats > 0) {

for (int j = 0; j < businessSeats; j++) {

if (b[j].getBookedStatus() == false) {

b[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

b[j].seatDetails();

available\_businessSeats--;

break;

}

}

}

else if (i == 1 && available\_businessSeats <= 0) {

cout << "\nNo Business Seat Availavle. Try Economy.\n";

}

else if (i == 2 && available\_economySeats > 0) {

for (int j = 0; j < economySeats; j++) {

if (e[j].getBookedStatus() == false) {

e[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

e[j].seatDetails();

available\_economySeats--;

break;

}

}

}

else if (i == 2 && available\_economySeats <= 0) {

cout << "\nNo Economy Seat Availavle. Try Business.\n";

}

else if (i == 3) {

m = true;

}

} while (m == false);

}

else {

cout << "No Seats available. Try another flight.";

}

}

};

class boeing :public aircraft {

string p;

public:

void flightDetails() {

cout << endl << "\t\t<<<<Flight Information>>>>\n\n";

cout << "Flight Name: Boeing-" << ID << endl;

cout << "Starting Point is: " << startingPoint << endl;

cout << "Destination is: " << destination << endl;

cout << "Available Business Seats: " << available\_businessSeats << endl;

cout << "Available Economy Seats: " << available\_economySeats << endl << endl << endl;

}

void updateFlightDetails() {

cout<<"Are you MANAGER"<<endl<<"Enter Pasword to prove your ID"<<endl;

cin>>p;

if(p=="UST")

{cout << "Enter Starting Point: ";

cin >> startingPoint;

cout << "Enter Destination: ";

cin >> destination;

do {

cout << "Total Number of Seats should be between 300 and 500\n";

cout << "Enter the number of business seats: ";

cin >> businessSeats;

cout << "Enter the number of economy seats: ";

cin >> economySeats;

totalSeats = businessSeats + economySeats;

} while (totalSeats <= 0 || totalSeats >= 500);

available\_businessSeats = businessSeats;

available\_economySeats = economySeats;

b = new business\_seat[businessSeats];

e = new economy\_seat[economySeats];

for (int i = 0; i < businessSeats; i++) {

b[i].setSeatNumber(i + 1);

b[i].setReservationNumber(++RESERVATION);

}

for (int i = 0; i < economySeats; i++) {

e[i].setSeatNumber(i + 1);

e[i].setReservationNumber(++RESERVATION);

}}

else

{cout<<"YOU HAVE ENTERED INVALID PASWORD"<<endl;

}

}

void showTicketStatus(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r) {

cout << "\nFlight Name: Boeing-" << ID;

b[i].seatDetails();

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r) {

cout << "\nFlight Name: Boeing-" << ID;

e[i].seatDetails();

break;

}

}

}

void flightSchedule() {

cout << "Boeing-" << ID << "\t" << ID << "\t\t" << startingPoint << "\t\t\t" << destination << "\t\t" << available\_businessSeats << "\t\t" << available\_economySeats << endl;

}

void cancelTicket(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r && b[i].getBookedStatus() == true) {

b[i].unBookSeat();

available\_businessSeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r && e[i].getBookedStatus() == true) {

e[i].unBookSeat();

available\_economySeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

}

void reserveTicket() {

if (available\_businessSeats > 0 || available\_economySeats > 0) {

int i = 0;

bool m = false;

do {

cout << "\nEnter 1 for Business Class\n";

cout << "Enter 2 for Economy Class\n";

cout << "Enter 3 to exit\n";

cin >> i;

if (i == 1 && available\_businessSeats > 0) {

for (int j = 0; j < businessSeats; j++) {

if (b[j].getBookedStatus() == false) {

b[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

b[j].seatDetails();

available\_businessSeats--;

break;

}

}

}

else if (i == 1 && available\_businessSeats <= 0) {

cout << "\nNo Business Seat Availavle. Try Economy.\n";

}

else if (i == 2 && available\_economySeats > 0) {

for (int j = 0; j < economySeats; j++) {

if (e[j].getBookedStatus() == false) {

e[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

e[j].seatDetails();

available\_economySeats--;

break;

}

}

}

else if (i == 2 && available\_economySeats <= 0) {

cout << "\nNo Economy Seat Availavle. Try Business.\n";

}

else if (i == 3) {

m = true;

}

} while (m == false);

}

else {

cout << "No Seats available. Try another flight.";

}

}

};

class ATR :public aircraft {

string p;

public:

void flightDetails() {

cout << endl << "\t\t<<<<Flight Information>>>>\n\n";

cout << "Flight Name: ATR-" << ID << endl;

cout << "Starting Point is: " << startingPoint << endl;

cout << "Destination is: " << destination << endl;

cout << "Available Business Seats: " << available\_businessSeats << endl;

cout << "Available Economy Seats: " << available\_economySeats << endl << endl << endl;

}

void updateFlightDetails() {

cout<<"Are you MANAGER"<<endl<<"Enter Pasword to prove your ID"<<endl;

cin>>p;

if(p=="UST")

{cout << "Enter Starting Point: ";

cin >> startingPoint;

cout << "Enter Destination: ";

cin >> destination;

do {

cout << "Total Number of Seats should be between 300 and 500\n";

cout << "Enter the number of business seats: ";

cin >> businessSeats;

cout << "Enter the number of economy seats: ";

cin >> economySeats;

totalSeats = businessSeats + economySeats;

} while (totalSeats <= 0 || totalSeats >= 500);

available\_businessSeats = businessSeats;

available\_economySeats = economySeats;

b = new business\_seat[businessSeats];

e = new economy\_seat[economySeats];

for (int i = 0; i < businessSeats; i++) {

b[i].setSeatNumber(i + 1);

b[i].setReservationNumber(++RESERVATION);

}

for (int i = 0; i < economySeats; i++) {

e[i].setSeatNumber(i + 1);

e[i].setReservationNumber(++RESERVATION);

}}

else

{cout<<"YOU HAVE ENTERED INVALID PASWORD"<<endl;

}

}

void showTicketStatus(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r) {

cout << "\nFlight Name: ATR-" << ID;

b[i].seatDetails();

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r) {

cout << "\nFlight Name: ATR-" << ID;

e[i].seatDetails();

break;

}

}

}

void flightSchedule() {

cout << "ATR-" << ID << "\t\t" << ID << "\t\t" << startingPoint << "\t\t\t" << destination << "\t\t" << available\_businessSeats << "\t\t" << available\_economySeats << endl;

}

void cancelTicket(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r && b[i].getBookedStatus() == true) {

b[i].unBookSeat();

available\_businessSeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r && e[i].getBookedStatus() == true) {

e[i].unBookSeat();

available\_economySeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

}

void reserveTicket() {

if (available\_businessSeats > 0 || available\_economySeats > 0) {

int i = 0;

bool m = false;

do {

cout << "\nEnter 1 for Business Class\n";

cout << "Enter 2 for Economy Class\n";

cout << "Enter 3 to exit\n";

cin >> i;

if (i == 1 && available\_businessSeats > 0) {

for (int j = 0; j < businessSeats; j++) {

if (b[j].getBookedStatus() == false) {

b[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

b[j].seatDetails();

available\_businessSeats--;

break;

}

}

}

else if (i == 1 && available\_businessSeats <= 0) {

cout << "\nNo Business Seat Availavle. Try Economy.\n";

}

else if (i == 2 && available\_economySeats > 0) {

for (int j = 0; j < economySeats; j++) {

if (e[j].getBookedStatus() == false) {

e[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

e[j].seatDetails();

available\_economySeats--;

break;

}

}

}

else if (i == 2 && available\_economySeats <= 0) {

cout << "\nNo Economy Seat Availavle. Try Business.\n";

}

else if (i == 3) {

m = true;

}

} while (m == false);

}

else {

cout << "No Seats available. Try another flight.";

}

}

};

class PIA :public aircraft {

string p;

public:

void flightDetails() {

cout << endl << "\t\t<<<<Flight Information>>>>\n\n";

cout << "Flight Name: PIA-" << ID << endl;

cout << "Starting Point is: " << startingPoint << endl;

cout << "Destination is: " << destination << endl;

cout << "Available Business Seats: " << available\_businessSeats << endl;

cout << "Available Economy Seats: " << available\_economySeats << endl << endl << endl;

}

void updateFlightDetails() {

cout<<"Are you MANAGER"<<endl<<"Enter Pasword to prove your ID"<<endl;

cin>>p;

if(p=="UST")

{cout << "Enter Starting Point: ";

cin >> startingPoint;

cout << "Enter Destination: ";

cin >> destination;

do {

cout << "Total Number of Seats should be between 300 and 500\n";

cout << "Enter the number of business seats: ";

cin >> businessSeats;

cout << "Enter the number of economy seats: ";

cin >> economySeats;

totalSeats = businessSeats + economySeats;

} while (totalSeats <= 0 || totalSeats >= 500);

available\_businessSeats = businessSeats;

available\_economySeats = economySeats;

b = new business\_seat[businessSeats];

e = new economy\_seat[economySeats];

for (int i = 0; i < businessSeats; i++) {

b[i].setSeatNumber(i + 1);

b[i].setReservationNumber(++RESERVATION);

}

for (int i = 0; i < economySeats; i++) {

e[i].setSeatNumber(i + 1);

e[i].setReservationNumber(++RESERVATION);

}}

else

{cout<<"YOU HAVE ENTERED INVALID PASWORD"<<endl;

}

}

void showTicketStatus(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r) {

cout << "\nFlight Name: PIA-" << ID;

b[i].seatDetails();

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r) {

cout << "\nFlight Name: PIA-" << ID;

e[i].seatDetails();

break;

}

}

}

void flightSchedule() {

cout << "PIA-" << ID << "\t\t" << ID << "\t\t" << startingPoint << "\t\t\t" << destination << "\t\t" << available\_businessSeats << "\t\t" << available\_economySeats << endl;

}

void cancelTicket(int r) {

for (int i = 0; i < businessSeats; i++) {

if (b[i].getReservationNumber() == r && b[i].getBookedStatus() == true) {

b[i].unBookSeat();

available\_businessSeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

for (int i = 0; i < economySeats; i++) {

if (e[i].getReservationNumber() == r && e[i].getBookedStatus() == true) {

e[i].unBookSeat();

available\_economySeats++;

cout << "\nSeat Cancelled\n";

break;

}

}

}

void reserveTicket() {

if (available\_businessSeats > 0 || available\_economySeats > 0) {

int i = 0;

bool m = false;

do {

cout << "\nEnter 1 for Business Class\n";

cout << "Enter 2 for Economy Class\n";

cout << "Enter 3 to exit\n";

cin >> i;

if (i == 1 && available\_businessSeats > 0) {

for (int j = 0; j < businessSeats; j++) {

if (b[j].getBookedStatus() == false) {

b[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

b[j].seatDetails();

available\_businessSeats--;

break;

}

}

}

else if (i == 1 && available\_businessSeats <= 0) {

cout << "\nNo Business Seat Availavle. Try Economy.\n";

}

else if (i == 2 && available\_economySeats > 0) {

for (int j = 0; j < economySeats; j++) {

if (e[j].getBookedStatus() == false) {

e[j].bookSeat();

m = true;

cout << "\nBooking Successful\n";

e[j].seatDetails();

available\_economySeats--;

break;

}

}

}

else if (i == 2 && available\_economySeats <= 0) {

cout << "\nNo Economy Seat Availavle. Try Business.\n";

}

else if (i == 3) {

m = true;

}

} while (m == false);

}

else {

cout << "No Seats available. Try another flight.";

}

}

};

int \_tmain(int argc, \_TCHAR\* argv[])

{

aircraft \*p[4];

p[0]=new airbus;

p[1]=new boeing;

p[2]=new ATR;

p[3]=new PIA;

int menu=0;

while (menu != 6) {

system("cls");

cout << "FLIGHT NAME\tFLIGHT ID\tSTARTING POINT\t\tDESTINATION\tBUSINESS SEATS\tECONOMY SEATS" << endl<<endl;

p[0]->flightSchedule();

p[1]->flightSchedule();

p[2]->flightSchedule();

p[3]->flightSchedule();

cout << "\n\n\tMAIN MENU\n";

cout << "------------------------------\n";

cout << "1. Flight Details\n";

cout << "2. Update Flight Details\n";

cout << "3. Reserving a ticket\n";

cout << "4. Cancelling a ticket\n";

cout << "5. Display the present ticket status\n";

cout << "6. Exit\n\n\n";

cout << "Enter your option: ";

cin >> menu;

if (menu == 1) {

int s;

cout << "Enter Aircraft ID: ";

cin >> s;

if (s == 1) {

p[0]->flightDetails();

}

if (s == 2) {

p[1]->flightDetails();

}

if (s == 3) {

p[2]->flightDetails();

}

if (s == 4) {

p[3]->flightDetails();

}

}

if (menu == 2) {

int s;

cout << "Enter Aircraft ID: ";

cin >> s;

if (s == 1) {

p[0]->updateFlightDetails();

}

if (s == 2) {

p[1]->updateFlightDetails();

}

if (s == 3) {

p[2]->updateFlightDetails();

}

if (s == 4) {

p[3]->updateFlightDetails();

}

}

if (menu == 3) {

int s;

cout << "Enter Aircraft ID: ";

cin >> s;

if (s == 1) {

p[0]->reserveTicket();

}

if (s == 2) {

p[1]->reserveTicket();

}

if (s == 3) {

p[2]->reserveTicket();

}

if (s == 4) {

p[3]->reserveTicket();

}

}

if (menu == 4) {

int c;

cout << "Enter Reservation Number: ";

cin >> c;

for (int i = 0; i < 4; i++) {

p[i]->cancelTicket(c);

}

}

if (menu == 5) {

int c;

cout << "Enter Reservation Number: ";

cin >> c;

for (int i = 0; i < 4; i++) {

p[i]->showTicketStatus(c);

}

}

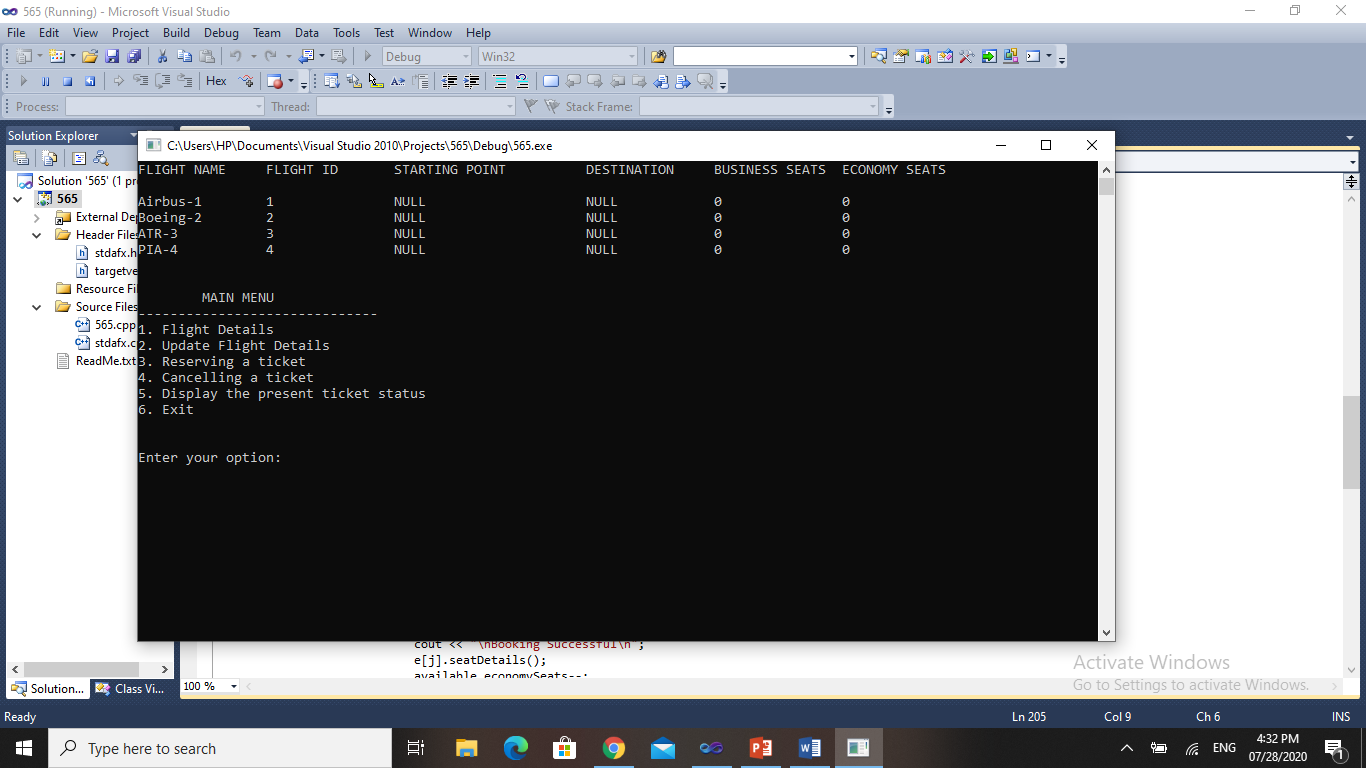
cout << endl;

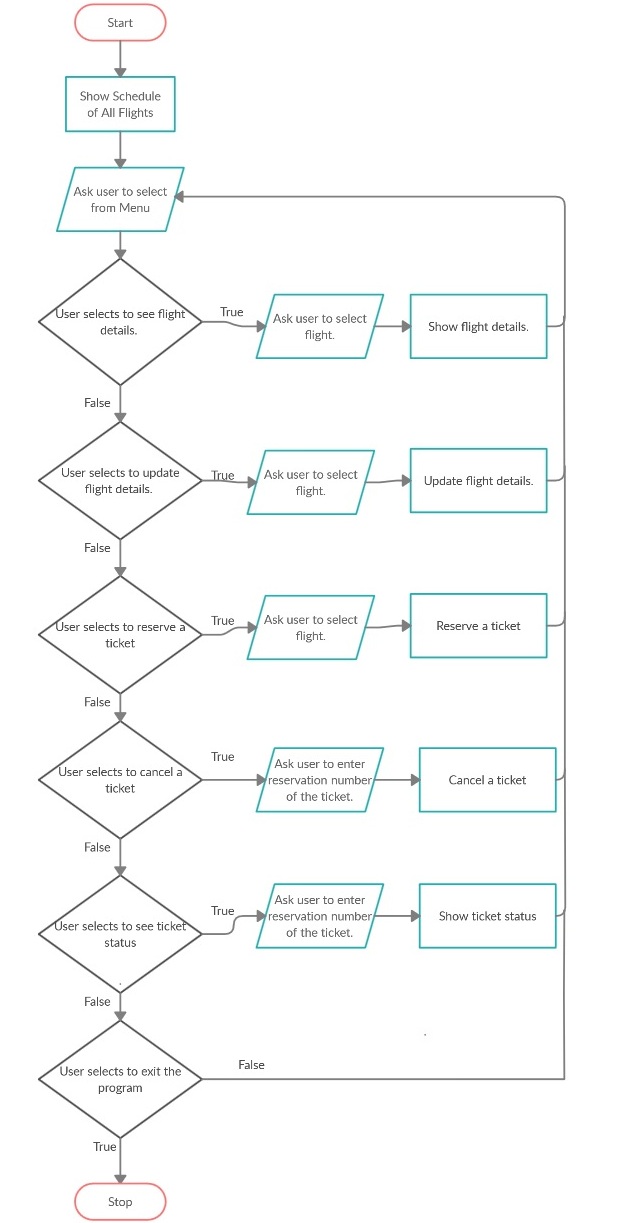
system("pause");

}

return(0);

}

**OUTPUT**

Flow Chart: