Flight reservation system

1.0 Introduction

Recently, the aviation industry has developed rapidly. Since there are more and more people choose to take planes, the flight reservation system becomes more important. To facilitate passenger travel, we plan to design a flight booking system. Passengers can inquire the flight through the online booking platform, and this system can arrange flights for passengers through the information systems such as origin, destinations and departure time. Passengers can choose flights according to the departure time and ticket price. If the booking is successful, the system will generate order information for passengers. The passenger can check their reservation information on the personal information platform.

2.0 Functional requirement

2.1 Requirement Analysis

After analysis, our flight reservation system mainly includes the following several functions:

(1) User interface

Enquiries: users query the flight information;

Sorting: users sort and filter query results;

Booking: processing and recording the reservation information of passengers and update

Refund: processing the refund application from users and updating database

View personal information: users view their personal ticketing information.

Exit: closing page.

(2) Administrator interface

Flight information management: the create, read, update and delete operation of flight information

Passenger information management: the create, read, update and delete operation of

passengers' information

Exit: close page.

2.2 Soft Function Design

(1) System primary function

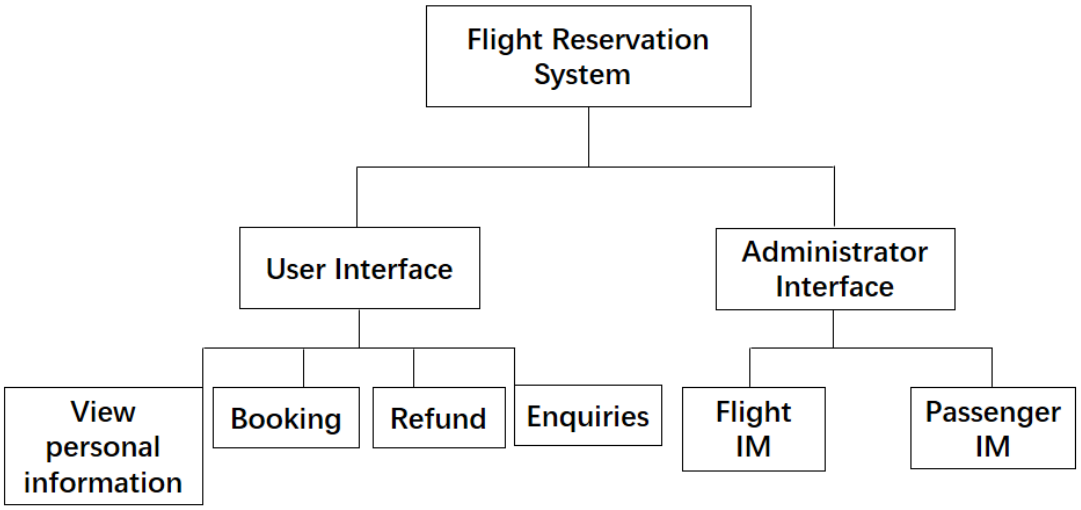


Figure 2.2-1

(2) Data Flow Diagram

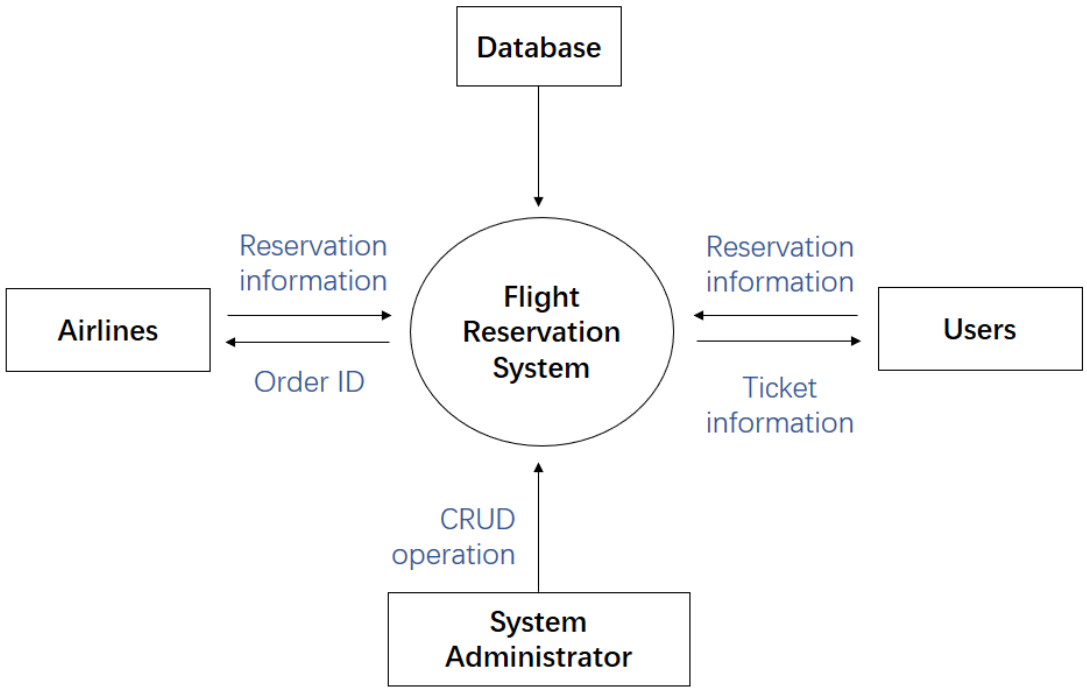
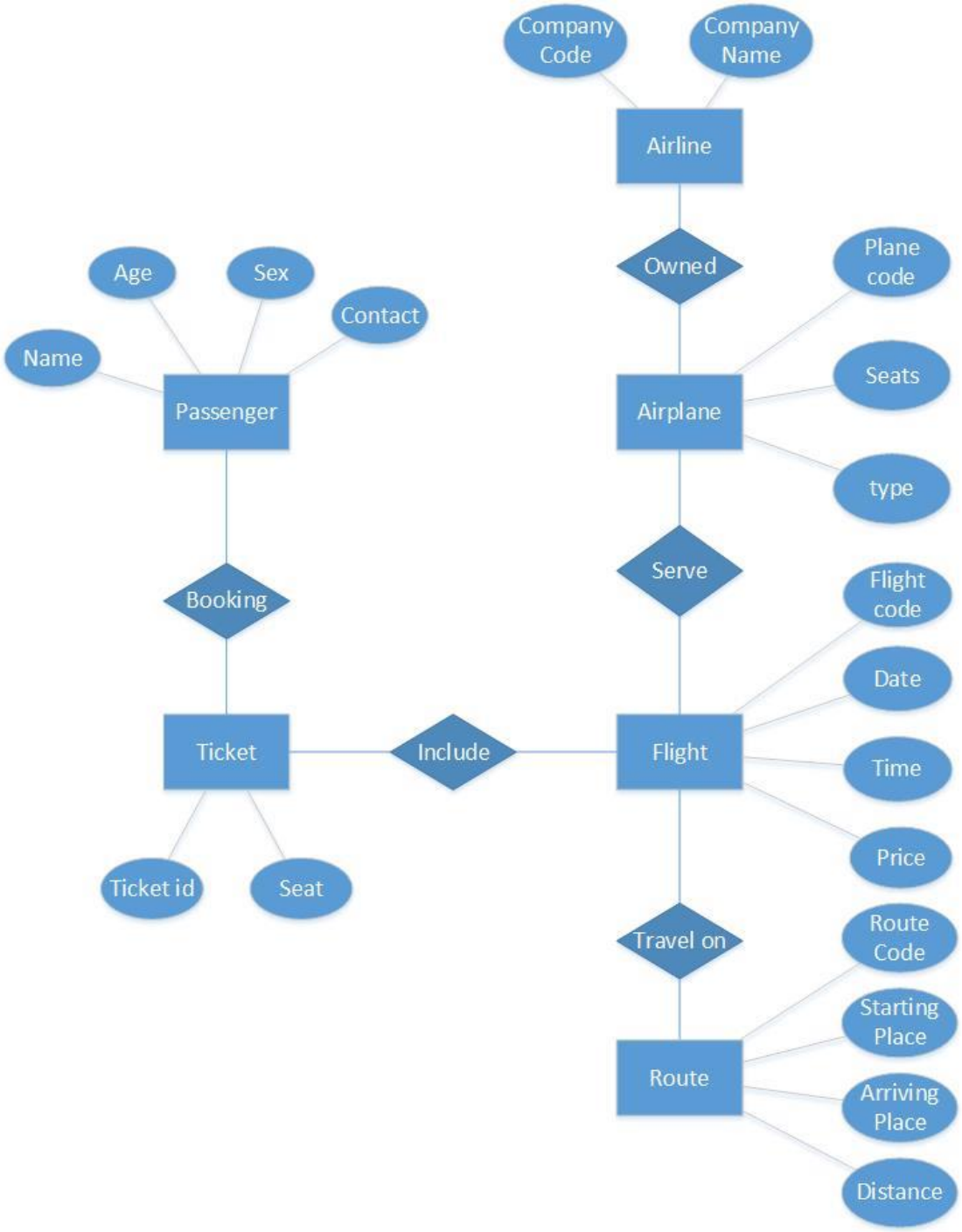


Figure 2.2-2

3.0 Database Structure Design

3.1 ER Diagram of the Database

According to the Demand analysis, the information structure can be obtained and form the ER diagram.



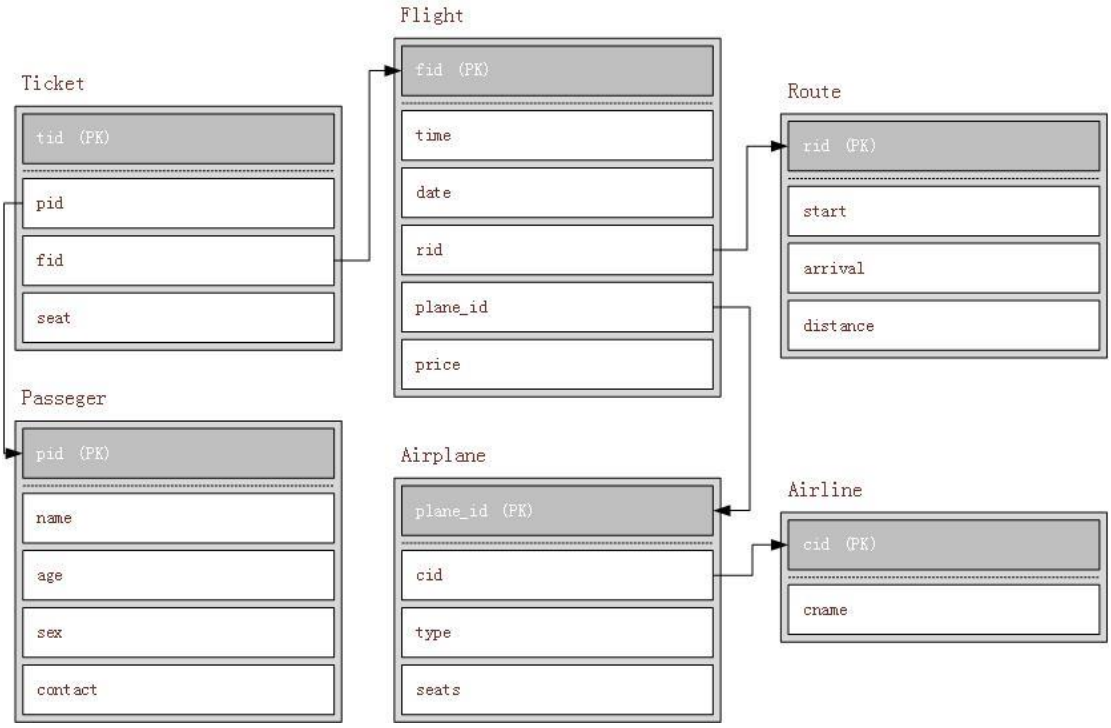
3.2 Translation to Relational Schema

The ER diagrame can be translation to a relational schema:

Passenger (pid, name, age, sex, contact)
Ticket (tid, pid, fid, seat)
Flight (fid, time, date, rid, plane_id, price)
Route (rid, start, arrival, distance)
Airline (cid, cname)
Airplane (plane_id, cid, type, seats)

3.3 Functional dependencies

Optimaize the translation relational schema, and determine the fuctional dependencies.



Ticket (tid, pid, fid, seat)
pid is a Foreign key from relation passenger, attribute pid.
fid is a Foreign key from relation flight, attribute fid.

Flight (fid, time, date, rid, plane_id, price)
rid is a Foreign key from relation route, attribute rid.
plane_id is a Foreign key from relation airplane, attribute plane_id.

Airplane (plane_id, cid, type, seats)
cid is a Foreign key from relation airline, attribute cid.

3.4 Database Structure

According to the overall database structure, the basic table structure of flight

reservation system is designed, and the corresponding target is defined as follows.

Table1 Passenger Information

Attribute	Type	Length	Can be null or not	explain
Pid	varchar	255	primary key	identity number of passenger
Name	varchar	255	not null	name of passenger
Age	int	11	not null	age of passenger
Sex	varchar	255	not null	sex of passenger
Contact	varchar	255	not null	contact of passenger

Table2 Ticket Information

Attribute	Type	Length	Can be null or not	explain
tid	varchar	255	primary key	identity number of ticket
pid	varchar	255	not null	identity number of passenger
fid	varchar	255	not null	identity number of flight
seat	varchar	255	not null	seat number

Table3 Flight Information

Attribute	Type	Length	Can be null or not	explain
fid	varchar	255	primary key	identity number of flight
time	time	0	not null	starting time
date	date	0	not null	starting date
rid	varchar	255	not null	flight route
plane_id	varchar	255	not null	identity number of plane
price	int	11	not null	ticket price

Table4 Route Information

Attribute	Type	Length	Can be null or not	explain
rid	varchar	255	primary key	identity number of route
start	varchar	255	not null	starting city
arrival	varchar	255	not null	arriving city
distance	float	0	not null	distance of the route

Table5 Airline Information

Attribute	Type	Length	Can be null or not	explain
cid	varchar	255	primary key	identity number of company

cname	varchar	255	not null	company name

Table6 Airplane Information

Attribute	Type	Length	Can be null or not	explain
plane_id	varchar	255	primary key	identity number of plane
cid	varchar	255	not null	identity number of owning company
type	varchar	255	not null	type of the plane
seats	int	11	not null	seats number of the plane

4.0 Systems Implementation

4.1 Database Implementation

First creat the database of flight reservation system, and then creat the six basic table in the database structure. The specific SQL code is showed as follows:

(1)creat the database

```
CREATE DATABASE flight;
```

(2) create the table of airline information

```
CREATE TABLE `airline` (
   `cid` varchar(255) NOT NULL DEFAULT '',
   `cname` varchar(255) NOT NULL,
   PRIMARY KEY (`cid`),
   KEY `cname` (`cname`)
);
```

(3) create the table of airplane information

```
CREATE TABLE `airplane` (
   `plane_id` varchar(255) NOT NULL,
   `cid` varchar(255) NOT NULL,
   `type` varchar(255) NOT NULL,
   `seats` int(11) NOT NULL,
   PRIMARY KEY (`plane_id`),
   KEY `cccc` (`cid`),
   CONSTRAINT `cccc` FOREIGN KEY (`cid`) REFERENCES `airline` (`cid`) ON
DELETE CASCADE ON UPDATE CASCADE
);
```

(4) create the table of route information

```
CREATE TABLE `route` (
   `rid` varchar(255) NOT NULL,
   `start` varchar(255) NOT NULL,
   `arrival` varchar(255) NOT NULL,
   `distance` float NOT NULL,
   PRIMARY KEY (`rid`)
);
```

(5) create the table of flight information

```
CREATE TABLE `flight` (
   `fid` varchar(255) NOT NULL,
```

```
  `time` time NOT NULL,
  `date` date NOT NULL,
  `rid` varchar(255) NOT NULL,
  `plane_id` varchar(255) NOT NULL,
  `price` int(11) NOT NULL,
  PRIMARY KEY (`fid`),
  KEY `flightrid` (`rid`),
  KEY `flightpl` (`plane_id`),
  CONSTRAINT `flightpl` FOREIGN KEY (`plane_id`) REFERENCES `airplane` (`plane_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `flightrid` FOREIGN KEY (`rid`) REFERENCES `route` (`rid`) ON DELETE CASCADE ON UPDATE CASCADE
);
```

(6) create the table of passenger information

```
CREATE TABLE `passenger` (
  `pid` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `age` int(11) NOT NULL,
  `sex` varchar(255) NOT NULL,
  `contact` varchar(255) NOT NULL,
  PRIMARY KEY (`pid`)
);
```

(7) create the table of ticket information

```
CREATE TABLE `ticket` (
  `tid` varchar(255) NOT NULL,
  `pid` varchar(255) NOT NULL,
  `fid` varchar(255) NOT NULL,
  `seat` varchar(255) NOT NULL,
  PRIMARY KEY (`tid`),
  KEY `tf` (`fid`),
  KEY `tp` (`pid`),
  CONSTRAINT `tf` FOREIGN KEY (`fid`) REFERENCES `flight` (`fid`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `tp` FOREIGN KEY (`pid`) REFERENCES `passenger` (`pid`) ON DELETE CASCADE ON UPDATE CASCADE
);
```

4.2 Data Loading

(1) Insert the data into table airline

```
INSERT INTO `airline` VALUES ('A4806', 'AAB_Alirline');
```

```
INSERT INTO `airline` VALUES ('A0874', 'AAH_Alirline');
INSERT INTO `airline` VALUES ('A3753', 'AAH_Alirline');
INSERT INTO `airline` VALUES ('A4913', 'AAK_Alirline');
INSERT INTO `airline` VALUES ('A1769', 'AAM_Alirline');
…
```

(2) Insert the data into table airplane

```
INSERT INTO `airplane` VALUES ('AP0001', 'A0845', '777', '240');
INSERT INTO `airplane` VALUES ('AP0002', 'A3726', '757', '100');
INSERT INTO `airplane` VALUES ('AP0003', 'A2386', 'A320', '250');
INSERT INTO `airplane` VALUES ('AP0004', 'A3268', '777', '260');
INSERT INTO `airplane` VALUES ('AP0005', 'A4833', 'M82', '130');
…
```

(3) Insert the data into table flight

```
INSERT INTO `flight` VALUES ('F0001', '20:44:00', '2018-06-02', 'R2498', 'AP4074',
'3277');
INSERT INTO `flight` VALUES ('F0002', '15:44:00', '2018-06-04', 'R1929', 'AP4529',
'3472');
INSERT INTO `flight` VALUES ('F0003', '05:10:00', '2018-11-25', 'R1511', 'AP0635',
'4329');
INSERT INTO `flight` VALUES ('F0004', '13:34:00', '2018-02-26', 'R3888', 'AP4567',
'5211');
INSERT INTO `flight` VALUES ('F0005', '07:04:00', '2018-12-30', 'R0624', 'AP3162',
'1027');
…
```

(4) Insert the data into table passenger

```
INSERT INTO `passenger` VALUES ('P0001', 'Andriu', '24', 'F', '14530192012');
INSERT INTO `passenger` VALUES ('P0002', 'Humph', '57', 'M', '10018585115');
INSERT INTO `passenger` VALUES ('P0003', 'Geoffrey', '50', 'F', '17417413005');
INSERT INTO `passenger` VALUES ('P0004', 'Gipp', '26', 'M', '12054232319');
INSERT INTO `passenger` VALUES ('P0005', 'Evelune', '47', 'M', '13469515966');
…
```

(5) Insert the data into table route

```
INSERT INTO `route` VALUES ('R0001', 'WSA', 'TBH', '6842.98');
INSERT INTO `route` VALUES ('R0002', 'BZF', 'KKX', '3577.48');
INSERT INTO `route` VALUES ('R0003', 'KJS', 'TXW', '3375.99');
INSERT INTO `route` VALUES ('R0004', 'VRK', 'CZX', '1361.49');
INSERT INTO `route` VALUES ('R0005', 'OSN', 'GIU', '1296.58');
…
```

(6) Insert the data into table ticket

```
INSERT INTO `ticket` VALUES ('T00001', 'P0920', 'F3457', '153C');
INSERT INTO `ticket` VALUES ('T00002', 'P0244', 'F1433', '116A');
INSERT INTO `ticket` VALUES ('T00003', 'P2641', 'F2719', '115F');
INSERT INTO `ticket` VALUES ('T00004', 'P3289', 'F4645', '92F');
INSERT INTO `ticket` VALUES ('T00005', 'P1650', 'F1878', '258G');
…
```

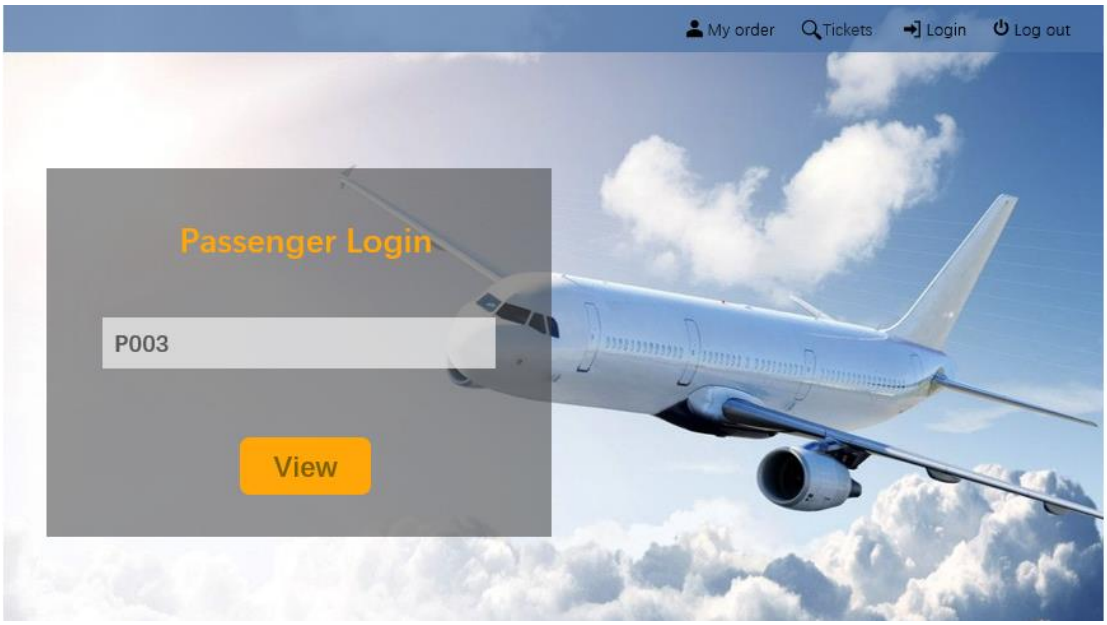
5.0 Run and Test

According to our funtional requirement, we test our flight reservation system in the following functions:

5.1 View personal information

Passenger inquiries about Personal Booking from the user interface:

Input the passenger id 'P003'.



With the SQL code:

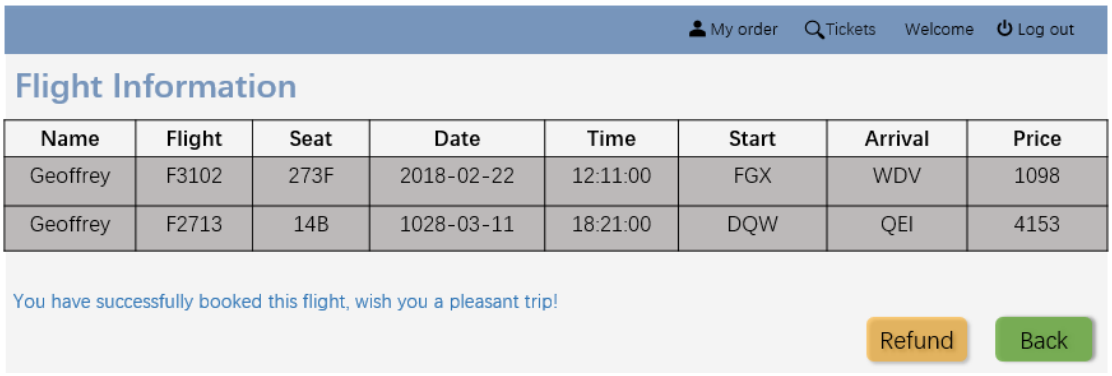
```
SELECT
passenger.name,
flight.fid,
flight.`date`,
flight.`time`,
ticket.seat,
route.start,
route.arrival,
flight.price
FROM passenger, ticket, flight, route
WHERE passenger.pid=ticket.pid and ticket.fid=flight.fid and flight.rid=route.rid and passenger.pid='P0003'
ORDER BY
flight.`date` ASC,
flight.`time` ASC;
```

The search process took 0.00s and the result from the database will be:

name	fid	date	time	seat	start	arrival	price
Geoffrey	F3102	2018-02-22	12:11:00	273F	FGX	WDV	1098
Geoffrey	F2713	2018-03-11	18:21:00	14B	DQW	QEI	4153
Geoffrey	F0238	2018-03-14	17:53:00	72D	ATL	MGF	5101
Geoffrey	F2441	2018-04-26	23:21:00	4F	JNU	PWP	3640

4 rows in set (0.00 sec)

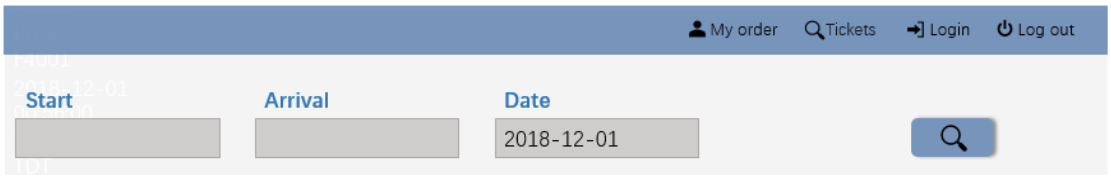
And the result shows in the user interface:



5.2 Flight enquiry

(1) Inquiring all the flights at specified date

Input the date '2018-12-01'.



With the SQL code:

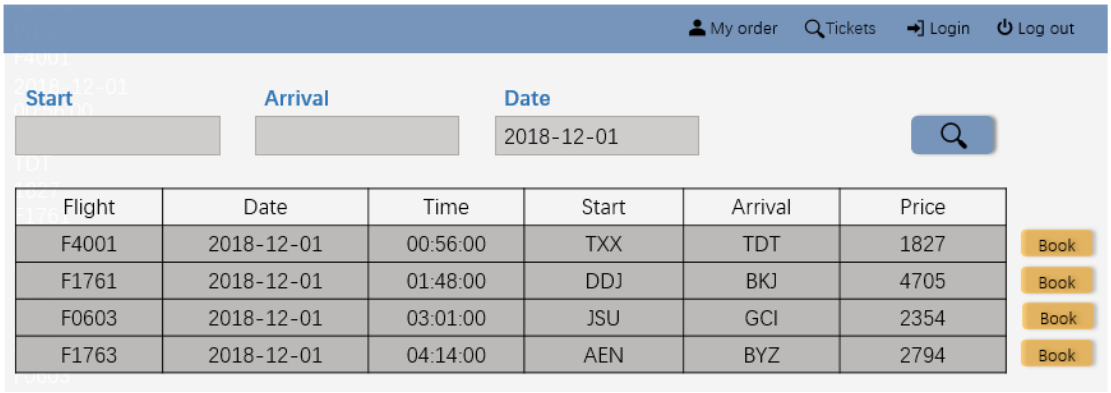
```
SELECT fid, time, date, start, arrival, price
FROM flight, route
WHERE flight.rid=route.rid and date='2018-12-01'
ORDER BY
flight.`time` ASC;
```

The search process took 0.01s and the result from the database will be:

fid	time	date	start	arrival	price
F4001	00:56:00	2018-12-01	TXX	TDT	1827
F1761	01:48:00	2018-12-01	DDJ	BKJ	4705
F0603	03:01:00	2018-12-01	JSU	GCI	2354
F1763	04:14:00	2018-12-01	AEN	BYZ	2794
F0421	04:37:00	2018-12-01	JKH	JIL	3194
F4106	05:15:00	2018-12-01	TCI	UGA	1981
F1615	05:25:00	2018-12-01	ATS	PAW	2719
F4277	11:19:00	2018-12-01	DSN	DIX	1791
F4901	11:19:00	2018-12-01	MDA	ALY	4977
F3064	11:51:00	2018-12-01	CIS	WUO	4610
F2777	13:41:00	2018-12-01	SSD	WYO	2310
F4169	14:22:00	2018-12-01	BRI	RYX	3184
F3005	15:46:00	2018-12-01	CIL	XEM	3592
F1684	16:46:00	2018-12-01	UXF	CCE	3640
F2507	18:16:00	2018-12-01	UEM	LPT	3579
F4049	18:43:00	2018-12-01	ECY	TSU	3672
F1613	19:37:00	2018-12-01	TCI	WEN	5455
F0765	21:21:00	2018-12-01	XCZ	JGK	1642
F2525	22:22:00	2018-12-01	DCD	EQQ	2760

19 rows in set (0.01 sec)

And the result shows in the user interface:



(2) Inquiring all flights from the specified departure point to destination

Input the start place 'NWA' and the arrival place 'XTE' in the user interface.

With the SQL code:

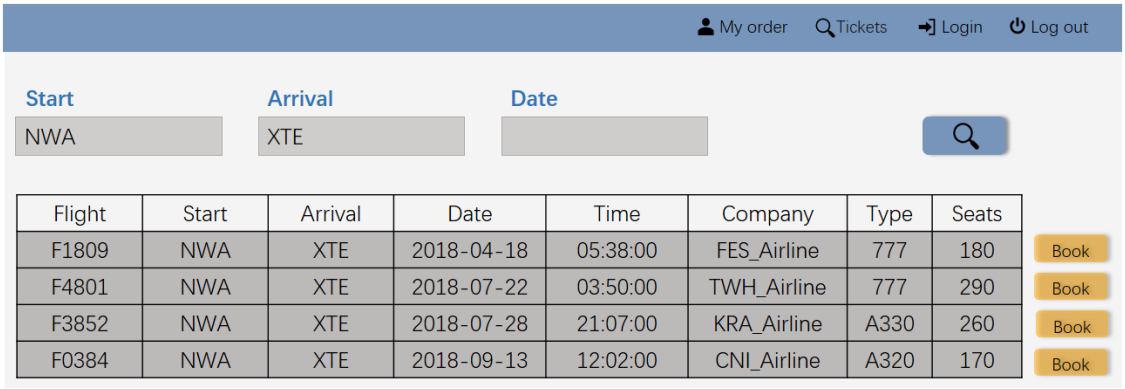
SELECT fid, start, arrival, date, time, cname,type,seats
FROM flight, route, airline, airplane
WHERE flight.rid=route.rid and airplane.plane_id=flight.plane_id and

airplane.cid=airline.cid and start='NWA' and arrival='XTE'
ORDER BY
flight.`date` ASC,
flight.`time` ASC;

The search process took 0.11s and the result from the database will be:

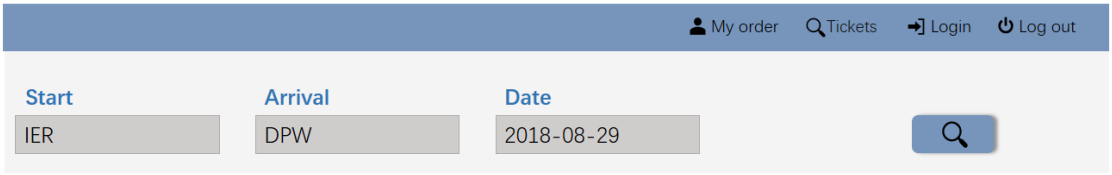
```
+-------+-------+---------+------------+----------+--------------+------+-------
+
| fid   | start | arrival | date       | time     | cname        | type | seats
|
+-------+-------+---------+------------+----------+--------------+------+-------
+
| F1809 | NWA   | XTE     | 2018-04-18 | 05:38:00 | FES_Alirline | 777  |   180
|
| F4801 | NWA   | XTE     | 2018-07-22 | 03:50:00 | TWH_Alirline | 777  |   290
|
| F3852 | NWA   | XTE     | 2018-07-28 | 21:07:00 | KRA_Alirline | A330 |   260
|
| F0384 | NWA   | XTE     | 2018-09-13 | 12:02:00 | CNI_Alirline | A320 |   170
|
+-------+-------+---------+------------+----------+--------------+------+-------
+
4 rows in set (0.11 sec)
```

And the result shows in the user interface:



(3) Inquiring all flights from the specified departure point to destination at fixed date.

Input the date '2018-08-29', the start place 'IER' and the arrival place 'DPW' in the user interface.



With the SQL code:

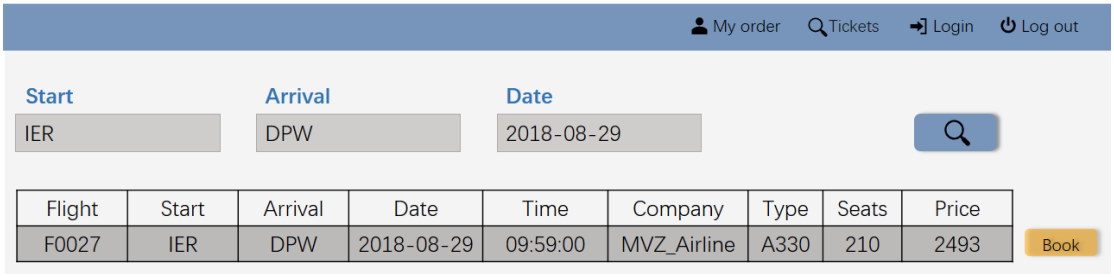
SELECT fid, start, arrival, date, time, cname,type,seats,price
FROM flight, route, airline, airplane
WHERE flight.rid=route.rid and airplane.plane_id=flight.plane_id and

airplane.cid=airline.cid and start='IER' and arrival='DPW' and date='2018-08-29'
ORDER BY
flight.`date` ASC,
flight.`time` ASC;

The search process took 0.08s and the result from the database will be:

```
+--------+-------+---------+------------+----------+--------------+------+-------
+--------+
| fid    | start | arrival | date       | time     | cname        | type | seats
| price  |
+--------+-------+---------+------------+----------+--------------+------+-------
+--------+
| F0027  | IER   | DPW     | 2018-08-29 | 09:59:00 | MVZ_Alirline | A330 |   210
|  2493  |
+--------+-------+---------+------------+----------+--------------+------+-------
+--------+
1 row in set (0.08 sec)
```

And the result shows in the user interface:



5.3 Creat/ Delete a new ticket information (ticket reservation/ ticket refund)

(1)Ticket reservation

Input the flight id and the passenger id, and input a new ticket id and a seat number in the user interface.

With the SQL code,

INSERT INTO `ticket` VALUES ('T50001', 'P0200', 'F1237', '151A');
The insert process took 0.13s.

(2) Ticket refund

They can also delete a ticket information, with the SQLcode:

DELETE FROM ticket
WHERE pid='P0200' and fid='F1237';
The delete process took 0.11s.

5.4 Administrator funtional requirement

(1) View passenger information of the specified flight

Input the flight id 'F0030' in the user interface.

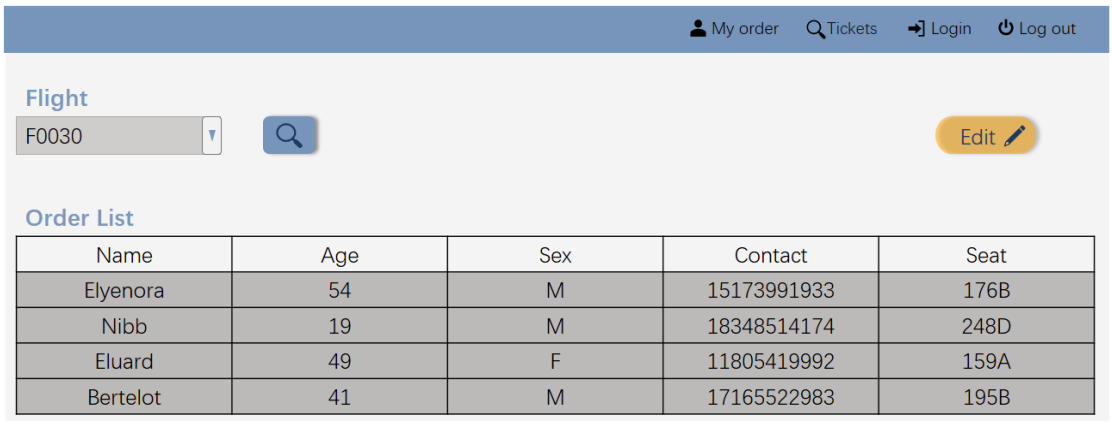
With the SQL code:

SELECT name,age,sex,contact,seat
from passenger, ticket, flight
where passenger.pid=ticket.pid and ticket.fid=flight.fid and flight.fid='F0030';

The search process took 0.00s and the result from the database will be:

```
+-------------+-----+-----+-------------+------+
| name        | age | sex | contact     | seat |
+-------------+-----+-----+-------------+------+
| Andreas     |  16 | M   | 15701785362 | 102C |
| Josep       |  40 | M   | 19530354040 | 22F  |
| Dickie      |  41 | M   | 15089499388 | 249F |
| Melchior    |  18 | M   | 17015541994 | 83A  |
| Elyenora    |  54 | M   | 15173991933 | 176B |
| Nibb        |  19 | M   | 18348514174 | 248D |
| Eluard      |  49 | F   | 11805419992 | 159A |
| Bertelot    |  41 | M   | 17165522983 | 195B |
| Jehannette  |  33 | F   | 17941469146 | 202F |
| Auizia      |  45 | F   | 17644315312 | 278A |
| Kit         |  27 | M   | 16989896419 | 221D |
| Joseph      |  59 | F   | 15032342928 | 66B  |
| Winnifred   |  20 | F   | 14618336343 | 203C |
| Cleremunda  |  22 | F   | 11488695356 | 167G |
+-------------+-----+-----+-------------+------+
14 rows in set (0.00 sec)
```

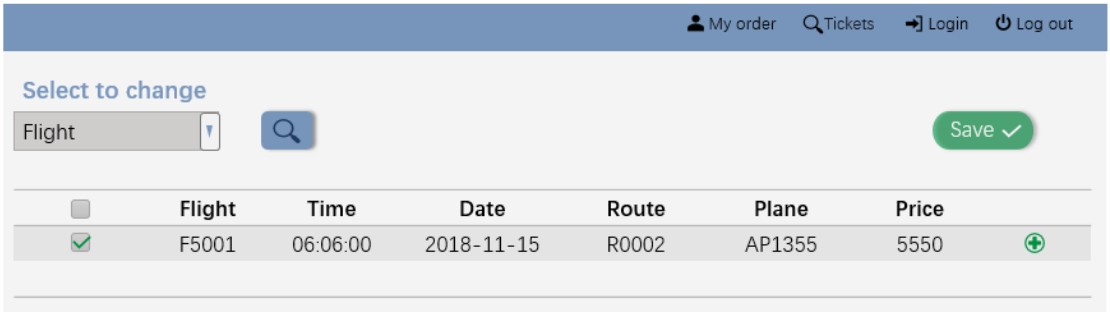
And the result shows in the user interface:



Name	Age	Sex	Contact	Seat
Elyenora	54	M	15173991933	176B
Nibb	19	M	18348514174	248D
Eluard	49	F	11805419992	159A
Bertelot	41	M	17165522983	195B

(2) Creat/ Delete a new flight information

The administrator can create a new flight information, input the flight id 'F5001', the time '06:05:00', the date '2018-11-15', the route id 'R0002', the plane id 'AP1355' and the price of the flight '5550',



with the SQLcode:

INSERT INTO `flight` VALUES ('F5001', '06:05:00', '2018-11-15', 'R0002', 'AP1355', '5550');

The insert process took 0.08s.

The administrator can also delete a flight information, with the SQLcode:

DELETE FROM flight
WHERE fid='F5001';

The delete process took 0.11s.

6.0 Difficulties

There are several difficulties we met when we doing the project:

The amount of the data is too large, and it is difficult to import data from Excel. And it is impractical to insert them to the database by typing. Therefore, we used the import function of Navicat for MySQL to complete the import of large number of data.

When we initially draw ER diagrams, we linked flight and airline, and then linked airline and airplane. But the there were some problems because we find that there may be more than one plane fly for a flight when we join the table flight, airline and airplane, so later we change this relation into ‘flight-airplane, airplane-airline’.

Failure to create a foreign key for table, and display ‘Can't create table’. We found that it is because the attibute we need to reference is not a primary key, and it also did not set up an index. And we find that if one of the foreign key we want to invoke is not a primary key, we have to create an index for it or change the reference attribute to a primary key.