Airline Management System

Information System Project

2nd year, First Semester

Abstract:

The purpose of the project is to build an application program allows scheduling, booking, customer management, and other functions. It saves the time of the customer when booking tickets and gets to know when the flight is delayed as the air ticket agencies will get an SMS alert when the flight is delayed and then they can inform the passenger.

Introduction:

Civil aviation where the departure and the arrival take place in the same country.

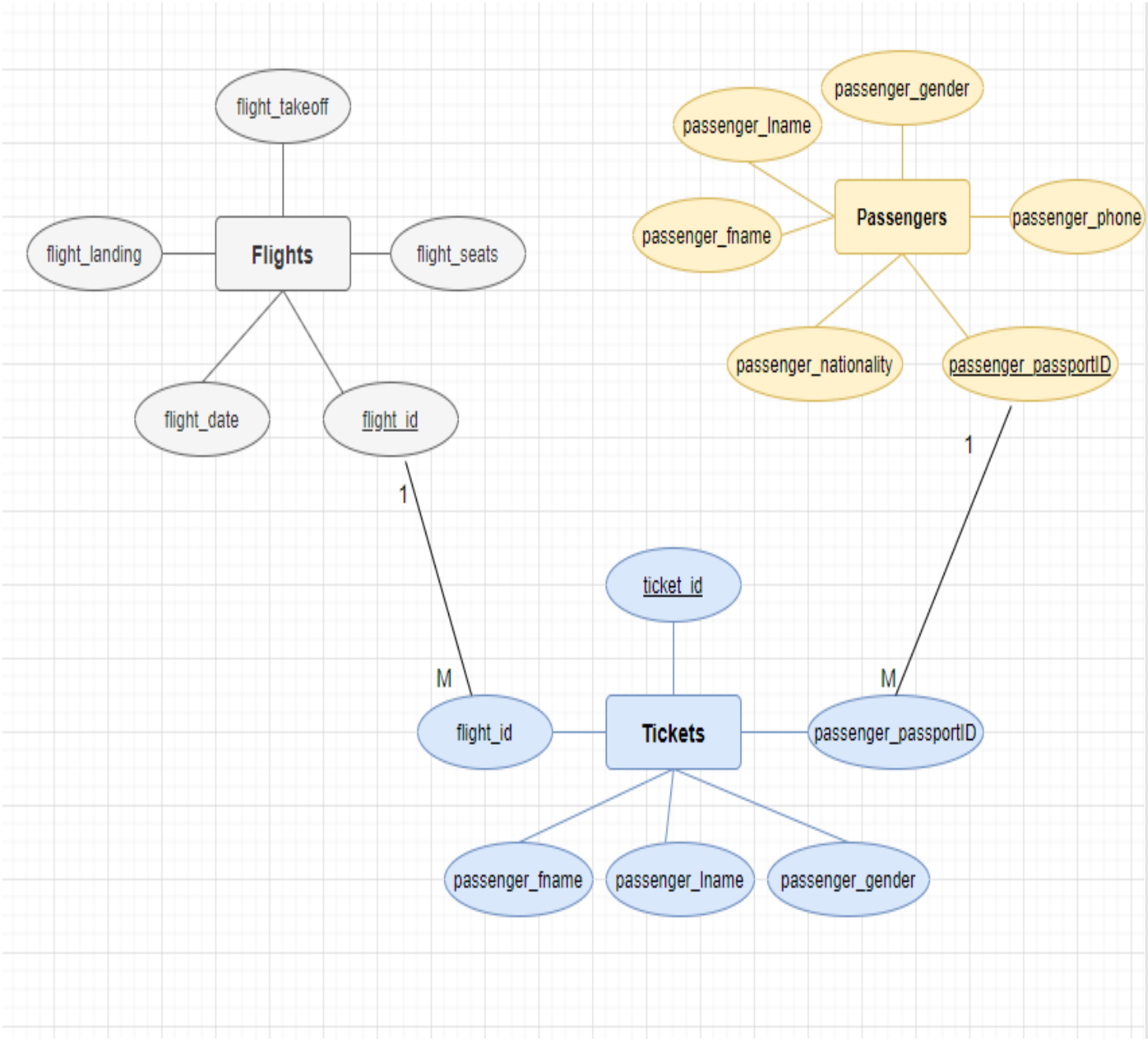
-this project will help the user reserving tickets.

-user will be aware of the most booked flights.

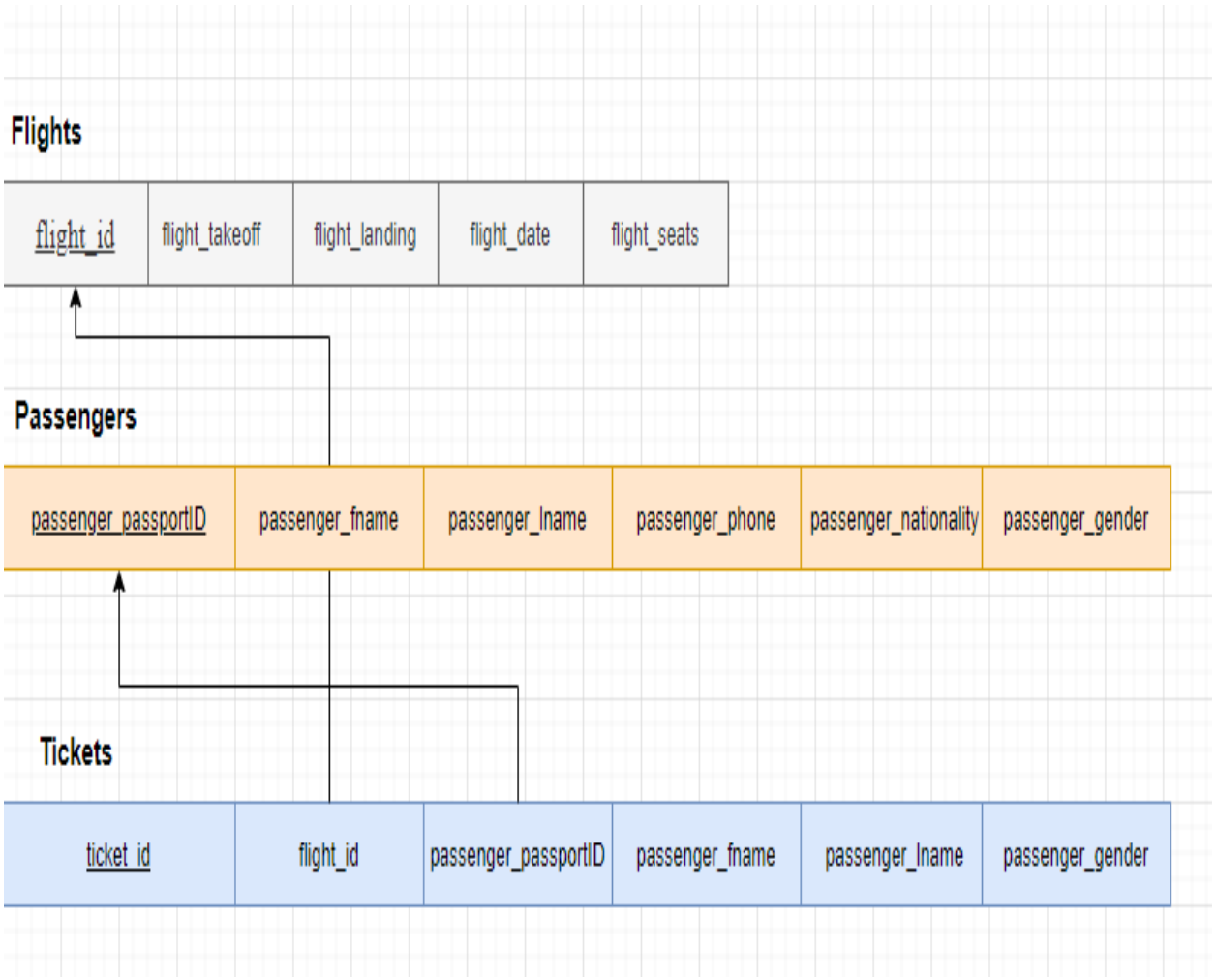
Database:

1. Entity relationship diagram of Airline .
2. Database Mapping.
3. Database Table .

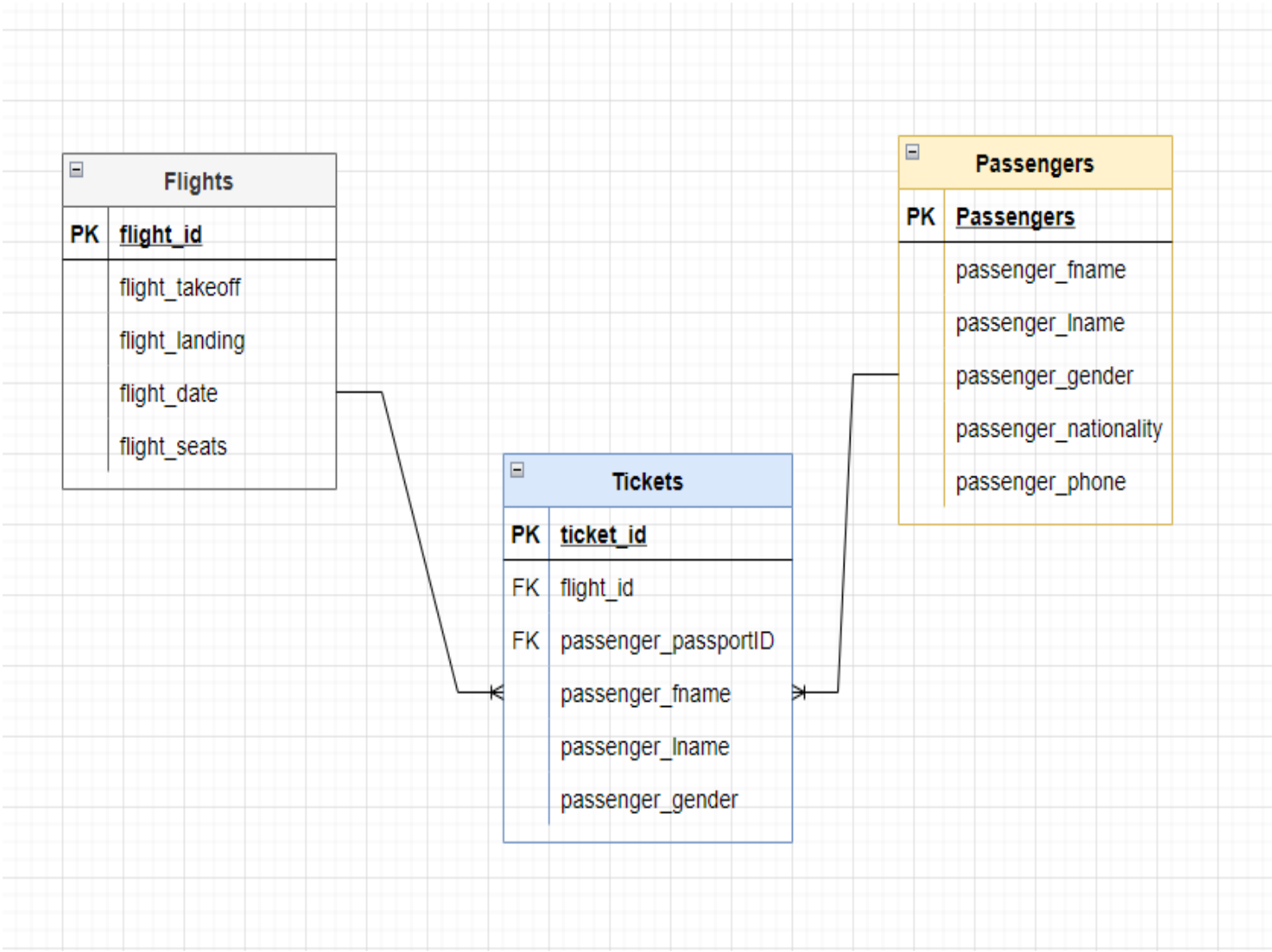
1. Entity relationship diagram of Airline :


flight_takeoff
flight_landing
Flights
flight_seats
flight_date
flight_id
passenger_gender
passenger_lname
passenger_fname
Passengers
passenger_phone
passenger_nationality
passenger_passportID
1
1
M
M
ticket_id
flight_id
Tickets
passenger_passportID
passenger_fname
passenger_lname
passenger_gender

2. Database Mapping :



3. Database Table :

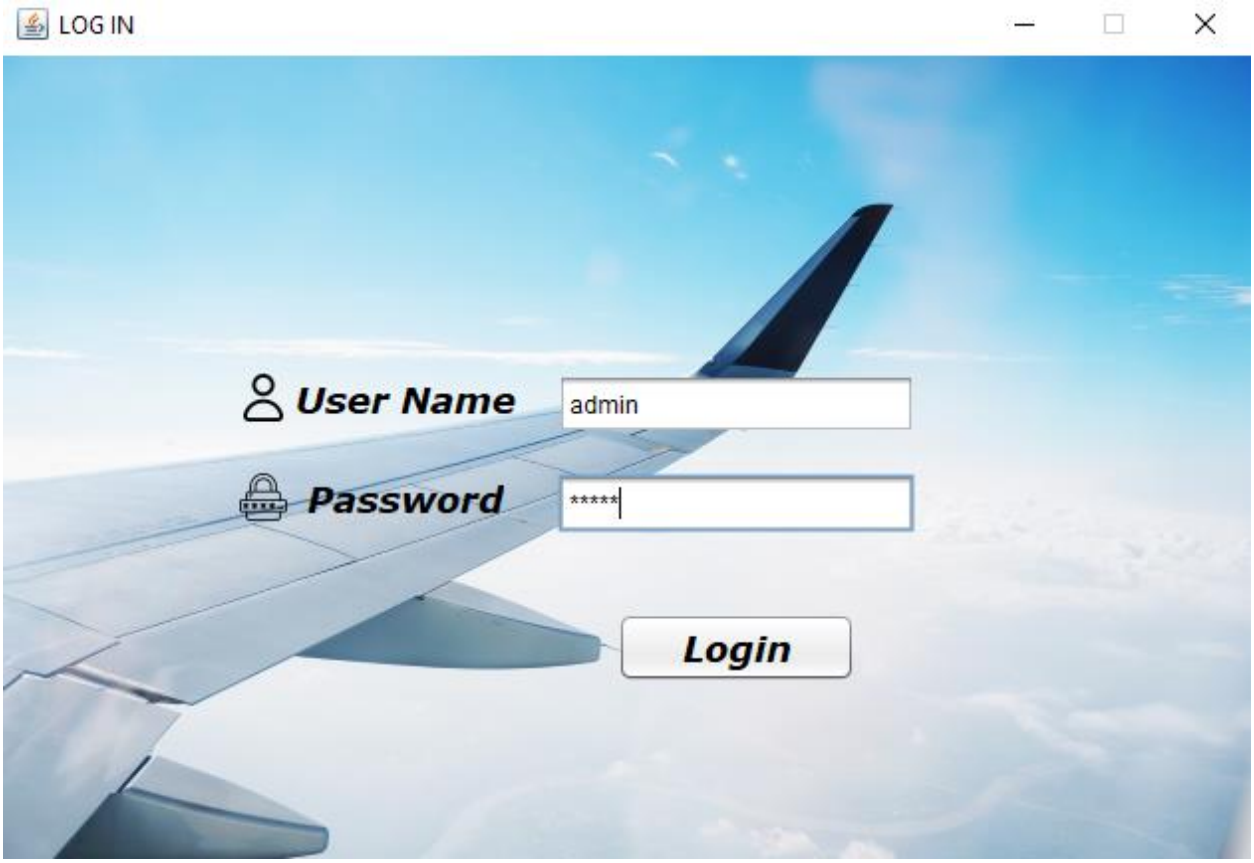


GUI :

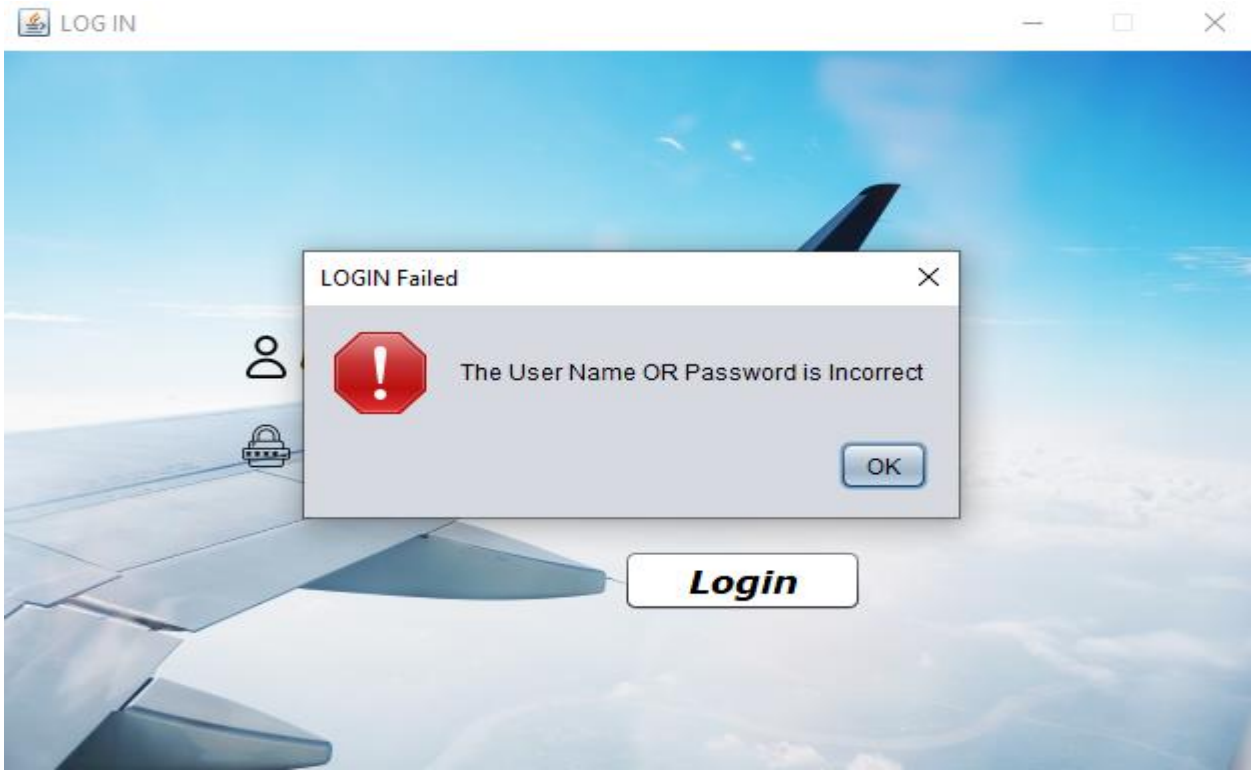
1. Frame One (LOG IN)
2. Frame Two (Main Window)
3. Frame Three (Flights)
4. Frame Four (Passengers)
5. Frame Five (Tickets)
6. Frame Six (Information)

1. Frame One :

- **LOG IN** , this page contains two text fields, one for typing the user's name and the other for typing the password and the log in button.



- When entering the username or password incorrectly.

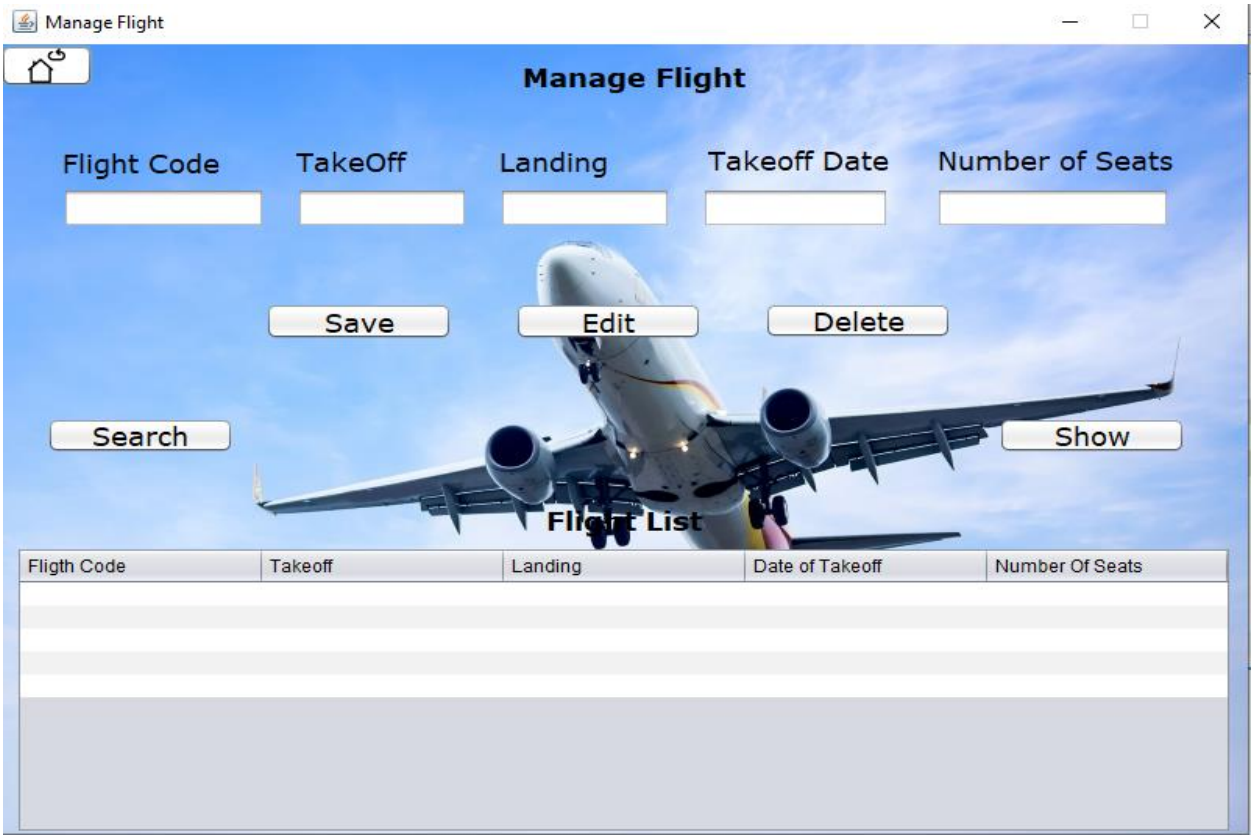


2. Frame Two :

- **Main Window** ,It contains four buttons, each button enters a page .

3. Frame Three :

- **Flight ,** it contains :-
  i. Text fields (**Flight Code , Takeoff , Landing , Takeoff Date , Number of Seats**).
  ii. Buttons (**Save ,Edit ,Delete ,Search, Show ,Back**).
  iii. Table to display the results.

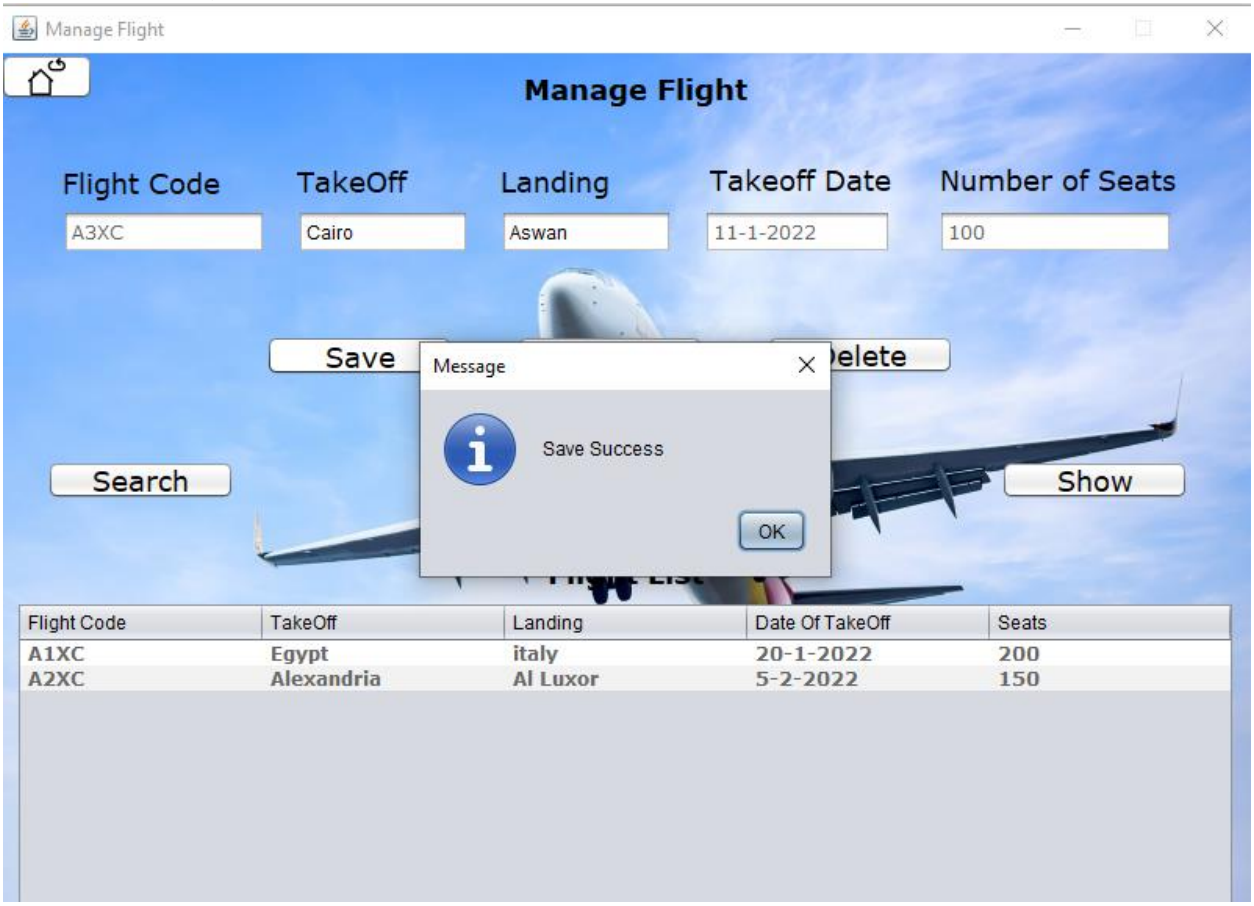


- ✓ **Save :**

```
//try, and catch (used for handling errors)
try {
    // TODO add your handling code here:
    PreparedStatement stmt =con.prepareStatement("INSERT INTO `airline`.`flights` (`fl_id`, `fl_takeoff`, `fl_landing`, `fl_date`, `fl_seats`) VALUES (?, ?, ?, ?,?);");
```

```
//   (used to insert flight's data into database)

        String  flightCode = tf_flightCode.getText();
        String  takeOff = tf_takeoff.getText();
        String  landing = tf_landing.getText();
        String  takeOffDate = tf_date.getText();
        int numSeats = Integer.parseInt(tf_numSeats.getText()) ;

                        //(assigning data into string for flight)


        stmt.setString(1, flightCode);
        stmt.setString(2, takeOff);
        stmt.setString(3, landing);
        stmt.setString(4, takeOffDate);
        stmt.setInt(5, numSeats);

       //putting the date the user put into the flight in  database


        stmt.executeUpdate();

                        //to update data into the flight table in the database


        JOptionPane.showMessageDialog(this, "Save Success");

                        //showing the massege of succsses of insertion



        tf_flightCode.setText("");
        tf_takeoff.setText("");
        tf_landing.setText("");
        tf_date.setText("");
        tf_numSeats.setText("");

                        //making the field text empty after insertion


      } catch (SQLException ex) {
```

Logger.getLogger(Flights.class.getName()).log(Level.SEVERE, null, ex);}



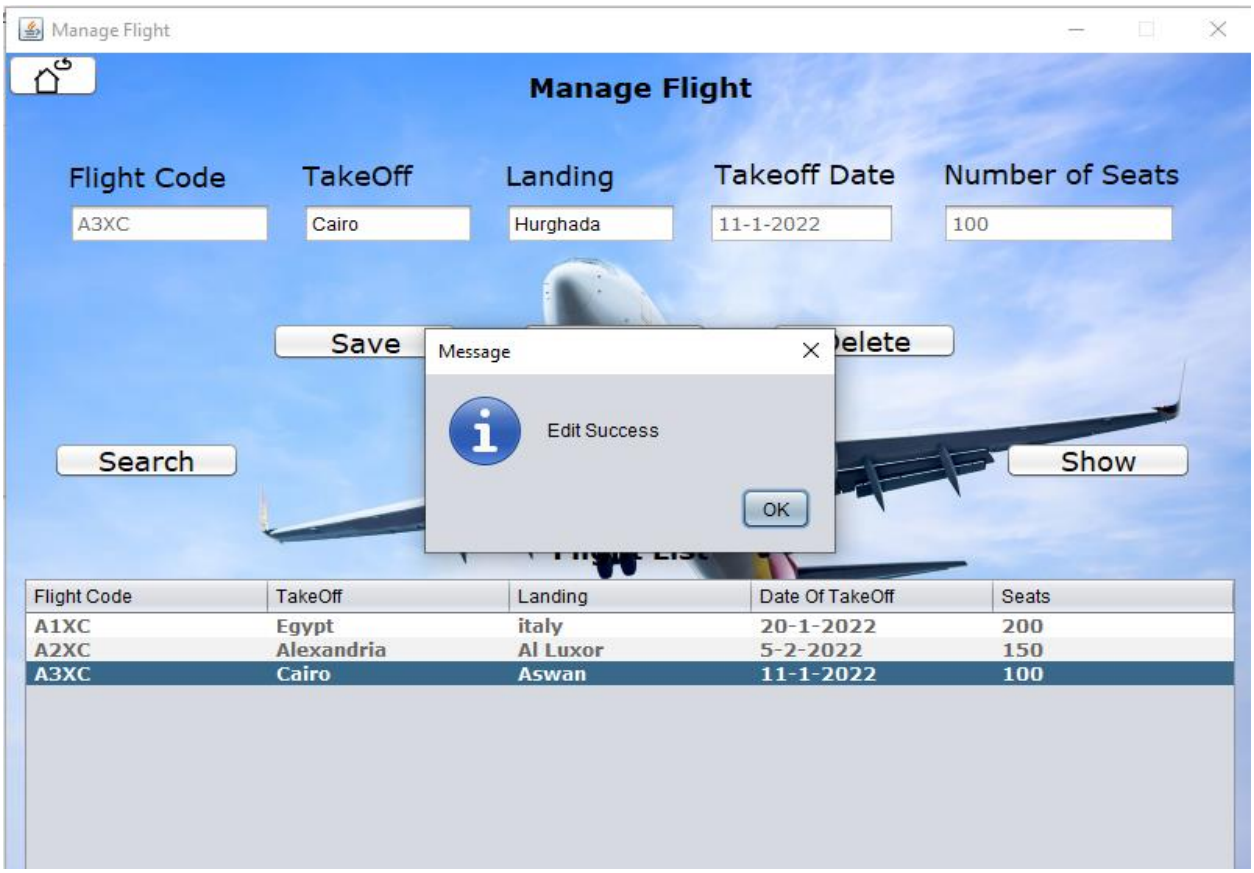
✓ Edit :

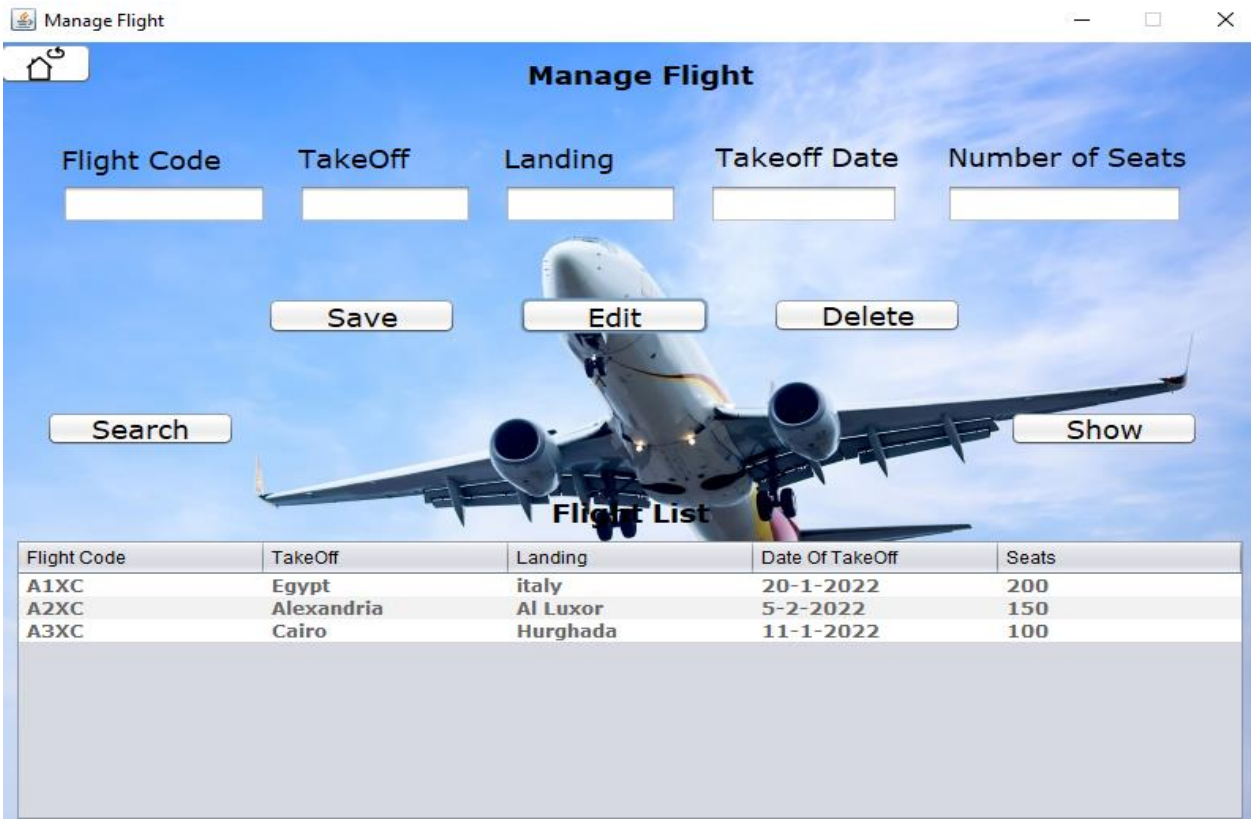
PreparedStatement stmt = con.prepareStatement("UPDATE `airline`.`flights` SET `fl_takeoff` = ?, `fl_landing` = ?, `fl_date` = ?, `fl_seats` = ? WHERE `fl_id` = ?");

// (used to change some info in flight's table)

String fl_takeoff = tf_takeoff.getText();
String fl_landing = tf_landing.getText();
String fl_date = tf_date.getText();
String fl_seats = tf_numSeats.getText();
String fl_id = tf_flightCode.getText();

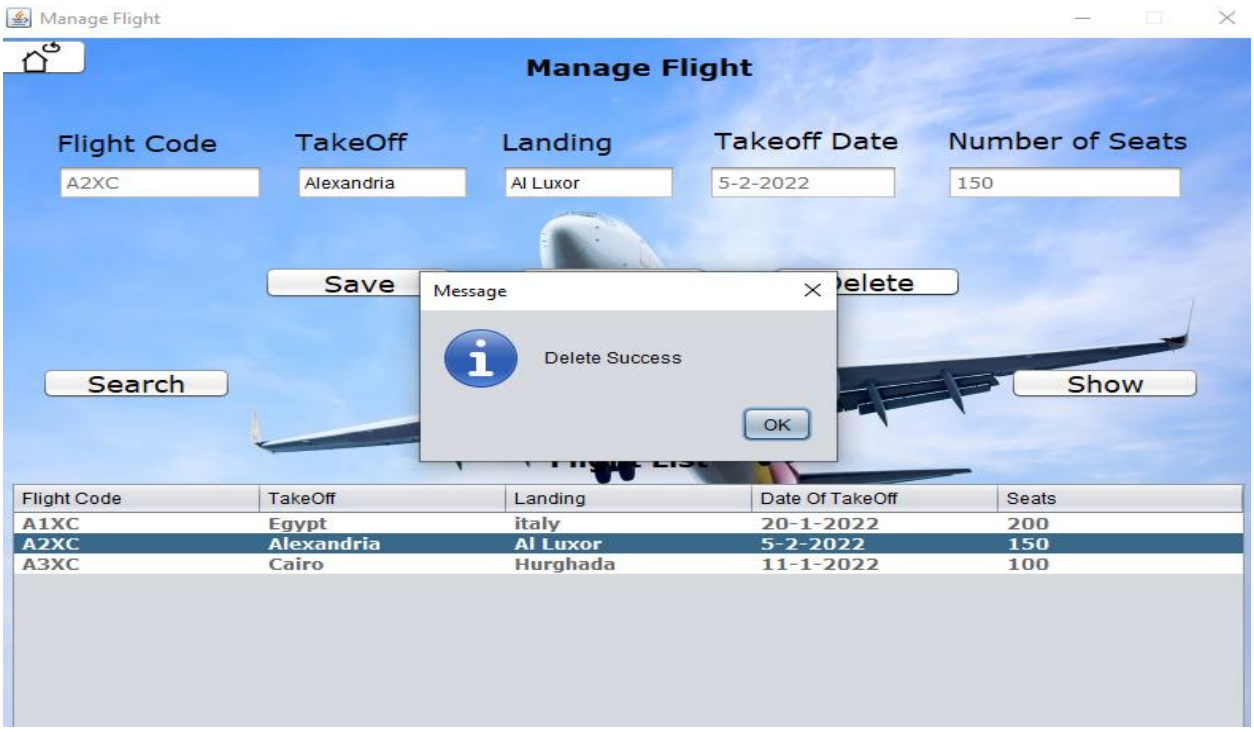
```
                    //(to get data from text fields)


            stmt.setString(1, fl_takeoff);
            stmt.setString(2, fl_landing);
            stmt.setString(3, fl_date);
            stmt.setString(4, fl_seats);
            stmt.setString(5,fl_id) ;
                                //put them in  database



            stmt.executeUpdate();
           //to update data into the flight table in the database


            JOptionPane.showMessageDialog(this, "Edit Success"); //showing
the massege of succsses of Update.
            btn_showActionPerformed(evt); // call function Show to display in
table .

            tf_flightCode.setText("");
            tf_takeoff.setText("");
            tf_landing.setText("");
            tf_date.setText("");
            tf_numSeats.setText("");
//making the field text empty after Edit

        } catch (SQLException ex) {
            Logger.getLogger(Flights.class.getName()).log(Level.SEVERE,
null, ex);
        }
```

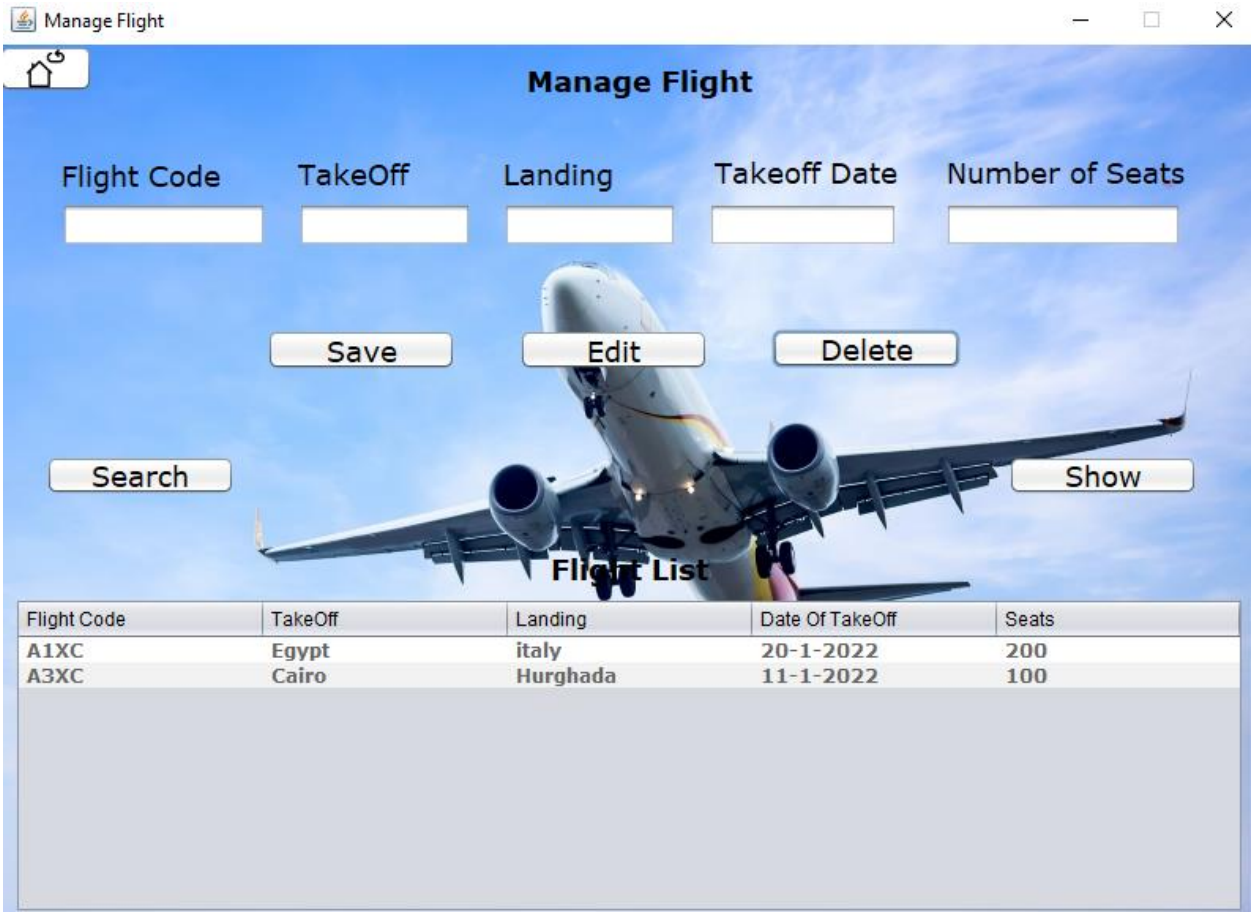

Manage Flight
Manage Flight
Flight Code
TakeOff
Landing
Takeoff Date
Number of Seats
A3XC
Cairo
Hurghada
11-1-2022
100
Save
elete
Search
Show
Message
Edit Success
OK
Flight Code
TakeOff
Landing
Date Of TakeOff
Seats
A1XC
Egypt
italy
20-1-2022
200
A2XC
Alexandria
Al Luxor
5-2-2022
150
A3XC
Cairo
Aswan
11-1-2022
100


Manage Flight
Manage Flight
Flight Code
TakeOff
Landing
Takeoff Date
Number of Seats
Save
Edit
Delete
Search
Show
Flight List
Flight Code
TakeOff
Landing
Date Of TakeOff
Seats
A1XC
Egypt
italy
20-1-2022
200
A2XC
Alexandria
Al Luxor
5-2-2022
150
A3XC
Cairo
Hurghada
11-1-2022
100

✓ **Delete :**

```
try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/airline", "root",
"root");  // connection to database
        PreparedStatement stmt = con.prepareStatement("delete from flights
where fl_id=?");
        String fl_id = tf_flightCode.getText();

                //To delete data from database in flight table


        stmt.setString(1, fl_id);

        stmt.executeUpdate();

            JOptionPane.showMessageDialog(this, "Delete Success");

                //to show flight data after deleting



        btn_showActionPerformed(evt);
         tf_flightCode.setText("");
        tf_takeoff.setText("");
        tf_landing.setText("");
        tf_date.setText("");
        tf_numSeats.setText("");
        //making the field text empty after Delete


    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
```


Manage Flight
Manage Flight
Flight Code
TakeOff
Landing
Takeoff Date
Number of Seats
A2XC
Alexandria
Al Luxor
5-2-2022
150
Save
Delete
Search
Show
Message
Delete Success
OK
Flight Code
TakeOff
Landing
Date Of TakeOff
Seats
A1XC
Egypt
italy
20-1-2022
200
A2XC
Alexandria
Al Luxor
5-2-2022
150
A3XC
Cairo
Hurghada
11-1-2022
100


Manage Flight
Manage Flight
Flight Code
TakeOff
Landing
Takeoff Date
Number of Seats
Save
Edit
Delete
Search
Show
Flight List
Flight Code
TakeOff
Landing
Date Of TakeOff
Seats
A1XC
Egypt
italy
20-1-2022
200
A3XC
Cairo
Hurghada
11-1-2022
100

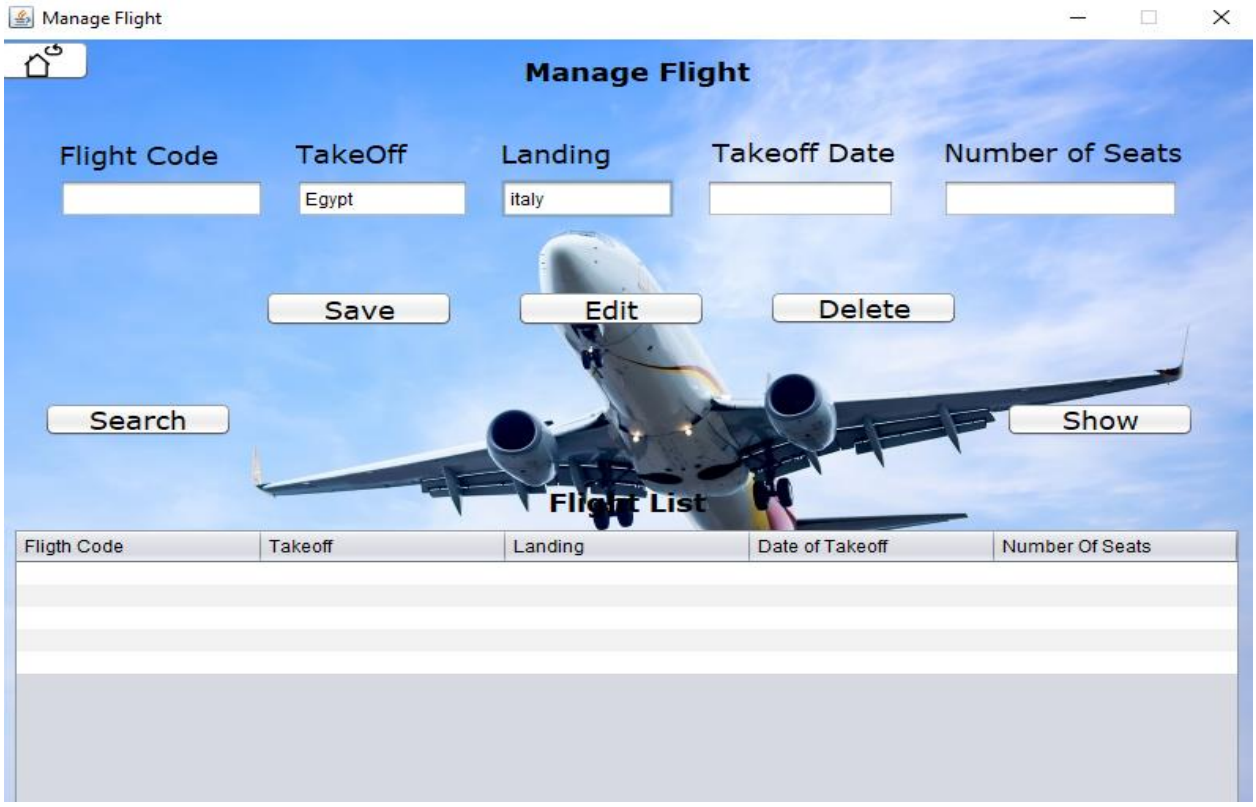
✓ **Show :**

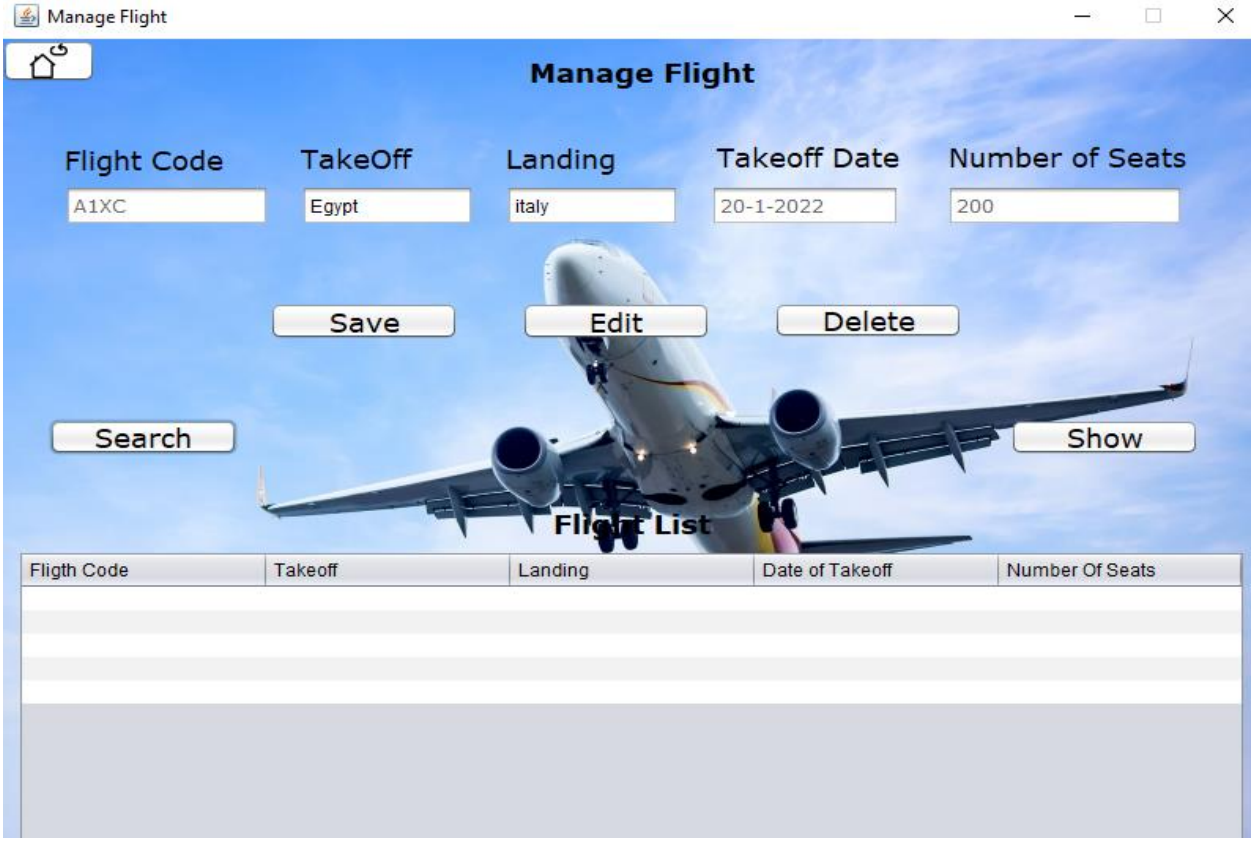
```
  PreparedStatement stat =con.prepareStatement("SELECT * FROM
airline.flights;"); // to get data from database in flights table .
          ResultSet rs =stat.executeQuery();

          dtm =new DefaultTableModel();
          dtm.addColumn("Flight Code");
          dtm.addColumn("TakeOff");
          dtm.addColumn("Landing");
          dtm.addColumn("Date Of TakeOff");
          dtm.addColumn("Seats");
         // create Table gui.




          while (rs.next()) {
             dtm.addRow(new Object []
{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getInt(5)});
// store values return from database and add row in table gui.
          }
              t_flight.setModel(dtm); // display in table gui

      } catch (SQLException ex) {
          Logger.getLogger(Flights.class.getName()).log(Level.SEVERE,
null, ex);
              }
```

✓ Search :

1. search by Takeoff and Landing .

```
try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/airline", "root", "root");
        PreparedStatement stmt = con.prepareStatement("select * from flights where fl_takeoff = ? and fl_landing = ?");
        String takeOff = tf_takeoff.getText();
        String landing = tf_landing.getText();
        stmt.setString(1, takeOff);
        stmt.setString(2, landing);
        ResultSet set = stmt.executeQuery();
        if (set.next()) {
           tf_flightCode.setText(set.getString("fl_id"));
           tf_date.setText(set.getString("fl_date"));
           tf_numSeats.setText(set.getString("fl_seats"));

        } else {
           JOptionPane.showMessageDialog(this, "Not Found"); //if the takeoff and landing not found in the table
        }

    } catch (SQLException ex) {       System.out.println(ex);}
```


Manage Flight
Manage Flight
Flight Code
TakeOff
Landing
Takeoff Date
Number of Seats
Egypt
italy
Save
Edit
Delete
Search
Show
Flight List
Fligth Code
Takeoff
Landing
Date of Takeoff
Number Of Seats


Manage Flight
Manage Flight
Flight Code
TakeOff
Landing
Takeoff Date
Number of Seats
A1XC
Egypt
italy
20-1-2022
200
Save
Edit
Delete
Search
Show
Flight List
Fligth Code
Takeoff
Landing
Date of Takeoff
Number Of Seats

- **Mouse Click Event (** in table Flight Frame**)**

```
int row_num = t_flight.getSelectedRow();
String fl_id = t_flight.getValueAt(row_num, 0).toString();
String fl_takeoff = t_flight.getValueAt(row_num,
1).toString();
String fl_landing= t_flight.getValueAt(row_num,
2).toString();
String fl_date = t_flight.getValueAt(row_num, 3).toString();
String fl_seats = t_flight.getValueAt(row_num, 4).toString();



tf_flightCode.setText(fl_id);
tf_takeoff.setText(fl_takeoff);
tf_landing.setText(fl_landing);
tf_date.setText(fl_date);
tf_numSeats.setText(fl_seats);
```

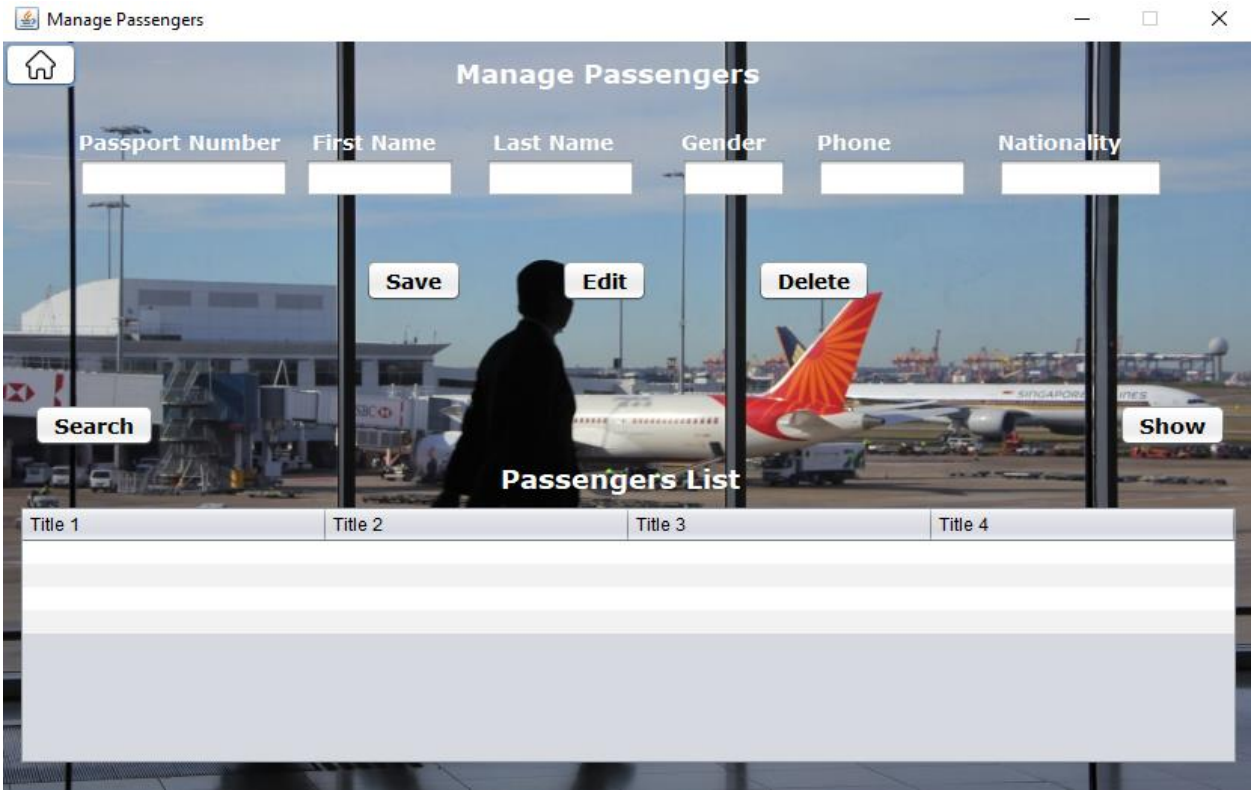
4. Frame Four :

Passengers , it contains :

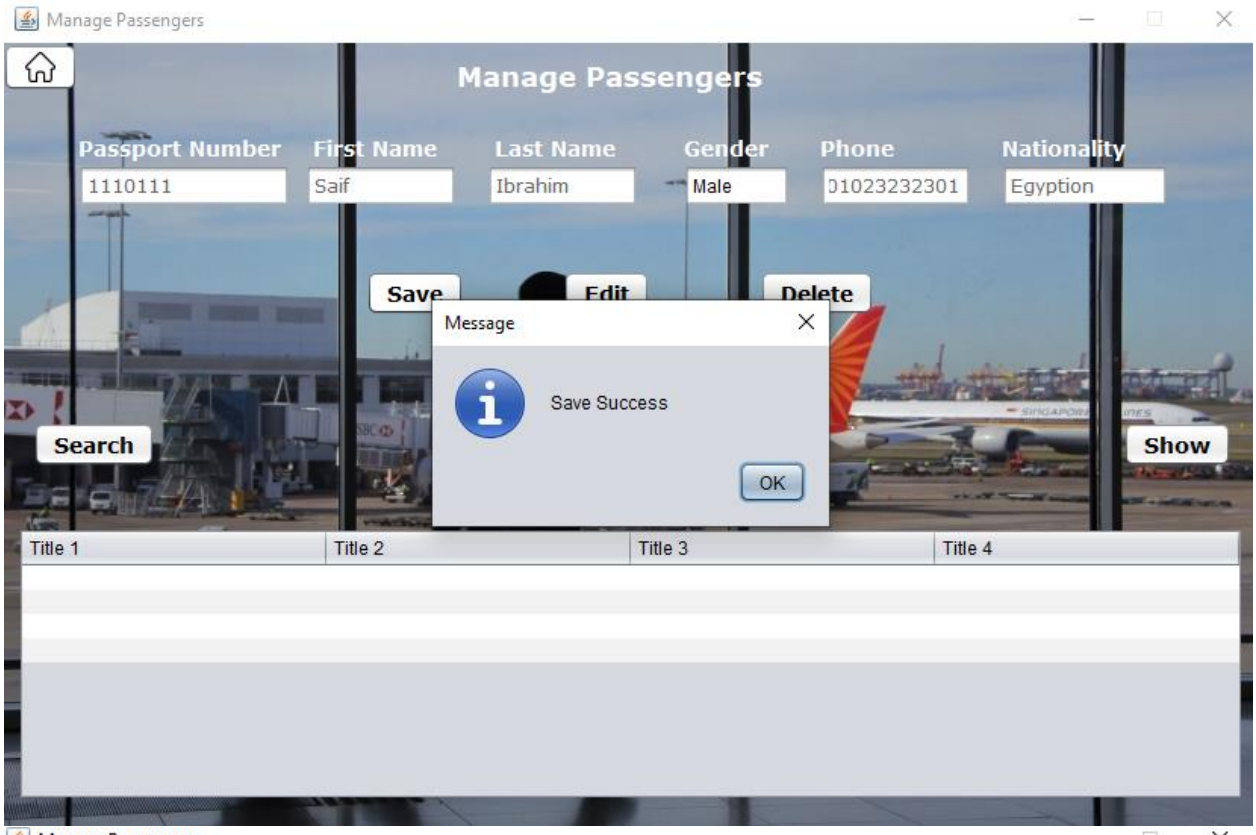
i. Text fields (**Passport Number, First Name ,Last Name ,Gender ,Phone , Nationality**)

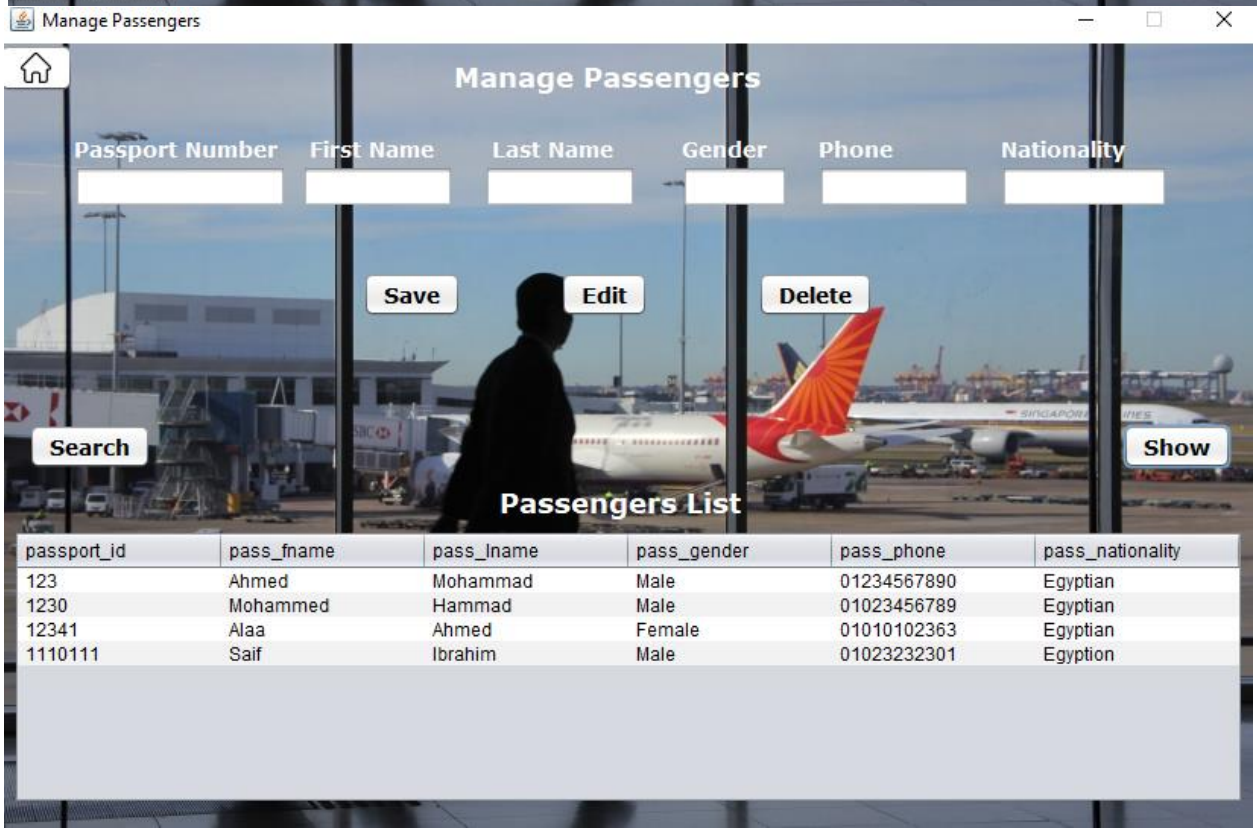
ii. Buttons (**Save ,Edit ,Delete ,Search, Show ,Back**)

iii. Table to display the results

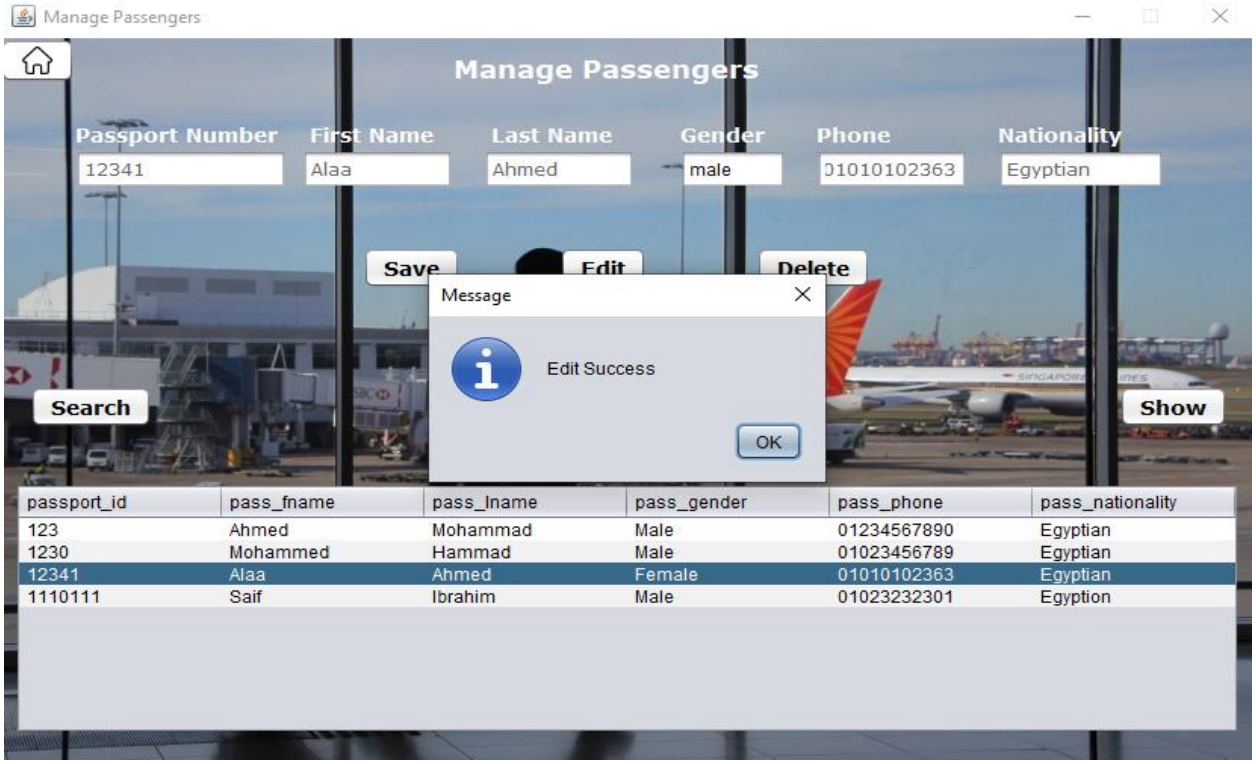


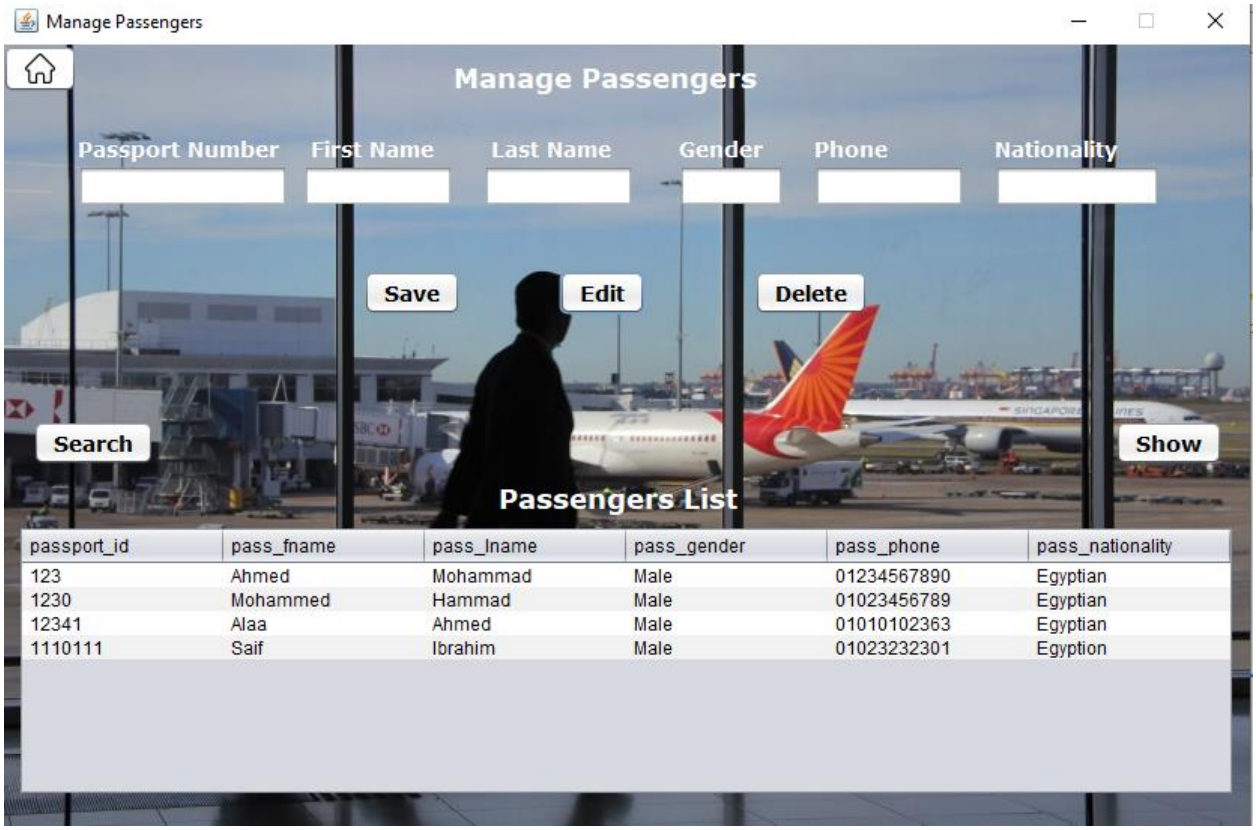
✓ Save :


Manage Passengers
Manage Passengers
Passport Number
First Name
Last Name
Gender
Phone
Nationality
1110111
Saif
Ibrahim
Male
01023232301
Egyption
Save
Edit
Delete
Message
Save Success
OK
Search
Show
Title 1
Title 2
Title 3
Title 4

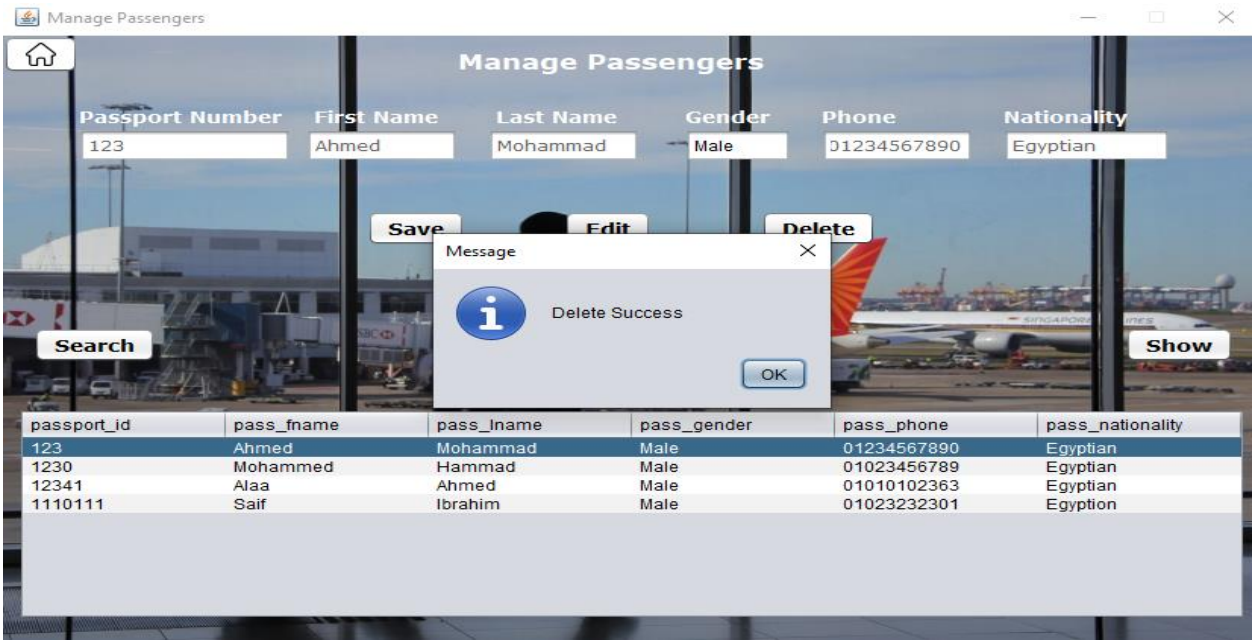

Manage Passengers
Manage Passengers
Passport Number
First Name
Last Name
Gender
Phone
Nationality
Save
Edit
Delete
Search
Show
Passengers List
passport_id
pass_fname
pass_lname
pass_gender
pass_phone
pass_nationality
123
Ahmed
Mohammad
Male
01234567890
Egyptian
1230
Mohammed
Hammad
Male
01023456789
Egyptian
12341
Alaa
Ahmed
Female
01010102363
Egyptian
1110111
Saif
Ibrahim
Male
01023232301
Egyption

✓ Edit :

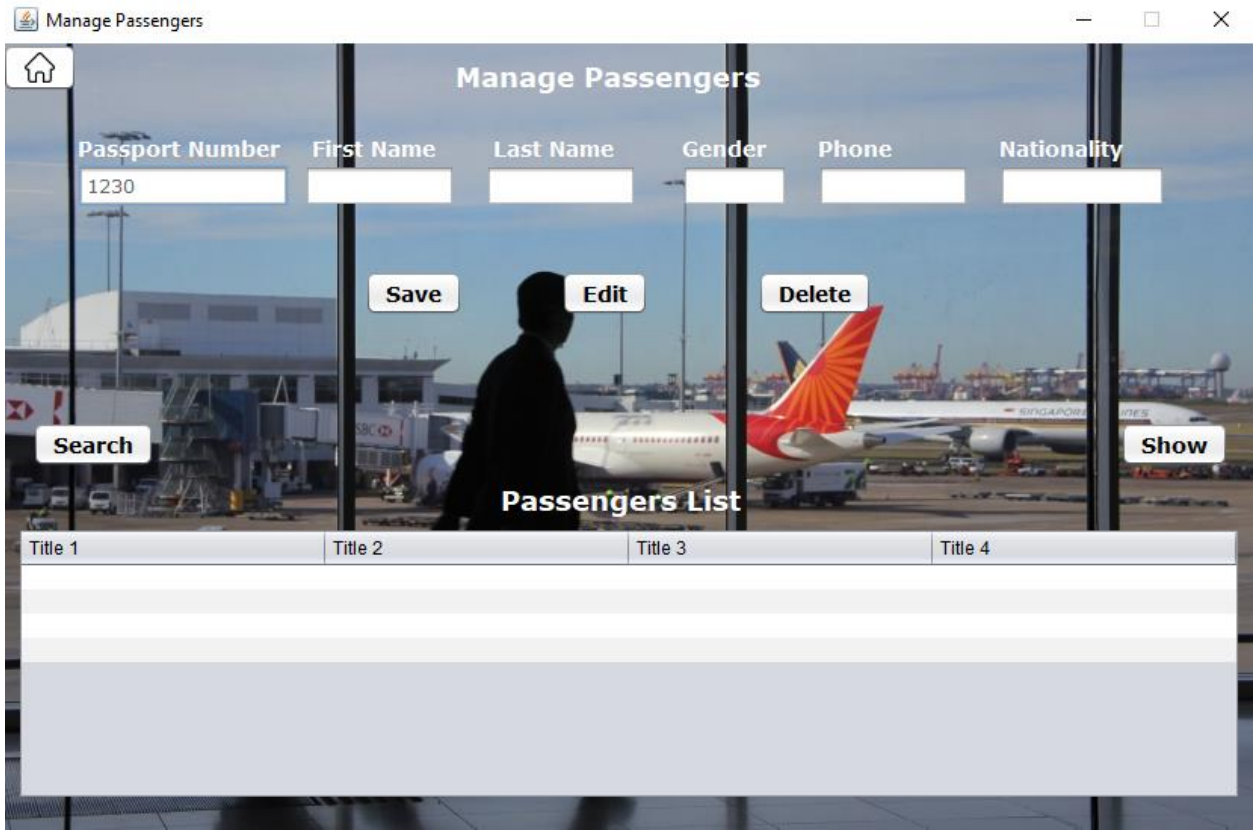

Manage Passengers
Manage Passengers
Passport Number
First Name
Last Name
Gender
Phone
Nationality
12341
Alaa
Ahmed
male
01010102363
Egyptian
Save
Edit
Delete
Search
Show
Message
Edit Success
OK
passport_id
pass_fname
pass_lname
pass_gender
pass_phone
pass_nationality
123 Ahmed Mohammad Male 01234567890 Egyptian
1230 Mohammed Hammad Male 01023456789 Egyptian
12341 Alaa Ahmed Female 01010102363 Egyptian
1110111 Saif Ibrahim Male 01023232301 Egyption

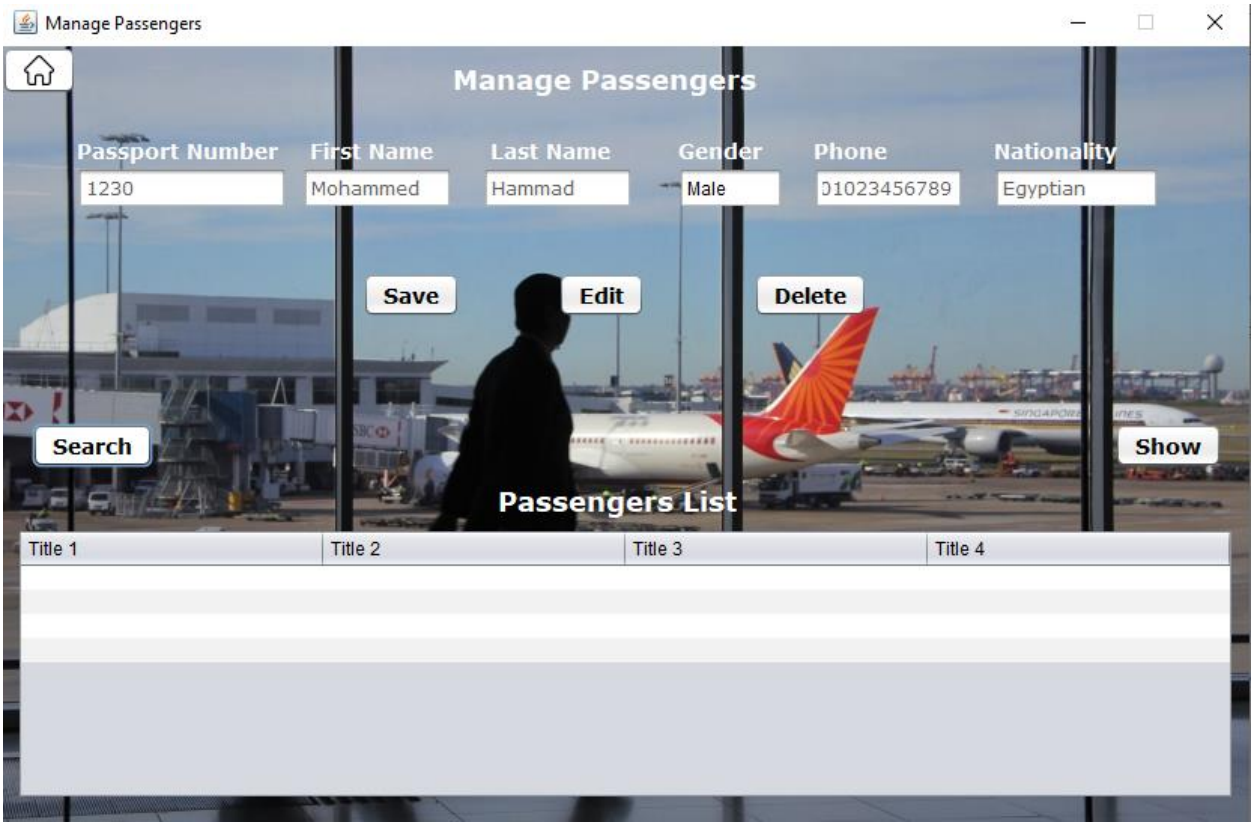

Manage Passengers
Manage Passengers
Passport Number
First Name
Last Name
Gender
Phone
Nationality
Save
Edit
Delete
Search
Show
Passengers List
passport_id
pass_fname
pass_lname
pass_gender
pass_phone
pass_nationality
123 Ahmed Mohammad Male 01234567890 Egyptian
1230 Mohammed Hammad Male 01023456789 Egyptian
12341 Alaa Ahmed Male 01010102363 Egyptian
1110111 Saif Ibrahim Male 01023232301 Egyption

✓ **Delete** :



✓ **Search (search by passport number):**





Mouse Click Event (table in passengers Frame)

```
int row_num = t_passengerss.getSelectedRow();
String passport_id  = t_passengerss.getValueAt(row_num, 0).toString();
String pass_fname = t_passengerss.getValueAt(row_num, 1).toString();
String pass_lname= t_passengerss.getValueAt(row_num, 2).toString();
String pass_gender = t_passengerss.getValueAt(row_num, 3).toString();
String pass_phone = t_passengerss.getValueAt(row_num, 4).toString();
 String pass_nationality = t_passengerss.getValueAt(row_num, 5).toString();


tf_passportNumber.setText(passport_id);
tf_Fname.setText(pass_fname);
tf_Lname.setText(pass_lname);
tf_gender.setText(pass_gender);
tf_phone.setText(pass_phone);
tf_nationality.setText(pass_nationality);
```

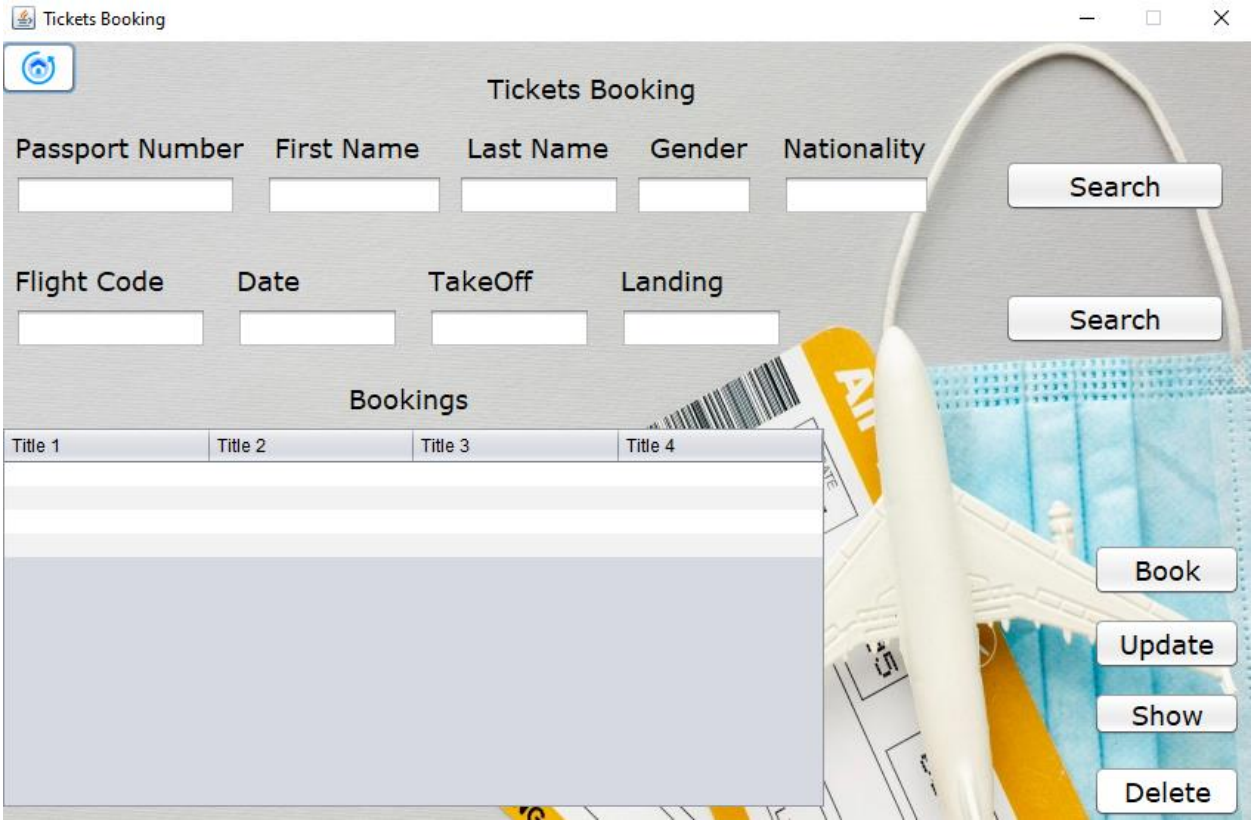
5. Frame Five :

Tickets , It contains :

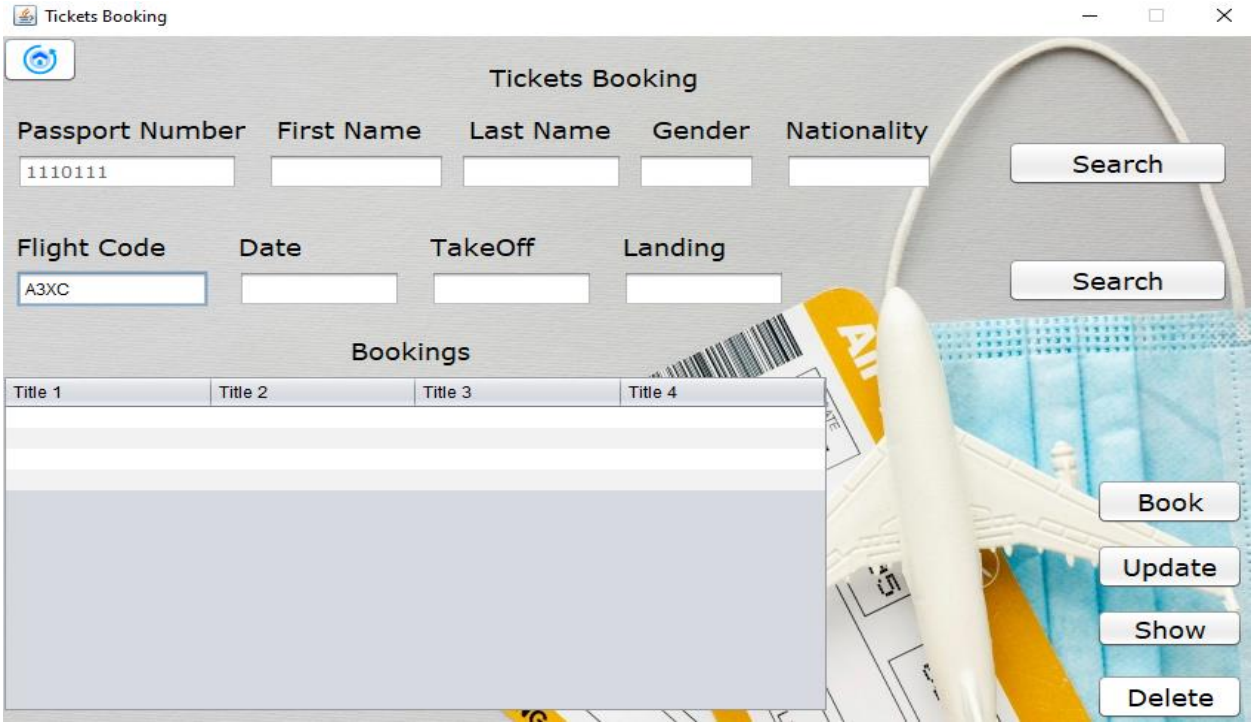
i. Text fields (**Passport Number, First Name ,Last Name ,Gender ,Phone , Nationality**)

ii. Buttons (**Save ,Edit ,Delete ,Search 1 , Search 2, Show ,Back**)

iii. Table to display the results



✓ Save (Search 1 & 2) :

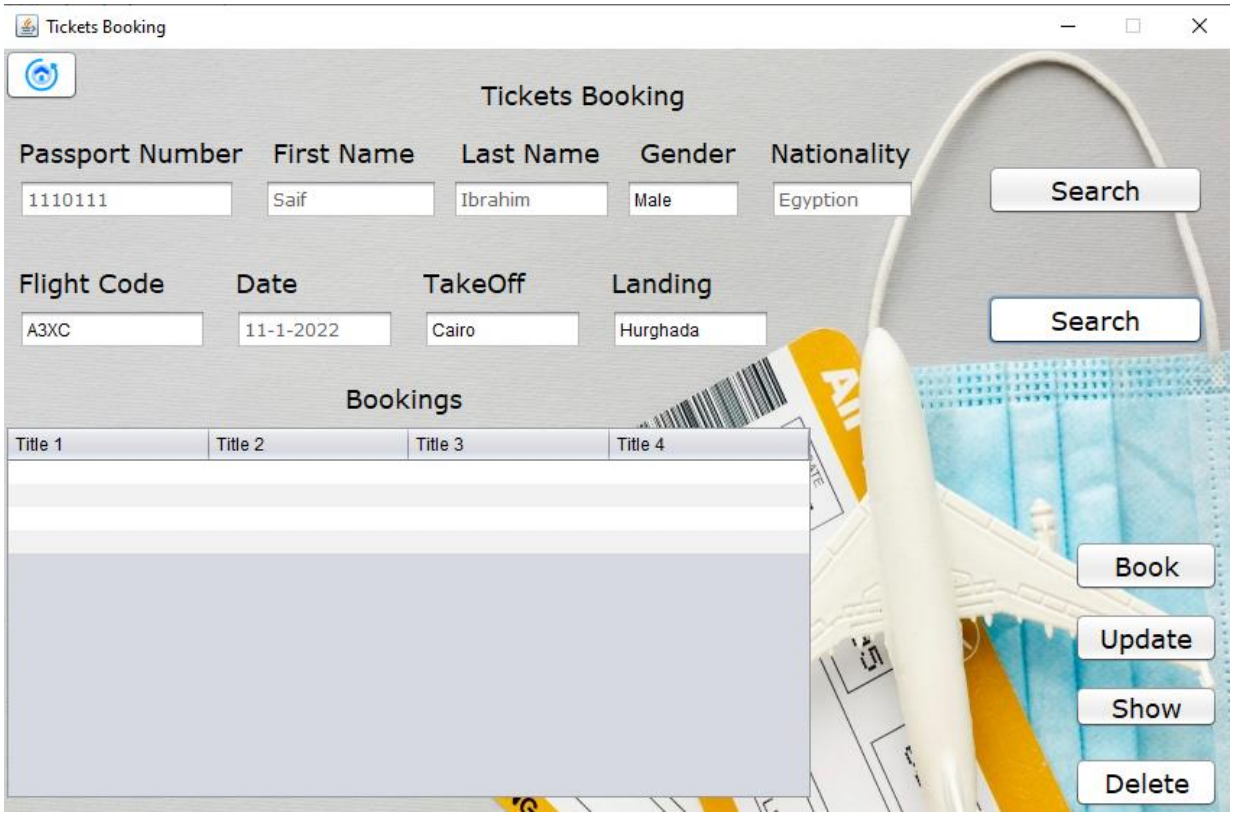


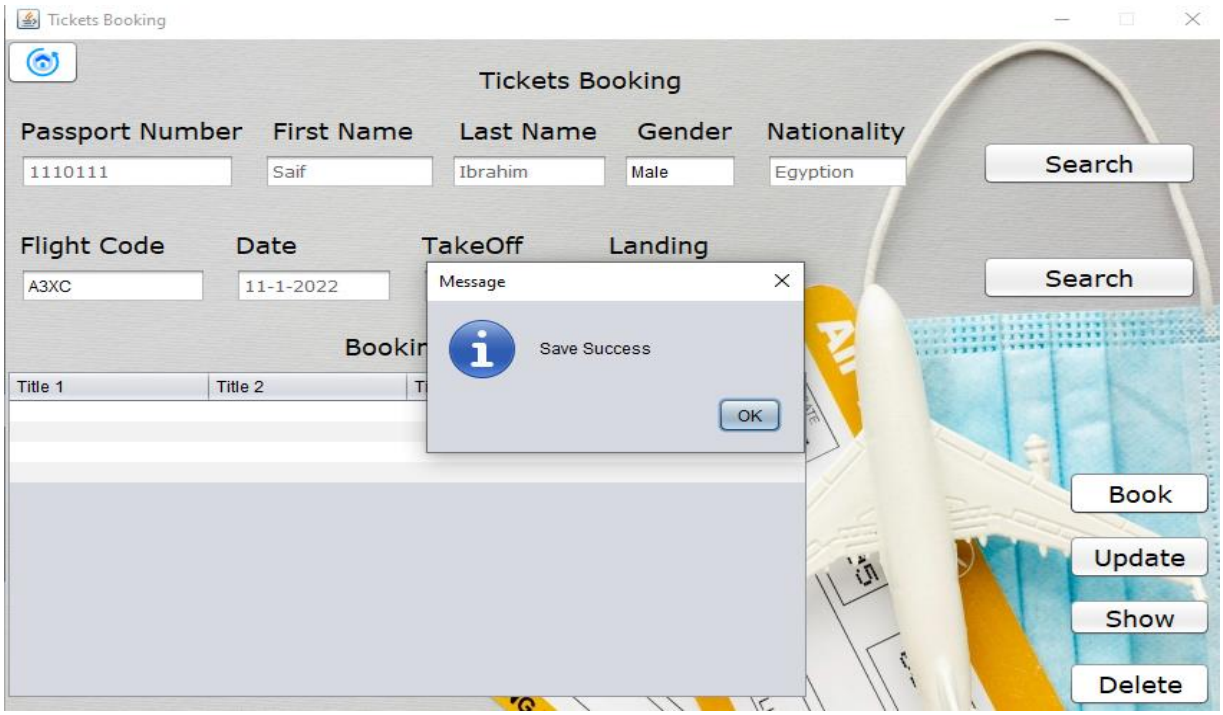
1. Search 1 (search by passport number) :

```
try
        {

            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/airline", "root", "root");
            PreparedStatement stmt = con.prepareStatement("select * from  passengers where
passport_id = ?");
            String passport_id = tf_passportNumber.getText();
            stmt.setString( 1, passport_id );
            ResultSet set = stmt.executeQuery();
            if (set.next()) {
               tf_Fname.setText(set.getString("pass_fname"));
               tf_Lname.setText(set.getString("pass_lname"));
               tf_gender.setText(set.getString("pass_gender"));
               tf_nationality.setText(set.getString("pass_nationality"));

            }
            else
            {
               JOptionPane.showMessageDialog(this, "Not Found"); //if the id not in the table
            }

        } catch (SQLException ex) {
            System.out.println(ex); }
```

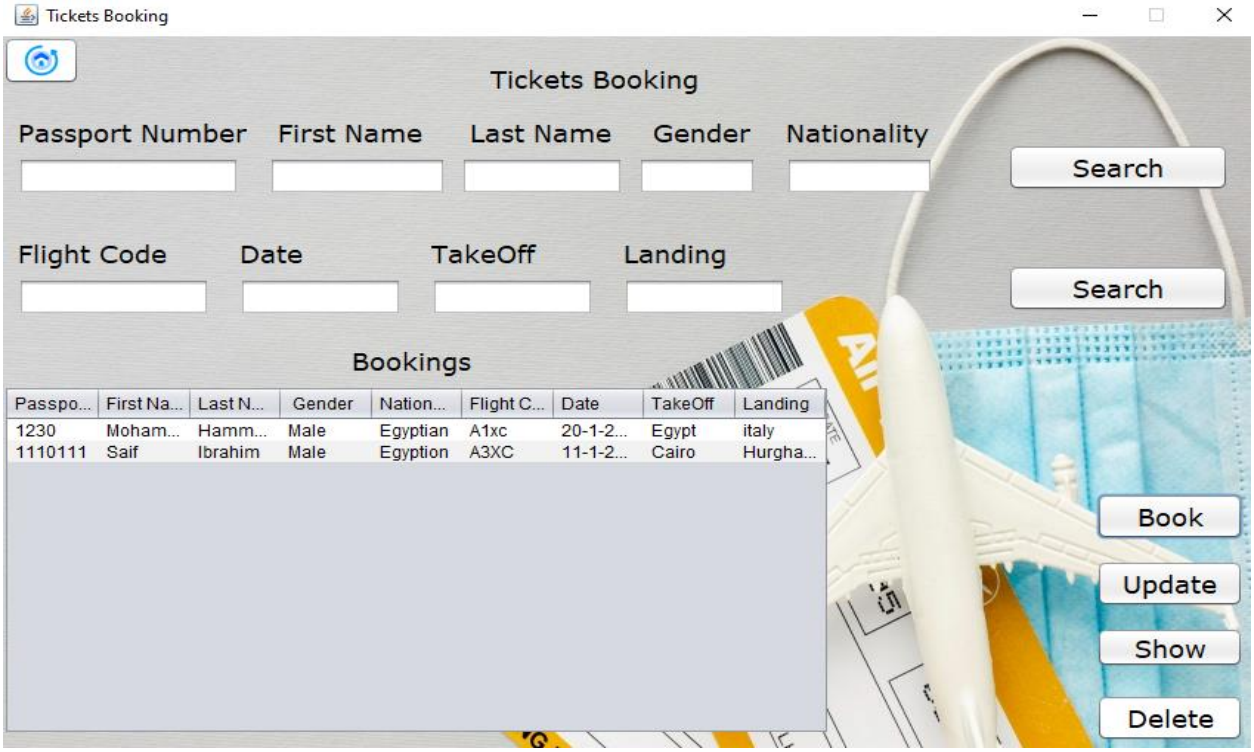
2. Search 2 (Search by flight code):

```
try {
        //String fl_id = tf_flightcode.getText();
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/airline", "root", "root");
        PreparedStatement stmt = con.prepareStatement("select fl_date,fl_takeoff,fl_landing
from flights  where fl_id = '" + tf_flightcode.getText() + "';");
        ResultSet rs = stmt.executeQuery();
        String date="" ,fl_takeoff="" ,fl_landing="";
        while(rs.next())
        {
            date =rs.getString(1);
            fl_takeoff =rs.getString(2);
           fl_landing=rs.getString(3);
        }

        if(!date.isEmpty()&&!fl_takeoff.isEmpty()&&!fl_landing.isEmpty())
        {
        tf_date.setText(date);
        tf_takeoff.setText(fl_takeoff);
        tf_landing.setText(fl_landing);

        }else{
              JOptionPane.showMessageDialog(this, "Not Found"); //if the id not in the table
        }



    } catch (SQLException ex) {
       System.out.println(ex);
                                  }
```


Tickets Booking
Tickets Booking
Passport Number
First Name
Last Name
Gender
Nationality
1110111
Saif
Ibrahim
Male
Egyption
Search
Flight Code
Date
TakeOff
Landing
A3XC
11-1-2022
Cairo
Hurghada
Search
Bookings
Title 1
Title 2
Title 3
Title 4
Book
Update
Show
Delete


Tickets Booking
Tickets Booking
Passport Number
First Name
Last Name
Gender
Nationality
1110111
Saif
Ibrahim
Male
Egyption
Search
Flight Code
Date
TakeOff
Landing
A3XC
11-1-2022
Search
Message
Save Success
OK
Title 1
Title 2
Book
Update
Show
Delete

✓ **Show :**


Tickets Booking
Tickets Booking
Passport Number
First Name
Last Name
Gender
Nationality
Search
Flight Code
Date
TakeOff
Landing
Search
Bookings
Passpo... First Na... Last N... Gender Nation... Flight C... Date TakeOff Landing
1230 Moham... Hamm... Male Egyptian A1xc 20-1-2... Egypt italy
1110111 Saif Ibrahim Male Egyption A3XC 11-1-2... Cairo Hurgha...
Book
Update
Show
Delete

Mouse Click Event (In table Ticket frame **):**

```
int row_num = t_ticket.getSelectedRow();
        String passport_id  = t_ticket.getValueAt(row_num,
0).toString();
        String pass_fname = t_ticket.getValueAt(row_num,
1).toString();
        String pass_lname= t_ticket.getValueAt(row_num,
2).toString();
        String pass_gender = t_ticket.getValueAt(row_num,
3).toString();
        String pass_nationalitypassengers =
t_ticket.getValueAt(row_num, 4).toString();
        String fl_date  = t_ticket.getValueAt(row_num, 6).toString();
        String fl_id  = t_ticket.getValueAt(row_num, 5).toString();
        String fl_takeoff  = t_ticket.getValueAt(row_num,
7).toString();
        String fl_landing  = t_ticket.getValueAt(row_num,
8).toString();



        tf_passportNumber.setText(passport_id);
        tf_Fname.setText(pass_fname);
        tf_Lname.setText(pass_lname);
        tf_gender.setText(pass_gender);
        tf_nationality.setText(pass_nationalitypassengers);
        tf_date.setText(fl_date);
        tf_flightcode.setText(fl_id);
        tf_takeoff.setText(fl_takeoff);
        tf_landing.setText(fl_landing);
```

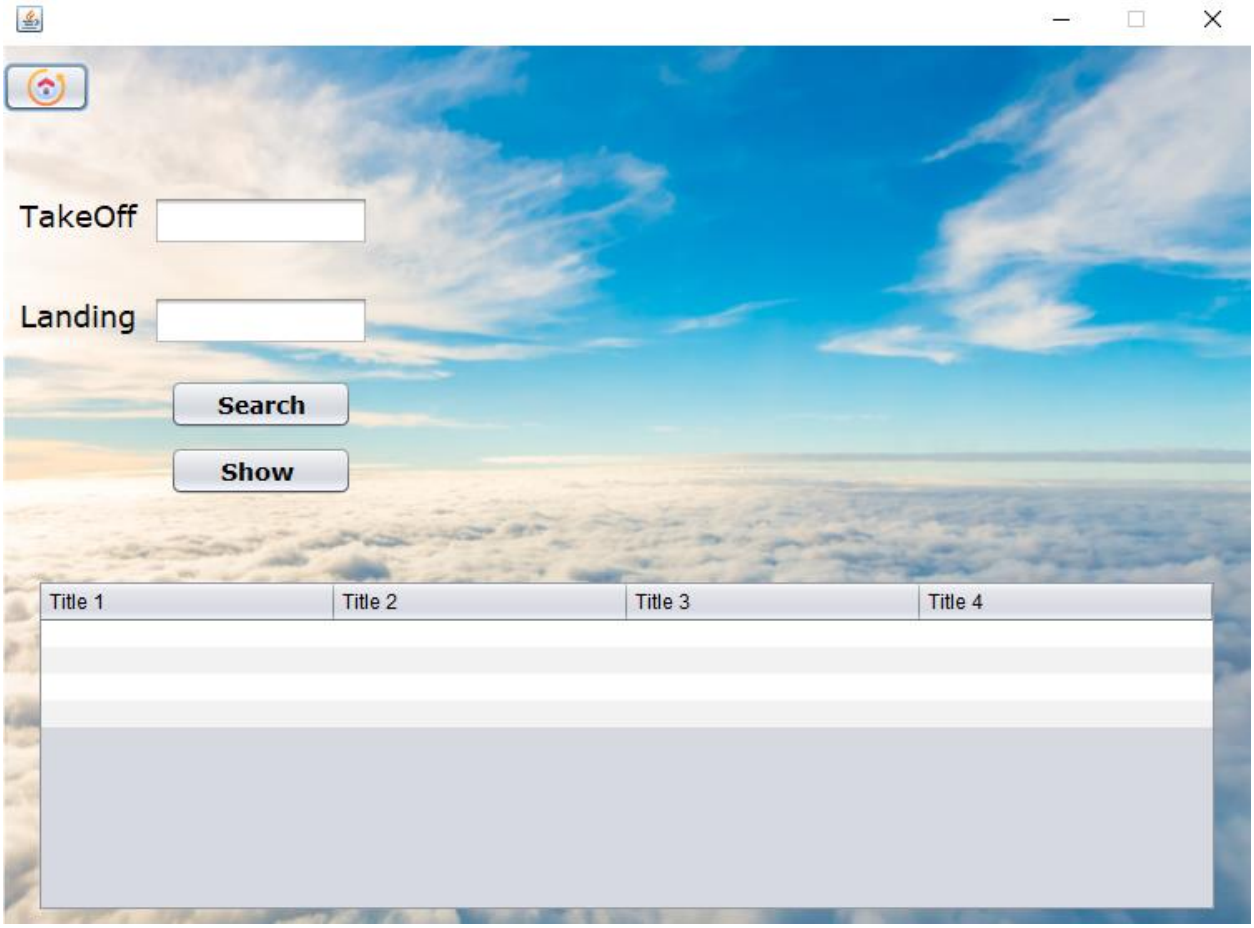
6.Frame Six :

Information , it contains :

iv. Text fields (**Takeoff ,Landing**).

v. Buttons (**Search , Show ,Back**).

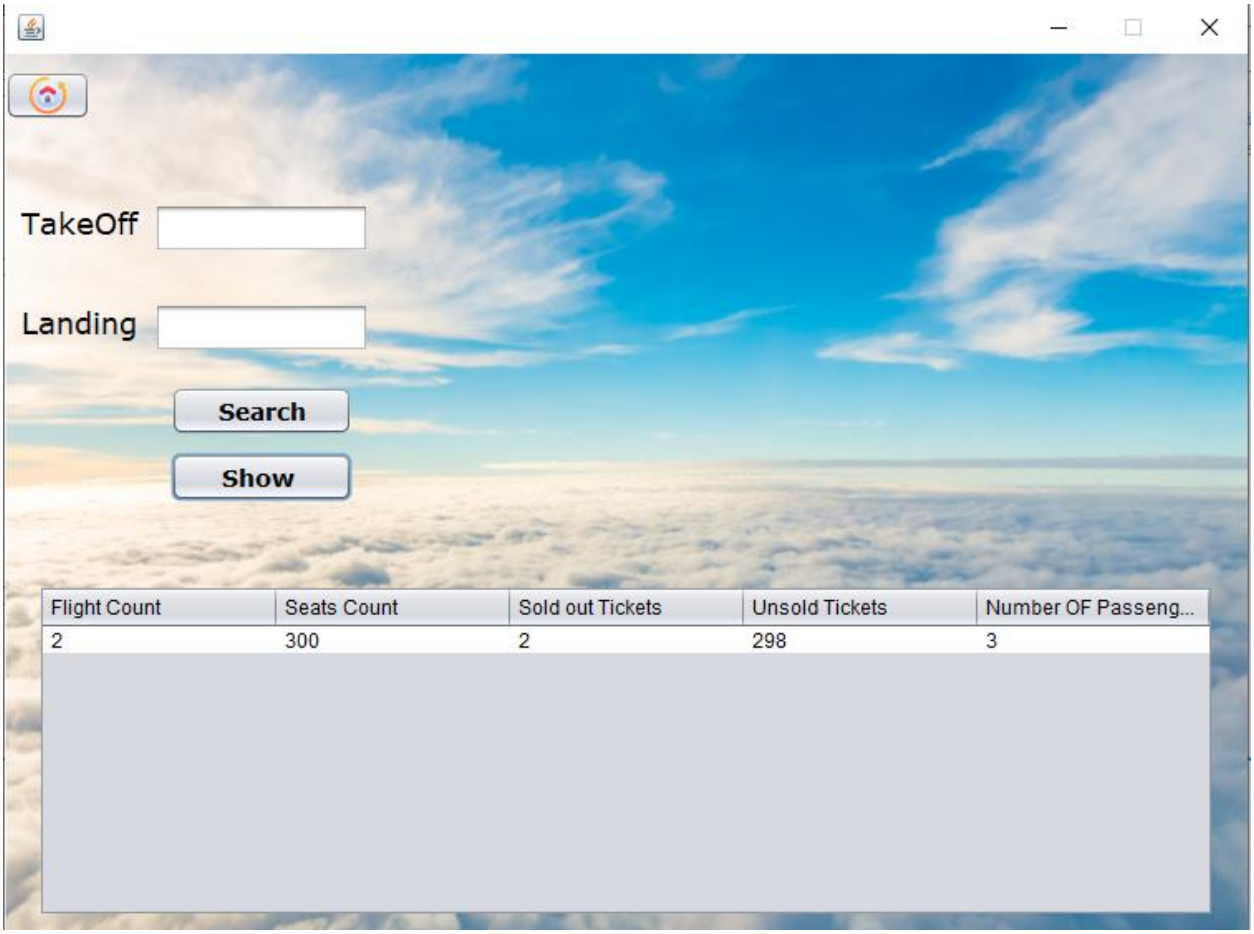
vi. Table to display the results



✓ **Show :**

```
try {
   dtm2=new DefaultTableModel();
   dtm2.addColumn("Flight Count");
   dtm2.addColumn("Seats Count");
   dtm2.addColumn("Sold out Tickets ");
   dtm2.addColumn("Unsold Tickets ");
   dtm2.addColumn("Number OF Passengers");
```

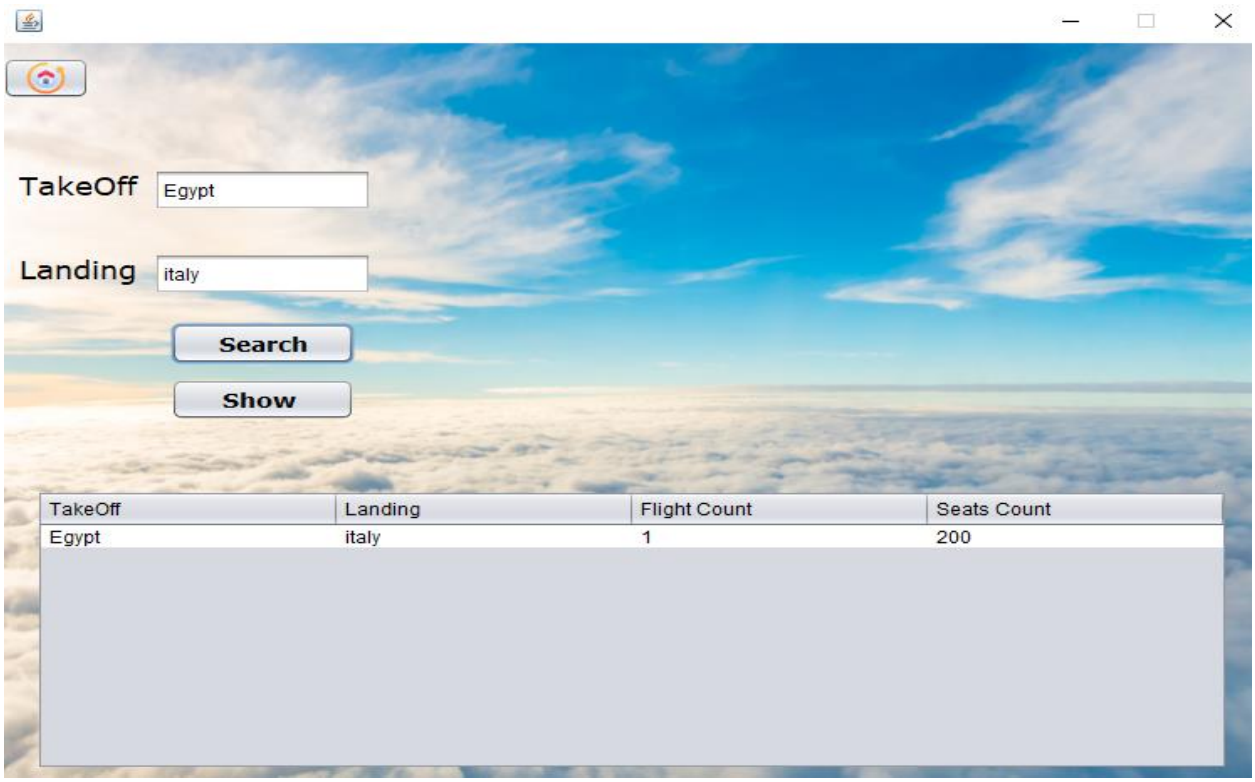
```
        int Fl_seats=0,Fl_count=0,SoutTicket=0,passenger=0;
       PreparedStatement stat1 =con.prepareStatement("SELECT
count(*),sum(fl_seats)  from   flights  ;");
        ResultSet rs1 = stat1.executeQuery();
           while (rs1.next()) {
                Fl_count=rs1.getInt(1);
               Fl_seats=rs1.getInt(2);
           }

      PreparedStatement stat2 =con.prepareStatement("select count(*) from
tickets ;");
        ResultSet rs2 =stat2.executeQuery();
         while(rs2.next())
         {
            SoutTicket = rs2.getInt(1);
         }
        PreparedStatement stat3 =con.prepareStatement("select count(*) from
passengers;");
        ResultSet rs3=stat3.executeQuery();
        while(rs3.next())
        {
           passenger=rs3.getInt(1);
        }
        int Unsold = Fl_seats-SoutTicket;



        dtm2.addRow(new
Object[]{Fl_count,Fl_seats,SoutTicket,Unsold,passenger});
        t_info.setModel(dtm2);


        } catch (SQLException ex) {
          Logger.getLogger(info.class.getName()).log(Level.SEVERE, null, ex);
}
```



✓ Search :

```
if(!tf_takeoff.getText().isEmpty()&&!tf_landing.getText().isEmpty()){


        try {
            dtm=new DefaultTableModel();
            dtm.addColumn("TakeOff");
            dtm.addColumn("Landing");
            dtm.addColumn("Flight Count");
            dtm.addColumn("Seats Count");

            PreparedStatement stat1 = con.prepareStatement("SELECT fl_takeoff
,fl_landing,count(*),sum(fl_seats)  from   flights  WHERE fl_takeoff =
'"+tf_takeoff.getText()+"' AND fl_landing ='"+tf_landing.getText()+"';");
```

```
            ResultSet rs = stat1.executeQuery();
            while(rs.next())
            {
               dtm.addRow(new
Object[]{rs.getString(1),rs.getString(2),rs.getInt(3),rs.getInt(4)});
            }
             t_info.setModel(dtm);

        } catch (SQLException ex) {
            Logger.getLogger(info.class.getName()).log(Level.SEVERE, null, ex);
        }
        }else{
              JOptionPane.showMessageDialog(frame,
      " Check TakeOff And Landing ",
     "Failed",
      JOptionPane.ERROR_MESSAGE);
        }
```



Conclusion:

The main objective of the Airlines Reservation System is to manage the details of Airlines Ticket,Flights,Customer,Booking Counter and to skip the long booking lines at the airport,skipping the person_contact in counter helps with stopping the panadimic.