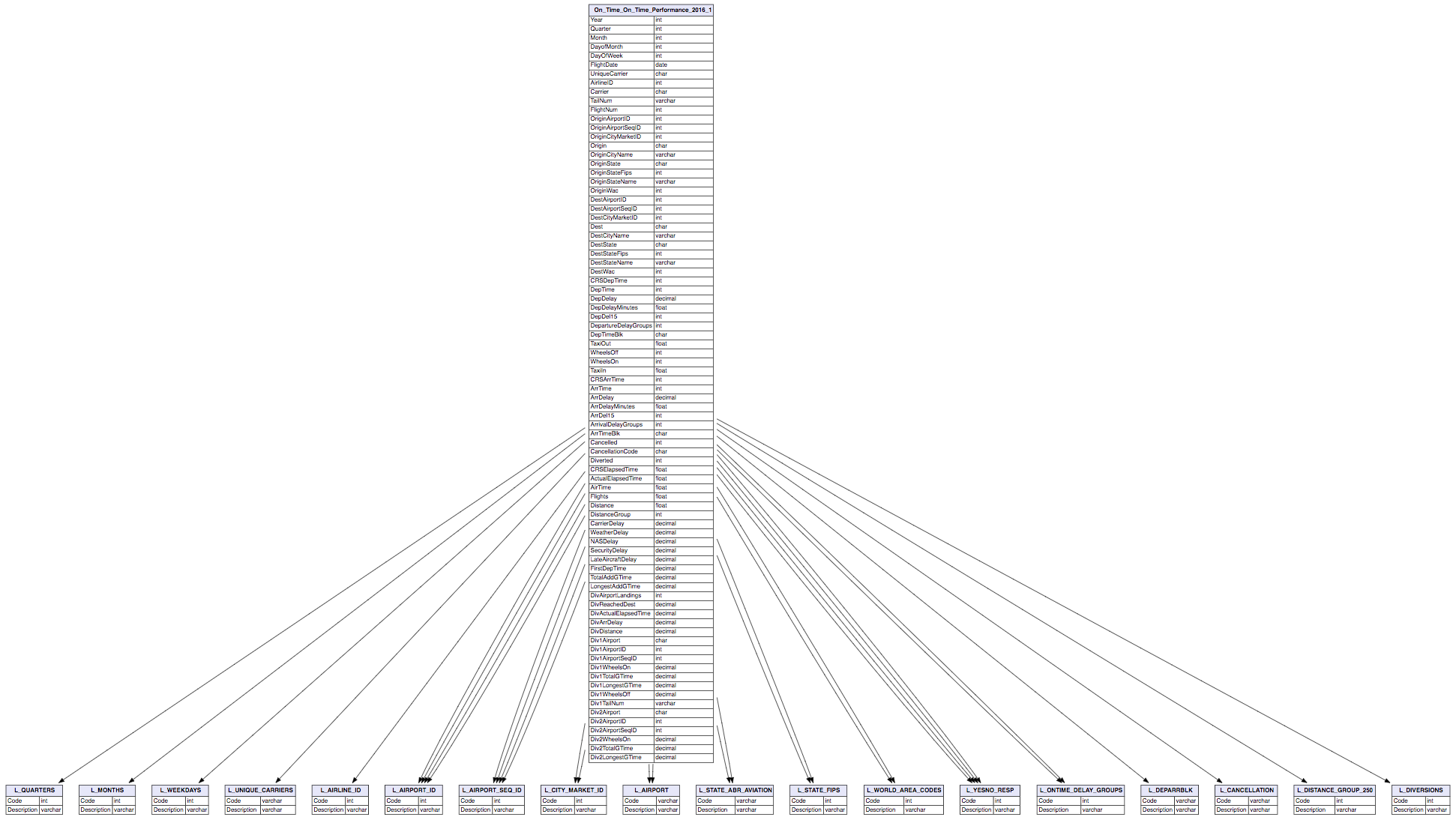
*DUPHENIEUX Valentin, LECOMTE Numa, QUENOT Akira*

Développement d’application Cloud

Projet

## Jeux de données

Nous avons choisi la base de donnée « Airline » parmi les différentes bases de données disponibles sur <https://relational.fit.cvut.cz/search>. Cette base de données est une base SQL contenant tous les vols aériens aux Etats-Unis en Janvier 2016 et dont le schéma relationnel est défini comme suit :



## 1.2 Spécification des besoins

### Spécification des besoins pour l’application

L’application doit proposer 3 interfaces à l’utilisateur comme le montre le schéma ci-dessous :

Application Home

Standard

Administrateur

Analyste

### Spécification des besoins pour les requêtes

L’application doit requêter la base de données afin de livrer des informations pertinentes à l’utilisateur. Dans le cadre de ce projet et en vue de la base de données choisie, les requêtes seront :

* Pour la partie « Standard » :
  + Obtenir la liste des vols par compagnie aérienne (paramètres : Compagnie, Date)
  + Obtenir la liste des vols entrants et sortants d’un aéroport (paramètres : Aéroport, Date)
  + Obtenir la liste des vols entre 2 aéroports (paramètres : Aéroport de départ, Aéroport d’arrivée, Date)
* Pour la partie « Analyste » :
  + Obtenir le pourcentage d’avoir un retard sur un vol en départ d’un aéroport (paramètre : Aéroport)
  + Obtenir le pourcentage d’avoir un retard sur un vol en arrivée d’un aéroport (paramètre : Aéroport)
  + Obtenir la moyenne des retards en minutes d’un vol en départ d’un aéroport (paramètre : Aéroport)
  + Obtenir la moyenne des retards en minutes d’un vol en arrivée d’un aéroport (paramètre : Aéroport)
  + Obtenir le top 10 des compagnies qui atterrissent le plus dans un aéroport (Paramètres : Aéroport)
  + Obtenir le top 10 des compagnies qui décollent le plus d’un aéroport (Paramètres : Aéroport)
* Pour la partie « Administrateur » :

## 1.3 Dénormalisation

### 1.3.1 Schéma

{

    "\_id" : ObjectId("5a2e90805498680c08fe6631"),

    "FL\_DATE" : ISODate("2017-01-01T00:00:00.000+0000"),

    "UNIQUE\_CARRIER" : "AA",

    "AIRLINE\_ID" : NumberInt(19805),

    "CARRIER" : "AA",

    "FL\_NUM" : NumberInt(307),

    "ORIGIN" : "DEN",

    "ORIGIN\_CITY\_NAME" : "Denver, CO",

    "ORIGIN\_STATE\_ABR" : "CO",

    "ORIGIN\_STATE\_NM" : "Colorado",

    "DEST" : "PHX",

    "DEST\_CITY\_NAME" : "Phoenix, AZ",

    "DEST\_STATE\_ABR" : "AZ",

    "DEST\_STATE\_NM" : "Arizona",

    "DEP\_TIME" : NumberInt(1135),

    "DEP\_DELAY" : -10.0,

    "DEP\_DEL15" : 0.0,

    "WHEELS\_OFF" : NumberInt(1153),

    "WHEELS\_ON" : NumberInt(1321),

    "ARR\_TIME" : NumberInt(1328),

    "ARR\_DELAY" : -17.0,

    "ARR\_DEL15" : 0.0,

    "CANCELLED" : 0.0,

    "DIVERTED" : 0.0,

    "AIR\_TIME" : 88.0,

    "FLIGHTS" : 1.0,

    "DISTANCE" : 602.0,

    "CARRIER\_DELAY" : null,

    "WEATHER\_DELAY" : null,

    "NAS\_DELAY" : null,

    "SECURITY\_DELAY" : null,

    "LATE\_AIRCRAFT\_DELAY" : null,

    "AIRLINE\_NAME" : "American Airlines Inc.: AA"

}

### 1.3.2 Transformation

Nous souhaitions rajouter le champ contenant le nom de la compagnie dans notre liste, celle-ci contenant déjà le code ID de la compagnie. Nous avons alors dû créer un nouveau champ « AIRLINE NAME », que nous remplissons pour chaque document avec le label correspondant en utilisant le code dans une table spécifiée. Voici le code qui a été exécuté pour effectuer cette modification :

db.FLIGHTS.update({},

{$set: {AIRLINE\_NAME: null}},

{multi: true});

db.FLIGHTS.find().forEach(function (flightInfo) {

var doc2 = db.L\_AIRLINE\_ID.findOne({ Code : flightInfo.AIRLINE\_ID }, { Description: 1 });

if (doc2 != null) {

flightInfo.AIRLINE\_NAME = doc2.Description;

db.FLIGHTS.save(flightInfo);

}

});

Le client MongoDB nous a donc retourné le résultat suivant :

WriteResult({ "nMatched" : 1349131, "nUpserted" : 0, "nModified" : 1349131 })

## 1.4 Vues

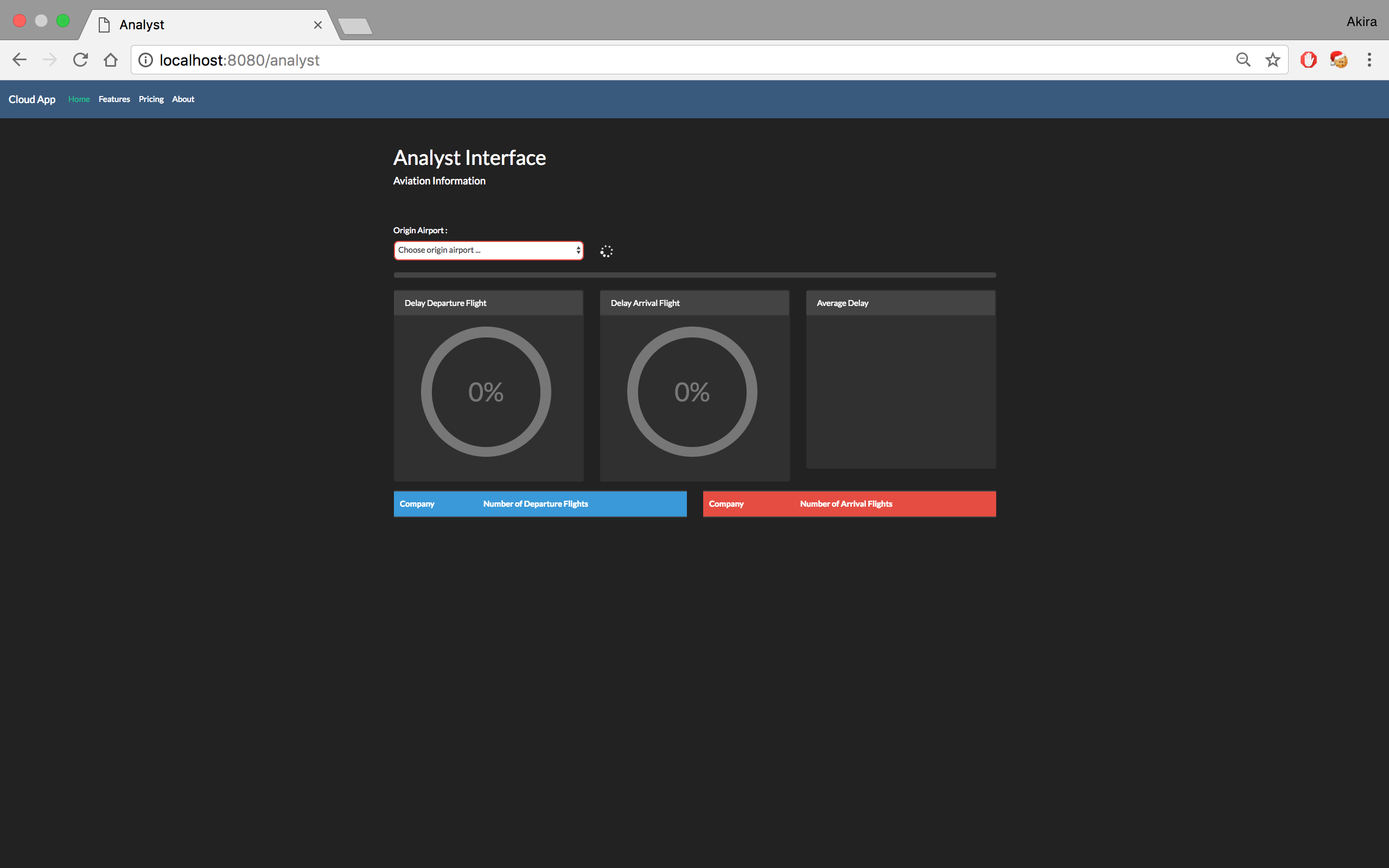
### Vue Home (index.html)

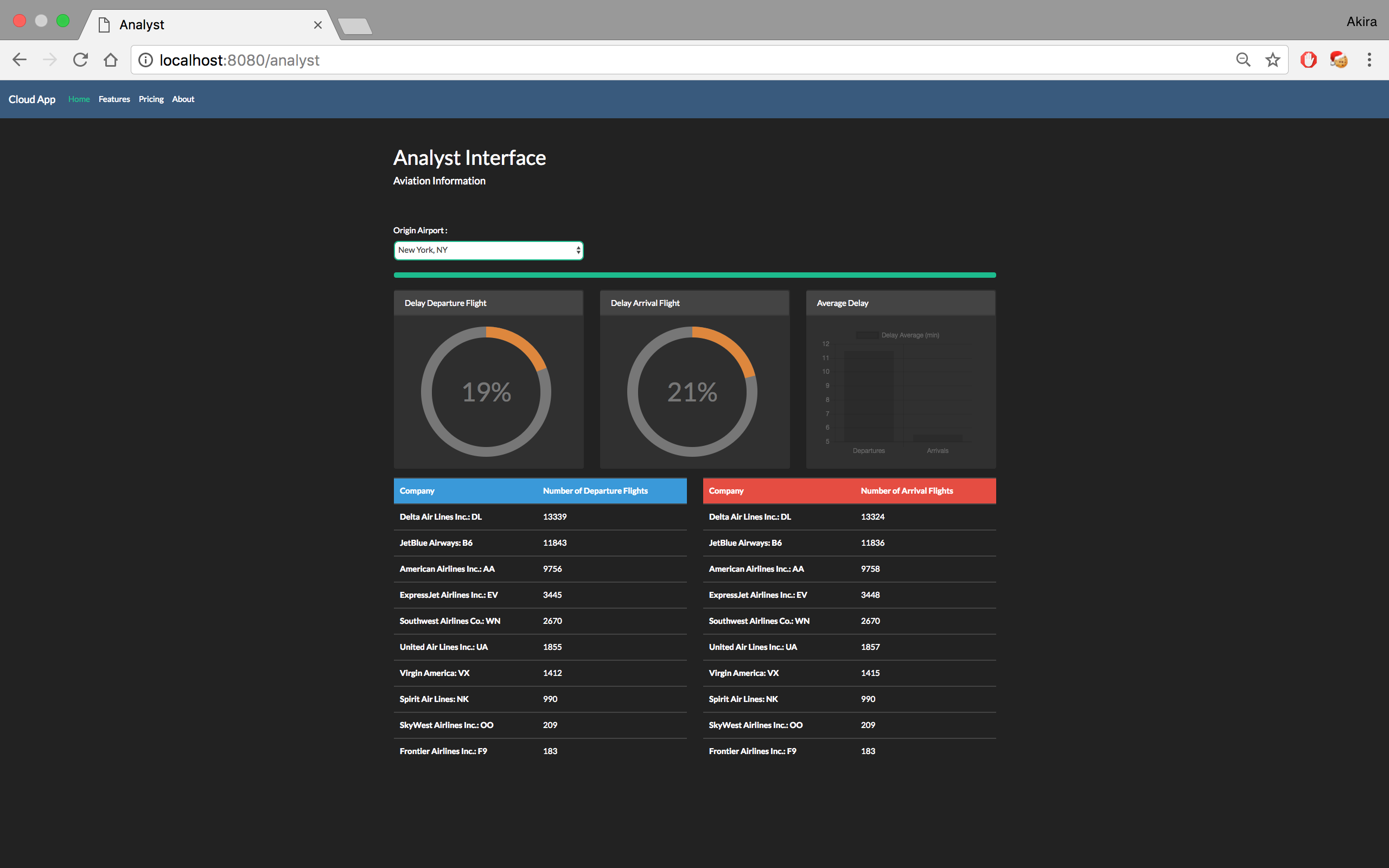


Cette vue est la page d’accueil de l’application. Elle laisse à l’utilisateur le choix d’accéder au 3 différentes interfaces que propose l’application.

### Vue Standard

### Vue Analyste (analyst.html)





Cette vue est la vue affichant à l’utilisateur les résultats de requêtes de type « Analyste ».

Sur cette vue, 5 requêtes sont effectuées :

* + **Liste déroulante des aéroports :**



* + **Proportion d’avions ayant du retard au départ de l’aéroport choisi + Moyenne de ces retards :**

****

* + **Proportion d’avions ayant du retard à l’arrivée de l’aéroport choisi + Moyenne de ces retards :**

****

* + **Top 10 des compagnies ayant le plus de vol en départ de cette aéroport :**



* + **Top 10 des compagnies ayant le plus de vol en arrivée de cette aéroport :**



### Vue Admin

Pour une compagnie + Entre 2 dates 🡪 tous les vols

Pour un trajet (Origine + Destination) 🡪 Obtenir tous les vols trié par temps

Pour un aéroport + Date 🡪 Arrivées + Départs

Pour un vol 🡪 Infos

Pour un aéroport 🡪 Moyenne des retards (départs + arrivés)

Pour une compagnie 🡪 Moyenne des retards (dep + arr)

Top 5 🡪 des compagnies ayant le plus de vols

Top 5 🡪 des aéroports accueillant le plus de vols

Performance indicators :

Temps de la requête

Dénormalisation :

* Rajouter en utilisant AirlineID par AIRLINE\_ID.Description