

TSIY

The Sky is Yours!

The sky has never been so
close...





Table of contents

1 System definition

2 Users

3 Business Object

4 Technical proposition

5 Navigation diagram

6 Mock-ups

7 Backlog

8 Persistance Layer

9 Web Services Layer

10 Organization



1

System Definition

“The Sky is yours”, the flight sharing platform, is a commercial platform accessible either by mobile application or on website. It is intended for professional/non professional pilots and particulars. It shows offers of trips entered by pilots who want to share the costs with few passengers.

It's designed on the same principle as car sharing, its object is either to move from point A to point B, or to fly over a region to discover it from the top.

Particulars can choose the convenient flight with the pilot they want and pay online. This platform will facilitate them the checking and long tails at airports.

System Definition

“The sky is yours” platform has different functionalities :

- Organizes private flights.
- Allows private pilots and passengers to share the cost of flights.
- Verifies all pilot profiles and their licenses before awarding them permission to publish flight-sharing posts.





- **Pilots**

They subscribe on the platform and send the documents needed, and once they are verified they post a description of the flights on the website such as what they would like to do, their destinations, the routes, the duration and number of seats they wish to share. They will accept or refuse any demand for a flight.

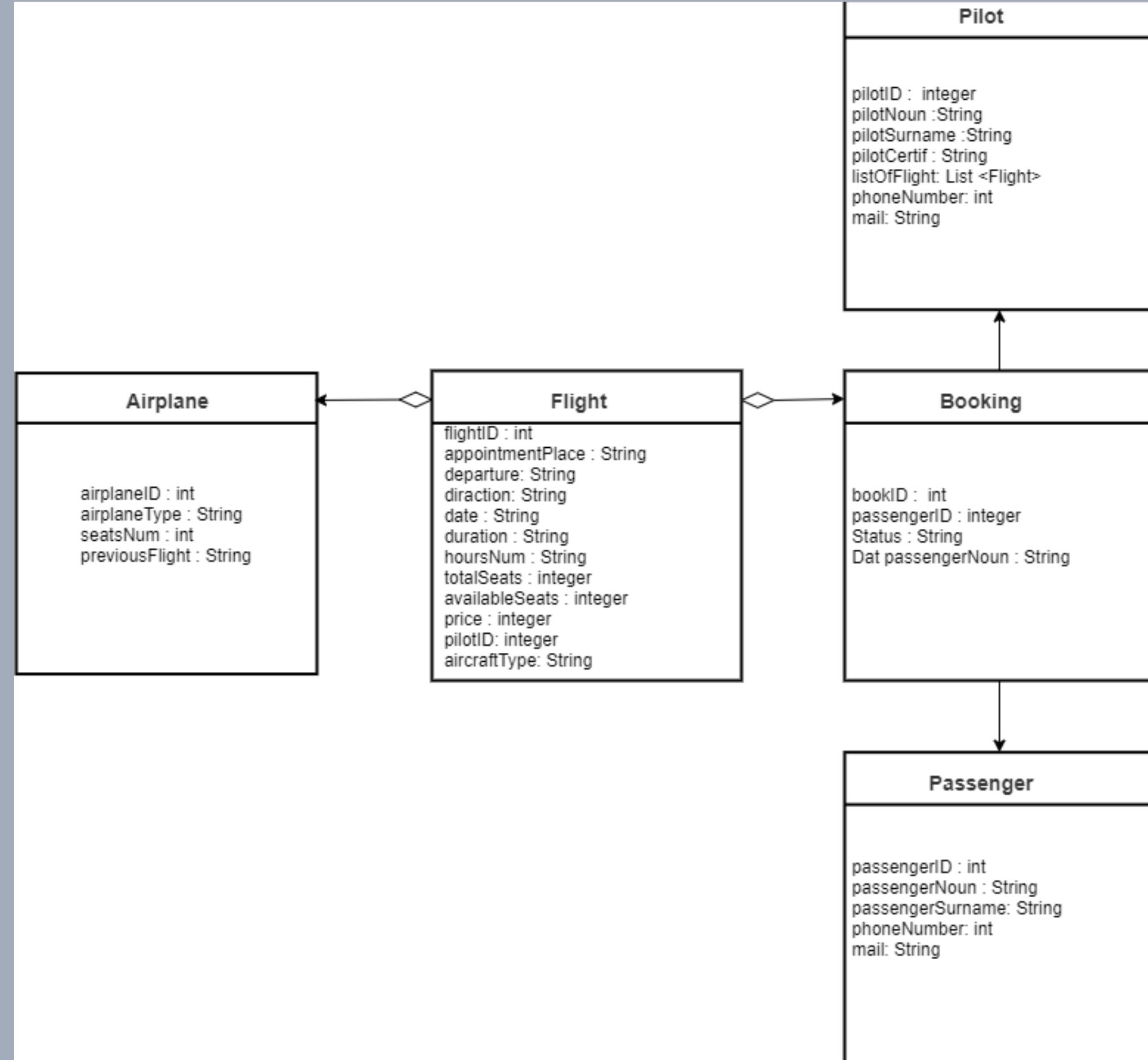
- **The passengers**

Each user can book a seat on the flight sharing and thus share the flight expenses with the pilot. The flight can be for a travel or in order to discover a region seen from the sky .



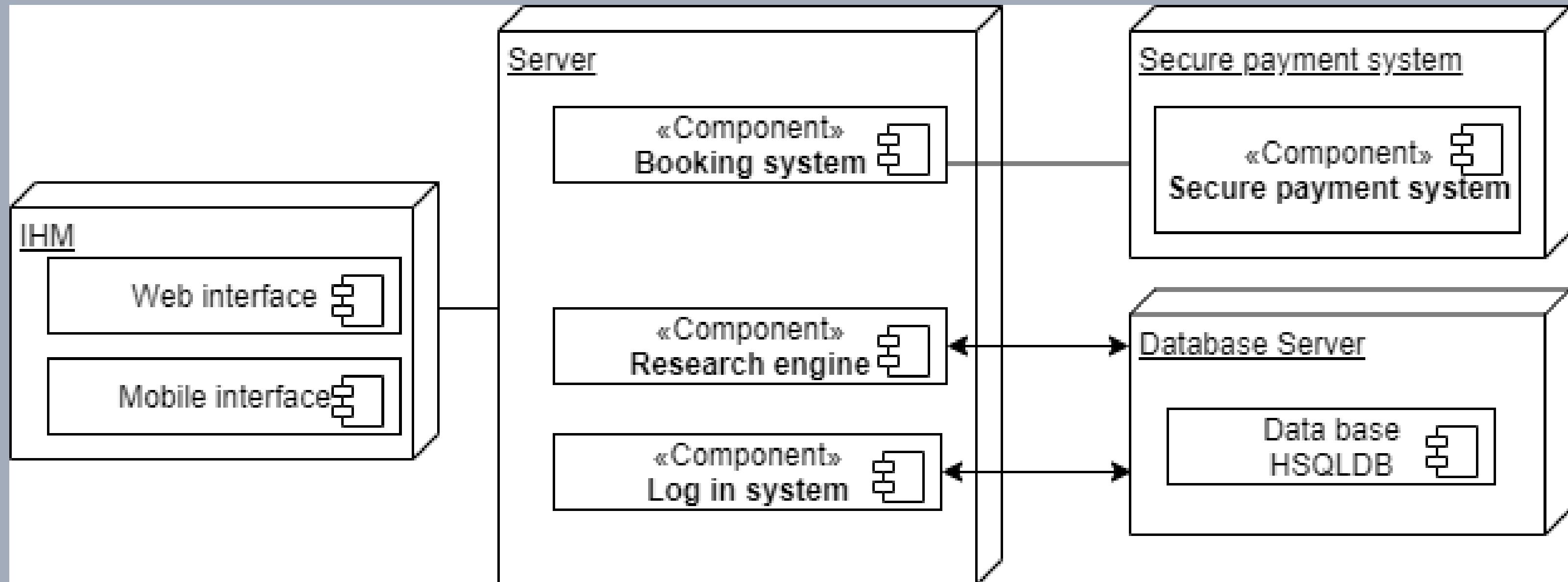
3

Buisness Object



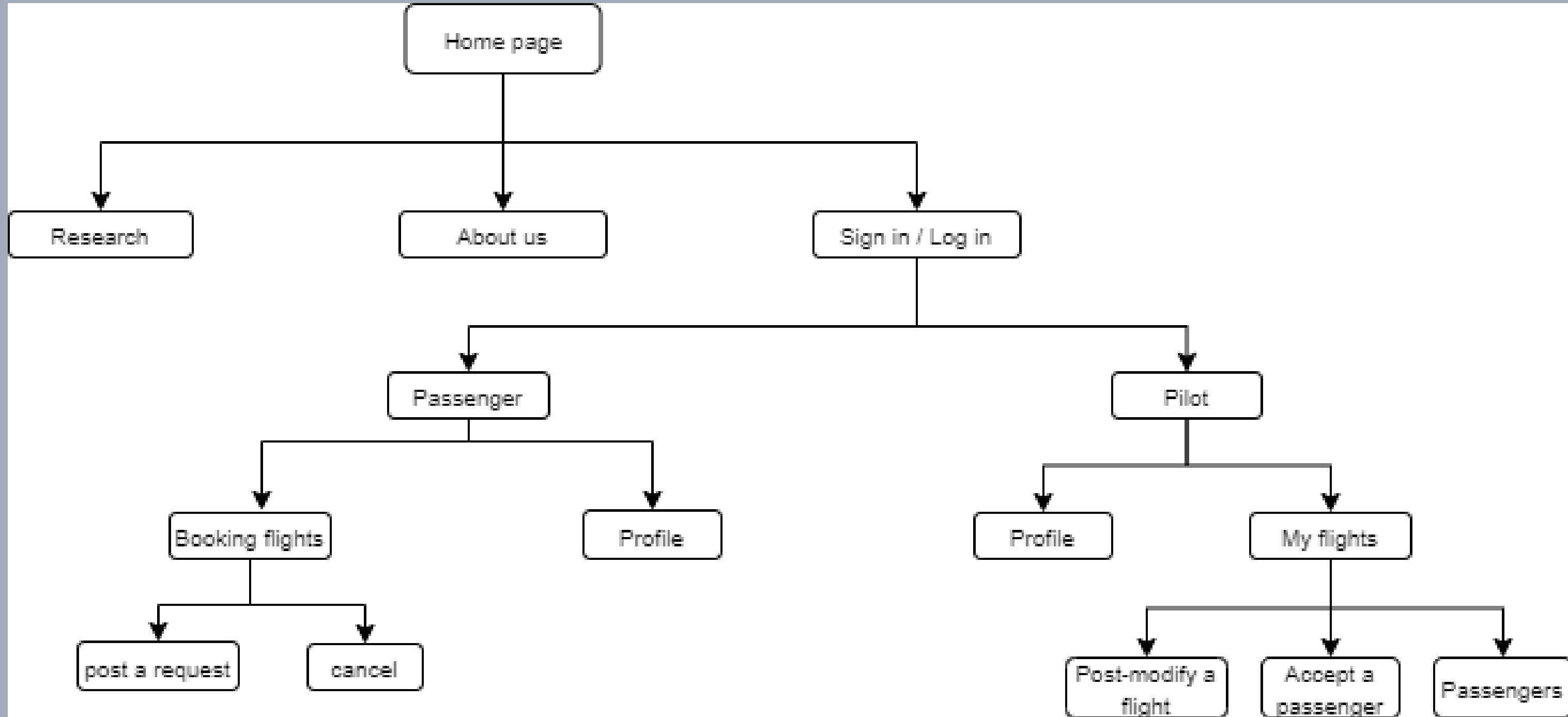


4 Technical proposition





5 Navigation Diagram



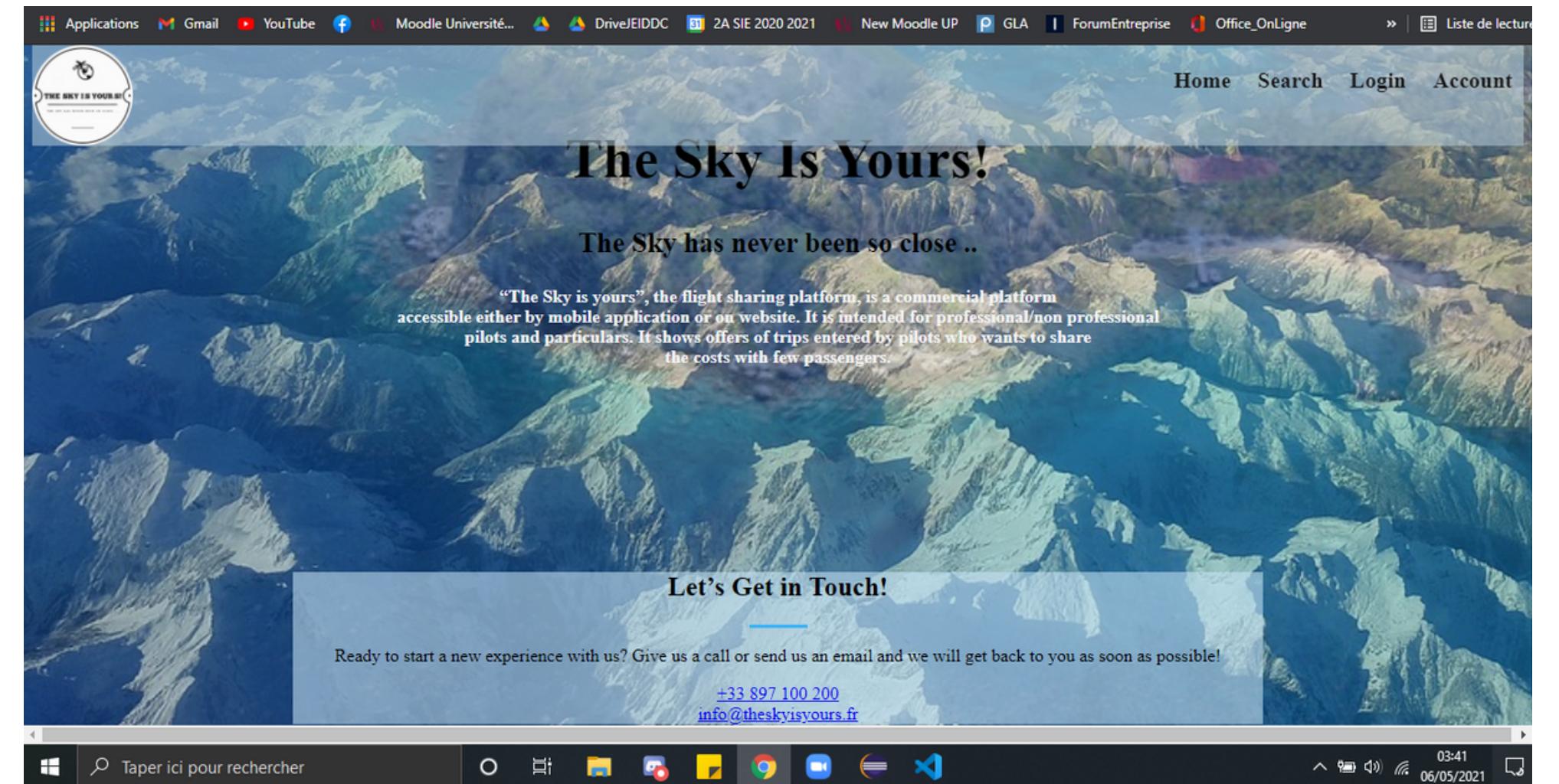
6

Mock_ups



• Home:

Here is the home page which includes an introduction of our interface and contact tools.



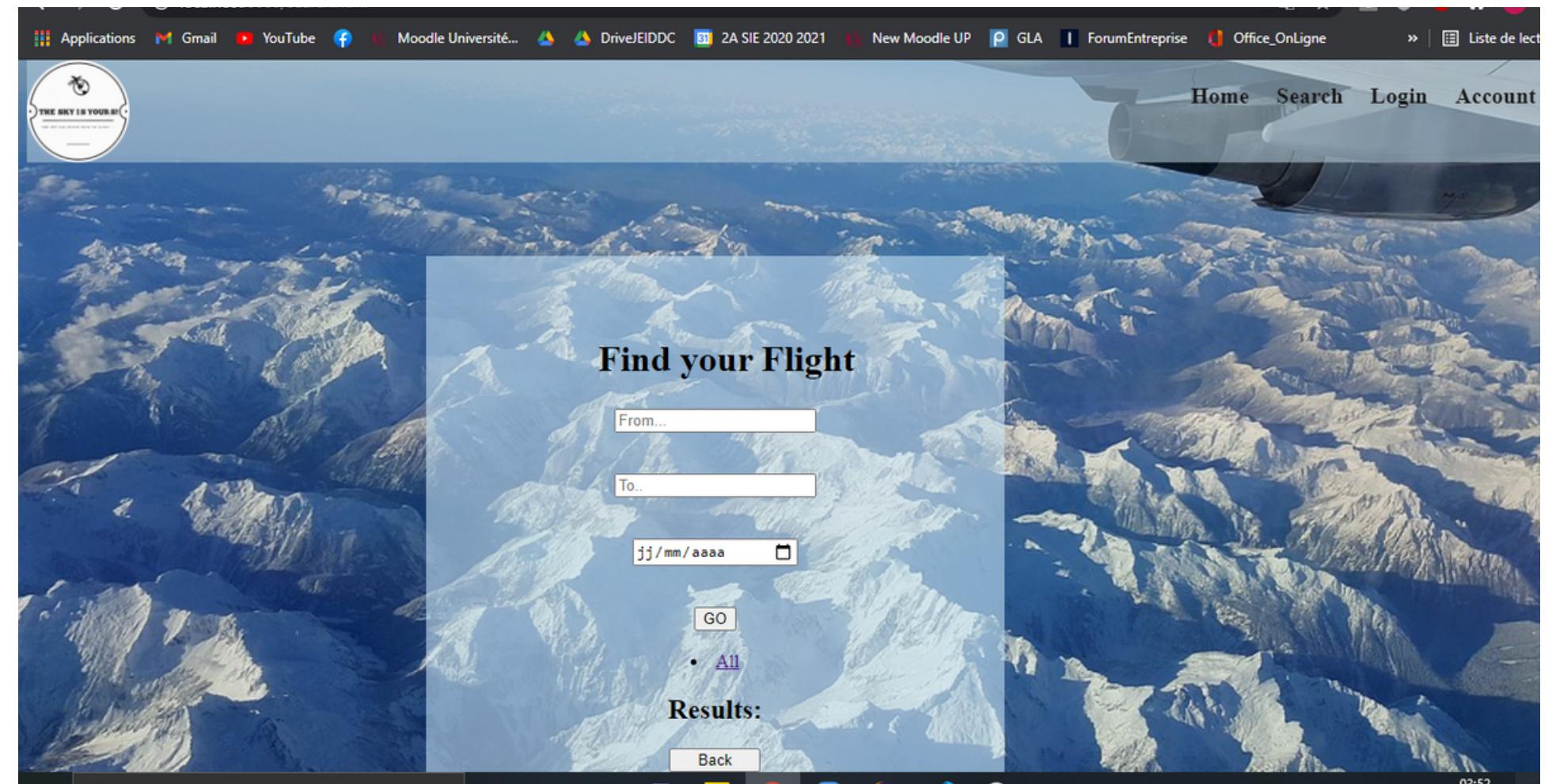
6

Mock_ups



- **Home > Search:**

Here is the search page which includes a navigation bar: Departure - Arival - Flying date. It will show a list of flights once you click on GO. The BACK button is to go back where you were in the website.

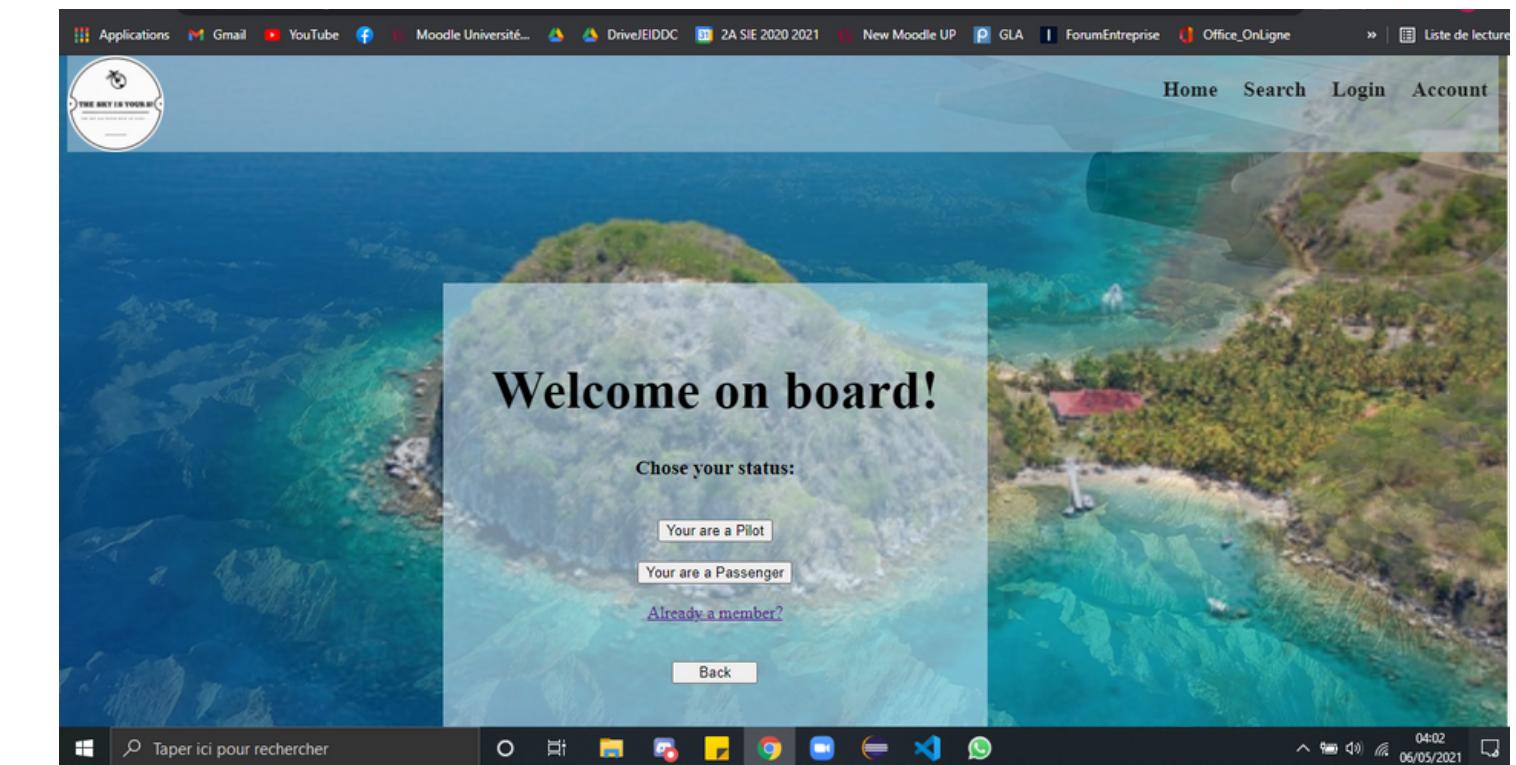
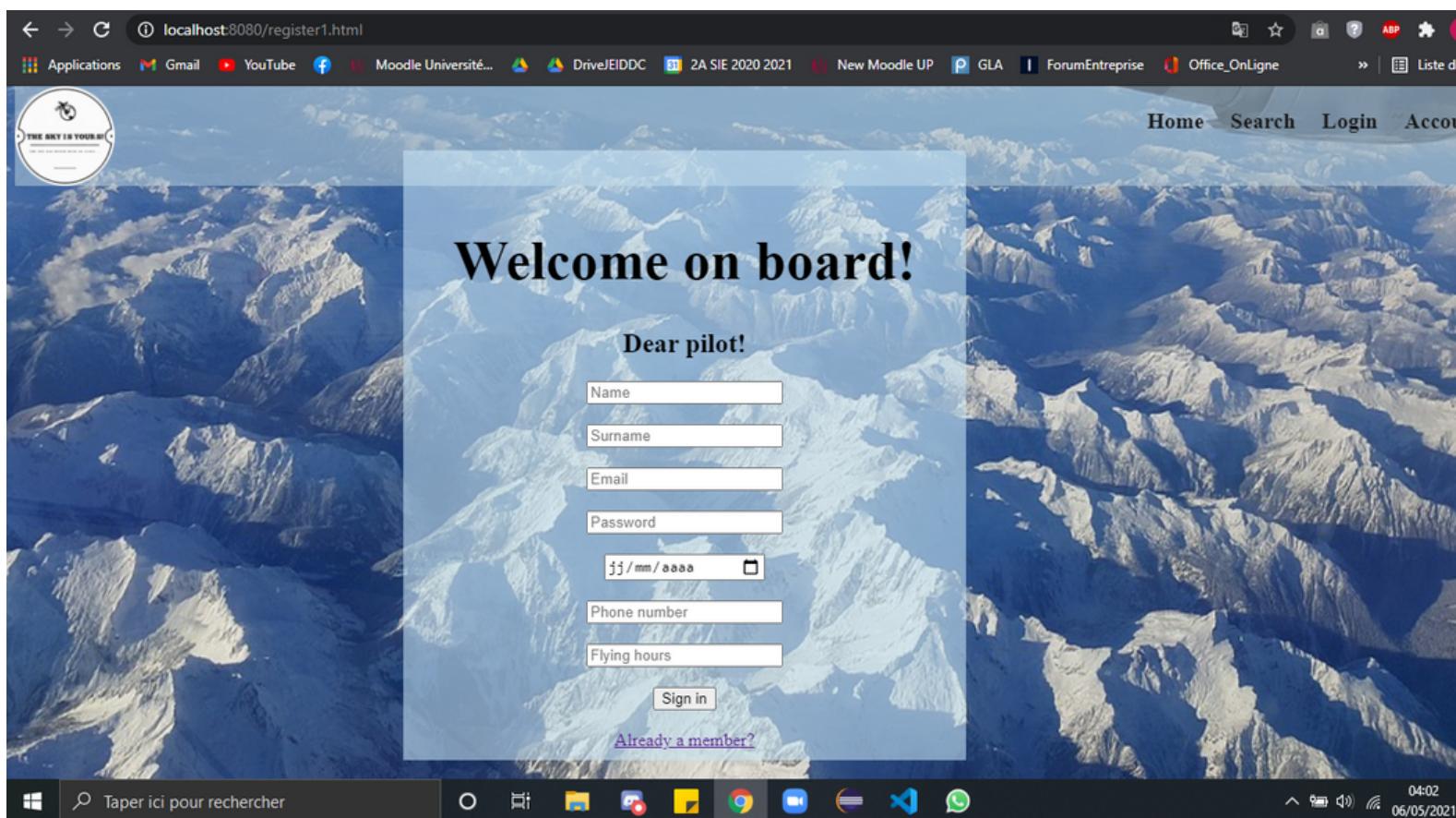


6

Mock_ups

- **Home > Sign-in:**

Here is the sign-in page which includes two choices: either you will sign in as a passenger or as pilot.



- **Home > Sign-in > Pilot:**

If you will chose to sign-in as a pilot you have to put your flying hours.

- **Home > Sign-in > Passenger:**

For the passenger account you won't be obliged to add it.



6

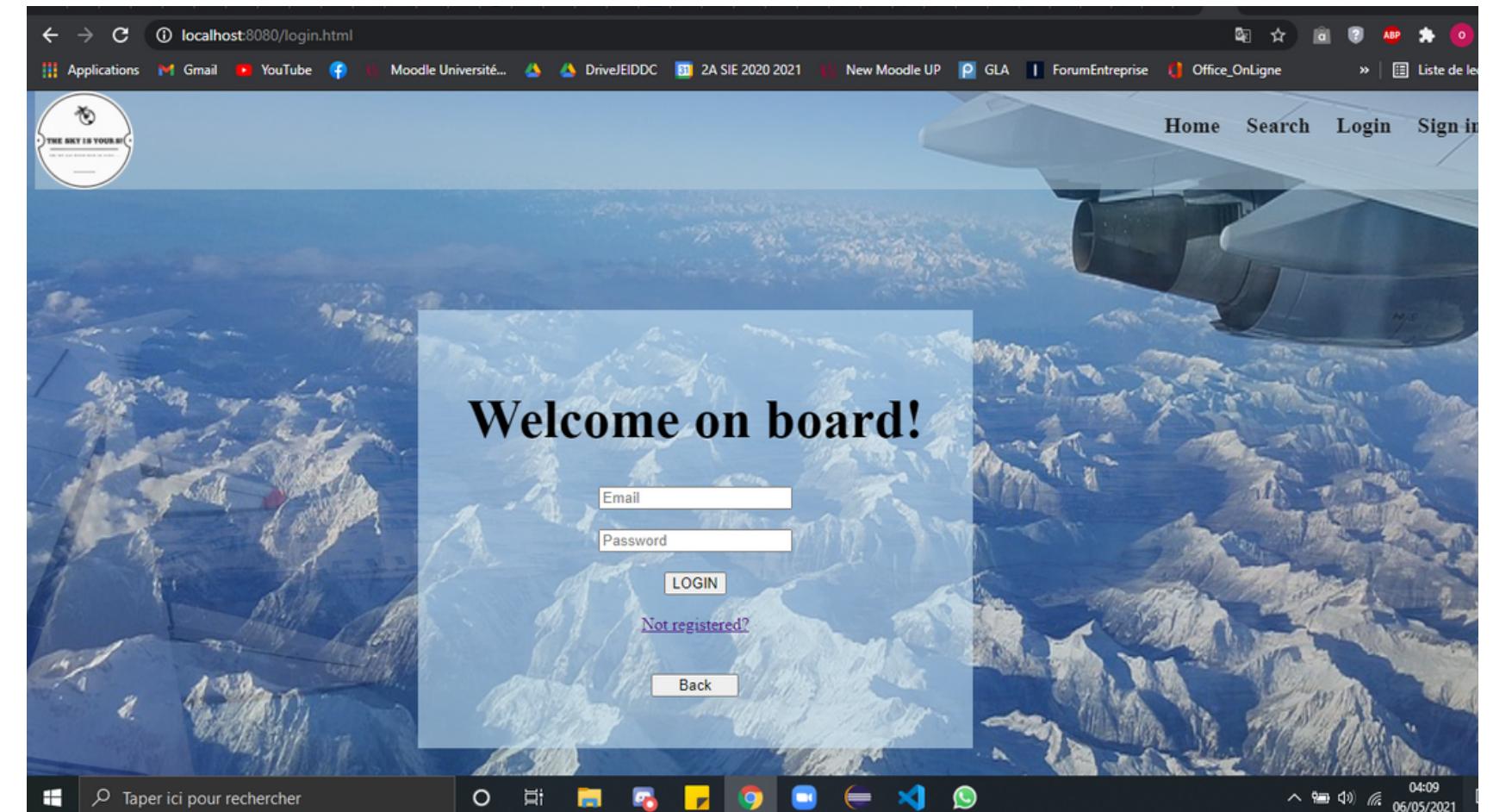
Mock_ups



- **Home > Log-in:**

Here is the log-in page which includes the ID and the Password of each user (Pilot or Passenger).

Once you click on the LOGIN button, you will be lead to your profile.



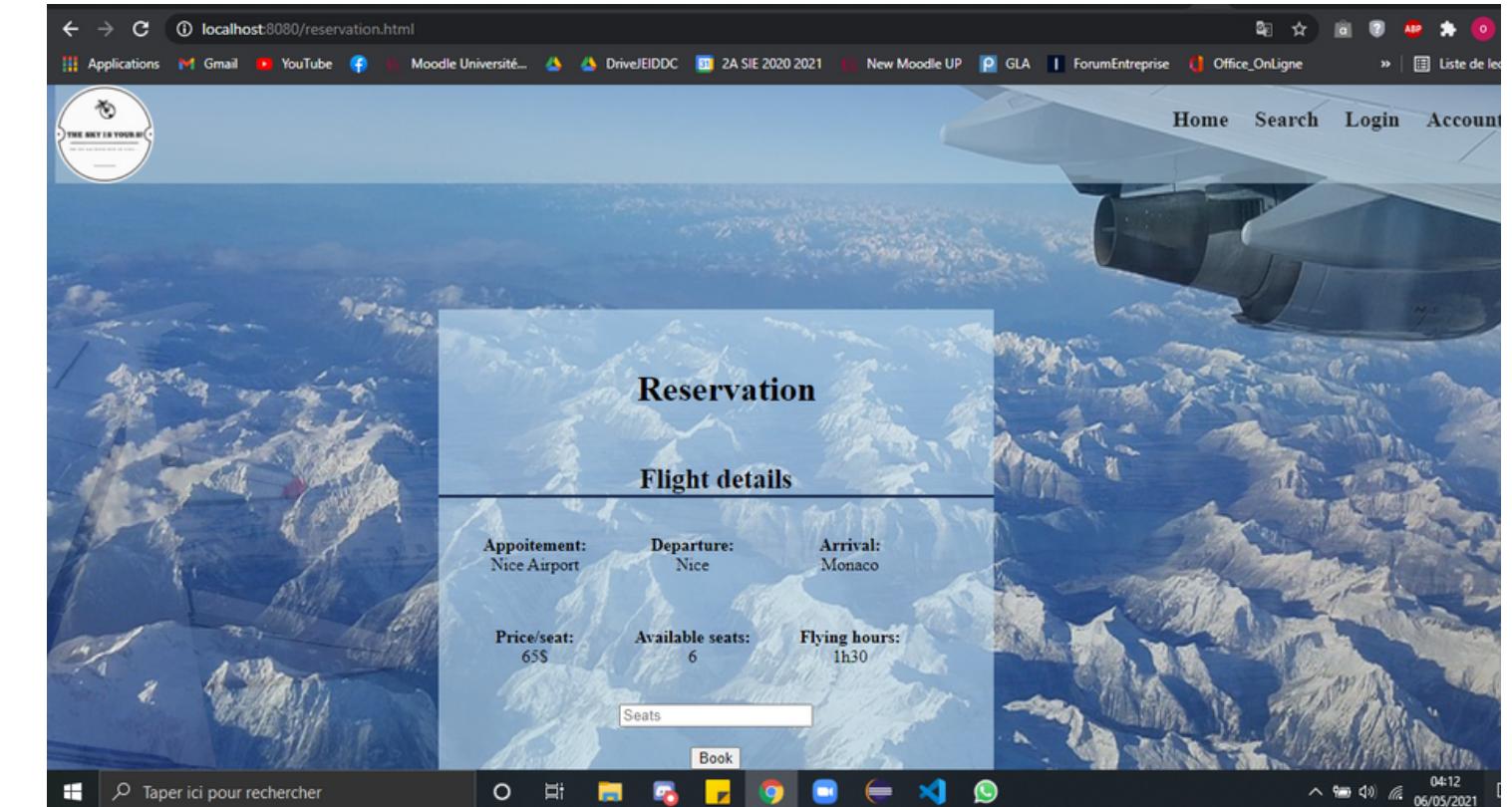
6

Mock_ups



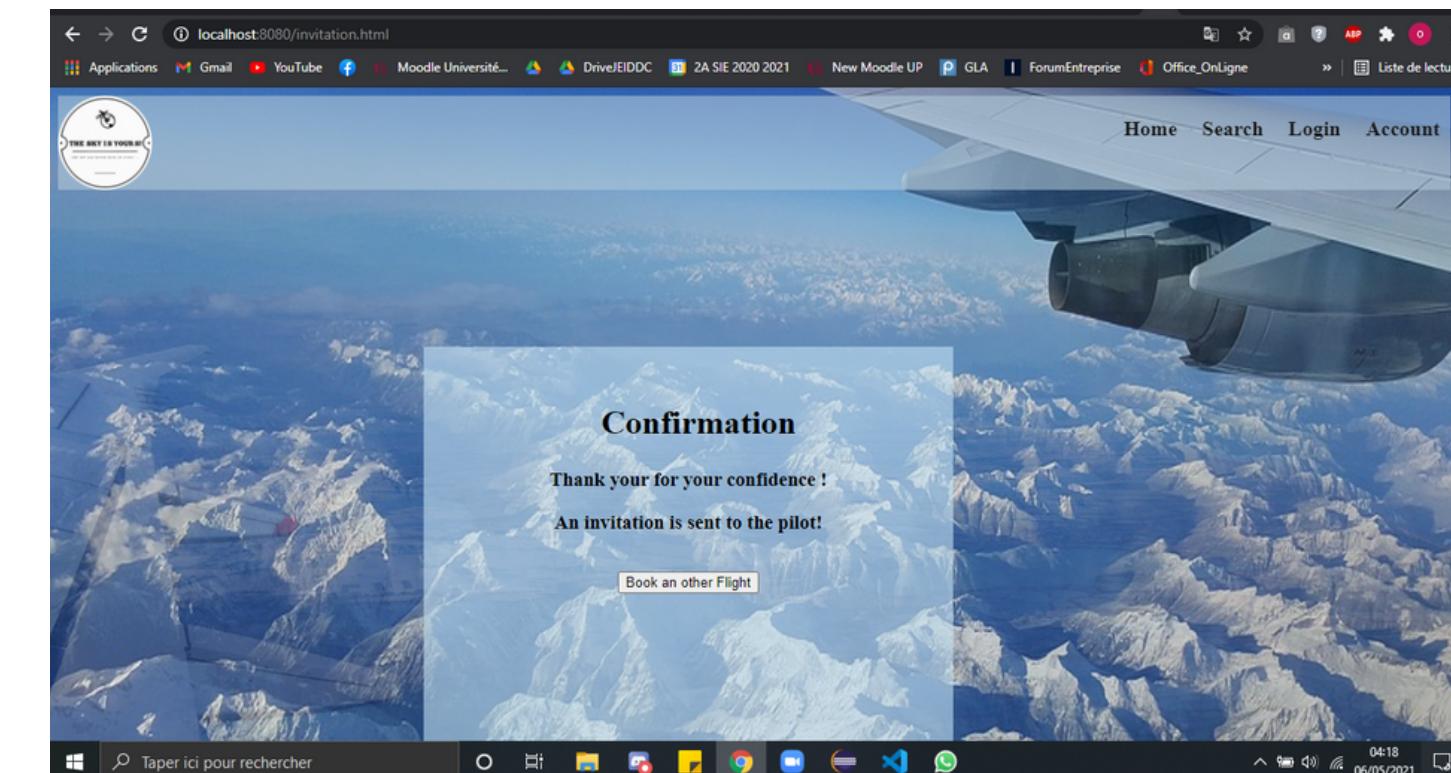
- **Home > Search > Reservation :**

Here is the flight table, including its details. The passenger will be able to choose the number of seats among the available ones and book them. Once he clicks on BOOK, an email will be sent to the pilot to confirm the reservation.



- **Home > Search > Reservation > BOOK :**

When clicking on BOOK, we will be lead to the confirmation page if there is enough available seats. Otherwise, we will be asked to book another flight.





	Features	Difficulty	Workload
1	Organize private flights: It contains all available flights uploaded by pilots and open for passengers to book. You search depending to the place you want to travel to or to your date flight. Also, you can choose the pilot profile you want.	4	3
2	Generate a list of trips available: The user can accede to a list containing all the flights proposed by pilots.	2	4
3	Distinguish two kinds of users: There is a difference in functionalities proposed depending on if the user is a pilot or a passenger.	3	4
4	Set a secure payment system: A secure payment system is what every app needs to be able to work. It's the main income for pilots. People won't put their private identities and their money card on a website which is not secure.	2	3
5	Pilots users can post a flight with his criteria: The pilot choose the day, the hour, the departure airport, number of seats available and the circuit he is following and post it in order to find passengers interested.	5	4
6	Passenger user can book a flight : If a passenger is interested by a flight, he can reserve his seat (in case there are still places free)	2	5



8

Persistance Layer

We have created five classes that contains the attributes which are:

- **Pilot** : with attributes id,password, number of flying hours and other personal information with methods and implements users interface
- **Passenger** : with attributes id,password and other personal information and methods and implements users interface
- **Flight** : which contains attributes concerning flight and aircraft details posted by the pilot: id, departure, arrival ,price...
- **Aircraft**: Which contains details about the airplanes.
- **Booking** : with attributes concerning the reserving platform: passengers on board and flight.

And we implemented for each class an **interface DAO**, which contains methods needed to run the application like searching a flight, adding a pilot or a passenger.. etc.

You can find more information about it in our git repository :

https://github.com/OumayaSatouri/GLA2021_YasmineOumaya/tree/main/datanucleus-master



We have created a WS for each class. They will contribute to send and to manage the link between the data and the implementation.

It includes the paths for the services used in Postman an in the front.

You can find more information about it in our git repository :

https://github.com/OumayaSatouri/GLA2021_YasmineOumaya/tree/main/datanucleus-master



Type	URL	Behavior
GET	/ws/Aircraft/All	
GET	/ws/Flight/All	Retun all the flight available in the website
GET	/ws/Aircraft/getAircraft/{tailnum}	Return the aircraft with id name
GET	ws/Booking/All	Return all the bookings
GET	ws/Booking/{id}	Return booking with id=id
GET	/ws/Flight/Available	Return the flights available
GET	/ws/Flight/Flightbyld/{id}	Return the flight by id
GET	/ws/Flight/FlightsFromTo	Return a list of flight with from a selected place to a selected direction and in a selected date
GET	/ws/Flight/getPassenger/{id}	Return the list of passengers of the flight {id}
GET	/ws/Passenger/getPassenger/{id}	Return the passenger with id=id
GET	/ws/Pilot/getPilot/{id}	Return the pilot with id=id
GET	/ws/Passenger/All	Return the list of all passengers
GET	/ws/Pilot/All	Return the list of all pilots
GET	/ws/Pilot/Request/{pilotid}	Return list of booking that were not valiate by the pilot
GET	/ws/Pilot/PlannedFlights/{pilotid}	Return list of plannedflights of the pilot with id=pilotid
GET	/ws/Passenger/login	Return passenger with email and password entered
GET	/ws/Passenger/getBookedFlights	Return the flights that has been booked by the passenger
PUT	/ws/Passenger/addPassenger	Add passenger to the database
PUT	/ws/Aircraft/addAircraft	Add an aircraft to the database
PUT	/ws/Booking/addBooking	Add a booking to the database
PUT	/ws/Flight/addFlight	Add a flight to the database
PUT	/ws/Pilot/addPilot	Add a pilot to the database
POST	/ws/Booking/Pilot/validate/{id}?pilotid	validate the booking id by the pilot with pilotid
POST	/ws/Flight/bookFlight	Book a flight
POST	/ws/Passengers	Modify the list of the passengers by replacing.
POST	/ws/Post-modifyaflight	Modify details about a flight.
DELETE	/ws/Aircraft/delete/{tailnum}	Delete an aircraft with the tailnumber
DELETE	/ws/Booking/delete/{id}	Delete the booking with id=id
DELETE	/ws/Booking/cancel/{id}	Cancel a booking with id=id
DELETE	/ws/Flight/delete/{id}	Delete a flight with id=id
DELETE	/ws/Passenger/delete/{id}	Delete a passenger from database
DELETE	/ws/Pilot/delete/{id}	Delete a pilot from database

10

Organization



It takes two hands to clap

Presentation & Github

- Repository
- Github organisation
- Logo
- Presentation
- Mock-ups
- Diagrams

BACK

- Persistance Layers
- DAO
- DAO implementations
- DataBase
- Tests
- Datagenerator

FRONT

- Web services
- Interface (html & css)
- Link (JS)



Team members



Oumaya

- Github organisation
- WS
- Tests

Shared

- Presentation
- Mock-ups
- Diagrams
- Persistance Layers
- Front and interfaces



Yasmine

- Logo
- DAO
- DAO implementation



Don't forget to download our app
available in Android and iOS!

You can find more on our Website:
www.theskyisyours.fr