
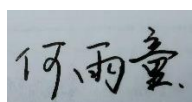
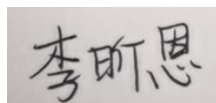
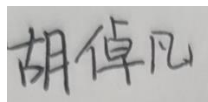


## Group Project

### Team leader

Student ID	1929013	Student Name	Yutong he
------------	---------	--------------	-----------

### Contribution table

Student ID	Student Name (Signed manually)	Tasks	Contribution (%)
1929962	 Hanyuan Ma	the program code	25%
1929013	 Yutong He	The specification and the maintain of the code and document	25%
1927130	 Xinen Li	problem, analysis and test	25%
1822764	 Zhuofan Hu	Design and draw the UML. Analysis each class.	25%
Total			100%

## the document of the code

Group Project.....	1
Specifications:.....	3
Analysis and design:.....	4
management of source code and lifecycle by software tools: .....	15
Test: .....	15
Bug report:.....	18
User manual: .....	18

# Specifications:

## **Overall spec.:**

In this project, to meet the requirement of airways company, our group creates a computer system which can be responsible for a large part of the operation of the airways company. The operations contain: book tickets, change or add the flights information,

## **System spec.:**

First, it matches the user id to the password, if they can be matched properly, then users can log in; otherwise, the users can not log in and achieve the steps after.

After logging in, users should choose the position of them. Therefore, here's the specific functions that meets the requirement of airways company.

## **For administrator:**

1. They can include new flight
2. They can change the price for the flight
3. They can change the dates for flights, which include leaving time and arriving time
4. They can change the source and destination for flights
5. Then can change the plane type for flights

## **For agent:**

1. They can use flight ID to search the flight information
2. They can find the flights which still have seats
3. They can search flights by using leaving time
4. They can search flights by using arriving time
5. They can search flights by using source of the airport
6. They can search flights by using destination of the airport
7. They can book tickets for customer

## **For manager:**

1. They can find the number of the certain type plane
2. They can get the total passenger number in certain flight
3. They can check the revenue of the airport

## **For customer:**

1. They can book seat

# Analysis and design:

## Analysis:

After reading the problem, there are the content we should design:

Enter the system with correct password and account number, then choice the role which you are.

Base on different role, there are different operations can be selected.

## Login system:

### Input:

The account number and password to login, the selection of user's operations.

### Output:

The result of input, the tips which ask people to input.

### Variable:

For the login system, the information of the account should be stored for login, thus, there are a file to store these data.

For the account number and password to login and the selection of user's operations, set them with variable type is "string"

The selection which user input should be a thin integral, set them with variable type is "int"

## Plane:

There are two attributes about "plane", they are the type of plane and the distributions of the seats in different type of plane. Thus, set the "plane" as a "class", the type of plane and the distributions of the seats in different type of plane are set as variable with the type is "string"

## Flight:

Set the "flight" as a class since it has plenty of attributes with it. They are ID, total number of the seat, remain number of the seat, leaving time, arriving time, leaving place, arriving place and price of one flight.

ID, total number of the seat, remain number of the seat, leaving time, arriving time, leaving place, arriving place of flight set as variables with type "string"

For the price of flight, set it as variable with type "double"

There are four types of roles.

### 1. Administrator:

#### In this role, the input should be:

the choice of operations;

for new flight: the information including dates, price, airline, airports and type of flight.

For existing flight: one or more change for the information which are included: dates, price, airports and type of flight, origin and destination.

#### The output should:

The selections of operations, the tips for the wrong input, the tips for the successful operations.

#### The variable should be:

The administrator its own, since there are plenty of different operators, set it as a "class".

The selections should be a number, set it with type of "int"

The dates, seats, type of flight, the ID of flight are string of data, use type "string" as the type of data for them.

The price may not be an integer, thus, use "double" as the type of the data.

The information including dates, price, airline, airports and type of flight should be stored, thus set a file to store them.

## 2. Agents

### Input:

For agents, this input should include: the selection which people choice, the ID, living time, arriving time, departure and the destination of the flight.

### Output:

For agents, the output should include: The selections of operations which agents can do, the result of the operations, the tips which ask people to input, the ID, living time, arriving time, seats, departure and the destination of the flight.

### Variable:

There are plenty of operations which agents can do, thus, set agents with the type "class".

The ID, living time, arriving time, departure and the destination of the flight and the name and ID of passengers are the string, thus, use "string" as the variable type for them. In addition, set them as the local variable in different function blocks.

The selections which people choice is a thin integral, thus, set it as a local variable with the variable type "int".

There are some of data which requires to store to preparing to use or change, thus, set a file to store these data.

## 3. Manager:

### Input:

For manager, the input should include: the operation which user select, the ID and type of the flight.

### Output:

For manager, the output should include: the selections of operations, the tips which ask for the input, the result of operation, the total passenger's number per flight, the total revenue.

### Variable:

The "Manager" can do variety of operations, thus set it as a "class"

The selections which user select should be a thin integral, set it as a local variable with the type is "int"

The ID and type of the flight should be used to get the input from keyboard, set them as local variable with variable type is "string"

The passengers' number and total revenue are used to output the result of them to the screen, set them as local variable and set the variable type "double" for them.

## 4. Customer:

### Input:

The input should include: the ID and name of passenger, the seat which passenger select.

### Output:

The tips which ask user to input, the information of the flight which passenger books, the seat, name and ID of passengers, the result after the operations.

**Variable:**

Since there are some information about passenger which are require to store and change, set a file to store them, set a "class" to input and output these data.

For customer, set a "class" for it to do the operation of booking ticket.

For ID and name of passenger, the seat which passenger select and the ID of the flight, in class "customer", they are used to get the data from keyboard, set them as local variable with variable type is "string"

For ID and name of passenger, the seat which passenger select and the ID of the flight, in class "passenger", they are use to input and output from the file "passenger", set them as the global variable in class "passenger" with the type of variable is "string"

**Design:**

**1. Class design:**

## FileOp

```

*****public*****

//for file planes
int FilePlanesLines();//return how many lines have in the text. each line is a type of plane
int getLineIndex_Planes(string planes_type);//get the line index for the file planes by using planes "type"
string FileOutPlanesType(int line_index);//return the type of each plane e.g. P62
string FileOutPlanesSeatArrangement(int line_index);//return the seat arrangement for each seat e.g. 6*2
void FileInNewPlane(string plane_type, string seat);//create a new plane type
int getPlaneNumber(int line_index);
void changePlaneNumbers(int line_index, int plane_number);//change the number of certain plane
//for file flights
/*flights.dat: flight_ID, delay, price, plane type, total number of seats, number of remain seats,source airport
destination airport, leaving time, arriving time, all seats("A1 A2 A3 A4 A5 A6 B1 B2.....")
*/
int FileFlightsLines();//get how many lines in this dat file
//get()
int getLineIndex_Flights(string flight_ID);//get the line index by find the flight ID
string FileOutFlightsPlaneType(int line_index);//get the type of the flights plane
int FileOutFlightsTotalNumberSeats(int line_index);//get the total number of seats
int FileOutFlightsNumberRemainSeats(int line_index);//get the remain seats for the flights
string FileOutFlightsSourceAirport(int line_index);//get the sourceAirport for the flights
string FileOutFlightsDestinationAirport(int line_index);//get the destination airport
string FileOutFlightsLeavingTime(int line_index);//get the leaving time
string FileOutFlightsArrivingTime(int line_index);// get the arriving time
string FileOutFlightsSeats(int line_index);// get all seats information for the flight
bool FileOutFlightsDelay(int line_index);//get the delay information of the flights
double FileOutFlightsPrice(int line_index);//get the price of the flight
string FileOutFlightsID(int line_index);//get the ID number for flight
//set()
void FileInFlightsNewFlight(string flight_ID, bool delay, double price, string plane_type, int total_seats, int remain_seats,
string source, string destination, string leaving_time, string arriving_time, string Seat);//create a new flights
void FileInFlightsChangeType_Seats_TotalSeats(int line_index, string type, int total_seats, string seats);//change the type of the flight's
plane and total seats
void FileInFlightsChangeRemainSeat(int line_index, int remain_seat);//change the number of remain seats
void FileInFlightsChangeSourceAirport(int line_index, string source);//change the source airport
void FileInFlightsChangeDestinationAirport(int line_index, string destination);//change the destination of the flight
void FileInFlightsChangeLeavingTime(int line_index, string leaving_time);//change the leaving time
void FileInFlightsChangeArrivingTime(int line_index, string arriving_time);//change the arriving time
void FileInFlightsChangeSeats(int line_index, string seat);//change a certian seat
void FileInFlightsChangeDelay(int line_index, bool delay);//change the delay information of the flight
void FileInFlightsChangePrice(int line_index, double price);//change the price of the flight
void FileInFlightsChangeFlightID(int line_index, string flight_ID);//change the ID number of flight

//for file passengers
/*passengers: ID, name, flight_ID, seat number */
int FilePassengersLines();//get the line of the passengers.dat
//get()
int getLineIndex_Passengers(string passenger_ID);//get the line index by find the passenger ID
string FileOutPassengersID(int line_index);//get the passenger ID
string FileOutPassengersSeatNumber(int line_index);// get the seat number(e.g. A2)
string FileOutPassengersFlight_ID(int line_index);//get the flight ID in passenger
string FileOutPassengersName(int line_index);//get the passenger name
//set()
void FileInPassengersNewPassengers(string ID, string name, string flight_ID, string seat_number);//create a new passenger
void FileInPassengersChangePassenger_ID(int line_index, string passenger_ID);//change passenger's ID
void FileInPassengersChangeName(int line_index, string name);//change passenger's name
void FileInPassengersChangeFlight_ID(int line_index, string flight_ID);//change the passengers Flight_ID
void FileInPassengersChangeSeatNumber(int line_index, string seat_number);//change passenger's seat number

//for file password
/*password: ID, password, permission*/
int FilePasswordLines();
int getLineIndex_Password(string user_ID);//get the line index by find the user ID
bool matchID_Password(int line_index, string password);//match the ID with password
string getPermission(int line_index); // by using ID to find the user's permission
string getUser_ID(int line_index);
void setNewID_Password(string ID, string password, string permission);//create new account
void change_password(int line_index, string password);//change the password

//for file agents
/*agent ID, time*/
int FileAgentsLines();
void createNew(string agent_ID, string local_time);//creat a new line
bool matchID(int line_index, string ID);
int getSoldTicketsNumber(string agent_ID, string first, string final);// how many tickets agent sold during a particular duration

```

## flights

\*\*\*\*\*protect\*\*\*\*\*

```
planes plane;
string flight_ID = ""; //primary key for flights
int total_seat_number=-1; //the total number of seats
int remain_seat_number=-1; //the number of remain seats
string source=""; // the source airport
string destination=""; // the destination airport
string leaving_time=""; // the time that the flight leave the source airport
string arriving_time=""; // the time that the flight arrive the destination airport
string seat = ""; //record seat number and whether it is been record. A2--not booked A2@--booked
bool delay=false; // show whether the flight is dealy
double price = 0; //the price of the flights
```

\*\*\*\*\*public\*\*\*\*\*

```
flights(); //default constructor
flights(planes Plane, string Source, string Destination,
        string Leaving_time, string Arriving_time, string Seat, double Price, string Flight_ID, bool Delay); //default flight
/*flights(planes Plane, int Total_seat_number, int Remain_seat_number, string Source,
        string Destination, string Leaving_time, string Arriving_time, bool Dealy);*/
void showFlightInformation(); // show all information of the flight
void operator =(flights a);
//java habit
//get()
int getTotalSeatNumber() { return total_seat_number; }
int getRemainSeatNumber() { return remain_seat_number; }
string getSource() { return source; }
string getDestination() { return destination; }
string getLeavingTime() { return leaving_time; }
string getArrivingTime() { return arriving_time; }
string getSeat() { return seat; }
bool whetherLeaving();
double getPrice() { return price; }
//set()
void setTotalSeatNumber(int Total_seat_number) { total_seat_number = Total_seat_number; }
void setRemainSeatNumber(int Remain_seat_number) { remain_seat_number = Remain_seat_number; }
void setSource(string Source) { source = Source; }
void setDestination(string Destination) { destination = Destination; }
void setLeavingTime(string Leaving_time) { leaving_time = Leaving_time; }
void setArrivingTime(string Arriving_time) { arriving_time = Arriving_time; }
void setSeat(string seat_number, string seat_deed); //change the seat information for the flight
void setPrice(double Price) { price = Price; }
void setAll(planes Plane, string Source, string Destination,
            string Leaving_time, string Arriving_time, string Seat, double Price, string Flight_ID, bool Delay);
```



administrator
*****public*****  void includeNewFlight();//include new flights void changePrice();//change the price for flights void changeDates();//change the leaving time or arriving time, also it can change both of them void changeAirport();//change source or destination, also both of them void changePlaneType();//change the type of plane for flights

agents
*****public*****  void searchFlights_flightID(); void searchFlights_haveRemainSeat(); void searchFlights_LeavingTime(); void searchFlights_ArrivingTime(); void searchFlights_Source(); void searchFlights_Destination(); void bookTicket();

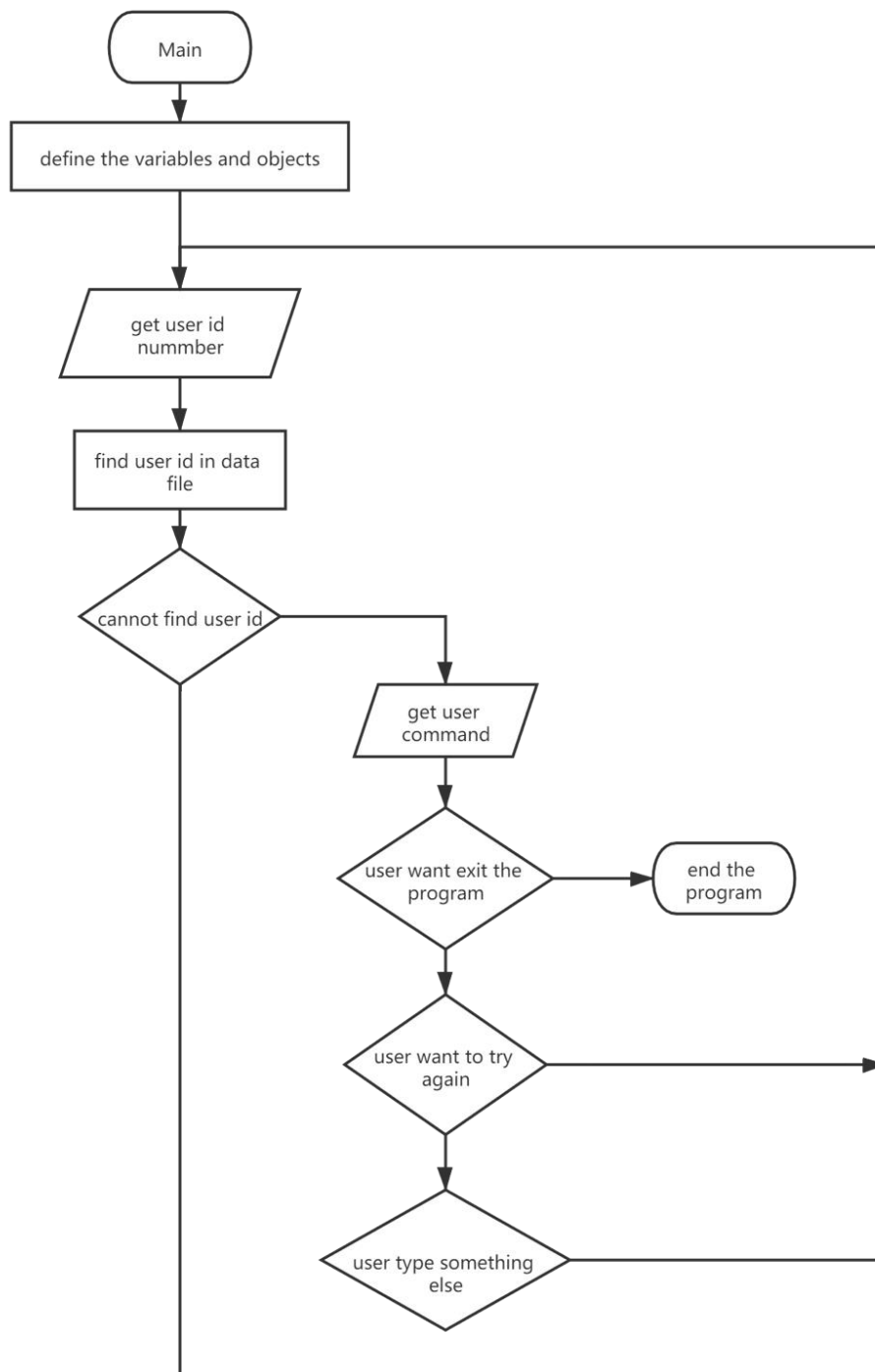
manager
*****public*****  void findNumberPlane(); void totalPassengers_certainFlight(); void revenue();

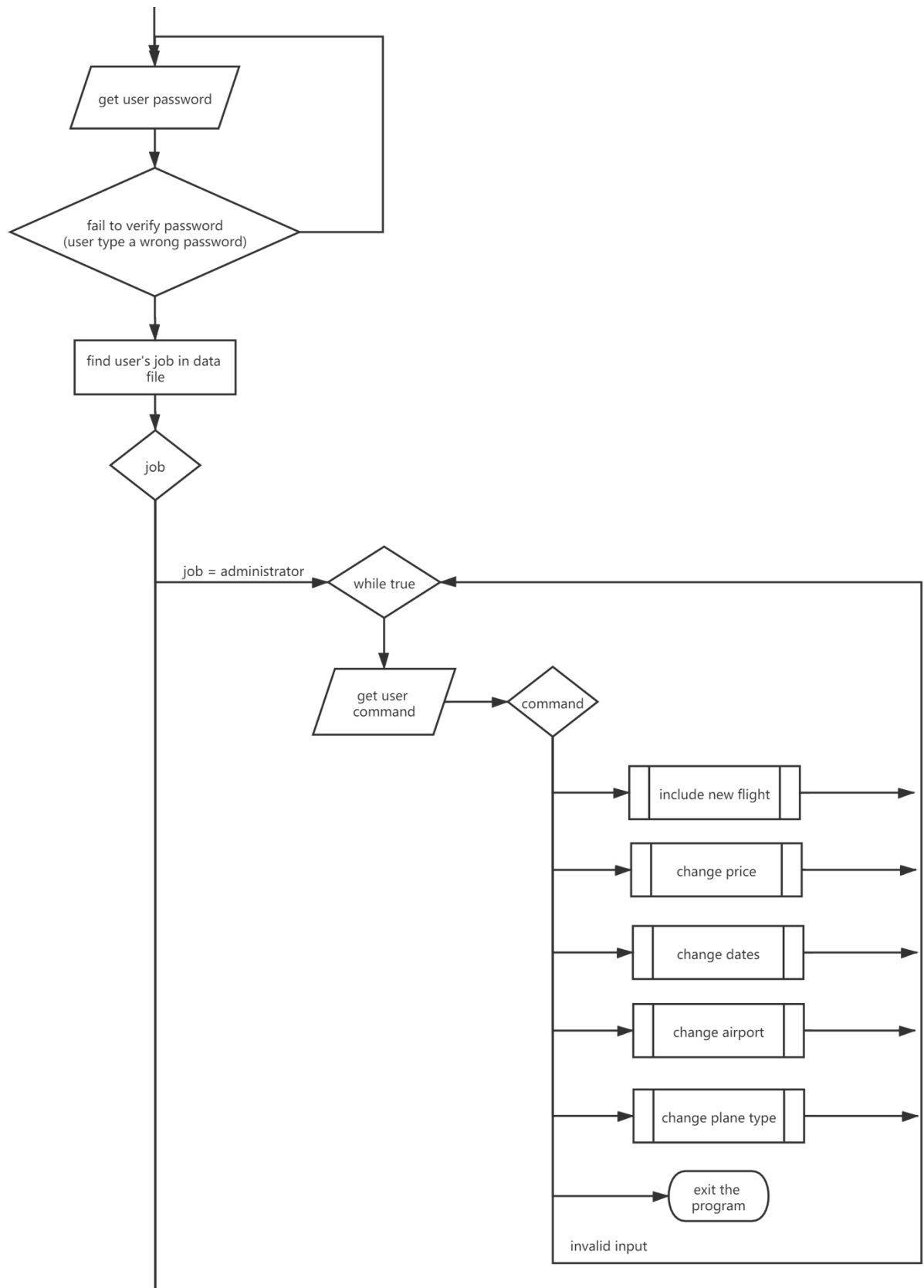
customer
<pre> *****public***** void bookSeat(); </pre>

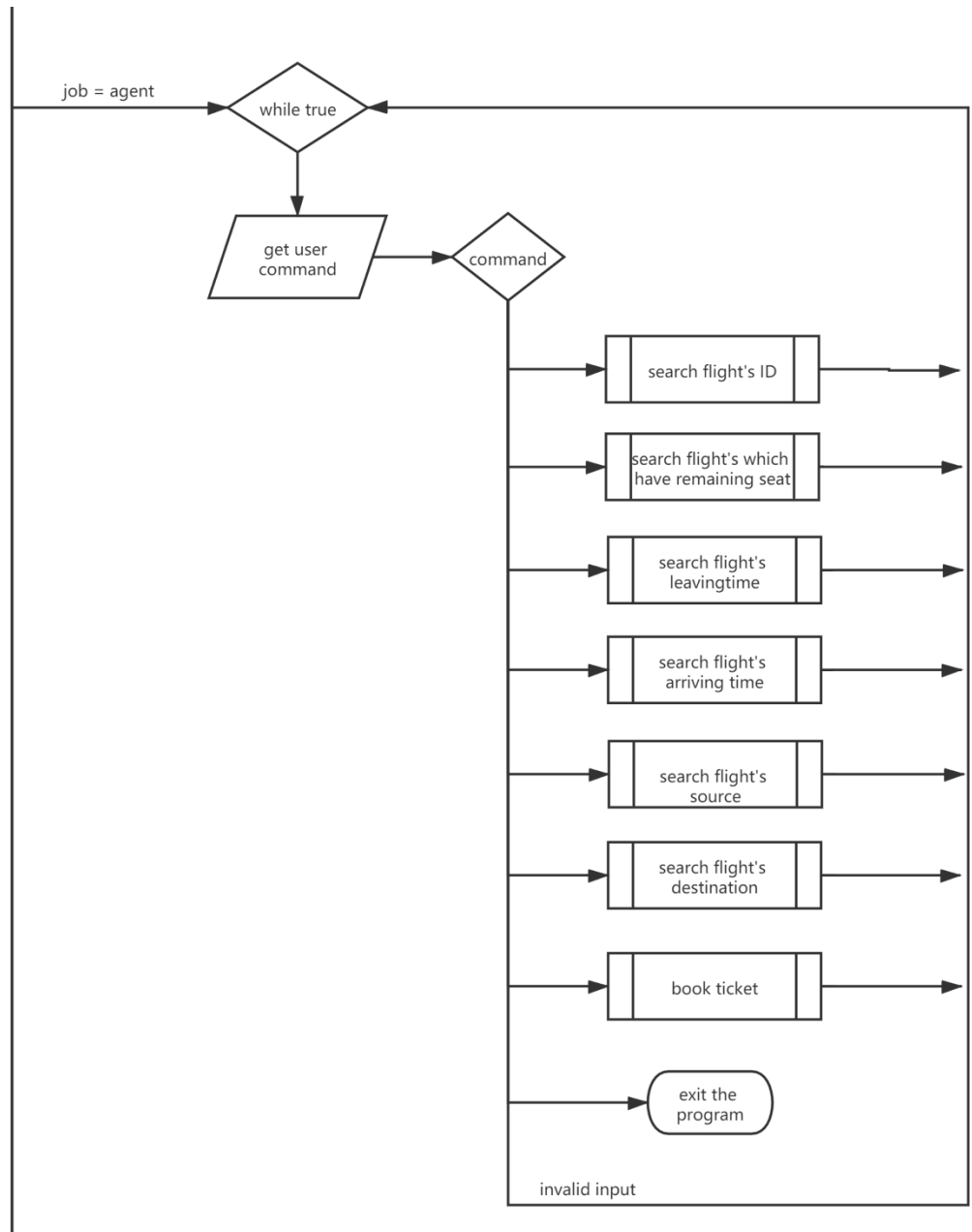
passengers
<pre> *****protect*****  flights flight;//the flight information for this passsenger string seat_number; string ID;//the only unique identification code string name;//passenger name </pre>
<pre> *****public*****  passengers();//default constructor passengers(flights Flight, string Seat_number, string id, string Name);//constructor void showPassengerInfo();// show passenger information //get string getSeatNumber() { return seat_number; } string getID() { return ID; } string getName() { return name; } //set void setSeatNumber(string input) { seat_number = input; } void setID(string input) { ID = input; } void setName(string input) { name = input; } </pre>

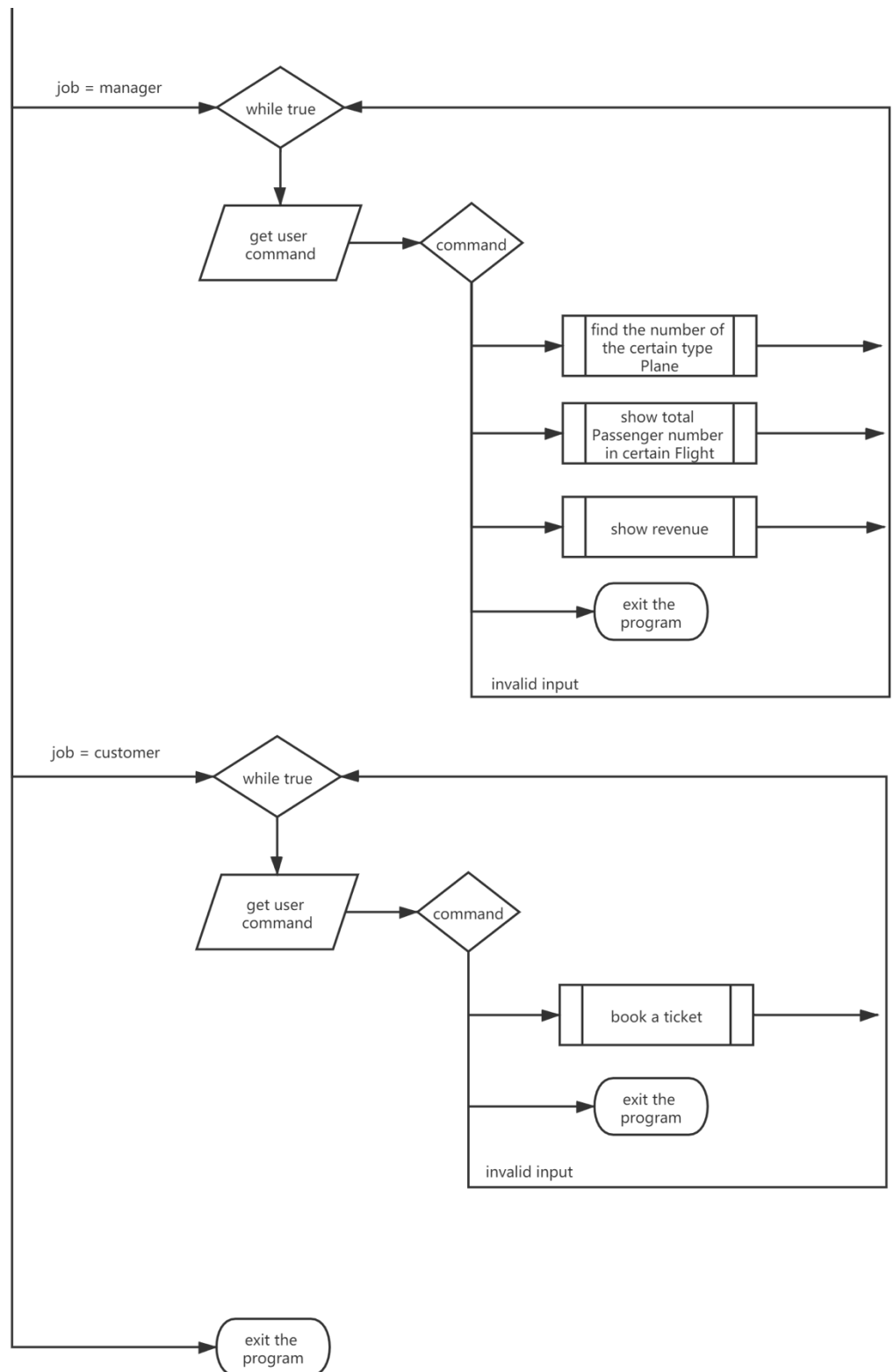
planes
<pre> *****protect*****  string type = "";//the type of the plane---P62 and P124 string seat arrangement = "";// show all seats for the certain plane 6*2 &amp; 12*4 </pre>
<pre> *****public*****  planes();//DEFAULT CONSTRUCTOR planes(string TYPE, string Seat_arrangement);//constructor string getPlaneType() { return type; };//get the type of plane void showPlaneSeatArrangement();//print a figure to let user know the set arrangement of plane //using java "get" and "set" habit void setType(string input) { type = input; } void setSeatArrangement(string input) { seat_arrangement = input; } string getType() { return type; } string getSeatArrangement() { return seat_arrangement; } void operator=(planes a);// overloping operator "=" </pre>

## 2. Main loop design:









# management of source code and lifecycle by software tools:

## **management of source:**

our group use GitHub to maintain our code.

The website addresses which can read the code and document are shown as below:

The address of the code:

The address of the document:

## **lifecycle of this program:**

this program is designed due to the requirement of the airport company, which is convenient for the company to maintain all the information of users and flights. Moreover, after the code is maintained in GitHub, the programmer can see all the changes after it was uploaded. Since it's an open-source project, everyone can see the codes. Therefore, if there are some bugs and insufficient in the program and others users report them, the programmer can change it instantly. Maybe with the develop of machine learning and AI technology, this code don't need to be improved by people, which means the information can be inputted automatically by computer and then we need to use another updated code. So, after this, the lifecycle of this product is finished.

## **Test:**

The testing is showed in below:

**For administration:**

```
CAUsers\DELL\Desktop\CPT106Solution\CPT106Solution\Debug\Assessment4.exe
Please input your id number in order to distribute accessibility
user id:1033891923

Please input password:hehaw2314
You have successfully login
Please choice operator
1. Include New Flight
2. Change Price for flight
3. Change Dates for flights(leaving time and arriving time)
4. Change Airport for flights(source and destination)
5. Change Plane Type for flights
6. Quit this permission
2
Please input the flight ID:XH0001-1
Please input the new price for this flight:200
The price has been successfully changed.
Please choice operator
1. Include New Flight
2. Change Price for flight
3. Change Dates for flights(leaving time and arriving time)
4. Change Airport for flights(source and destination)
5. Change Plane Type for flights
6. Quit this permission
3
Please input the flight ID:XH0001-1
Please chose whihc date you want to change(input 1 or 2 or 3)
1. change arriving time
2. change leaving time
3. change both of them
3
Please input the new leaving time for this flight(YYYY-MM-DD HH:MM):2021-6-9 19:20_
```

```
CAUsers\DELL\Desktop\CPT106Solution\CPT106Solution\Debug\Assessment4.exe
user id:1033891923

Please input password:hehaw2314
You have successfully login
Please choice operator
1. Include New Flight
2. Change Price for flight
3. Change Dates for flights(leaving time and arriving time)
4. Change Airport for flights(source and destination)
5. Change Plane Type for flights
6. Quit this permission
2
Please input the flight ID:XH0001-1
Please input the new price for this flight:200
The price has been successfully changed.
Please choice operator
1. Include New Flight
2. Change Price for flight
3. Change Dates for flights(leaving time and arriving time)
4. Change Airport for flights(source and destination)
5. Change Plane Type for flights
6. Quit this permission
3
Please input the flight ID:XH0001-1
Please chose whihc date you want to change(input 1 or 2 or 3)
1. change arriving time
2. change leaving time
3. change both of them
3
Please input the new leaving time for this flight(YYYY-MM-DD HH:MM):2021-6-9 19:20
The leaving time has been successfully changed.
Please input the new arriving time for this flight(YYYY-MM-DD HH:MM):2021-6-9 21:20
The arriving time has been successfully changed.
Please choice operator
1. Include New Flight
2. Change Price for flight
3. Change Dates for flights(leaving time and arriving time)
4. Change Airport for flights(source and destination)
5. Change Plane Type for flights
6. Quit this permission
```

For Agent:



```
CAUsers\DELL\Desktop\CPT106Solution\CPT106Solution\Debug\Assessment4.exe
1. using flight ID to search
2. find all flights that still have seats
3. search Flights by using Leaving Time
4. search Flights by using Arriving Time
5. search Flights by using Source Airport
6. search Flights by using Destination Airport
7. book A Ticket
8. quit this permission
2
Press Enter to start search请按任意键继续. . .
flight ID: XH0001-1



| plane type        | plane seat                            |
|-------------------|---------------------------------------|
| P62               | A1 A2@ A3 A4 A5 A6@ B1 B2 B3 B4 B5 B6 |
| total seat number | remain seat number                    |
| 12                | 12                                    |
| source            | destination                           |
| PEK               | SHA                                   |
| leaving time      | arriving time                         |
| 2021-7-18 10:35   | 2021-7-18 13:20                       |
| situation         |                                       |
| on time           |                                       |


*****
flight ID: XH0001-2



| plane type        | plane seat                          |
|-------------------|-------------------------------------|
| P124              | A1 A2 A3 A4 A5 A6 B1 B2 B3 B4 B5 B6 |
| total seat number | remain seat number                  |


```

## For manager:

```
Microsoft Visual Studio 调试控制台
Please input the type of plane:P62
the total number for this type of plane is 231Please choice operator
1. find the number of the certain type Plane
2. total Passenger number in certain Flight
3. revenue
4. quit this permission
2
Please input the ID number of flight:1234234
the input ID is not matching
Press 1 to exist
Press 2 to try again
1
Please choice operator
1. find the number of the certain type Plane
2. total Passenger number in certain Flight
3. revenue
4. quit this permission
3
Press Enter to calculate the revenue
请按任意键继续. . .
the revenue is 221709Please choice operator
1. find the number of the certain type Plane
2. total Passenger number in certain Flight
3. revenue
4. quit this permission
4
C:\Users\DELL\Desktop\CPT106Solution\CPT106Solution\Debug\Assessment4.exe (进程 19152)已退出, 代码为 0。
请按任意键关闭此窗口. . .
```

## For customer:

```
CAUsers\DELL\Desktop\CPT106Solution\CPT106Solution\Debug\Assessment4.exe
Welcome to the Airways Management System
please input your id number in order to distribute accessability
user id:1023172737

Please input password:uebfi3422
You have successfully login
Please choice operator
1. book A Seat
2. quit this permission
1
Press Enter to start booking
请按任意键继续. . .
Passenger ID:12004987320
Passenger name:Xian.An
Flight ID:XH0001-1
Seat:A5
Pleasing waiting for booking
Your booking has been done
Please choice operator
1. book A Seat
2. quit this permission
```

# Bug report:

There exist several bugs when I write this program. This part will discuss what these bugs are and how I fixed them.

I use type “int” to get the input in the interactive system. But there exists a problem when user input a string type value, such as “a”. The program will get into drop-dead halt. In order to find why this problem happened. I use the “step in” debugging method to find the reason. By looking at the memory and the value of the input. I find that the problem is the “cin”. The “cin” stream has some problem. Comparing “cin” stream with string stream, I decided that the problem is I didn’t clear the “cin” stream and clear the buffer. This cause the problem of the drop-dead halt. So we use function “clear()” and “Clear\_buffer()” to deal. However the function “Clear\_buffer()” is not exist in the standard library. The code for this function is shown below.

```
void Clear_buffer()
{
    int count = 0;
    char c;
    while ((c = getchar()) != '\n');
    return;
}
```

(This solved solution will not be discussed in this report because it seems a bit complex)

# User manual:

1. Administrator:

Step	Describe						
1	Input ID and passwords						
2	Input correspond number to choose command <table><tr><td>1</td><td>Include new flight</td></tr><tr><td>2</td><td>Change price for flight</td></tr><tr><td>3</td><td>Change</td></tr></table>	1	Include new flight	2	Change price for flight	3	Change
1	Include new flight						
2	Change price for flight						
3	Change						

		dates for flights	
	4	Change airport for flights	
	5	Change plane type for flights	
	6	Quit	

## 2. Agent

Step	Describe	
1	Input ID and students	
2	Input correspond number to choose command	
	1	Search by flight ID
	2	Search flights which still have seats
	3	Search flight by leaving time
	4	Search flight by arriving time
	5	Search flight by source airport
	6	Search flight by destination airport
	7	Book a ticket
8	Quit	

## 3. Manager

Step	Describe
------	----------

1	Input ID and students	
2	Input correspond number to choose command	
	1	Find the number of certain type plane
	2	Total passenger number in certain flight
	3	Show revenue
	4	Quit

#### 4. Customer

Step	Describe	
1	Input ID and students	
2	Input correspond number to choose command	
	1	Book a ticket
	2	quit