

# Airport Management System

## Contents

Airport Management System.....	1
1. Login class : .....	2
2. Home class:.....	3
3. Passengers package: .....	4
a. Passengers class:.....	4
b. AddPassForm class:.....	4
c. searchPassenger class:.....	5
d. updatePass class: .....	5
4. Staff package:.....	6
1. Staff class: .....	6
2. ManageStaff class: .....	6
3. searchStaff class: .....	7
5. Airplanes class:.....	8
6. Flights package:.....	9
a. Flights class: .....	9
b. manageFlights class: .....	9
c. searchFlight class: .....	10

# 1.Login class :

A login algorithm under Login.java class with two methods: signup & Login.

>Signup:

```
//this method is for adding new users in the database
private void signupbtnActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/samer", "samer", "samer");
        String query = "insert into LOGIN values(?,?)";
        PreparedStatement stmt = con.prepareStatement(query);

        String name = username.getText();//get username from textfield
        String password = passw.getText();//get password from textfield

        stmt.setString(1, name);//insert username in LOGIN table
        stmt.setString(2, password);//insert password in LOGIN table

        stmt.execute();//execute query
        JOptionPane.showMessageDialog(null, "Thank you for your registration "+name+", You can now login to our");

    } catch (SQLException ex) {
        Logger.getLogger(AddPassForm.class.getName()).log(Level.SEVERE, null, ex);
    }
    System.exit(0);//exit java application after signup
}
```

>login:

```
//once the login button is clicked this method is on
private void loginbtnActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/samer", "samer", "samer");

        String user = username.getText();//get the written username from the username textfield
        String passw = passw.getText();//get the written username from the password textfield

        //after we get the written username and password this query will get them from LOGIN table
        String query = "SELECT USERNAME,PASSWORD FROM LOGIN where USERNAME='"+user+"' and PASSWORD='"+passw+"'";
        Statement stmt = con.createStatement();
        ResultSet result = stmt.executeQuery(query);

        //check if the result returned by the query is empty or not
        if(result.next()){
            String userc = result.getString("USERNAME");
            JOptionPane.showMessageDialog(null, "Welecome "+userc);
            //if username and password true then the user will be redirected to travel management system frame
            Traveling_management_system backBtn = new Traveling_management_system();
            backBtn.setVisible(true);
        } else{
            //if username and password false then the system will exit
            JOptionPane.showMessageDialog(null, "Wrong password or username, please try again. bye bye.");
            System.exit(0);
        }
    } catch (SQLException ex) {
        Logger.getLogger(AddPassForm.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

## 2.Home class:

The home class displays a menu containing 4 buttons, each will redirect you to a class:

- ✓ Button1: manage Passengers(Passengers class)
- ✓ Button2: manage Staff(staff class)
- ✓ Button3: manage Airplanes(Airplane class)
- ✓ Button4: manage Flights(Flights class)

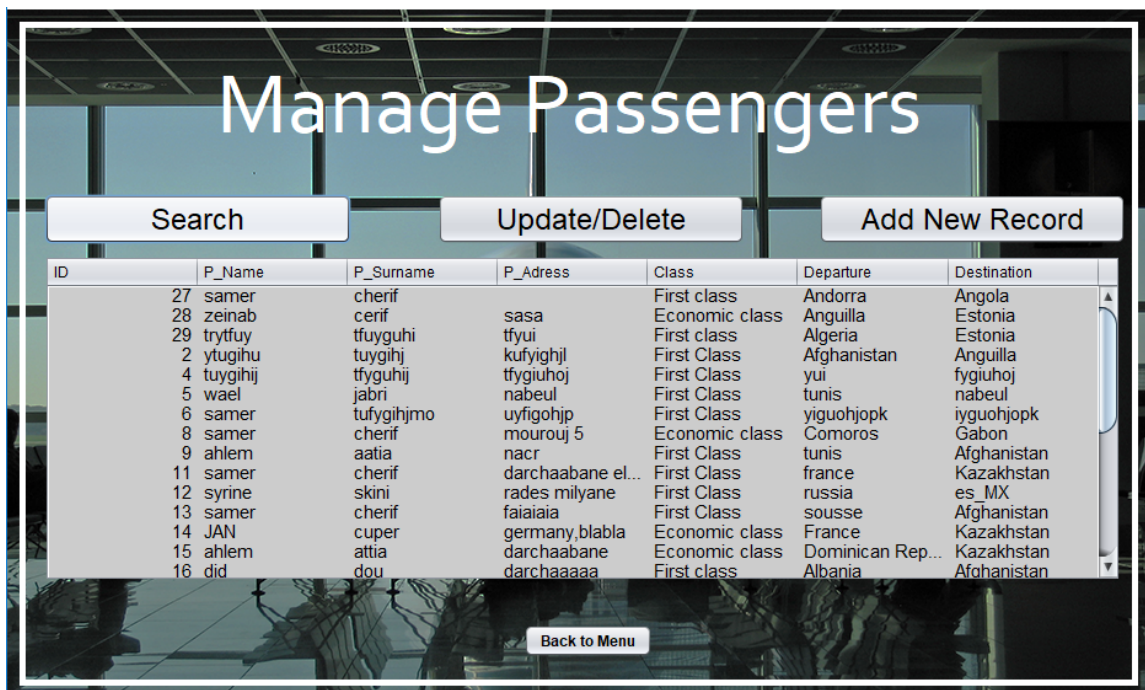


### 3.Passengers package:

This package contains four classes to manage passengers.

#### a. Passengers class:

This class displays all the info about the passengers from the database into a table.



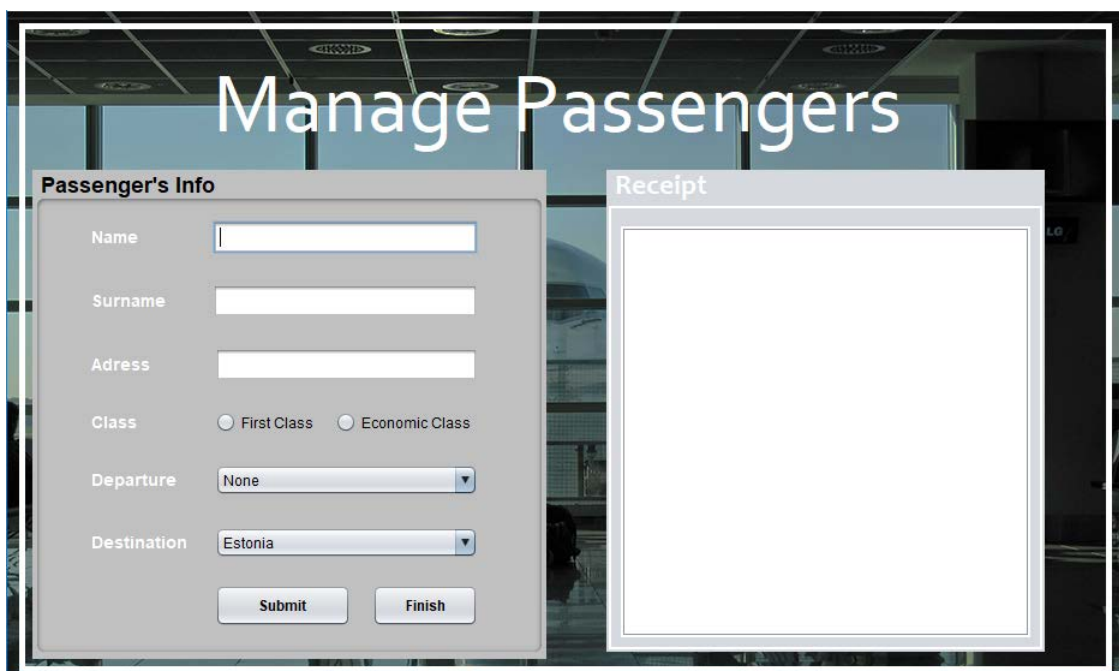
The screenshot shows a window titled "Manage Passengers" with three buttons at the top: "Search", "Update/Delete", and "Add New Record". Below these buttons is a table with the following data:

ID	P_Name	P_Surname	P_Adress	Class	Departure	Destination
27	samer	cherif		First class	Andorra	Angola
28	zeinab	cerif	sasa	Economic class	Anguilla	Estonia
29	trytuy	tfuyguhi	tfyui	First class	Algeria	Estonia
2	ytugihu	tuygiyh	kufyighjl	First Class	Afghanistan	Anguilla
4	tuygi hij	tfygu hij	tfygiuhoj	First Class	yui	fygiuhoj
5	wael	jabri	nabeul	First Class	tunis	nabeul
6	samer	tufygi hmo	uyfigohjp	First Class	yiguohjopk	iyguohjopk
8	samer	cherif	mourouj 5	Economic class	Comoros	Gabon
9	ahlem	aatia	nacr	First Class	tunis	Afghanistan
11	samer	cherif	darchaabane el...	First Class	france	Kazakhstan
12	syri ne	skini	rades milyane	First Class	ru ssia	es_MX
13	samer	cherif	faiaiaia	First Class	sousse	Afghanistan
14	JAN	cuper	germany,blabla	Economic class	France	Kazakhstan
15	ahlem	attia	darchaabane	Economic class	Dominican Rep...	Kazakhstan
16	did	dou	darchaaaaa	First class	Albania	Afohanistan

At the bottom of the window is a "Back to Menu" button.

#### b. AddPassForm class:

This class is responsible for adding new passengers to table PASSENGERS2 in the database.



The screenshot shows a window titled "Manage Passengers" with two main sections: "Passenger's Info" and "Receipt".

The "Passenger's Info" section contains the following fields and controls:

- Name:
- Surname:
- Adress:
- Class: ☐ First Class ☐ Economic Class
- Departure:
- Destination:
- Submit:
- Finish:

The "Receipt" section is a large empty box for displaying the receipt.

### c. searchPassenger class:

Once we click on “Search”, this class is responsible for the search and display part of the passengers data from the database.

The screenshot shows a web application titled "Manage Passengers". It features a search form with the following fields and controls:

- Search By:** Name (text input), Surname (text input), Class (dropdown menu set to "None"), and a "search" button.
- Departure:** A dropdown menu set to "None".
- Arrival:** A dropdown menu set to "None".
- A "refresh" button.

Below the search form is a table with the following headers: ID, P\_Name, P\_Surname, P\_Address, Class, Departure, and Destination. The table body is currently empty. At the bottom center of the interface is a "Finish" button.

### d. updatePass class:

this class is where we update or delete passengers from the database.

The screenshot displays two overlapping web application interfaces. On the left is the "Edit Passenger" form, and on the right is the "Passengers" list view.

**Edit Passenger Form:**

- Passenger ID:** A dropdown menu showing "27".
- Address:** A text input field containing "sa sa".
- Departure:** A dropdown menu showing "Estonia".
- Destination:** A dropdown menu showing "Estonia".
- Class:** A dropdown menu showing "Economic class".
- Address:** A text input field containing "sa".
- Buttons: "Update" and "Delete".

**Passengers List View:**

At the top, there are two buttons: "Update/Delete" and "Add New Re". Below these is a table with the following headers: ID, P\_Name, P\_Surname, P\_Address, Class, Departure, and Destination. The table contains several rows of passenger data.

ID	P_Name	P_Surname	P_Address	Class	Departure	Destination
15	ahlem	attia	darchaabane	Economic class	Dominican Rep...	Kazakhstan
16	did	dou	darchaaaaa	First class	Albania	Afghanistan

At the bottom center of the interface is a "Back to Menu" button.



## 4. Staff package:

This package contains three classes to manage passengers.

### 1. Staff class:

This class displays all the info about the staff from the database into a table.



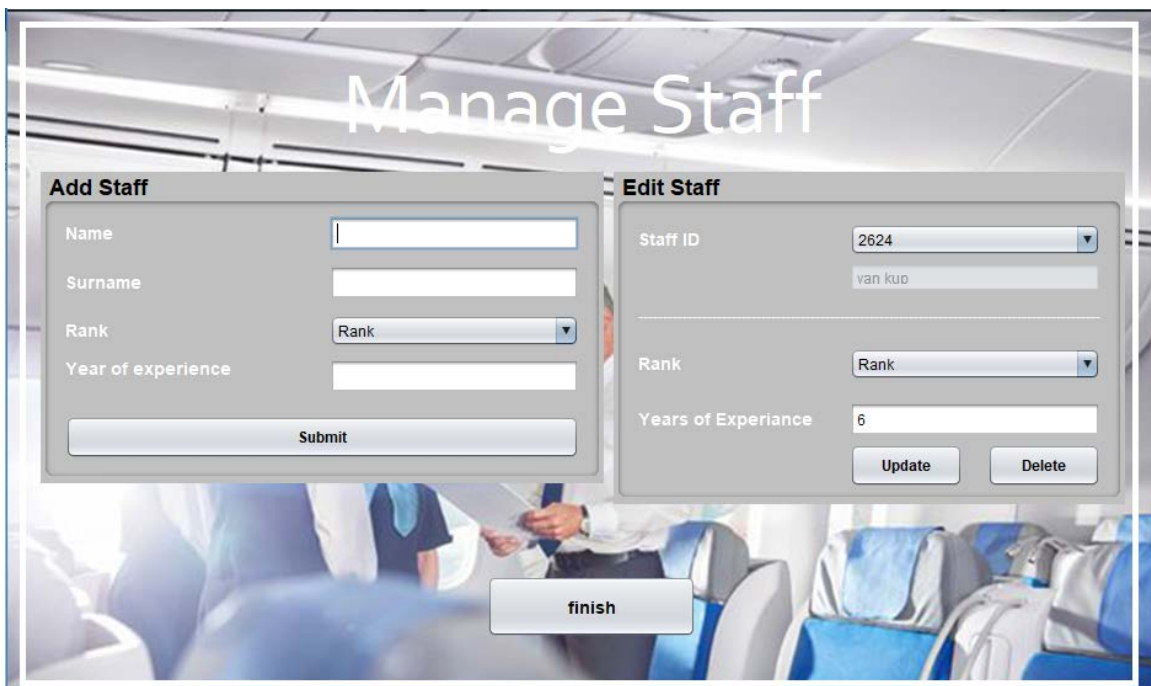
The screenshot shows a web application titled "Manage Staff". It features a "Search" button and an "Edit/Remove Staff" button. Below these buttons is a table displaying staff information. The table has five columns: ID, Name, Surname, Rank, and YearsExp. The data rows are as follows:

ID	Name	Surname	Rank	YearsExp
2897	jan	kuper	Airline Administrative	50
2624	yan	kup	Rank	6
4464	sana	attia	Rank	30
3579	maisaaa	chacha	Aviation Meteorologist	52

At the bottom of the interface is a "Back to Menu" button.

### 2. ManageStaff class:

Once we click on "Edit/Remove Staff", This class is where we can add, update or remove staff members from the table STAFF from database.



The screenshot shows the "Manage Staff" interface with two main forms: "Add Staff" and "Edit Staff".

**Add Staff Form:**

- Name:
- Surname:
- Rank:
- Year of experience:
- Submit button

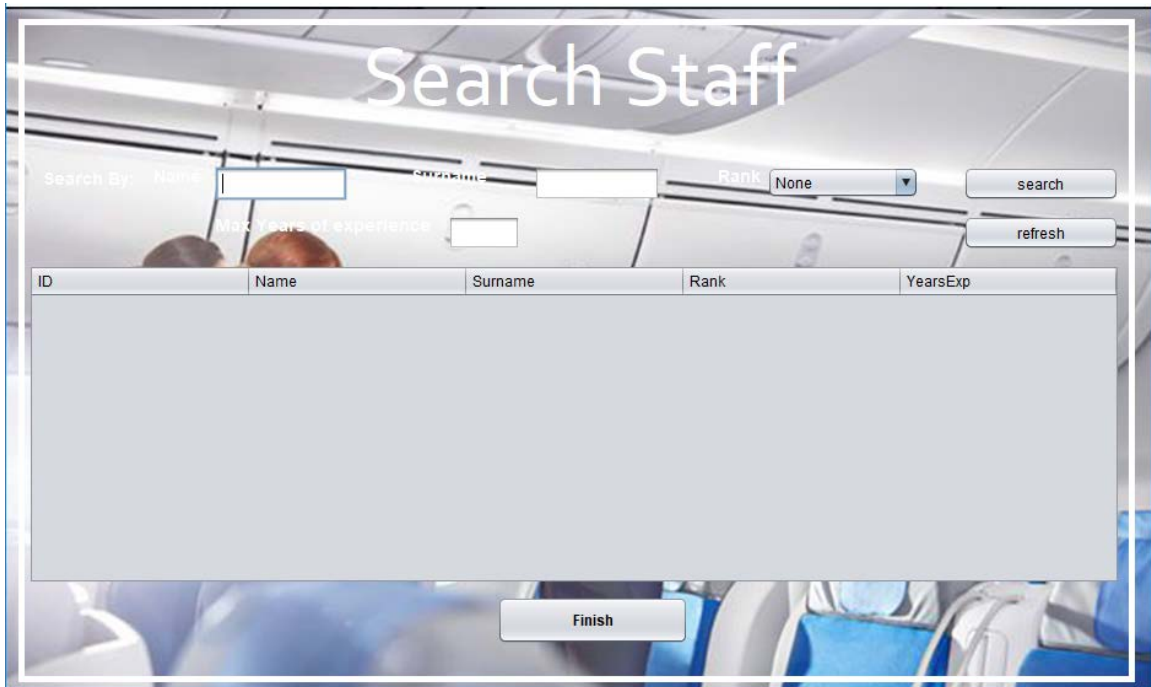
**Edit Staff Form:**

- Staff ID:
- 
- Rank:
- Years of Experience:
- Update button
- Delete button

At the bottom of the interface is a "finish" button.

### 3. searchStaff class:

Once we click on “Search”, this class is responsible for the search and display part of the Staff’s data from the database.



The screenshot displays a web application titled "Search Staff". The interface includes a search form with the following elements:

- Search By:** A label followed by a text input field for "Name".
- Surname:** A text input field.
- Rank:** A dropdown menu currently set to "None".
- Max Years of experience:** A text input field.
- Buttons:** "search" and "refresh" buttons.

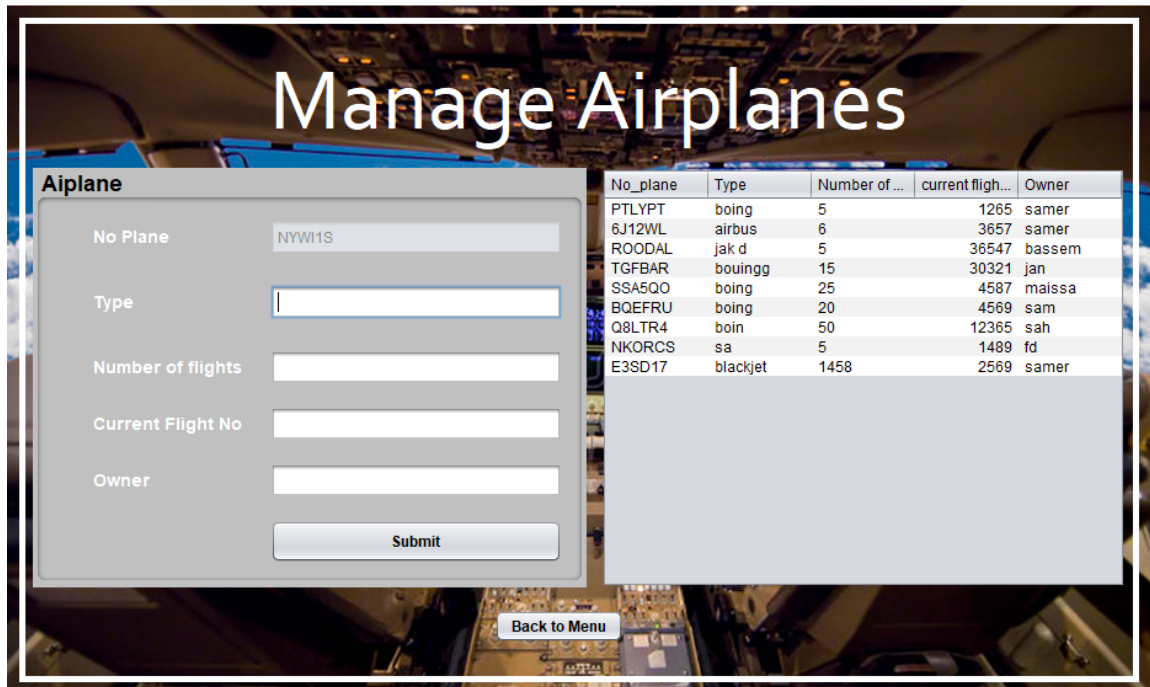
Below the search form is a table with the following headers:

ID	Name	Surname	Rank	YearsExp

At the bottom center of the interface is a "Finish" button.

## 5. Airplanes class:

This class contains only one method to add new airplanes the database table AIRPLANE then displays the data from the database into a table.



# Manage Airplanes

**Aiplane**  

No PlaneNYWI1S

Type

Number of flights

Current Flight No

Owner

Submit

No_plane	Type	Number of ...	current fligh...	Owner
PTLYPT	boing	5	1265	samer
6J12WL	airbus	6	3657	samer
ROODAL	jak d	5	36547	bassem
TGFBAR	bouingg	15	30321	jan
SSA5QO	boing	25	4587	maissa
BQEFRU	boing	20	4569	sam
Q8LTR4	boin	50	12365	sah
NKORCS	sa	5	1489	fd
E3SD17	blackjet	1458	2569	samer

Back to Menu

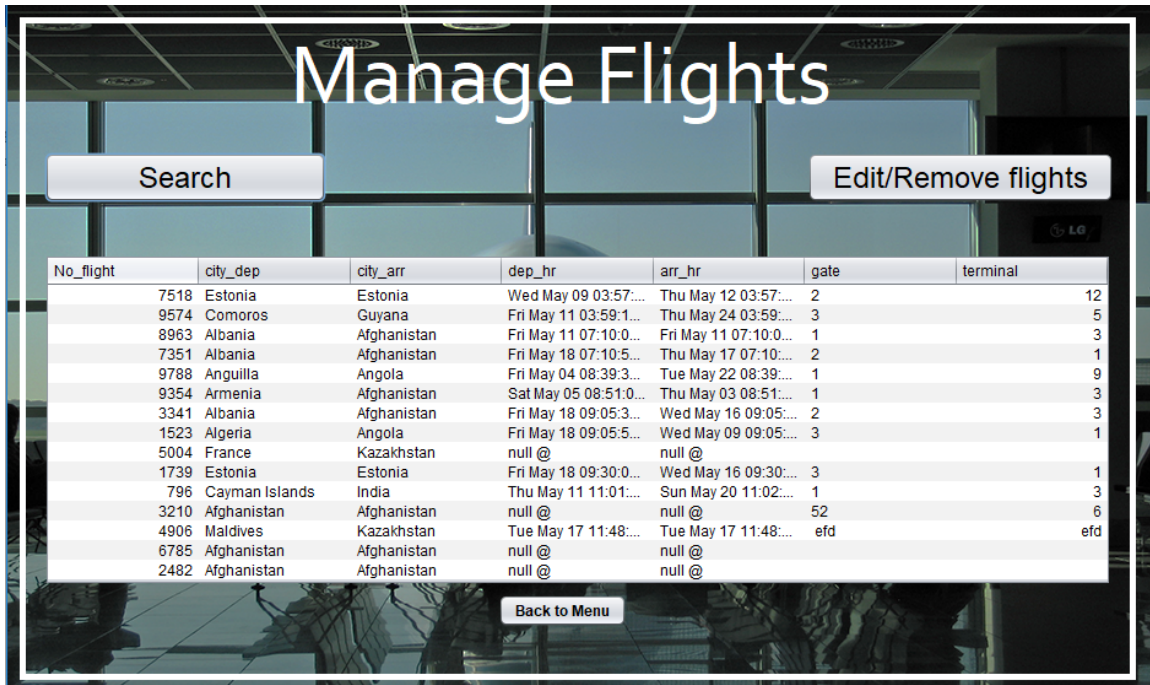


## 6. Flights package:

This package contains three classes to manage flights.

### a. Flights class:

This class displays all the info about the staff from the database into a table.



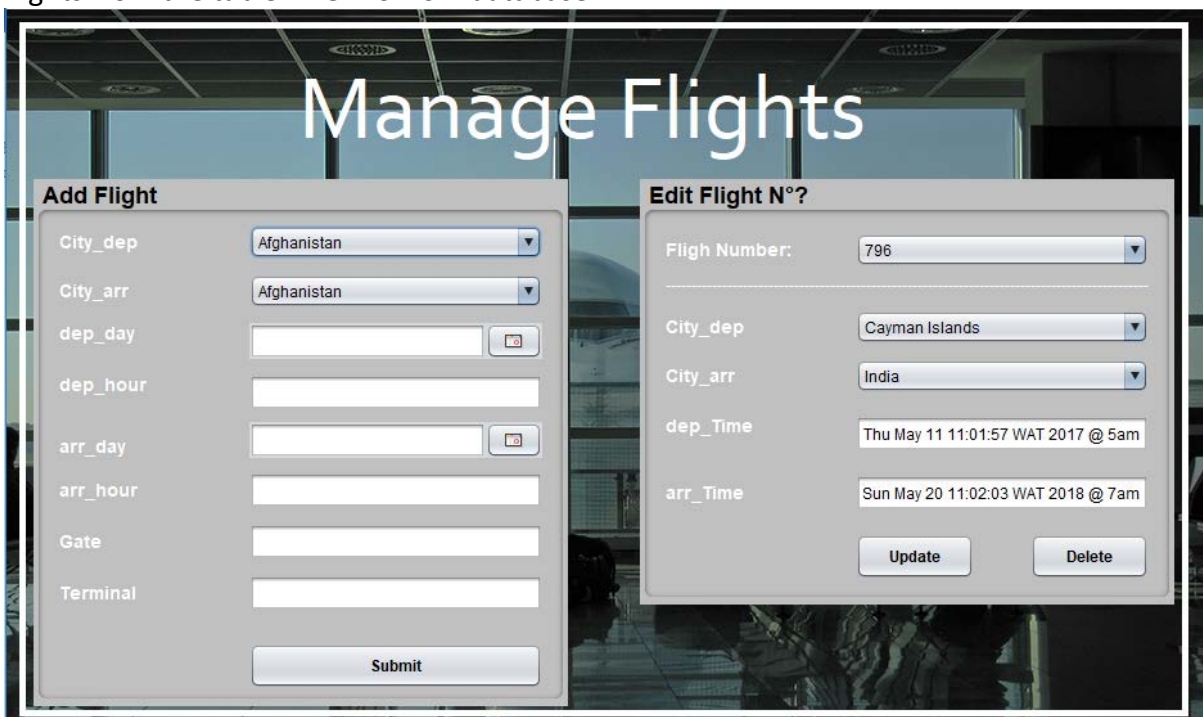
The screenshot shows a web application titled "Manage Flights". It features a "Search" button on the left and an "Edit/Remove flights" button on the right. Below these buttons is a table displaying flight information. The table has columns for No\_flight, city\_dep, city\_arr, dep\_hr, arr\_hr, gate, and terminal. The data is as follows:

No_flight	city_dep	city_arr	dep_hr	arr_hr	gate	terminal
7518	Estonia	Estonia	Wed May 09 03:57:...	Thu May 12 03:57:...	2	12
9574	Comoros	Guyana	Fri May 11 03:59:1...	Thu May 24 03:59:...	3	5
8963	Albania	Afghanistan	Fri May 11 07:10:0...	Fri May 11 07:10:0...	1	3
7351	Albania	Afghanistan	Fri May 18 07:10:5...	Thu May 17 07:10:...	2	1
9788	Anguilla	Angola	Fri May 04 08:39:3...	Tue May 22 08:39:...	1	9
9354	Armenia	Afghanistan	Sat May 05 08:51:0...	Thu May 03 08:51:...	1	3
3341	Albania	Afghanistan	Fri May 18 09:05:3...	Wed May 16 09:05:...	2	3
1523	Algeria	Angola	Fri May 18 09:05:5...	Wed May 09 09:05:...	3	1
5004	France	Kazakhstan	null @	null @		
1739	Estonia	Estonia	Fri May 18 09:30:0...	Wed May 16 09:30:...	3	1
796	Cayman Islands	India	Thu May 11 11:01:...	Sun May 20 11:02:...	1	3
3210	Afghanistan	Afghanistan	null @	null @	52	6
4906	Maldives	Kazakhstan	Tue May 17 11:48:...	Tue May 17 11:48:...	efd	efd
6785	Afghanistan	Afghanistan	null @	null @		
2482	Afghanistan	Afghanistan	null @	null @		

At the bottom of the table, there is a "Back to Menu" button.

### b. manageFlights class:

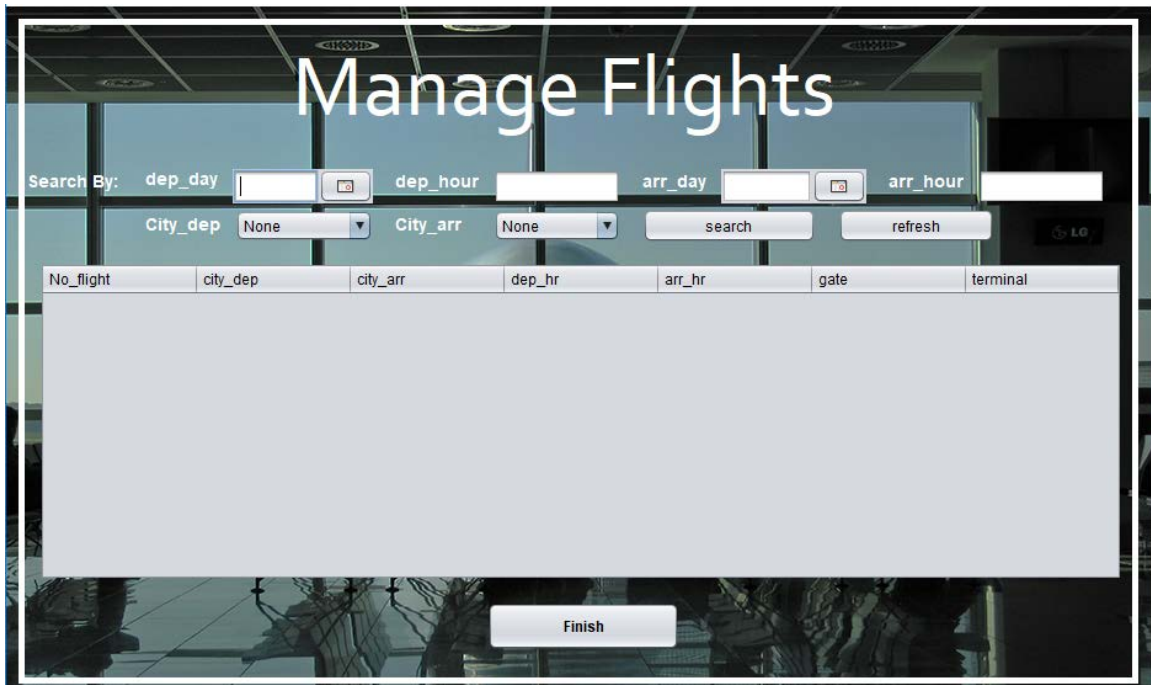
Once we click on “Edit/Remove flights”, This class is where we can add, update or remove flights from the table FLIGHTS from database.



The screenshot shows the "Manage Flights" application with two forms visible. The "Add Flight" form on the left has fields for City\_dep (Afghanistan), City\_arr (Afghanistan), dep\_day, dep\_hour, arr\_day, arr\_hour, Gate, and Terminal. The "Edit Flight N°?" form on the right has a Flight Number field (796), City\_dep (Cayman Islands), City\_arr (India), dep\_Time (Thu May 11 11:01:57 WAT 2017 @ 5am), and arr\_Time (Sun May 20 11:02:03 WAT 2018 @ 7am). Both forms have "Update" and "Delete" buttons at the bottom.

**c. searchFlight class:**

Once we click on “Search”, this class is responsible for the search and display part of the Flights’ data from the database.



The screenshot displays a web application titled "Manage Flights". The interface includes a search section with the following elements:

- Search By:** A label followed by four input fields: `dep_day`, `dep_hour`, `arr_day`, and `arr_hour`. Each input field has a small calendar icon to its right.
- City Selection:** Two dropdown menus labeled `City_dep` and `City_arr`, both currently set to "None".
- Buttons:** A "search" button and a "refresh" button.

Below the search section is a table with the following headers:

No_flight	city_dep	city_arr	dep_hr	arr_hr	gate	terminal

At the bottom center of the interface is a "Finish" button.