OPP PROJECT REPORT

MAIN FUNCTION:

The function begins by displaying a menu of options to the user, and it uses a switch statement to execute the corresponding block of code based on the user's choice.

In the first case, the code displays the available flight routes by iterating over two-dimensional arrays of international and local destinations. The user is then prompted to go back to the main menu.

In the second case, the code displays information about the airline company. The user is then prompted to go back to the main menu.

In the third case, the code prompts the user to select either passenger or admin login. If the user selects passenger login, they are prompted to enter their username and password, and the code uses the User class to verify their login credentials. If the user enters valid credentials, the code displays a home page for the user. If the user selects the admin login option, the code prompts the user to enter the admin username and password, and the code uses the Admin class to verify the credentials. If the login is successful, the admin can access the admin panel to manage flights and passenger information.

In the fourth case, the code exits the program.

LOGIN CLASS:

There are three classes: Login, User and Admin. The User and Admin classes inherit from the Login class.

```
class Login{
    protected:
        string username;
        string password;
        string fullName;
        string email;
        int cnic[13];
        int a=0;
};
class User:public Login{
class Admin:public Login{
class HomePage{
    public:
```

The Login class has four private member variables: username, password, fullName, email, and cnic (an array of 13 integers). The cnic variable is initialized to zero. There is no constructor or other member functions defined for this class.

USER CLASS:

The User class has one public member function named signUp(), which prompts the user to enter their username, password, full name, email and cnic number. It then validates the cnic number to make sure it has exactly 12 digits. If the cnic number is invalid, the user is prompted to enter it again. If it is valid, the user's information is appended to a text file named "users.txt".

The User class also has one private member function named verifyUser(), which takes a username and password as input and checks if they match the ones in the "users.txt" file. If they do, the function returns true, otherwise it returns false.

The User class also has a public member function named login(), which prompts the user to enter their username and password. It then calls the verifyUser() function to check if the username and password are valid. If they are, the function returns true and displays a message saying that the login was successful. If they are not, the function returns false and displays a message saying that the login was unsuccessful.

ADMIN CLASS:

The Admin class has a public member function named login(), which prompts the user to enter an admin username and password. It then hides the password by displaying asterisks for each character entered, and checks if the username and password are valid. If they are, the function displays a message saying that the login was successful and displays an admin panel with two options: manage complains or exit. If the user chooses manage complains, the resolve_complain() function is called. If the user chooses exit, the program terminates. If the user enters an invalid choice, the program prompts the user to enter a valid one.

The Admin class also has two private member functions: view_complains() and resolve_complain(). The view_complains() function reads from a file named "complain.txt" and displays all the contents of the file on the console. The resolve_complain() function prompts the user to enter the complaint ID they want to mark as resolved, searches for it in the "complain.txt" file and displays a message saying that the complaint was resolved.

HOMEPAGE CLASS:

The HomePage class contains a constructor that is executed when an object of the class is created. The constructor displays a welcome message for an airline named "NUCES FLY Airline".

The class also contains member variables such as f_name (string), l_name (string), add (string), pass_no (string), phone_no (string), choice (integer), datedep (integer), date (integer), month (integer), year (integer), ans (boolean), and ch (character).

The class also contains member functions such as the switch statement that performs different actions based on the user's input, case 1 is for booking, case 2 is for displaying the routes of the airline, case 3 is for ticket information, and case 4 is for complaints.

The booking process includes taking input from the user such as first name, last name, address, phone number, and the date of departure. Then the user is asked to select either local or international destinations. The system checks if the airline flies to the selected destination or not. If it does, the system displays seat availability and meal options. Then the user is asked to confirm the booking, and if confirmed, the system displays a message thanking the user for flying with the airline.

The ticket information process allows the user to change or cancel a booking by entering the passenger's first and last name.

PASSENGER CLASS:

passenger is an abstract class that provides the basic information and functionalities for a passenger such as first and last name, phone number, address, passport number, date of departure, seat reservation, meal preferences, registration, and displaying passenger details. booking is a derived class of passenger that inherits all the properties of the base class and adds some more functionalities such as checking available seats, booking, and displaying local or international destinations.

```
    protected:
        string firstname,lastname, phone, mealtype, passport, Address;
        int bus, eco, datedep, payment;
        char origin[30];
        char destination[30];
    public:
        passenger(string f_name, string l_name, string phone_no, string add="N/A", string pass_no="N/A", int dd=000000): bseats(50), eseats(250)
        {
            datedep=dd;
            firstname=f_name;
            lastname=l_name;
            Address=add;
            phone=phone_no;
            passport=pass_no;
            bus = bseats;
            eco = eseats;
        }

        virtual bool ldest()=0;
        virtual bool idest(string pass)=0;
        virtual void seats()=0;
        virtual void Meals()=0;
        virtual void Registration()=0;
        virtual bool display()=0;
        virtual void drive()=0;

};

class booking:public passenger
{
    protected:
        int i,j,n,m,k,NoOfSeats,choice;
```

The passenger class has the following private data members:

BusinessClassSeats: an integer constant that represents the total number of business seats.

EconomyClasSeats: an integer constant that represents the total number of economy seats.

The passenger class also has the following protected data members:

firstname: a string that stores the first name of the passenger.

lastname: a string that stores the last name of the passenger.

phone: a string that stores the phone number of the passenger.

mealtype: a string that stores the meal type preferred by the passenger.

passport: a string that stores the passport number of the passenger.

Address: a string that stores the address of the passenger.

bus: an integer that represents the number of booked business seats.

eco: an integer that represents the number of booked economy seats.

datedep: an integer that represents the date of departure.

The passenger class also has the following public virtual member functions:

ldest(): a pure virtual function that displays and checks availability of local destinations.

idest(): a pure virtual function that displays and checks availability of international destinations.

seats(): a pure virtual function that allows booking of seats for the passenger.

Meals(): a pure virtual function that allows the passenger to choose a meal preference.

Registration(): a pure virtual function that allows the passenger to complete registration.

display(): a pure virtual function that displays the passenger details.

drive(): a pure virtual function that allows the passenger to drive.

The booking class is a derived class of the passenger class and has the following public member functions:

booking(): a constructor that initializes the base class properties and sets the booked business and economy seats to zero.

ldest(): a member function that displays and checks availability of local destinations.

idest(): a member function that displays and checks availability of international destinations.

seats(): a member function that allows booking of seats for the passenger.

Meals(): a member function that allows the passenger to choose a meal preference.

Registration(): a member function that allows the passenger to complete registration.

display(): a member function that displays the passenger details.

The booking class also has the following private data members:

NoOfSeats: an integer that represents the number of seats booked.

choice: an integer that represents the meal preference of the passenger.

The complaint process allows the user to register a complaint by entering the passenger's first and last name and the details of the complaint.

ROUTES CLASS:

The class called "routes". It contains a public member function called "print" which returns a boolean value.

The "print" function is used to display information about international and local flights available. It starts by printing a message to the console indicating that it is displaying the available flights. Then, it loops through an array of international destinations and prints out all the possible flights between them. The same is done for local destinations.

Finally, it prompts the user to book a flight and waits for the user to input a response (y/n). If the user responds with 'y' or 'Y', the function returns true, otherwise, it returns false.

COMPLAINT CLASS:

The class called complain which represents a customer complaint for an airline company. It has private member variables f_name, l_name, and p_num to store the first name, last name, and phone number of the customer respectively. It also has choice, sentence, types, and ch member variables which will be used to store the user's input and choices.

The complain class has a constructor that takes three parameters f, l, and num, which initializes the private member variables f_name, l_name, and p_num.

The type() function is a member function that allows the user to enter their complaint. It displays a menu of complaint types and prompts the user to enter their complaint. It then stores the type of complaint in the types variable and the customer's complaint in the sentence variable.

The function then prompts the user if they want to edit their complaint. If the user chooses to edit, the function loops back to the beginning. If the user chooses not to edit, the function writes the customer's name, phone number, and complaint to a text file called complain.txt using an ofstream.

CLASS MANAGE:

The class named "Manage". The class has a private member variable named "temp" which is of type string, and "y" and "c2" which are also of type string. Additionally, it has a private member variable named "option" which is of type integer.

The class has one public member function named "change()". This function reads a file named "flight.txt" and asks the user to input a ticket number. If the user enters "0", the function returns 1, indicating that the program should exit. If the user enters a valid ticket number, the function searches the file for the corresponding record.

If the record is found, the function displays the entire contents of the file and asks the user to choose one of the following options: Change date, Cancel the flight,Go Back If the user selects option 1, the function prompts the user to enter a new date of departure in the format "DDMMYYYY". The function then creates a temporary file named "temp.txt" and copies the contents of "flight.txt" to this file, replacing the original date of departure with the new one. The function then closes both files, deletes "flight.txt", renames "temp.txt" to "flight.txt", and returns 1, indicating that the operation was successful.

If the user selects option 2, the function deletes "flight.txt", clears the console, displays a message indicating that the booking has been canceled, and returns 1, indicating that the operation was successful.

If the user selects option 3, the function clears the console and returns 1, indicating that the user wants to go back to the main page.

If the record is not found in the file, the function displays a message indicating that it was unable to find any record and returns 1, indicating that the user wants to go back to the main page.

CLASS DIAGRAM: