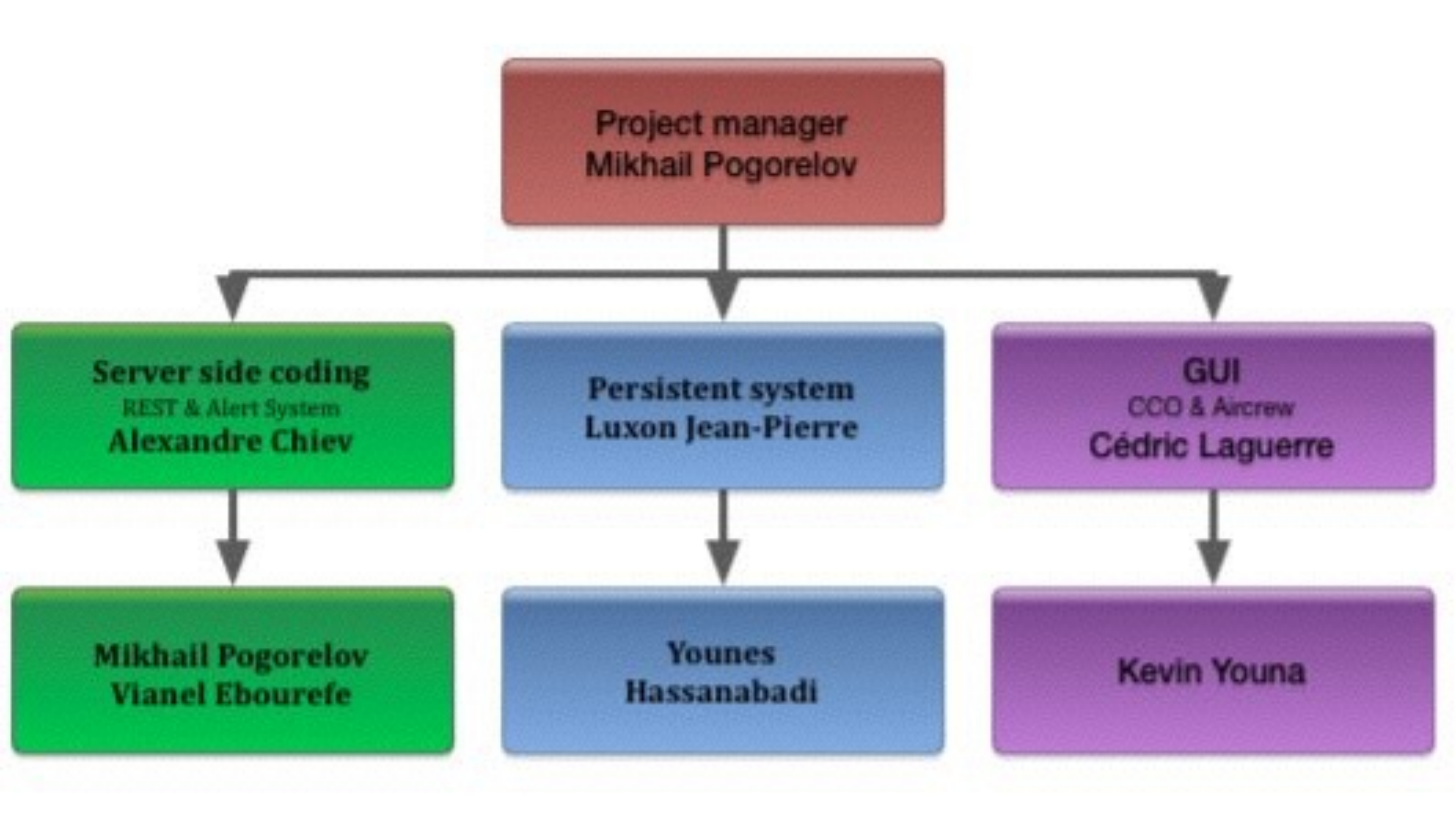
Flight Planning System

By L.M.A.C.K.V.Y

Mikhail Pogorelov

Head Chief of L.M.A.C.K.V.Y

The Team



Project manager
Mikhail Pogorelov

Server side coding
REST & Alert System
Alexandre Chiev

Persistent system
Luxon Jean-Pierre

GUI
CCO & Aircrew
Cédric Laguerre

Mikhail Pogorelov
Vianel Ebourefe

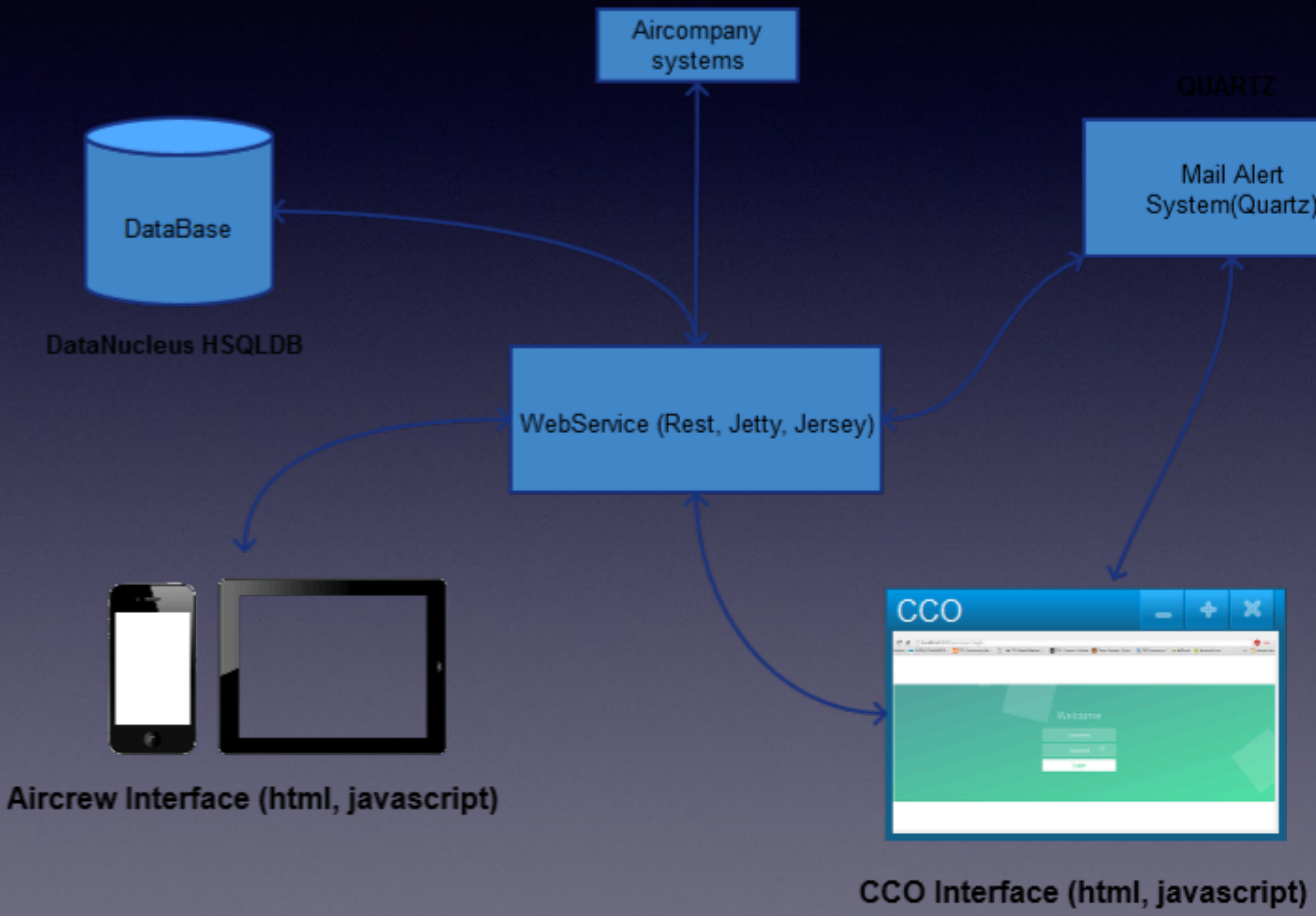
Younes
Hassanabadi

Kevin Youna

What you’ve asked.

- A web app
- two types of user:
  - aircrew
  - CCO
- Different types of permission:
  - the aircrew can read
  - the COO can totally manage(read,write,delete)
- Different way to create flight:
  - Excel import
  - Manually
- Alert system:
  - A week before the flight in case of missing plane or crew
  - Twelve hours before the flight in case of missing information
- Authentification system

Voila



Aircompany systems

QUARTZ

DataBase

Mail Alert System(Quartz)

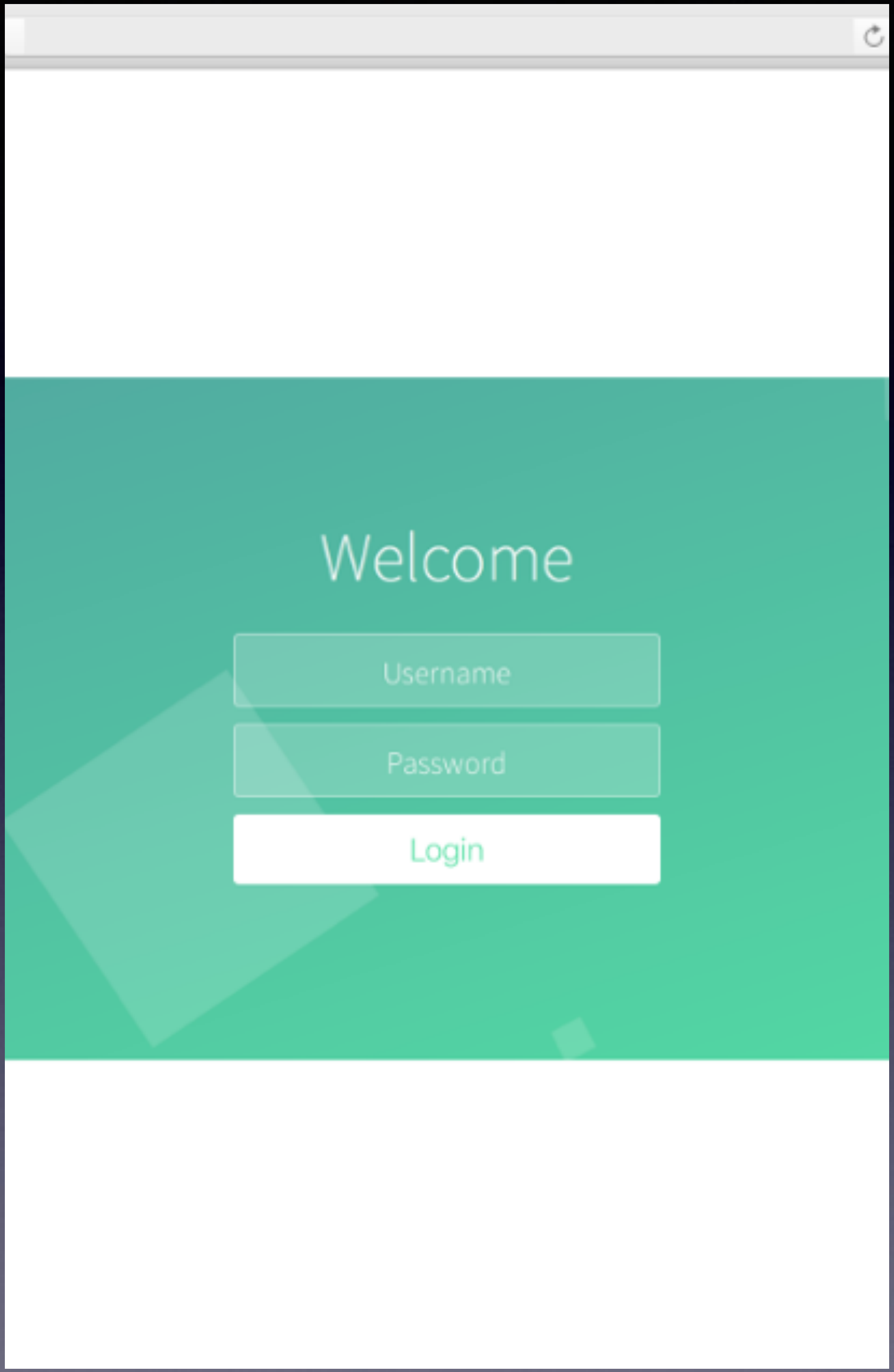
DataNucleus HSQLDB

WebService (Rest, Jetty, Jersey)

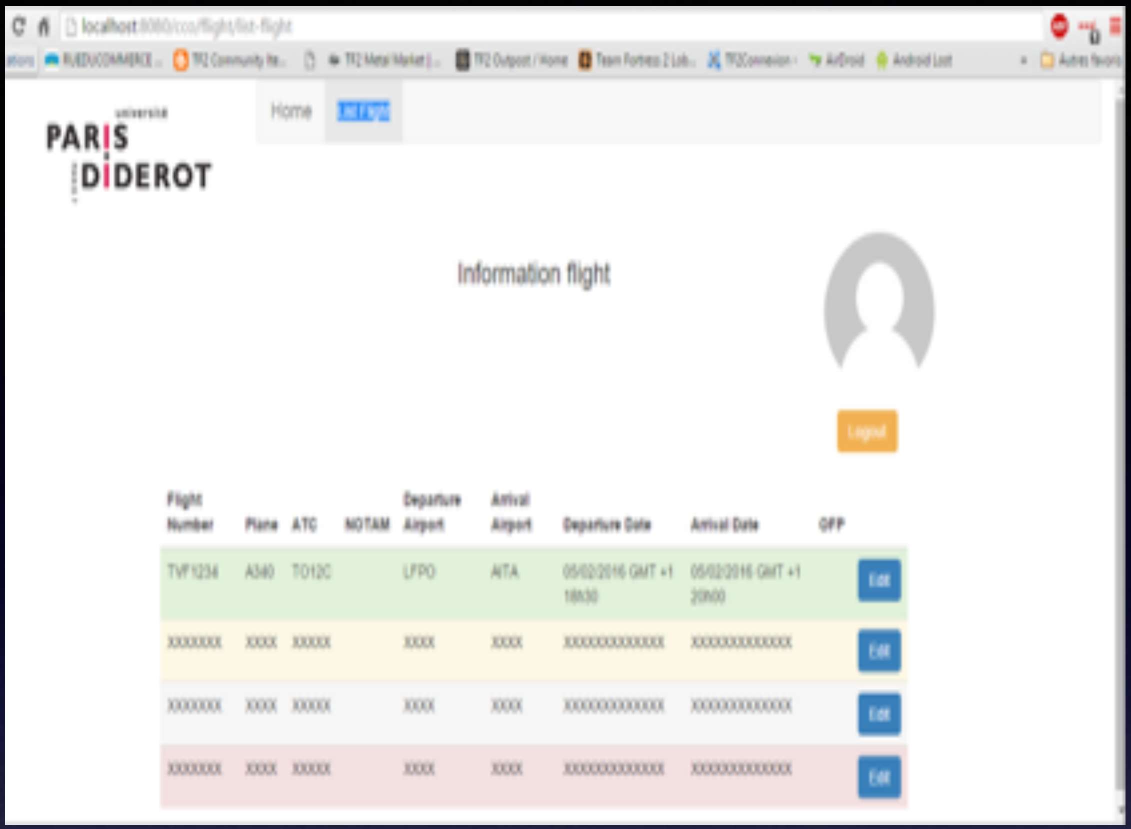
CCO

Aircrew Interface (html, javascript)

CCO Interface (html, javascript)

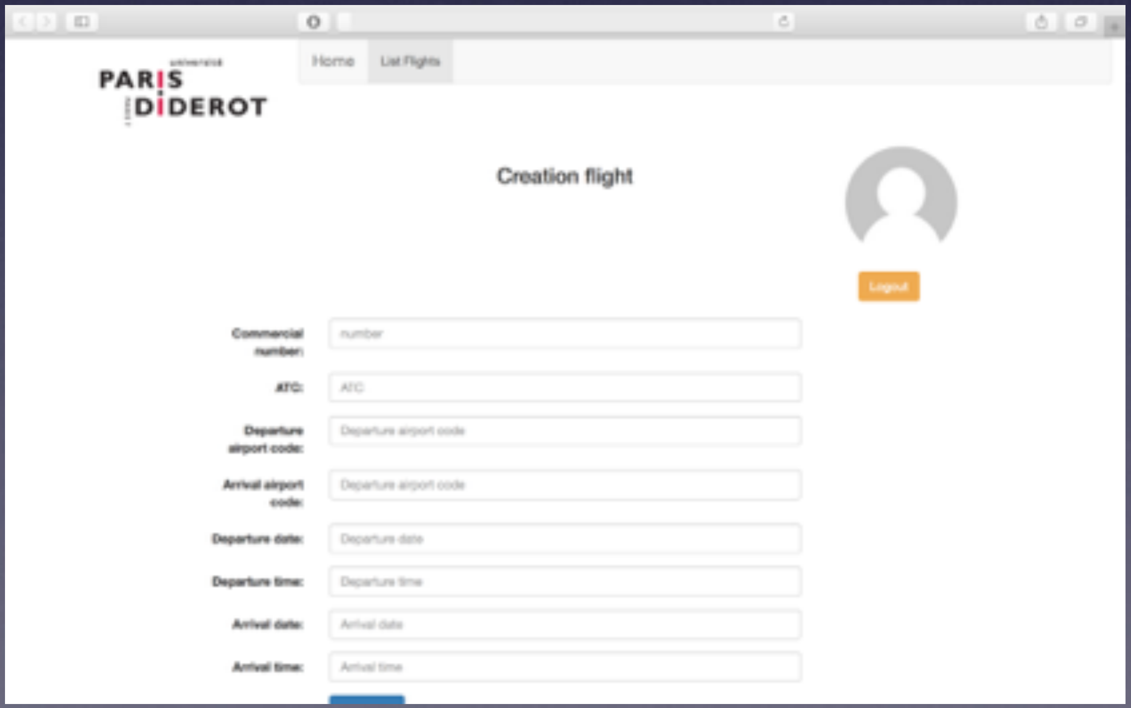


Welcome



Information flight

Flight Number	Plane	ATC	NOTAM	Departure Airport	Arrival Airport	Departure Date	Arrival Date	OFP	
TVF1234	A340	TO12C		LFPO	AITA	05/02/2016 GMT +1 18h30	05/02/2016 GMT +1 20h00		Edit
XXXXXXX	XXXX	XXXXX		XXXX	XXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX		Edit
XXXXXXX	XXXX	XXXXX		XXXX	XXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX		Edit
XXXXXXX	XXXX	XXXXX		XXXX	XXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX		Edit



Creation flight

Commercial number:	number
ATC:	ATC
Departure airport code:	Departure airport code
Arrival airport code:	Departure airport code
Departure date:	Departure date
Departure time:	Departure time
Arrival date:	Arrival date
Arrival time:	Arrival time

Web services

Module	View	Fonction	Method	Json	Path Param1	Path Param2
user	login	User authentification	POST	{‘type’ : ‘cco’} or {‘type’ : ‘crew’}	userid	token = password token
flight	List-flight	Return the list of flight	GET			
flight	create-flight	Add a flight	PUT	flight data in Json format	userid	
flight	show-flight	Return the detail for the matching id flight	GET		id = id of flight	
flight	show-flight	Delete a flight	DELETE		userid	
flight	modify-flight	modify the detail of the flight	POST	flight data in Json format	userid	
leaflet	list-leaflet	Return the list of the commercial leaflet	GET			
leaflet	create-leaflet	Add a commercial leaflet	PUT	leaflet data in Json format		
leaflet	show-leaflet	Delete a commercial leaflet	DELETE		id = id of leaflet	
leaflet	show-leaflet	Return details of leaflet for the corresponding	GET		id = id of leaflet	

« L'homme bien préparé au combat a déjà vaincu à demi. »

Cervantes

Vianes

(Via Airlines)

What you’ve asked.

- A web app
- two types of user:
  - aircrew
  - CCO
- Different types of permission:
  - the aircrew can read
  - the COO can totally manage(read,write,delete)
- Different way to create flight:
  - Excel import
  - Manually
- Alert system:
  - A week before the flight in case of missing plane or crew
  - Twelve hours before the flight in case of missing information
- Authentification system

What you get

- A web app
- two types of user:
  - aircrew
  - CCO
- Different types of permission:
  - the aircrew can read
  - the COO can totally manage(read,write,delete)
- Different way to create flight:
  - Excel import
  - Manually
- Alert system:
  - A week before the flight in case of missing plane or crew
  - Twelve hours before the flight in case of missing information
- Authentification system
- Pdf and pictures import (for free)

Web services

Module	View	Fonction	Method	Json	Path Param1	Path Param2
user	login	User authentification	POST	{‘type’ : ‘cco’} or {‘type’ : ‘crew’}	userid	token = password token
flight	List-flight	Return the list of flight	GET			
flight	create-flight	Add a flight	PUT	flight data in Json format	userid	
flight	show-flight	Return the detail for the matching id flight	GET		id = leaflet id	
flight	show-flight	Delete a flight	DELETE		userid	
flight	modify-flight	modify the detail of the flight	POST	flight data in Json format	userid	
leaflet	list-leaflet	Return the list of the commercial leaflet	GET			
leaflet	create-leaflet	Add a commercial leaflet	PUT	leaflet data in Json format		
leaflet	show-leaflet	Delete a commercial leaflet	DELETE		id = leaflet id	
leaflet	show-leaflet	Return details of leaflet for the corresponding id	GET		id = leaflet id	
aircraft	list-aircraft	return the aircraft available for the flight	POST	Fight data		
airport	list-airport	return all the airport in the database	GET			
upload	file	allow the user to import excel file and to integrate them	POST	{« file » : base64_string}		
upload	file-pdf	allow the user to upload a file	POST	{« file » : base64_string}		
User	list-user	return all the user in the database	GET			
Upload	file-ofp	affect a specific ofp to a specific flight	POST	{« file » : base64_string}	id = flight id	

Live Demo

Please connect to the « insert name here » wifi network
go to http://insert_ip_here:8080

Thank You !

https://github.com/mikhailpgrl/G-nie-Lociel-FPS/