



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

Chapitre 1 - 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 boîte Yakoub Parking, est une boîte 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 boîte 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 administratives.

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 (créer 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 paiement 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.

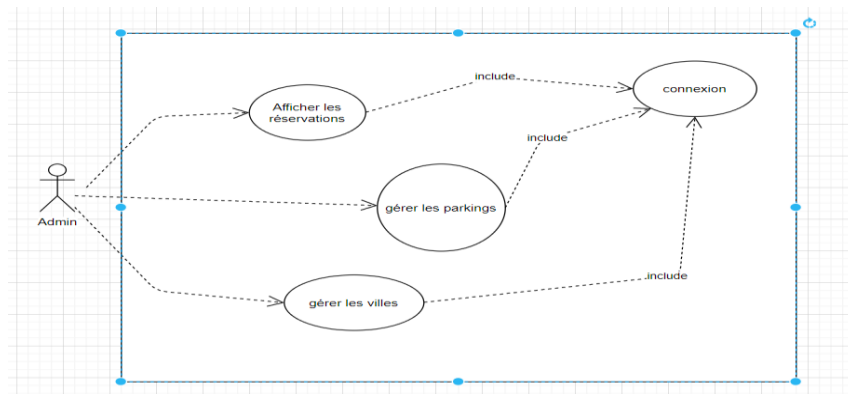
Chapitre2 - Conception du projet:

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

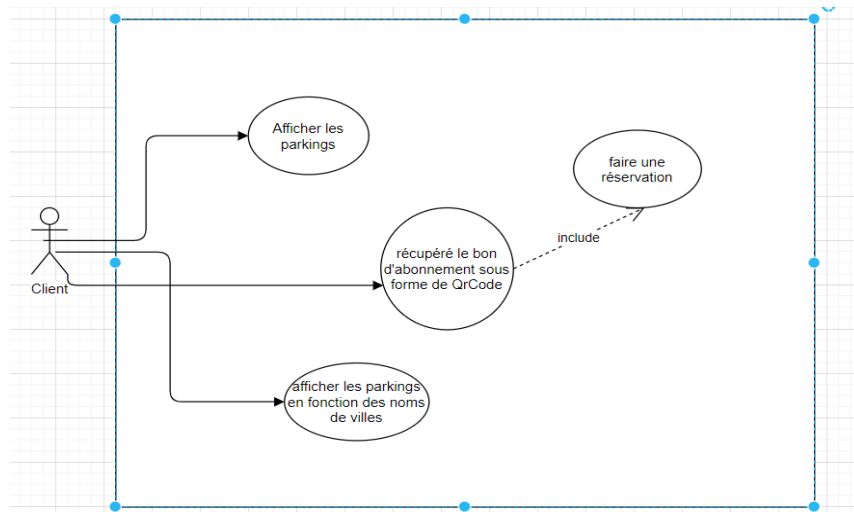
I)Conception :

1- diagramme de cas d'utilisation :

. Pour Admin :

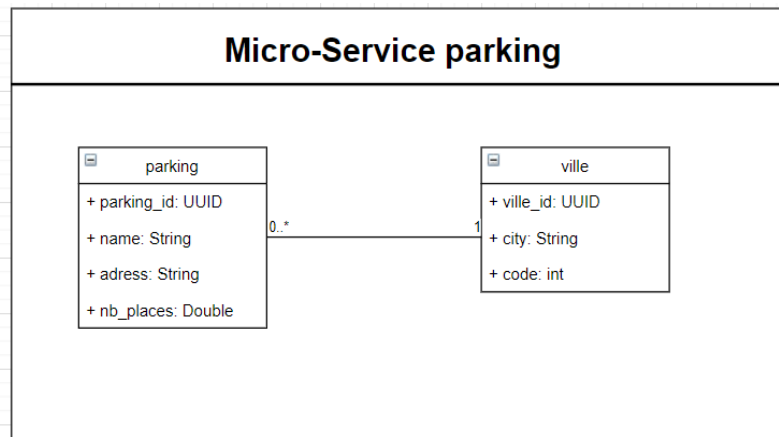


. Pour Client :

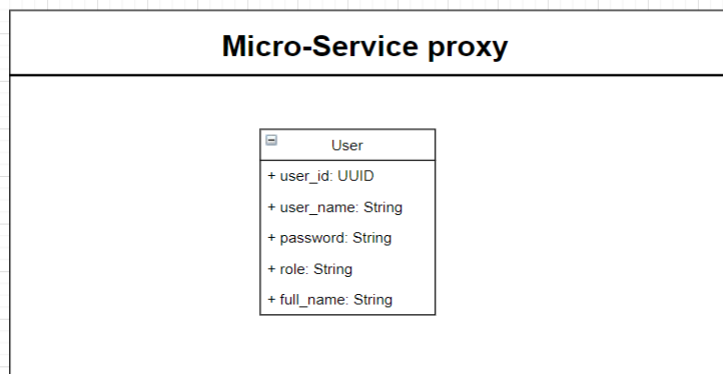


2- diagramme de classe :

. 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 :

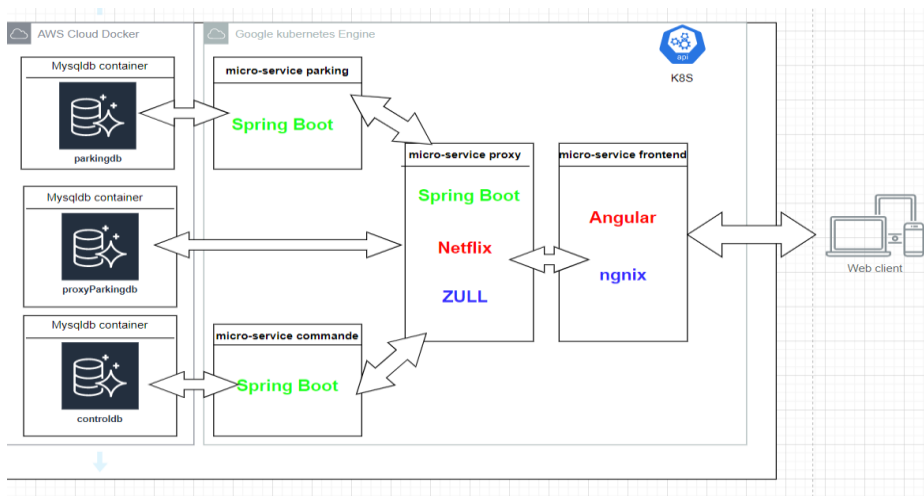
Dans ce chapitre nous avons fait l'étude conceptionnelle pour savoir les taches qu'on doit réaliser, dans le prochain chapitre on entame la réalisation de l'application.

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 boîte 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.

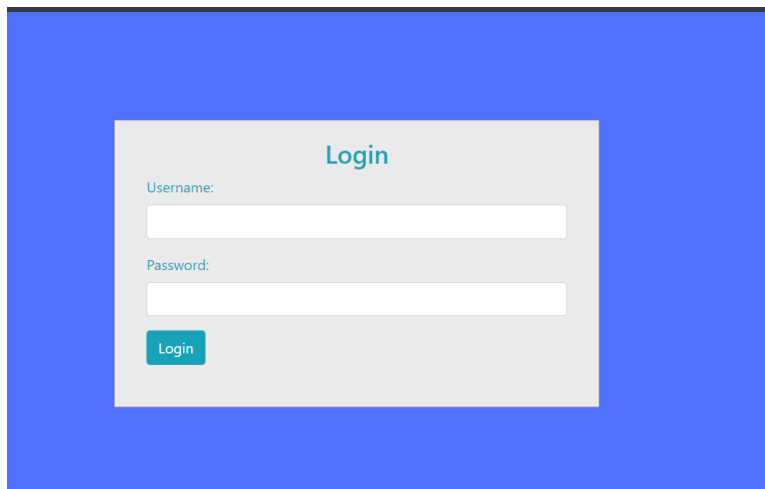
Or que pour l'infrastructure on a le backend et le frontend qui sont déployer sur Kubernetes (k8s), et les bases de données qui ont pour rôle de stocker les info des parkings et de l'admin etc...

III)Démonstration :

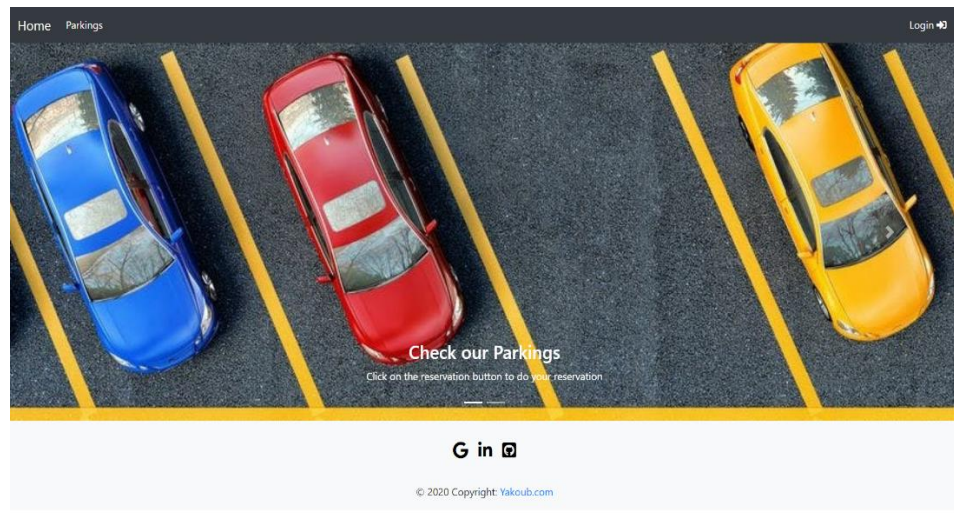
Dans ce titre je vais présenter quelque fonctionnalité :

1) Frontend :

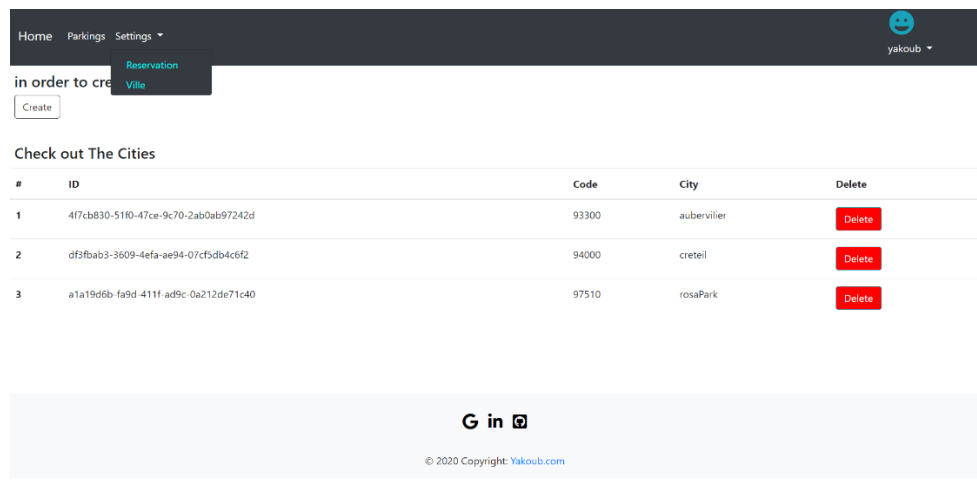
La page de login de l'admin :



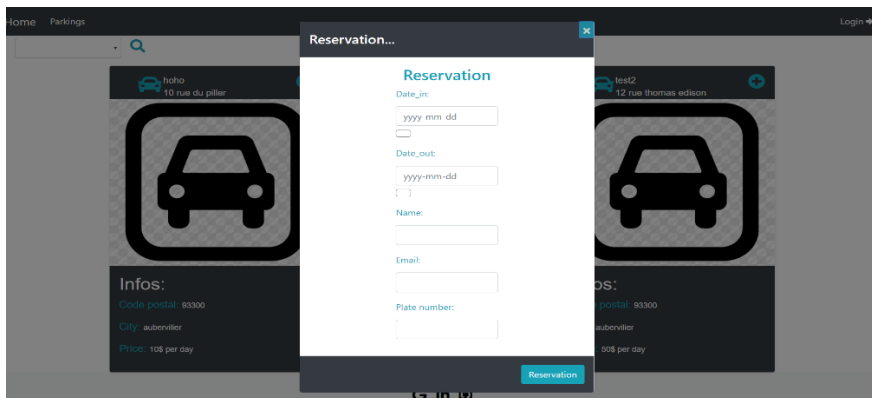
Home :



Gestion des villes pour l'admin :

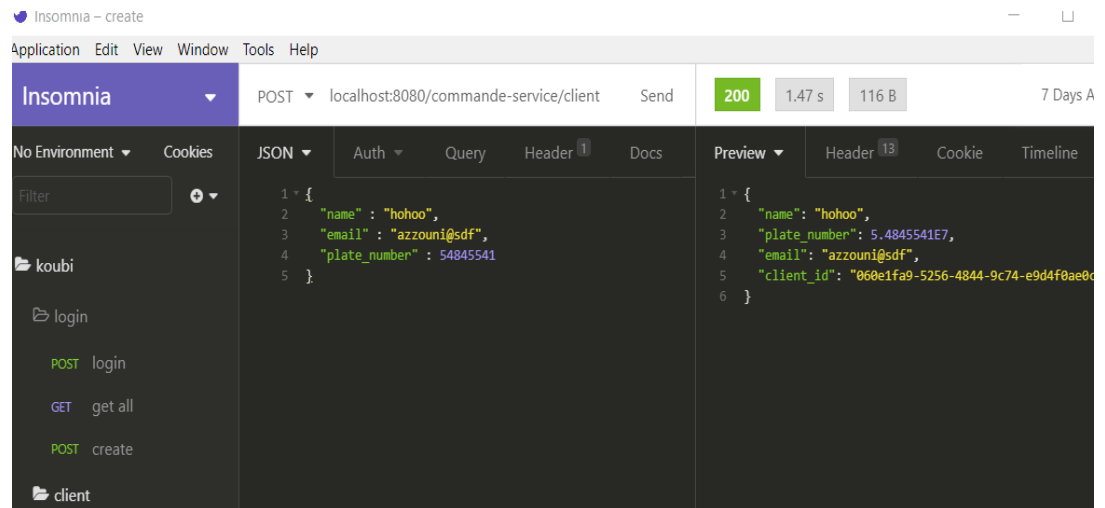


La page parking pour client :

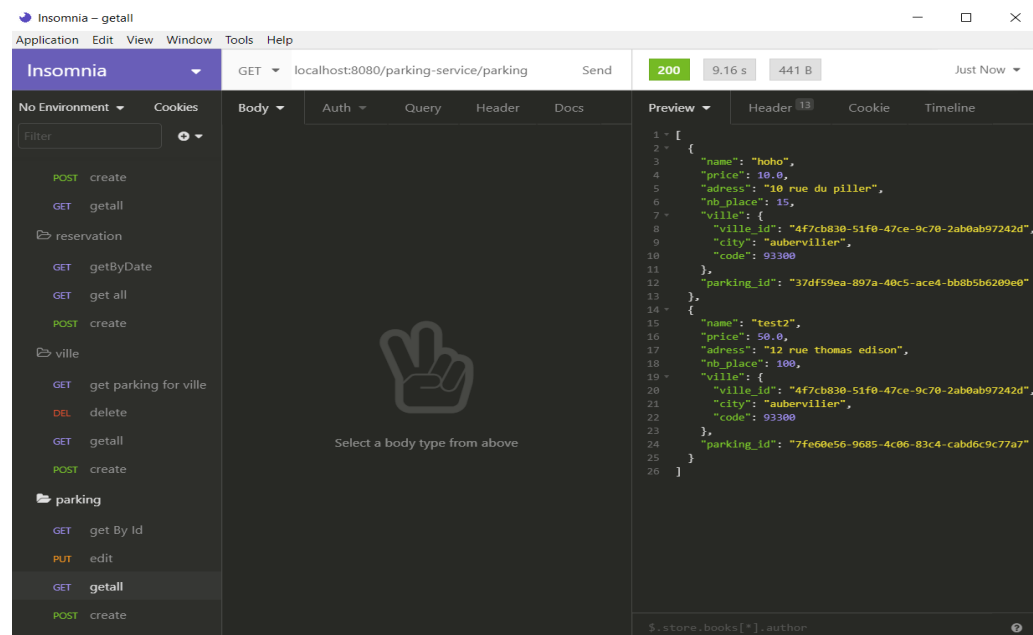


2) Backend :

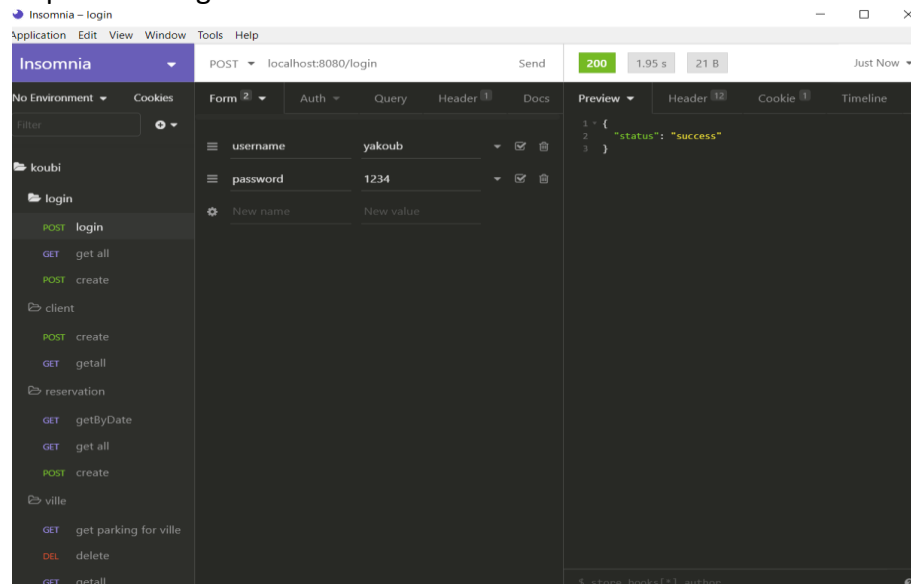
Réponse de création d'un client :



Réponse de lecture des parkings :



Réponse du login :



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.