OBJECT ORIENTIED METHODOLOGY

AIRLINES RESERVATION SYSTEM

Team Members :

LCI2020079 – Srikanth

LIT2020078 – Manasa

LCS2020059 – Sindhuja

LCS2020058 - Meenakshi

PROJECT REPORT

Introduction:

Nowadays there are many ways for traveling from one place to another. In those flights are very convenient for traveling long distances in a short period .our project aids in booking the flight tickets. we have constructed this project in such a way that anyone can access it easily even a non-programmer can access it very conveniently. We have designed our project using structured & modular techniques.

We have provided various options to the user. they can select accordingly to their convenience. We have designed our project in the favor of the user. There is no role of admin.

Objective:

Our project is a computer-based management system that is designed to handle all the primary information of the user. We created a separate database to store and maintain all the details of the user. We store each and every information of the user.

How our code executes :

The user can operate this code in a very convenient way. The user is given 6 options to be chosen according to their convenience The first phase of our project deals with booking flight tickets :-

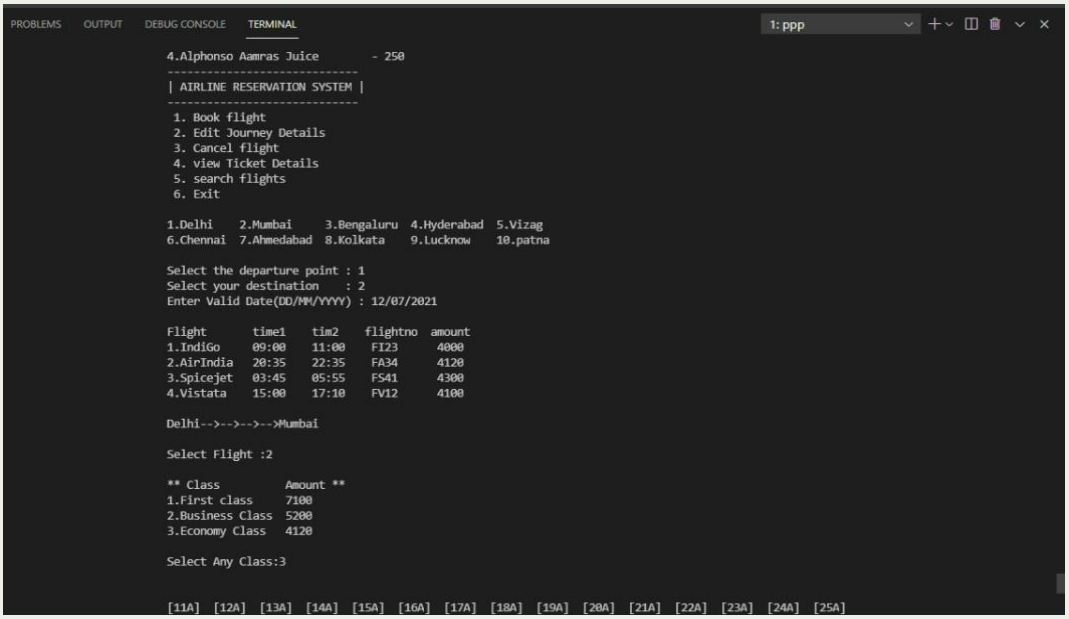
1) **Book the flight ticket:**

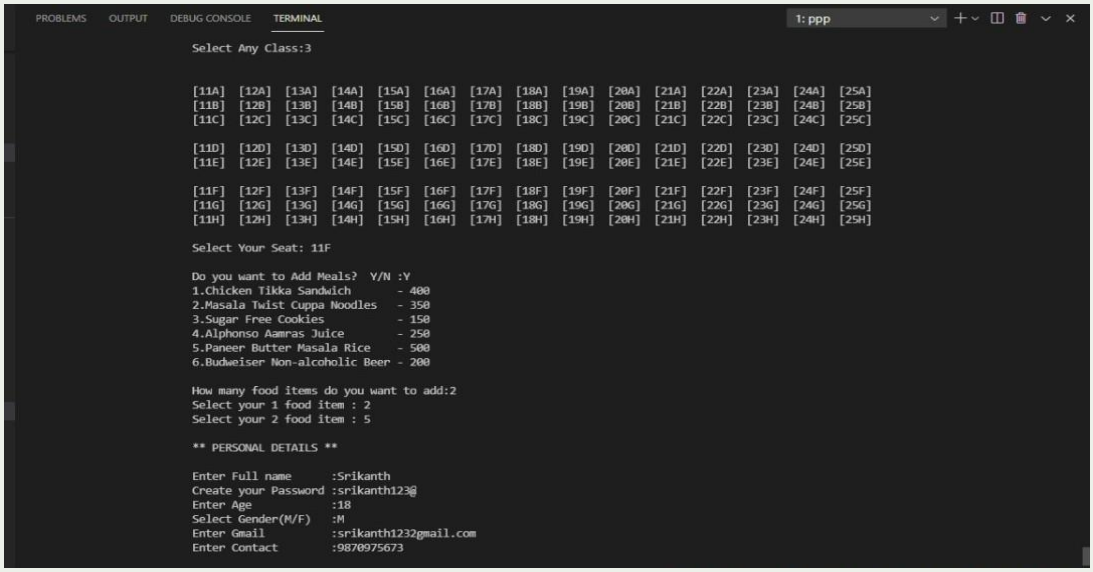
To book the ticket for a flight the user needs to go through 7 steps:

- The user is given two options to select their domain accordingly .

a) Domestic
b) International

- Once the user selected their domain. we provide the information about the available flights to reach their destination .user can select through the availability.
- The user is given 3 classes they can select favorable classes accordingly
  a) First class
  b) Business class
  c) Economy class
- The user can select their seat according to availability and convenience
- The user is given the option to select their meal which can be consumed through their journey. This step is absolutely in the opinion of the user
- In this step, the user should provide their personal information
- Here comes the last phase of booking the flight the user can proceed with the payment process. At the very end, the user receives the payment form with all the billing details.





2) Edit journey Details:

The second phase of the project helps the user to edit their booking in many ways:-

In this option user can change their journey details such as they can change their destination or date or time or class.

- Choosing the 2nd option given to the user and selecting one among domestic and international three options will be displayed:

i. Change_destination :

In this option user can change his/her destination just by entering their name and choosing destination of their choice. If the user gives correct name, then the details will be edited, if not then it displays not found.

ii. Change date:

In this option user can change his/her date just by entering their name and changing date. If the user gives correct name, then their details will be edited, if not then it displays not found.

iii. Change class:

In this option user can change his/her time just by entering their name and entering new class name. If the user gives correct name, then their details will be edited, if not then it displays not found.

iv. Change seat:

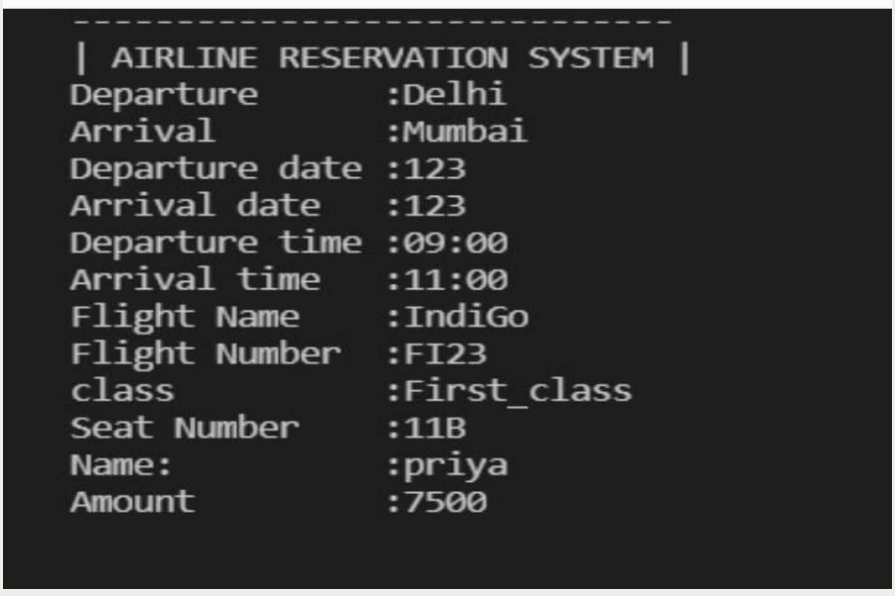
In this option user can change his/her seat just by entering their name and selecting seat according their convince. If user gives their correct name, then their details will be edited, if not then it displays not found.

3)Cancel flight:

- We are providing an option where user can cancel their flight in very convent way.

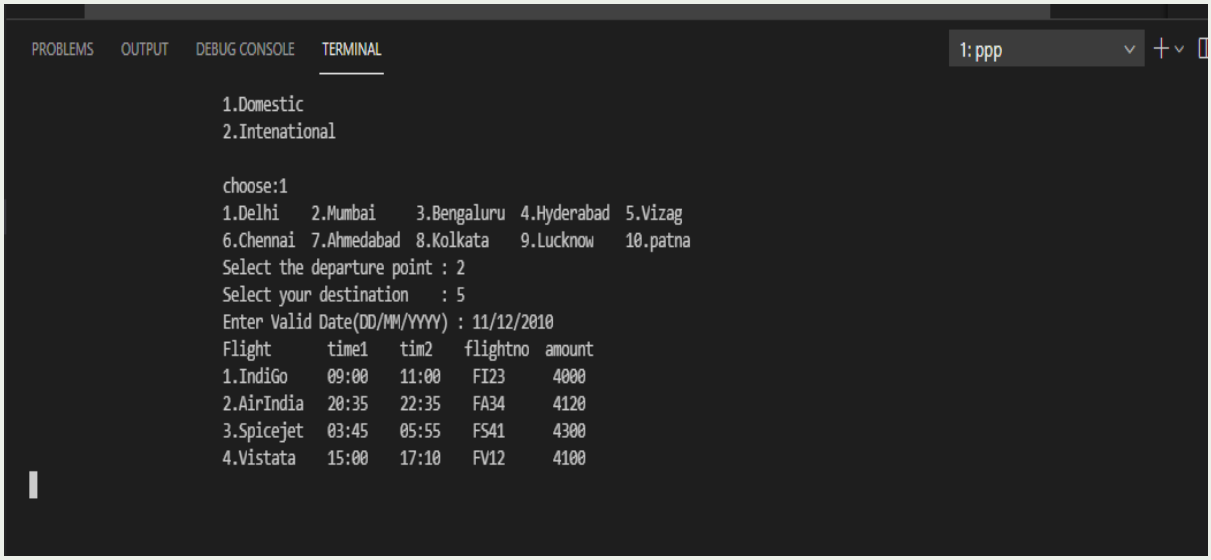
4)View ticket details:

- In this option user can view their ticket details by providing their full name. It displays all details of your journey



5)Search flights:

- In this option we need to first we need to choose domestic or international the departureand destination point and date then list of flights will be displayed.

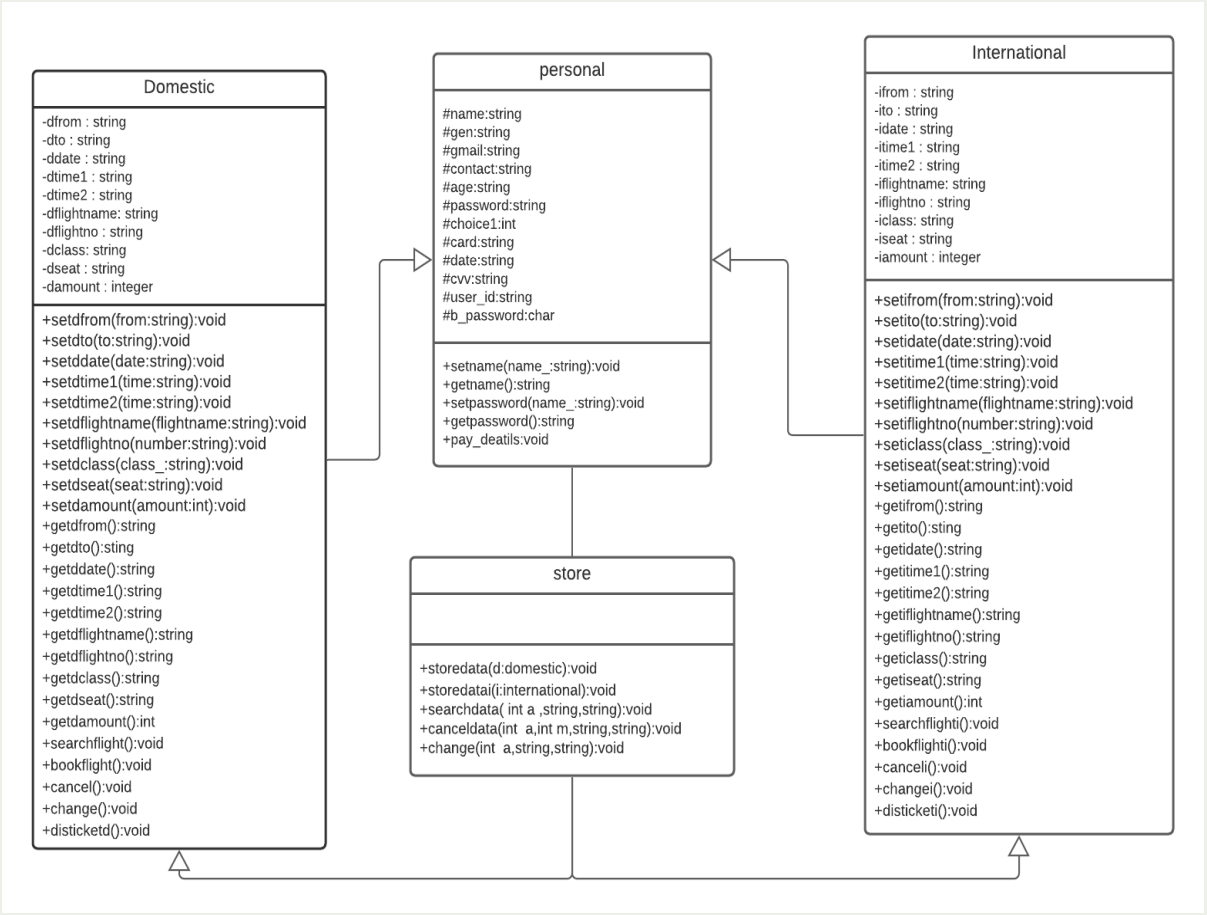


Design :

Structural UML Diagram

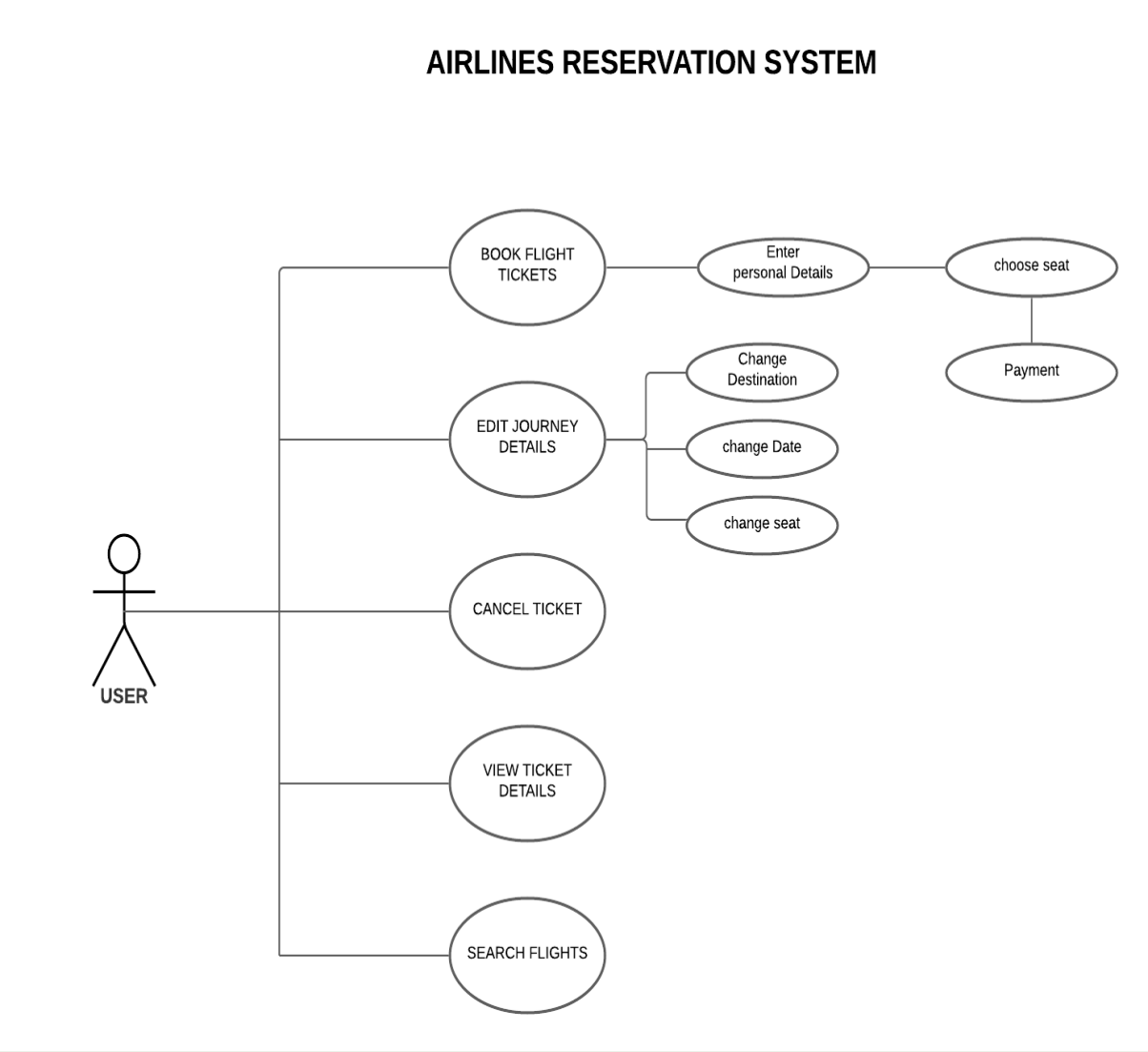
UML CLASS DIAGRAM

This class Diagram includes all the Declarations of the functions and the variables in the code. This defines the structure of the code


Domestic
-dfrom : string
-dto : string
-ddate : string
-dtime1 : string
-dtime2 : string
-dflightname: string
-dflightno : string
-dclass: string
-dseat : string
-damount : integer
+setdfrom(from:string):void
+setdto(to:string):void
+setddate(date:string):void
+setdtime1(time:string):void
+setdtime2(time:string):void
+setdflightname(flightname:string):void
+setdflightno(number:string):void
+setdclass(class_:string):void
+setdseat(seat:string):void
+setdamount(amount:int):void
+getdfrom():string
+getdto():sting
+getddate():string
+getdtime1():string
+getdtime2():string
+getdflightname():string
+getdflightno():string
+getdclass():string
+getdseat():string
+getdamount():int
+searchflight():void
+bookflight():void
+cancel():void
+change():void
+disticketd():void
personal
#name:string
#gen:string
#gmail:string
#contact:string
#age:string
#password:string
#choice1:int
#card:string
#date:string
#cvv:string
#user_id:string
#b_password:char
+setname(name_:string):void
+getname():string
+setpassword(name_:string):void
+getpassword():string
+pay_deatils:void
store
+storedata(d:domestic):void
+storedatai(i:international):void
+searchdata(int a ,string,string):void
+canceldata(int a,int m,string,string):void
+change(int a,string,string):void
International
-ifrom : string
-ito : string
-idate : string
-itime1 : string
-itime2 : string
-iflightname: string
-iflightno : string
-iclass: string
-iseat : string
-iamount : integer
+setifrom(from:string):void
+setito(to:string):void
+setidate(date:string):void
+setitime1(time:string):void
+setitime2(time:string):void
+setiflightname(flightname:string):void
+setiflightno(number:string):void
+seticlass(class_:string):void
+setiseat(seat:string):void
+setiamount(amount:int):void
+getifrom():string
+getito():sting
+getidate():string
+getitime1():string
+getitime2():string
+getiflightname():string
+getiflightno():string
+geticlass():string
+getiseat():string
+getiamount():int
+searchflighti():void
+bookflighti():void
+canceli():void
+changei():void
+disticketi():void

Use Case Diagram:

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



REQUIREMENTS :

- It divides user into 2 categories :
  1. Domestic
  2. International
- Store and display information during booking a ticket:
  1. Departure place of user
  2. Destination of user
  3. Date of journey
  4. Class
  5. Name of user
  6. Age of user(stores but never displays)
  7. Gender of user(stores but never displays)
  8. G mail of user(stores but never displays)
  9. Contact of user(stores but never displays)
- Information stored during payment
  1. Card number
  2. CVV
  3. Expiry date
- Information stored during change of destination or seat or time (or)cancel ticket (or)view details
  1. Personal details
- Tasks that can be performed by user
  1. Book flight ticket
  2. Change destination/date/seat
  3. Cancel tickets
  4. View details
  5. Search flights

Note: Personal details includes Name of passenger and password(which was created in personal details)

conclusion :

while working on the project we have learned and understood the importance of classes. even we gained much knowledge in the concepts of inheritance, abstraction, structured programming.

we have also learned how to divide the work and the importance of teamwork throughout the journey we have experienced and learned a lot of new things which are going to be helpful in our future. Throughout the journey we have experienced and learned a lot of object oriented methodology which are going to be helpful in our future