



# Master 1

## Vision et Machine Intelligente

Rapport Du projet Web Et Programmation distribuée (Réalisation d'une plateforme de gestion de parking)

Réaliser par : Mohamed Yakoub Azzouni.

Github Repository : <https://github.com/YakoubAzzouni/Webapp>

# Plan de rapport

- I)Contexte générale du projet
- II)Conception et Outils utilisés
- III)Réalisation de l'application

## Chapitre1 - contexte générale du projet :

Dans ce chapitre on introduit le contexte général du projet dont il présente le but du projet, la problématique, l'étude du besoin et la planification.

### I)Introduction :

#### 1)Problématique :

La boite Yakoub Parking, est une boite qui a plusieurs parkings, et qui a besoin d'automatiser sa gestion des parkings par l'admin et le processus de réservation de place.

#### 2)Etude des besoins :

Le principal besoin de la boite est d'avoir une solution informatique permettant d'automatiser la gestion des parkings et des réservations.

La solution est de développer une plateforme qui répond au besoin de l'entreprise ou l'application est accessible par tout le monde sans devoir créer un compte sauf pour l'admin ou il peut gérer les parkings et autres tâches administratif.

L'application doit permettre au client de voir tous les parkings ainsi de faire une réservation ou il choisit la date de début et de fin de l'abonnement.

L'admin gère les parkings et les villes (crée supprimer éditer), et il consulte la liste des réservations.

### 3)Solution :

La plateforme Yakoub parking satisfait les besoins des clients et de l'admin ou elle optimise le temps par remplacer tout le travail manuel par une application web.

## II)Les fonctionnalité :

### 1)Admin :

- Faire le login (connecter)
- Ajouter/supprimer/éditer/afficher les parkings
- Ajouter/supprimer/éditer/afficher les villes
- Afficher les réservations
- Log out (déconnection)

### 2)Client :

- Afficher les parkings et les villes
- Faire une réservation
- Récupère la confirmation de payement sous forme de Qrcode
- Afficher par nom de ville les parkings
- Consulter l'application web

## III)Conclusion de ce chapitre :

Dans ce chapitre nous avons présenté en générale le contexte du projet, la problématique et la solution proposé, dans le prochain chapitre nous parlons de la conception du projet.

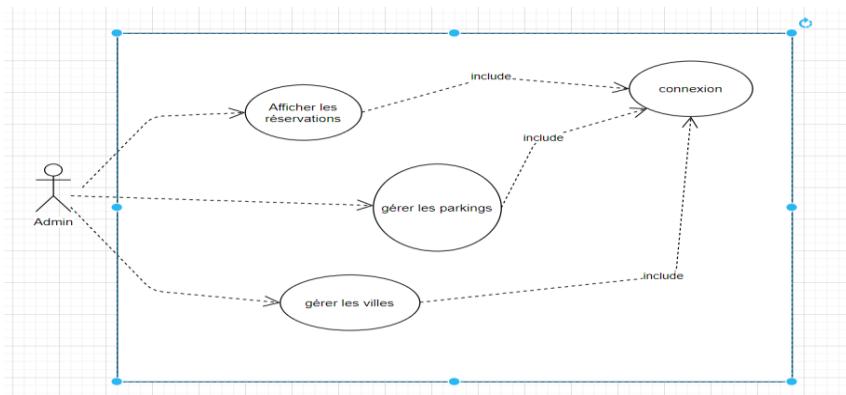
## Chapitre2 - Conception du projet:

Dans ce chapitre en parle de la conception du projet et les technologies utilisées dans l'application web.

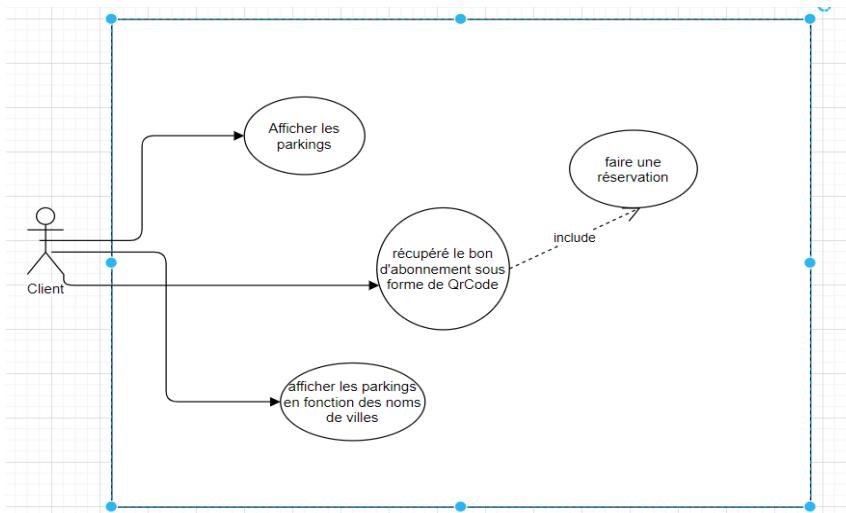
### I)Conception :

#### 1- diagramme de cas d'utilisation :

. Pour Admin :

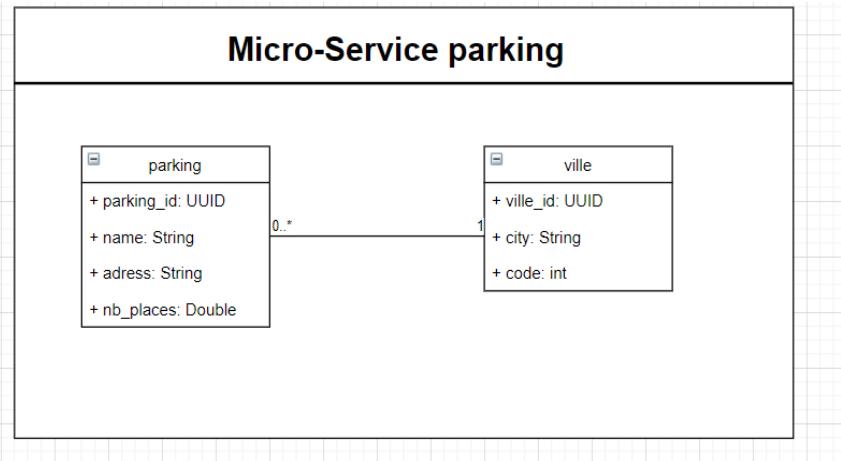


. Pour Client :

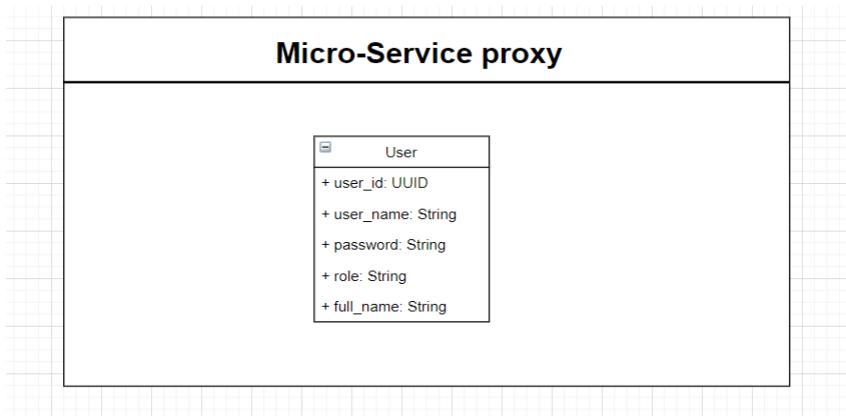


## 2- diagramme de classe :

. Micro-service parking :



. Micro-service proxy :



## 3- Outils et technologies :

. Backend :

- Java 8
- Spring Boot
- Spring Data
- Spring Security
- ZUUL
- MySql
- REST
- Eureka (dans la version local)

. Frontend :

- Angular 8
- Angular material design
- Angular Flex Layout
- Bootstrap 4
- Font Awesome

. infrastructure :

- Docker
- Nginx
- Aws Ec2
- Kubernetes (k8s)

## II) Conclusion de ce chapitre :

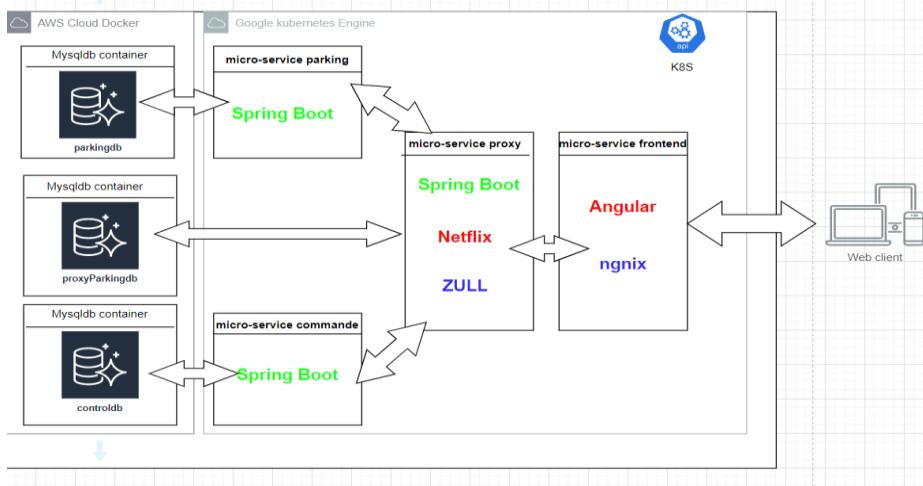
Dans ce chapitre nous avons fait l'étude conceptionnelle pour savoir les tâches qu'on doit réaliser, dans le prochain chapitre on entame la réalisation de l'application.

## Chapitre3 - Réalisation:

Après avoir introduit le contexte général du projet et faire l'étude conceptionnelle, la prochaine étape sera de faire la réalisation de l'application.

### I)L'infrastructure de la plateforme :

La figure ci-dessous montre la structure qui a été mise en place lors de la réalisation du projet, elle est la plus efficace et généralisable dans les projets Web sur le cloud.



## II)Explication :

Dans le **backend** on a trois micro-services ou le rôle de chacun est comme suite :

- . Micro-service **PARKING** : fournit un service pour gérer les parkings et les villes.
- . Micro-service **COMMANDE** : fournit un service pour gérer les réservations et les clients.
- . Micro-service **PROXY** (Gateway) : fournit une API de type service web au micro-service **FRONTEND**, il a aussi le rôle de donner l'accès au autres micro-service de backend de l'extérieur et via l'internet en transmettant indirectement les requêtes qui lui sont adressées (ZUUL), cela aide la boite de ne pas exposer leur micro-service en les plaçant derrière le micro-service proxy qui et protéger (Spring Security).

Pour le coté **frontend** on a un seul micro-service **FRONTEND** qui fournit une interface web pour le client ou les technologies utilisées dans ce micro-service sont Angular et Nginx.

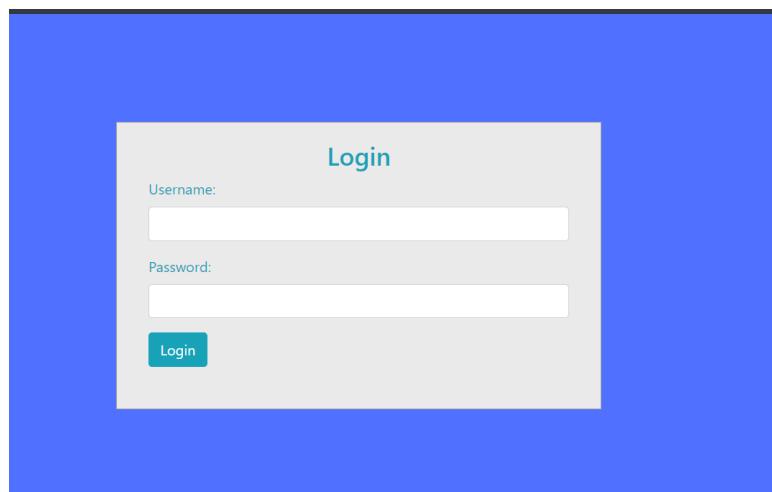
Or que pour l'infrastructure on a le backend et le frontend qui sont déployer sur Kubernetes (k8s), et les bases de données qui ont pour rôle de stocker les info des parkings et de l'admin etc...

## III)Démonstration :

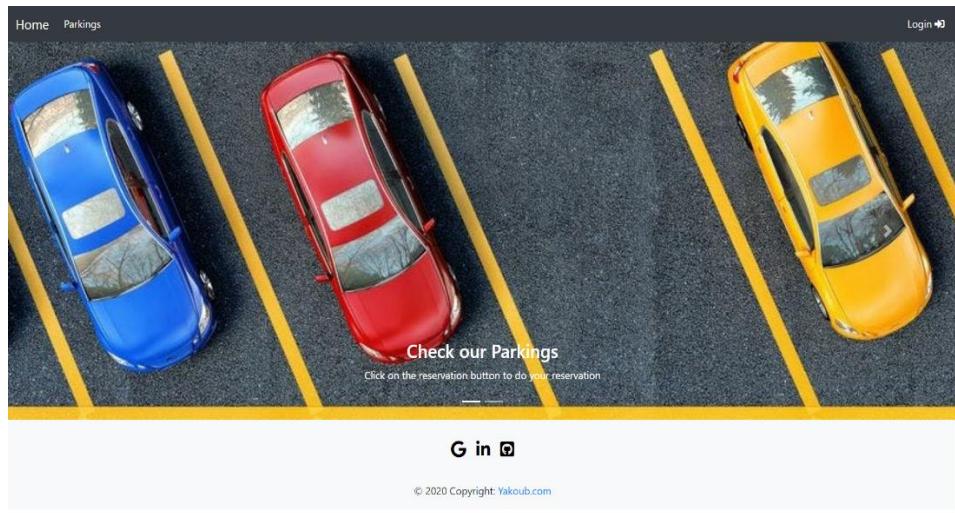
Dans ce titre je vais présenter quelque fonctionnalité :

### 1) Frontend :

La page de login de l'admin :



## Home :



## Gestion des villes pour l'admin :

A screenshot of an admin panel titled "Check out The Cities". The interface includes a navigation bar with "Home", "Parkings", and "Settings" and a user profile icon. A modal window titled "Reservation" is open, with the message "in order to create a new Ville" and a "Create" button. The main table lists three cities with columns for ID, Code, City, and Delete. The data is as follows:

#	ID	Code	City	Delete
1	4f7cb830-51f0-47ce-9c70-2ab0ab97242d	93300	auberviller	<button>Delete</button>
2	df3fbab3-3609-4efa-ae94-07cf5db4c6f2	94000	creteil	<button>Delete</button>
3	a1a19d6b-fa9d-411f-ad9c-0a212de71c40	97510	rosaPark	<button>Delete</button>

At the bottom, there is a footer with social media icons and a copyright notice: "© 2020 Copyright: Yakoub.com".

## La page parking pour client :

A screenshot of a client-facing parking reservation page. The background shows a parking lot with a car icon. In the center, a modal window titled "Reservation..." is open, containing fields for "Date\_in" (with placeholder "yyyy-mm-dd") and "Date\_out" (with placeholder "yyyy-mm-dd"). It also includes fields for "Name" and "Email". Below these fields is a "Plate number:" input field. At the bottom of the modal is a "Reservation" button. To the left and right of the modal are two cards with parking information: one for "hoho" at "10 rue du pilier" and another for "test2" at "12 rue thomas edison". Both cards show a car icon and "Infos" sections with "Code postal: 93300", "City: auberviller", and "Price: 10\$ per day". The footer of the page includes social media icons and a copyright notice: "© 2020 Copyright: Yakoub.com".

## 2) Backend :

Réponse de création d'un client :

The screenshot shows the Insomnia REST Client interface. The top bar indicates a successful response (200), a duration of 1.47s, and a body size of 116B. The timeline shows a single entry for 7 Days. The left sidebar lists environments (No Environment, Cookies), a filter, and a file structure with 'koubi' and 'client' folders. Under 'client', there are three requests: 'login' (POST), 'get all' (GET), and 'create' (POST). The 'create' request is selected, showing its JSON payload:

```
1 var {
2   "name": "hohoo",
3   "email": "azzouni@sdf",
4   "plate_number": 54845541
5 }
```

Réponse de lecture des parkings :

The screenshot shows the Insomnia REST Client interface. The top bar indicates a successful response (200), a duration of 9.16s, and a body size of 441B. The timeline shows a single entry for Just Now. The left sidebar lists environments (No Environment, Cookies), a filter, and a file structure with 'reservation', 'ville', and 'parking' folders. Under 'parking', there are four requests: 'create' (POST), 'getall' (GET), 'getall' (POST), and 'create' (POST). The 'getall' (POST) request is selected, showing its JSON response:

```
1 var [
2   {
3     "name": "hoho",
4     "price": 10.0,
5     "address": "10 rue du piller",
6     "nb_place": 15,
7     "ville": {
8       "ville_id": "4f7cb830-51f0-47ce-9c70-2ab0ab97242d",
9       "city": "aubervillier",
10      "code": 93300
11    },
12    "parking_id": "37df59ea-897a-40c5-ace4-bb8b5b6209e0"
13  },
14  {
15    "name": "test2",
16    "price": 50.0,
17    "address": "12 rue thomas edison",
18    "nb_place": 100,
19    "ville": {
20      "ville_id": "4f7cb830-51f0-47ce-9c70-2ab0ab97242d",
21      "city": "aubervillier",
22      "code": 93300
23    },
24    "parking_id": "7fe60e56-9685-4c06-83c4-cabd6c9c77a7"
25  }
26]
```

## Réponse du login :

The screenshot shows the Insomnia REST client interface. The left sidebar shows a project structure with environments, cookies, and a list of API endpoints. The main area shows a POST request to 'localhost:8080/login'. The 'Form' tab is selected, showing fields for 'username' (yakoub) and 'password' (1234). The 'Preview' tab shows the JSON response: { "status": "success" }. The top bar indicates a 200 status code, 1.95 s duration, 21 B size, and 'Just Now' timestamp.

Pour plus d'image consulter le git hub ou le lien ce trouve dans la première page du rapport.

## IV)Conclusion :

Ce chapitre avait pour but de présenter la réalisation du projet, on a présenté la structure de l'application par suite nous avons présenté quelque images qui présent l'interface de notre application.