M1 Informatique
2015 - 2016

Flight Planning System

Project team 6

Project Leader:
Aslam Senouci-Bereksi

Ahmed Guettou, Amine Belkhechine, Helmi Zegaya, Jordan Capaldi, Karthik Sampath, Lyes Hadjed, Mamoudou Tallibe Diallo, Mehdi Necibi, Raphael Trancoso, Timera Ibrahima

System Presentation

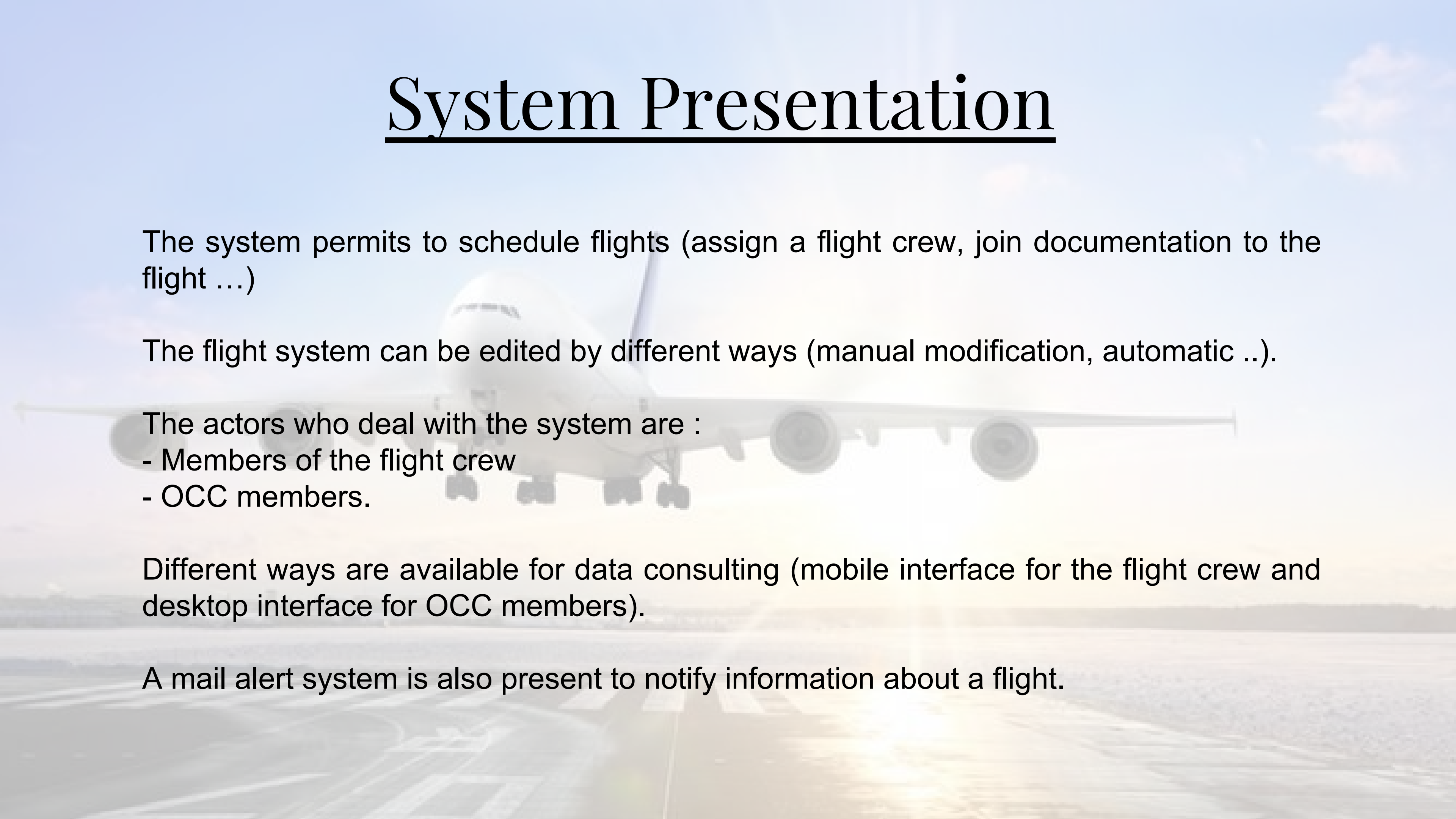
The system permits to schedule flights (assign a flight crew, join documentation to the flight …)

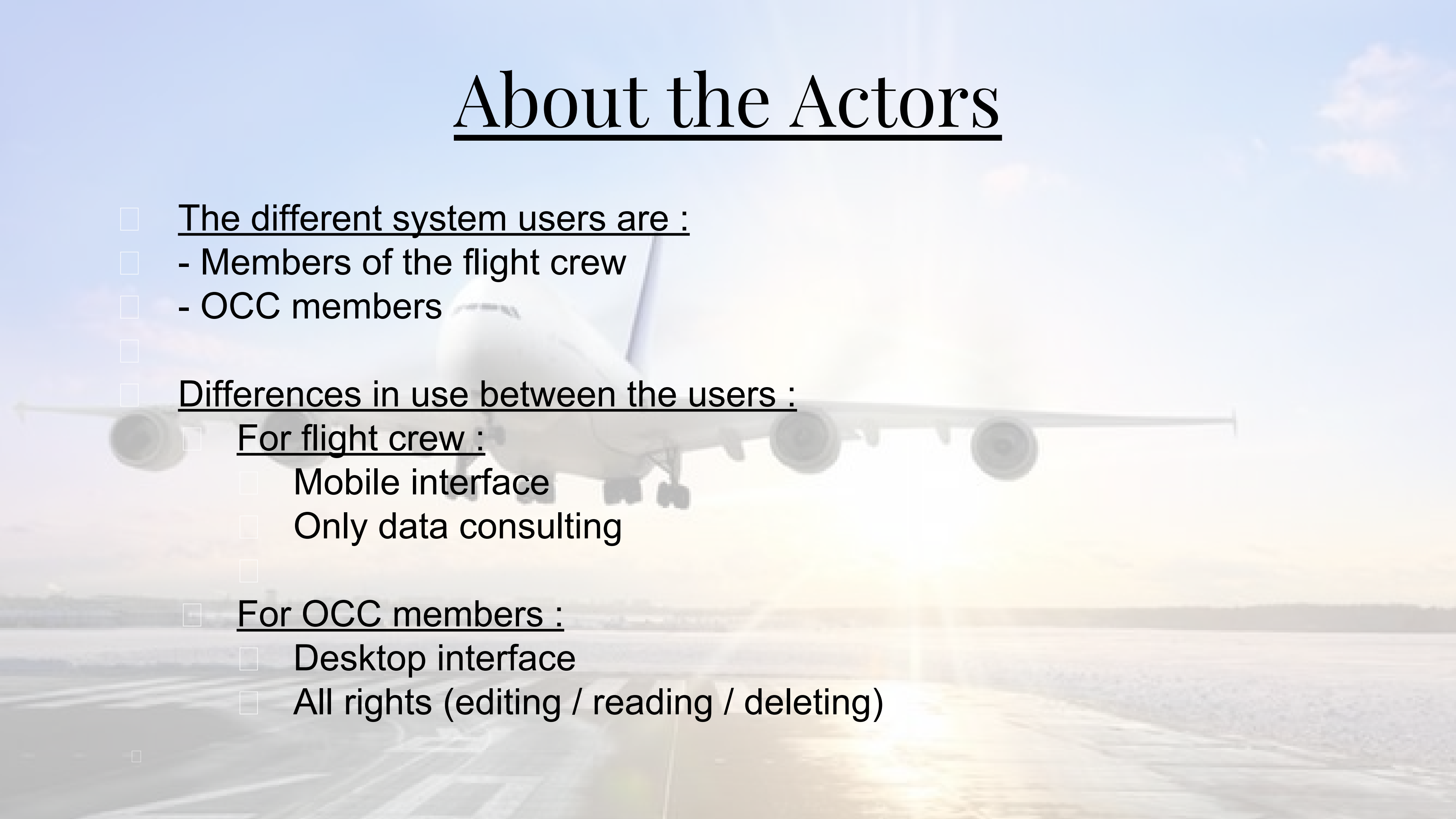
The flight system can be edited by different ways (manual modification, automatic ..).

The actors who deal with the system are :
- Members of the flight crew
- OCC members.

Different ways are available for data consulting (mobile interface for the flight crew and desktop interface for OCC members).

A mail alert system is also present to notify information about a flight.





About the Actors

- The different system users are :
- - Members of the flight crew
- - OCC members

- Differences in use between the users :
  - For flight crew :
    - Mobile interface
    - Only data consulting

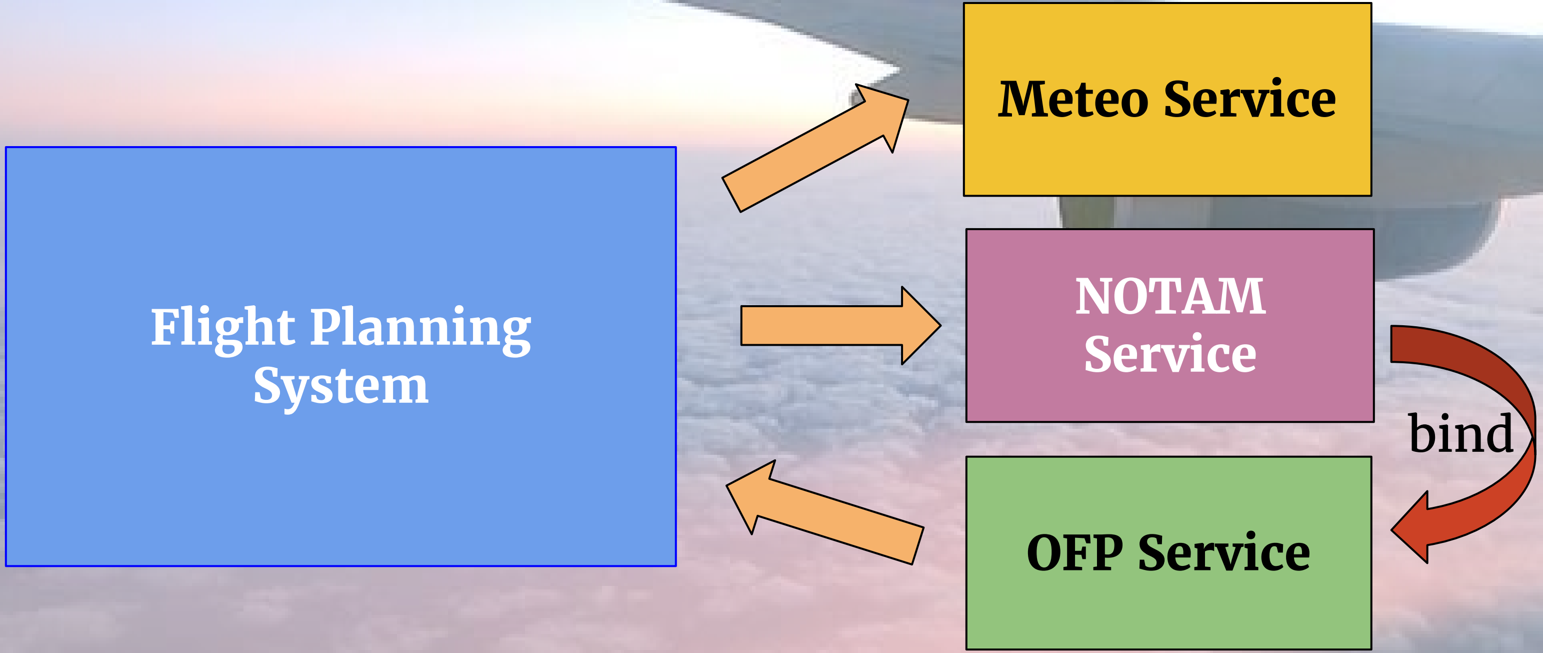
  - For OCC members :
    - Desktop interface
    - All rights (editing / reading / deleting)

Business Objects

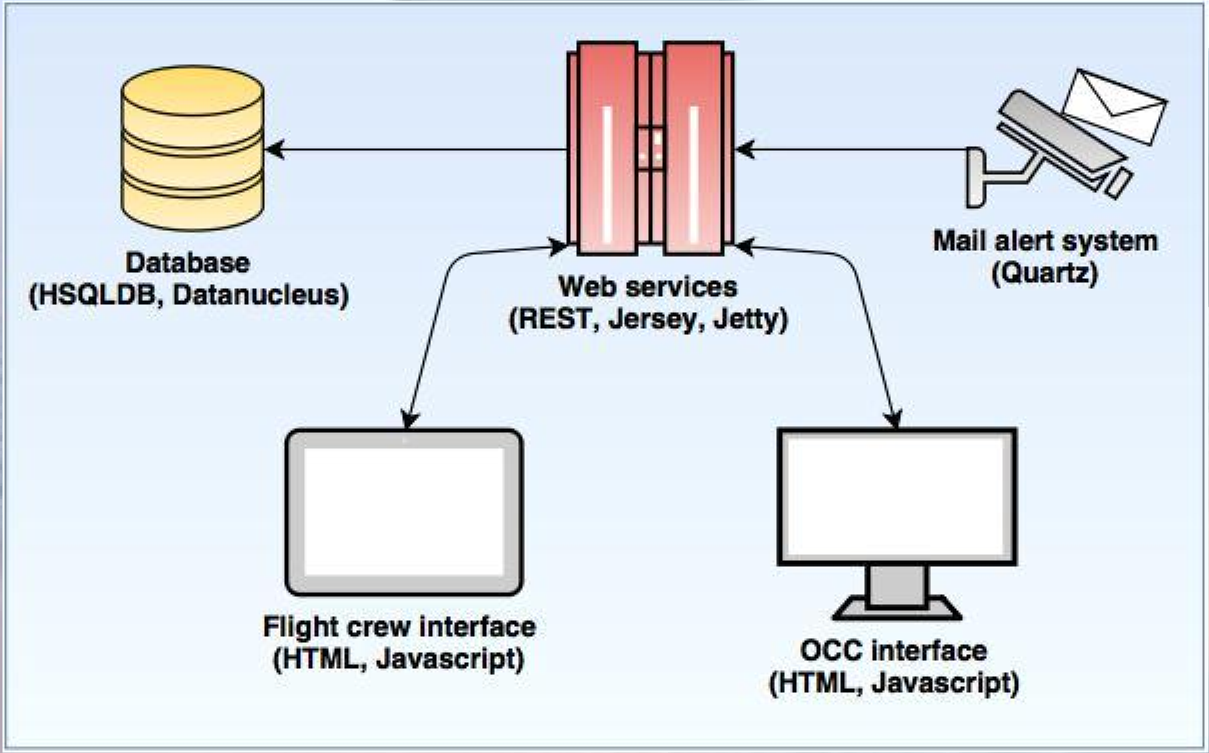
Presented with the following form : ***Object (attributes)***

- **Crew** (Pilot, Co-pilot, Hostess/Steward)
- **Airplane** (Id, Type, Weight, Size, Capacity)
- **Airport** (OACI, Name)
- **Person** (Login, Type, Name, Surname, Password)
- **Flight** (Commercial number, ATC number, Arrival date, NOTAM Departure date, Crew crew,Arrival OACI, Departure OACI, Meteo, OFP)

Global System Diagram

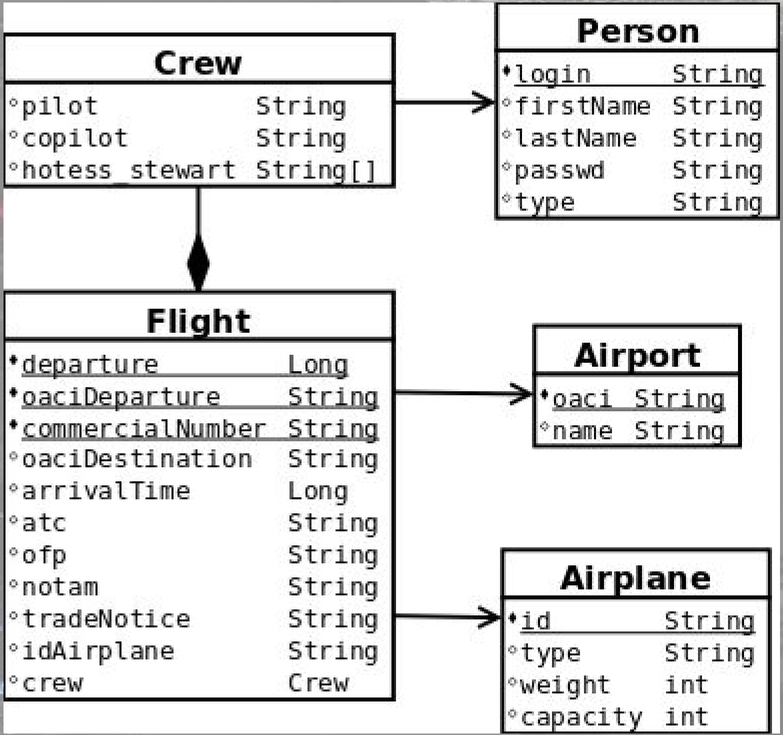


Subsystems

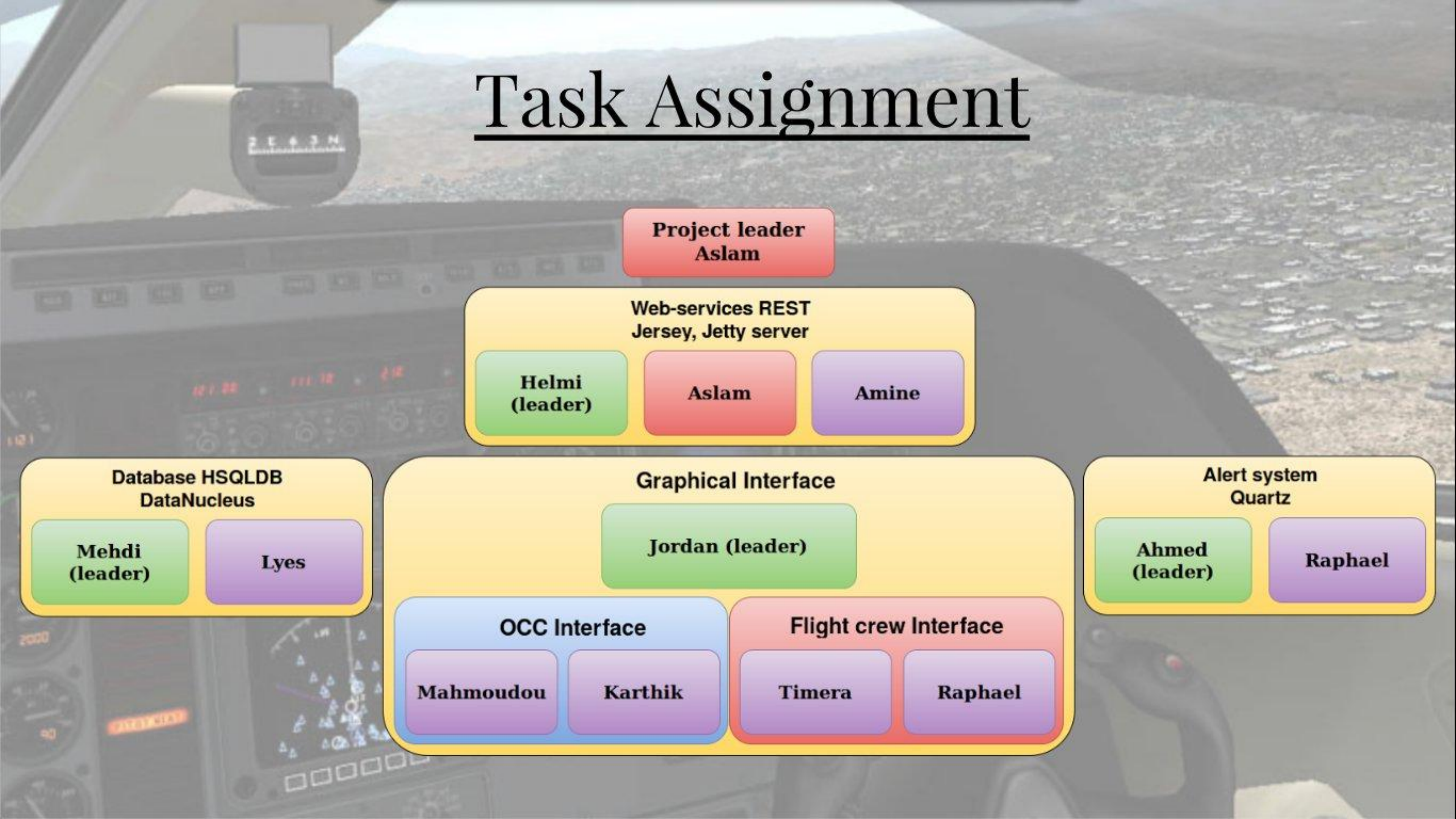


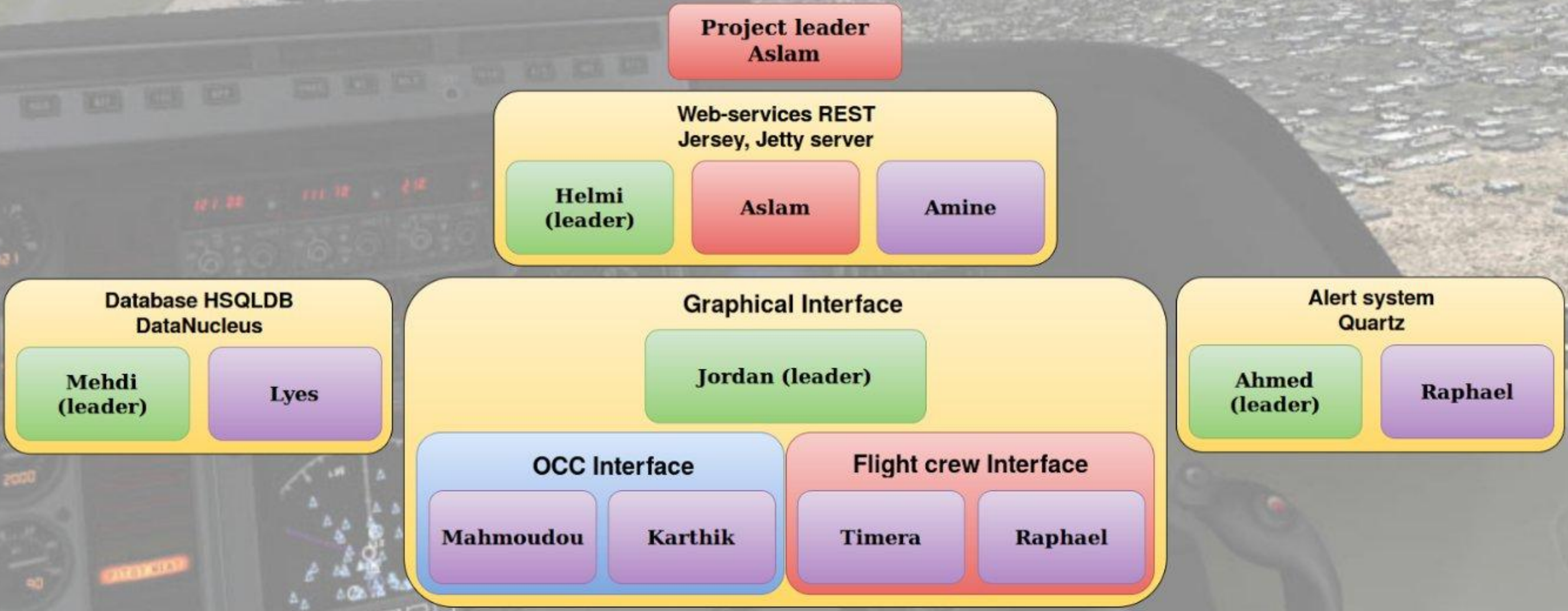
Database
(HSQLDB, Datanucleus)
Web services
(REST, Jersey, Jetty)
Mail alert system
(Quartz)
Flight crew interface
(HTML, Javascript)
OCC interface
(HTML, Javascript)

Technical Propositions

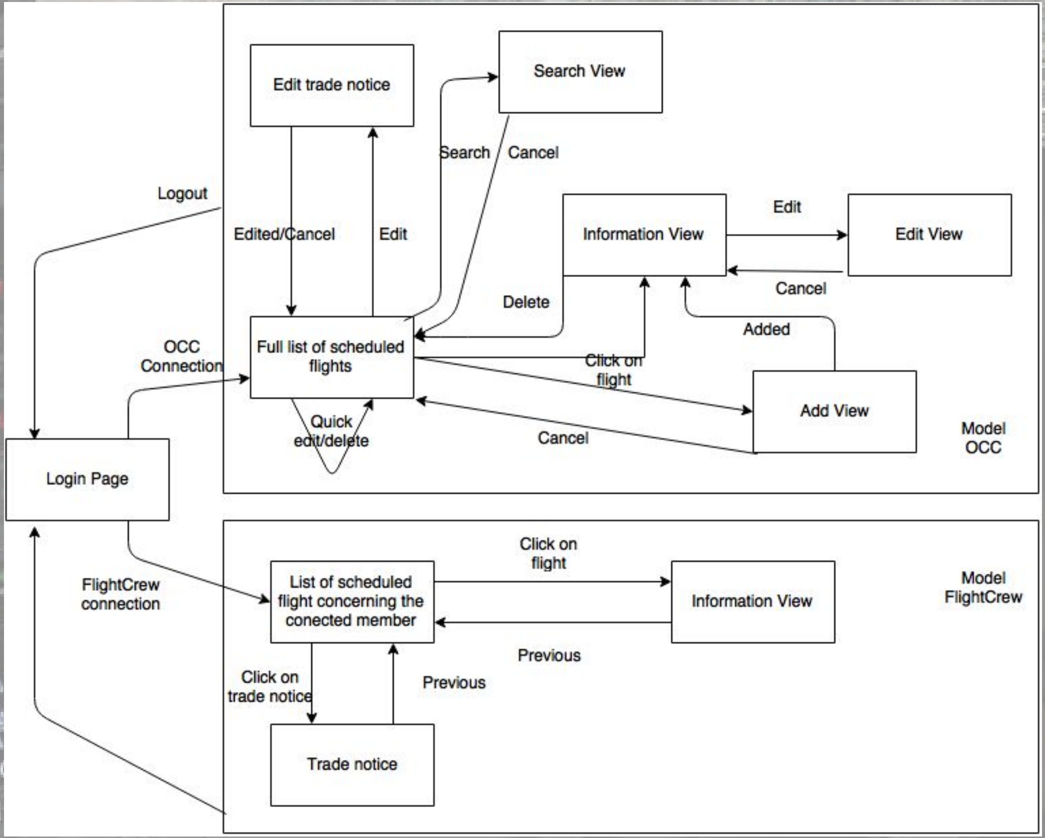


Task Assignment

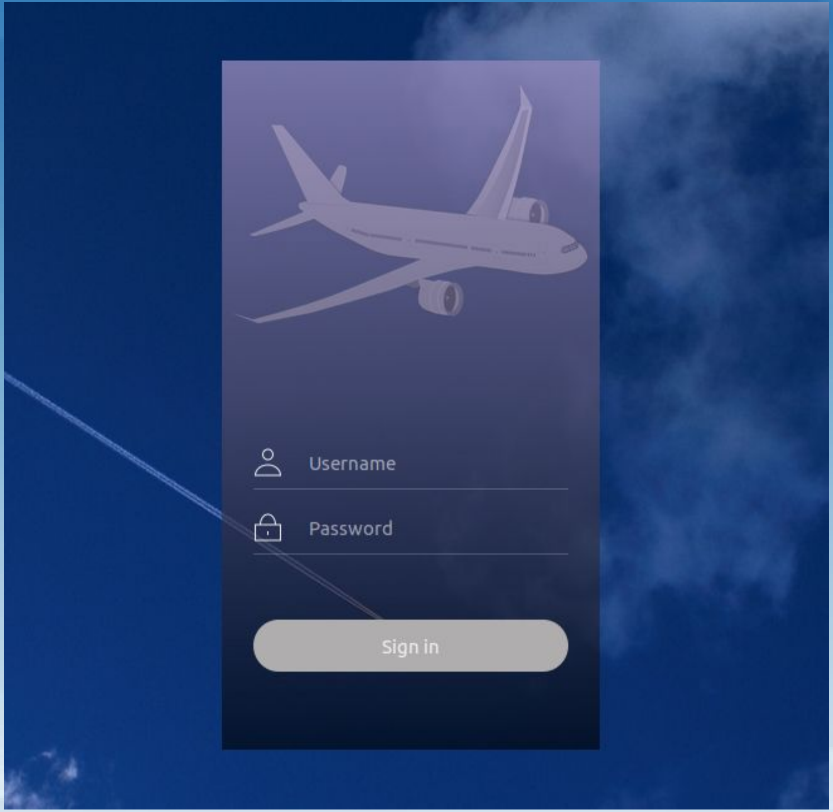


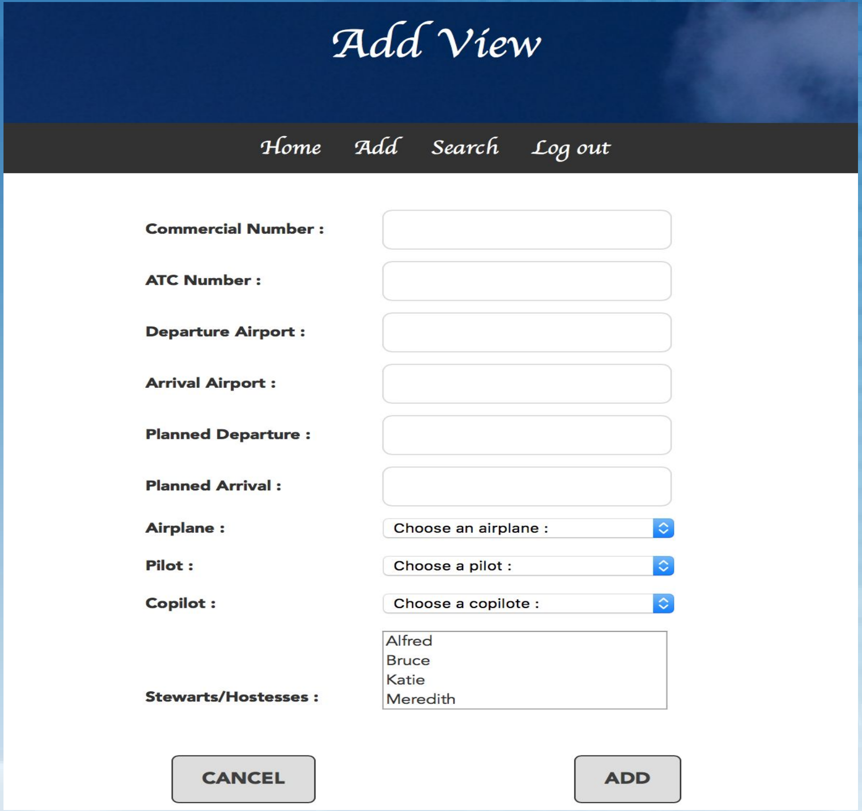


Navigation Diagram



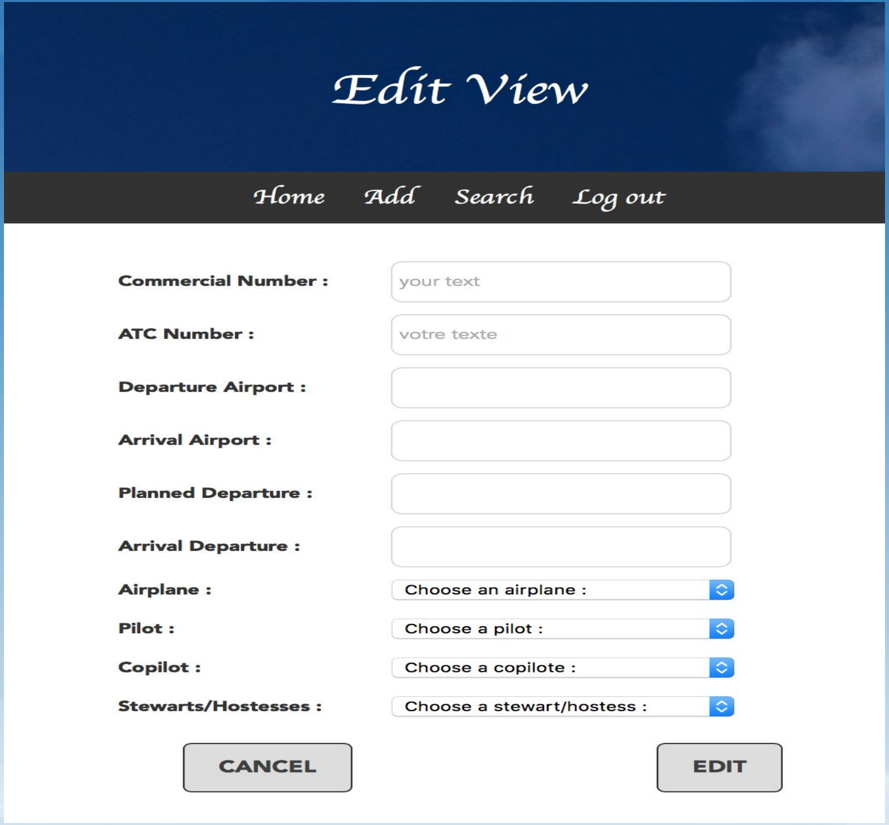
Mock-ups





Commercial Number :	
ATC Number :	
Departure Airport :	
Arrival Airport :	
Planned Departure :	
Planned Arrival :	
Airplane :	Choose an airplane :
Pilot :	Choose a pilot :
Copilot :	Choose a copilote :
Stewarts/Hostesses :	Alfred Bruce Katie Meredith

Mock-ups

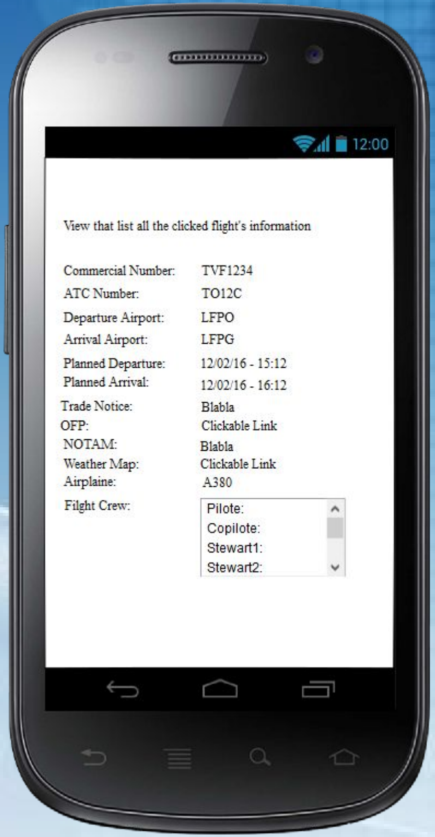


Edit View

Home Add Search Log out

Commercial Number :	your text
ATC Number :	votre texte
Departure Airport :	
Arrival Airport :	
Planned Departure :	
Arrival Departure :	
Airplane :	Choose an airplane :
Pilot :	Choose a pilot :
Copilot :	Choose a copilote :
Stewarts/Hostesses :	Choose a stewart/hostess :

CANCEL EDIT

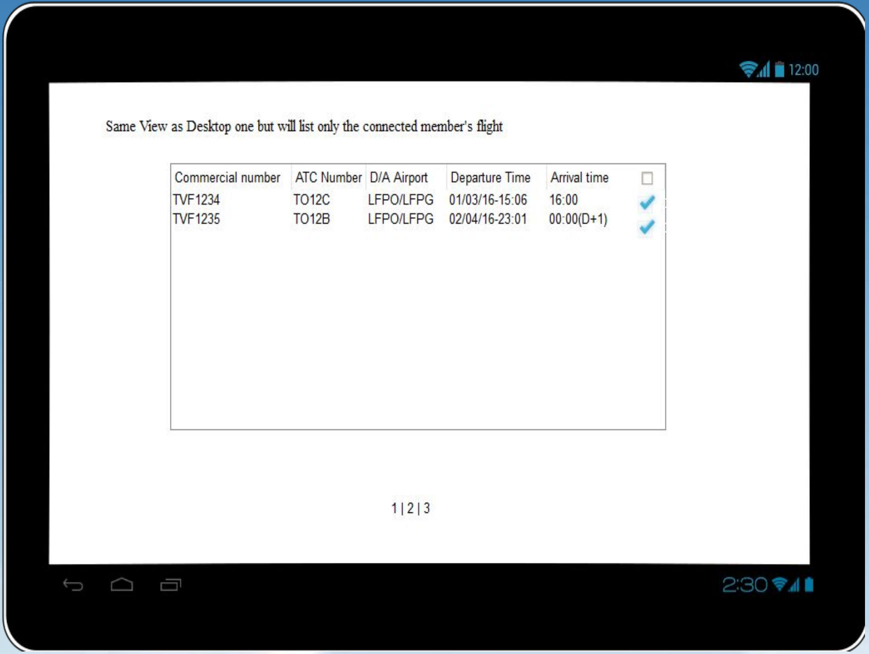


12:00

View that list all the clicked flight's information

Commercial Number:	TVF1234
ATC Number:	TO12C
Departure Airport:	LFPO
Arrival Airport:	LFPG
Planned Departure:	12/02/16 - 15:12
Planned Arrival:	12/02/16 - 16:12
Trade Notice:	Blabla
OFP:	Clickable Link
NOTAM:	Blabla
Weather Map:	Clickable Link
Airplaine:	A380
Filght Crew:	Pilote: Copilote: Stewart1: Stewart2:

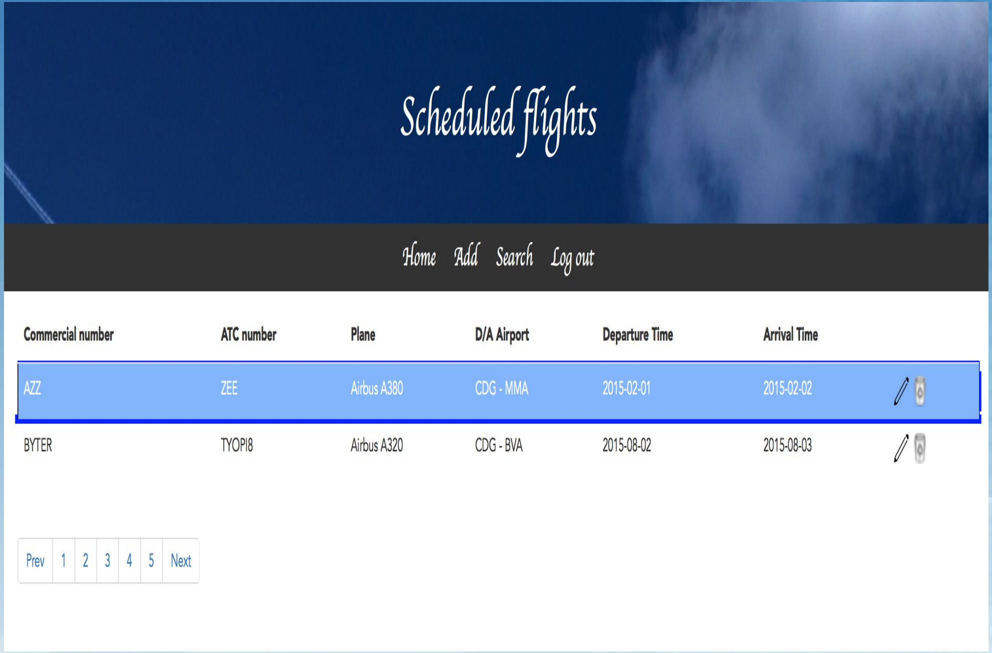
Mock-ups



Same View as Desktop one but will list only the connected member's flight

Commercial number	ATC Number	D/A Airport	Departure Time	Arrival time	☐
TVF1234	TO12C	LFPO/LFPG	01/03/16-15:06	16:00	✓
TVF1235	TO12B	LFPO/LFPG	02/04/16-23:01	00:00(D+1)	✓

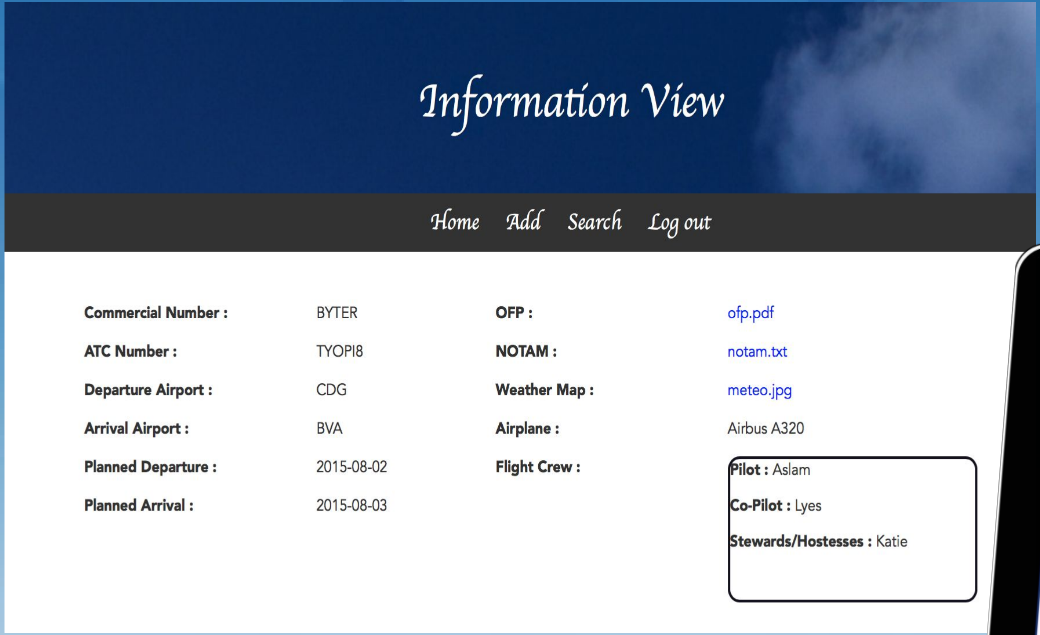
1 | 2 | 3



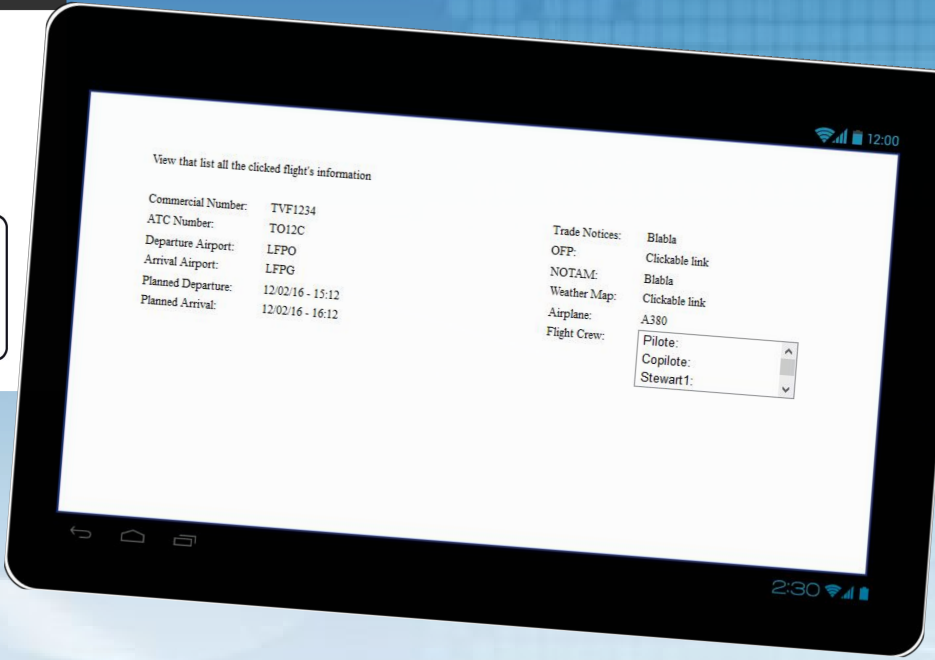
Scheduled flights

Commercial number	ATC number	Plane	D/A Airport	Departure Time	Arrival Time
AZZ	ZEE	Airbus A380	CDG - MMA	2015-02-01	2015-02-02
BYTER	TYOPI8	Airbus A320	CDG - BVA	2015-08-02	2015-08-03

Mock-ups



Commercial Number :	BYTER	**OFP :**	ofp.pdf
ATC Number :	TYOPI8	**NOTAM :**	notam.txt
Departure Airport :	CDG	**Weather Map :**	meteo.jpg
Arrival Airport :	BVA	**Airplane :**	Airbus A320
Planned Departure :	2015-08-02	**Flight Crew :**	**Pilot :** Aslam **Co-Pilot :** Lyes **Stewards/Hostesses :** Katie
Planned Arrival :	2015-08-03		



View that list all the clicked flight's information

Commercial Number:	TVF1234	Trade Notices:	Blabla
ATC Number:	TO12C	OFP:	Clickable link
Departure Airport:	LFPO	NOTAM:	Blabla
Arrival Airport:	LFPG	Weather Map:	Clickable link
Planned Departure:	12/02/16 - 15:12	Airplane:	A380
Planned Arrival:	12/02/16 - 16:12	Flight Crew:	Pilote: Copilote: Stewart1:

JavaScript

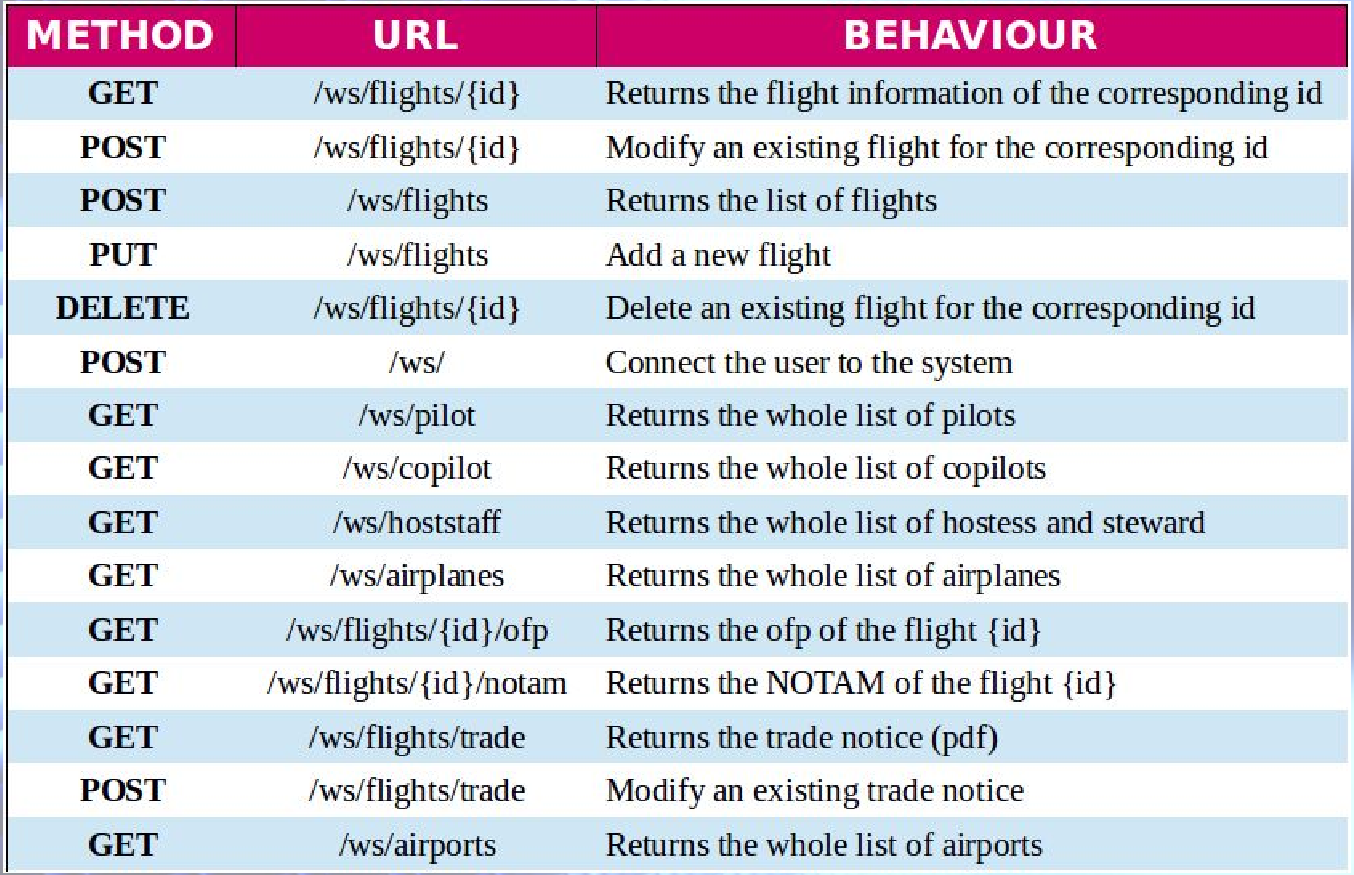
```
function callGetDone(result){
   var templateExample = _.template($('#template').html());
   var flightList = $.parseJSON(JSON.stringify(result));
   sessionStorage.myValue = JSON.stringify(flightList[0].atc);
   for(var i = 0; i<flightList.length; i++){
      getAirplaneType(flightList[i].idAirplane);
      var html = templateExample({
            "commercialNumber": flightList[i].commercialNumber,
            "atcNumber":  flightList[i].atc,
            "airplane":  airplaneType,
            "oaciDeparture":  flightList[i].oaciDeparture,
            "oaciDestination":  flightList[i].oaciDestination,
            "departure":  flightList[i].departure,
            "arrivalTime":  flightList[i].arrivalTime
      });
      $("#newLine").append(html);
   }
}
```

```
<script id="template" type="text/template">
 <tr class="clickableRow" data-href="url://info.html">
  <td>
    <%= commercialNumber %>
  </td>
  <td>
    <%= atcNumber %>
  </td>
  <td>
    <%= airplane %>
  </td>
  <td>
    <%= oaciDeparture %> - <%= oaciDestination %>
  </td>
  <td>
    <%= departure %>
  </td>
  <td>
    <%= arrivalTime %>
  </td>
 </tr>
</script>
```

JavaScript

```
$.ajax({
        contentType: "application/json",
        url: url,
        data: JSON.stringify({
        "departure": departure, "oaciDeparture": oaciDep,
        "commercialNumber":commNum, "oaciDestination": oaciArr,
        "arrivalTime": arrival, "atc": atc, "ofp": "ofp.pdf",
        "notam": "notam.txt", "meteo": "meteo.jpg", "tradeNotice": tradeNot,
        "crew":
        { "loginPilot": pilot,"loginCopilot": copilot,
                "loginHostStaff":  staffVal },"idAirplane": plane }),
        type: "PUT",
        processData: false

}).done(function(){
        alert( "The flight has been added." );
        // Cleaning of input fields here

}).fail(function(){
        alert("An error has occured. Please check informations.");
});
```

Web-services



METHOD	URL	BEHAVIOUR
GET	/ws/flights/{id}	Returns the flight information of the corresponding id
POST	/ws/flights/{id}	Modify an existing flight for the corresponding id
POST	/ws/flights	Returns the list of flights
PUT	/ws/flights	Add a new flight
DELETE	/ws/flights/{id}	Delete an existing flight for the corresponding id
POST	/ws/	Connect the user to the system
GET	/ws/pilot	Returns the whole list of pilots
GET	/ws/copilot	Returns the whole list of copilots
GET	/ws/hoststaff	Returns the whole list of hostess and steward
GET	/ws/airplanes	Returns the whole list of airplanes
GET	/ws/flights/{id}/ofp	Returns the ofp of the flight {id}
GET	/ws/flights/{id}/notam	Returns the NOTAM of the flight {id}
GET	/ws/flights/trade	Returns the trade notice (pdf)
POST	/ws/flights/trade	Modify an existing trade notice
GET	/ws/airports	Returns the whole list of airports

Password encryption

- Client: "I want to login".
- Server: Generates a random number ***#S*** and sends it to the Client.
- Client
    - Reads username and password typed by the user.
    - Calculates the hash of the password, getting ***h(pw)*** (which is what is stored in the DB).
    - Generates another random number ***#C***.
    - Concatenates ***h(pw) + #S + #C*** and calculates its hash, call it ***h(all)***.
    - Sends to the server username, ***#C*** and ***h(all)***.
- Server
    - Retrieves **h(pw)'** for the specified username, from the DB.
    - Now it has all the elements to calculate ***h(all')***, like Client did.
    - If ***h(all) = h(all')*** then ***h(pw) = h(pw)'***, almost certainly.

Jersey Implementation

```
@POST
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Response authenticate(Person person) throws
WebApplicationException{

    HttpSession session = req.getSession(true);
    // Set timeout value in second
    session.setMaxInactiveInterval(20*60);
    Person retrieved = pdi.checkUser(person);
    if (retrieved == null)
        throw new WebApplicationException(...);
    session.setAttribute("username", retrieved.getLogin());
    session.setAttribute("ptype", retrieved.getPtype());
    return Response.status(Response.Status.ACCEPTED).build();

}
```

```
@POST
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
@Path("/flights")
List<Flight> getFlights(@QueryParam("page") int page){
        List<Flight> flights = null;

        if (ResourceManager.isConnectedUser(req))
                flights = isCcoMember()
                ? fdi.getFlights(page)
                : fdi.getCrewFlights(page, getUsername());
        return flights;
}
```

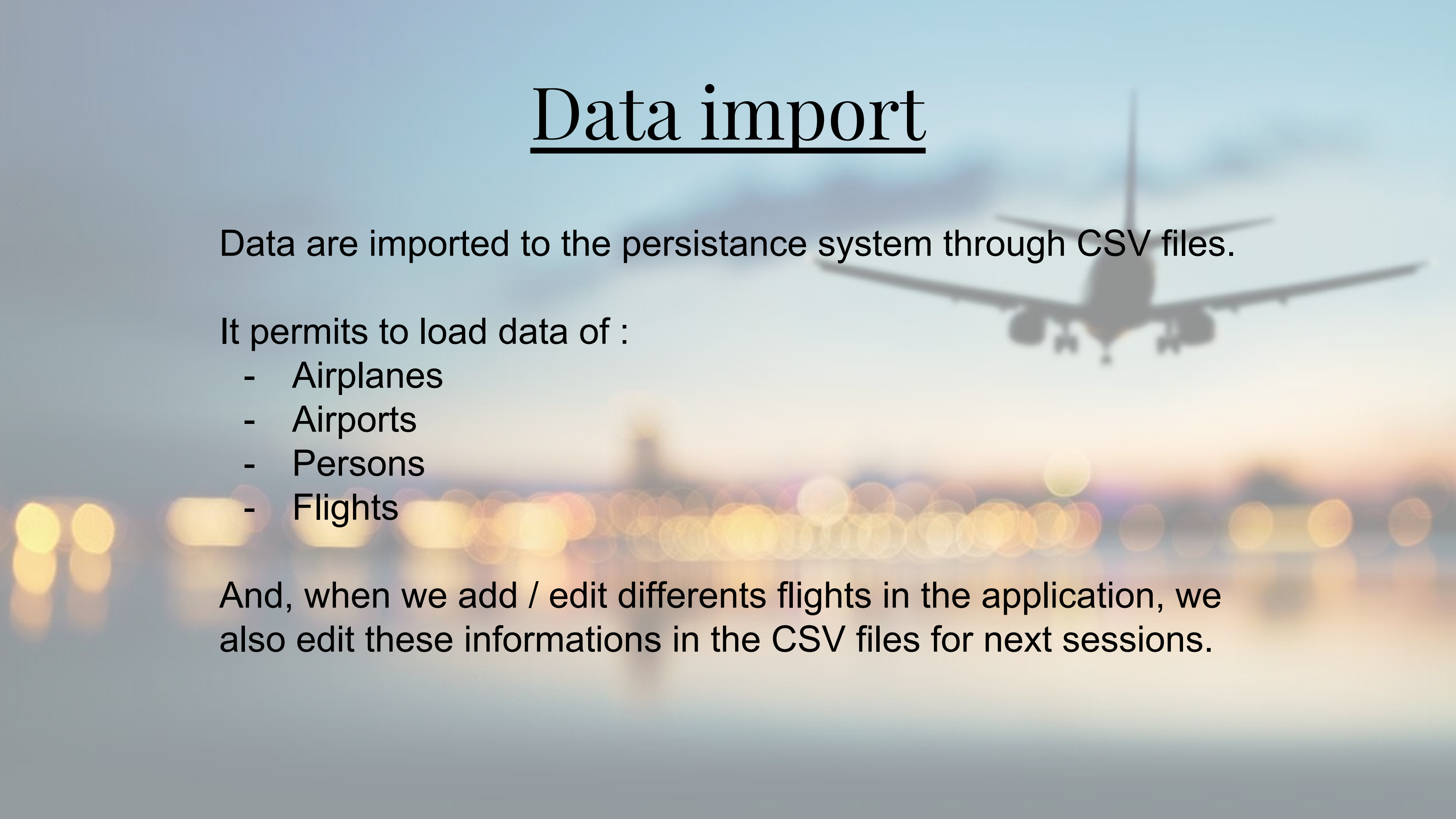
Java interfaces

```
public interface PersonDao {
    /**
    * @return the list of pilots
    */
   List<Person> getPilots();

   /**
    * @return the list of co-pilots
    */
   List<Person> getCoPilots();

   /**
    * @return the list of staff
    */
   List<Person> getHostStaff();

   /**
   * @param the user who wants to log in
   */
   Response authenticate(Person person);
}
```

```
public interface AirplaneDao {

    /**
     * @return the list of all available airplanes
     */
    List<Airplane> getAirplanes();
}
```

Java interfaces

```java
public interface FlightDao {

    /**
     * @return this list of Flight
     */
    List<Flight> getFlights();

    /**
     * @param departure
     * @return the list of Flights assigned to
          a specific departure date.
     */
    Flight getFlight(String id);

    /**
     * @return a flight crew
     * @param departure
     */
    Crew getCrew(String id);
    ...

    /**
     * @param departure
     * @return The URI of the OFP
     */
    String getOFP(String id);

    /**
     * @param departure
     * @return The URI of the NOTAM
     */
    String getNOTAM(String id);

    /**
     * @return The URI of the trade notice
     */
    String getTradeNotice();
    ...
}
```



Data import

Data are imported to the persistance system through CSV files.

It permits to load data of :

- Airplanes
- Airports
- Persons
- Flights

And, when we add / edit differents flights in the application, we also edit these informations in the CSV files for next sessions.

Alert System

Send mail to the CCO members:

- If airplane or crew is missing 7 days before flight departure
- If OFP is missing 12 hours before flight departure