*Ce projet a été réalisé dans le cadre de la préparation opérationnelle à l’emploi collective organisée par L’EPSI et Pôle Emploi Haut-de-France.*

Projet « Parking.App »

*par stephan duponchelle – soude adrien – defontaines quentin – Dens mike*

ANALYSE DU PROJET

Après l’étude du projet, nous avons choisi d’exploiter l’open data de le métropole Lilloise.

Plus particulièrement les données correspondantes aux parkings payants.

Le projet s’orientera donc vers une application web, qui permettra à l’utilisateur de visualiser les parkings ainsi que leur localisation. L’application devra permettre à l’utilisateur d’avoir les données des parkings en temps réel, de connaître son nombre de places et de savoir si le parking est ouvert ou non.

L’Équipe

Notre équipe se compose de quatre personnes qui ont toutes eu des rôles différents dans la conception de ce projet.

**Stéphane Duponchelle** : Qui a endossé le rôle de **Scrumaster** pour veiller au respect des rituels agiles, et qui a participé au développement de l’application.

**Quentin Deffontaines :** Qui a endossé le rôle de **chef de projet**, et qui a participé au développement de l’application.

**Adrien Soudé :** Qui a endossé le rôle du **Product Owner** pour analyser les besoins du projet, et qui a participé au développement de l’application.

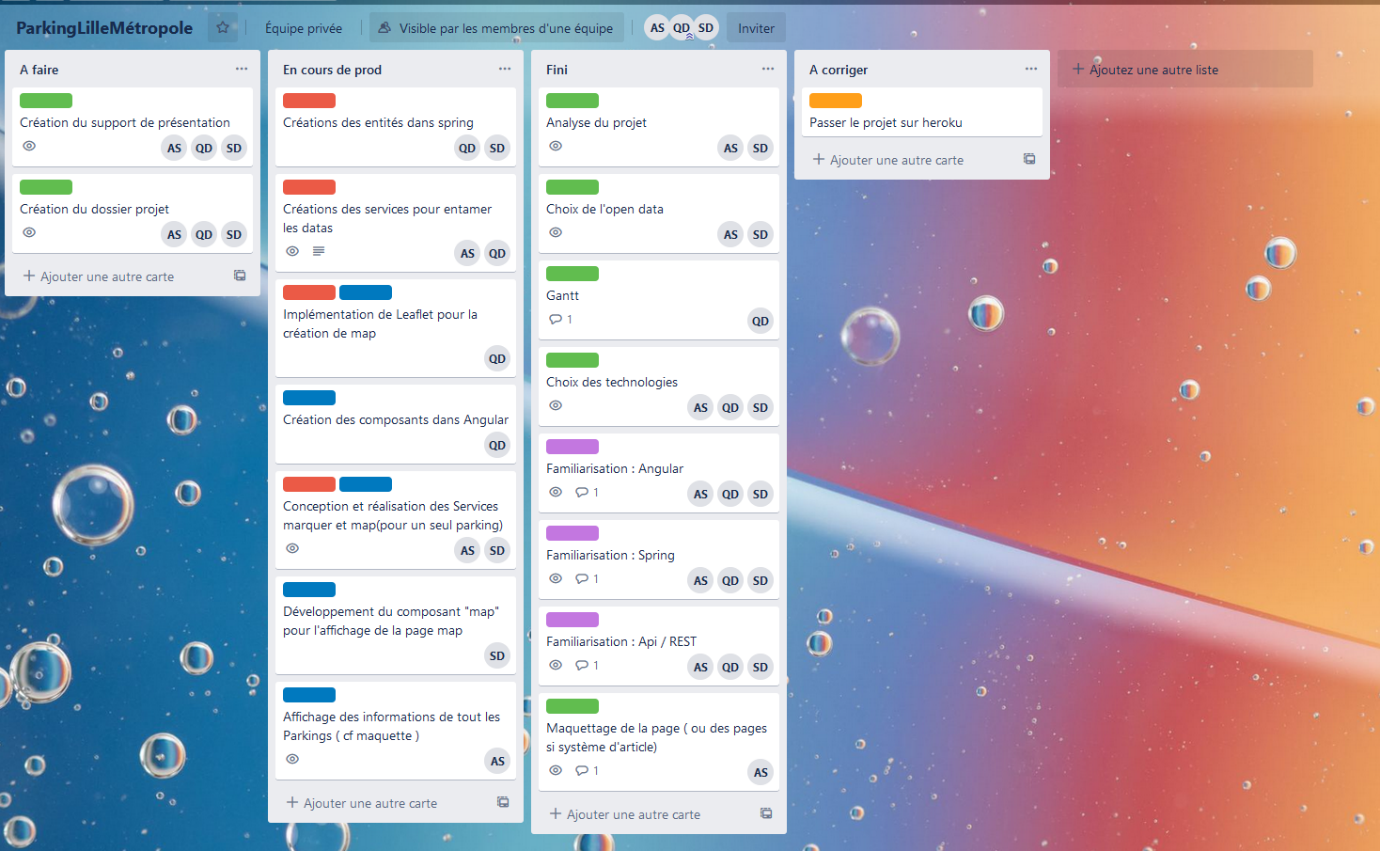
**Mike Dens :** Qui a participé au développement de l’application.

Gestion Du Projet

La gestion du projet s’est directement orientée vers l’utilisation d’un tableau KABAN, en effet, c’est ce qui nous est apparu comme le plus adapté à l’envergure du projet et à la taille de notre équipe.

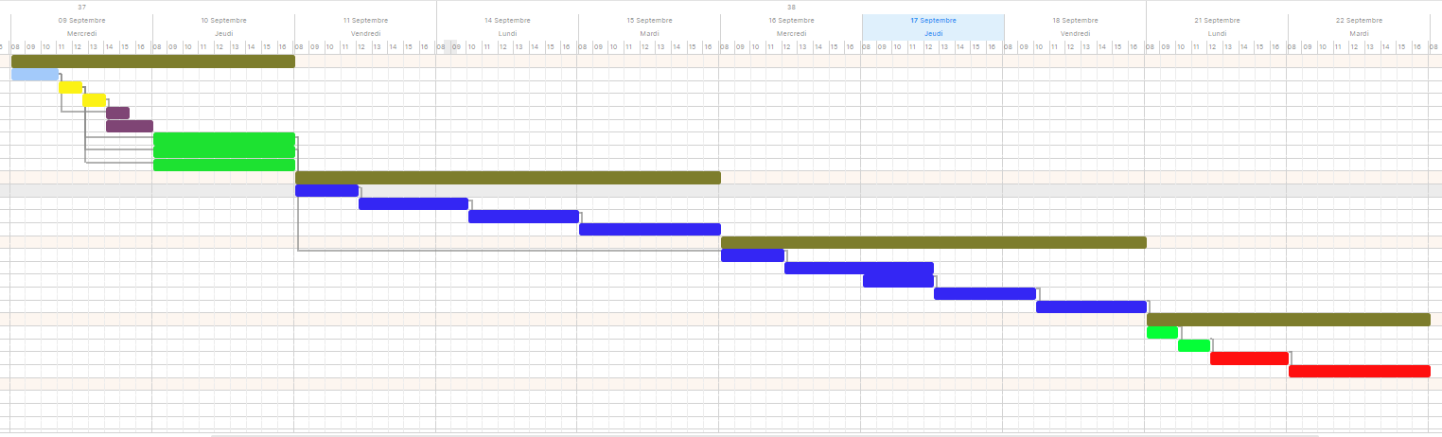
Après avoir analysé les besoins du projet nous avons découpé le projet en plusieurs tâches. Notre équipe étant hétérogène, nous avons décidé de prendre un peu de temps pour tous se familiariser avec les différents outils que nous allions utiliser.

Nous avons pris la décision de travailler tout en étant en conférence audio sur l’application Discord, ce qui a grandement facilité la communication au sein de l’équipe.

Nous avons mis en place un Trello que nous avons utilisé comme un tableau KABAN :

Le Trello nous a permis d’avoir un développement plus efficace et de gérer facilement toutes les étapes du projet malgré le distanciel.

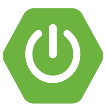
Nous avons ensuite synthétisé les étapes du développement et estimée le coût par tâches afin de pouvoir mieux estimer les charges de chaque tâche.

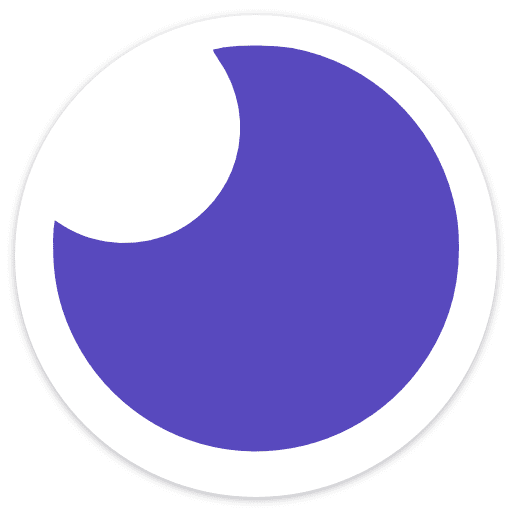
Par la suite nous avons synthétisé tout cela dans un diagramme de Gantt afin d’avoir une vision globale des charges du projet ainsi que de son déroulement.

Les Technologies

Après analyse des besoins du projet nous nous sommes mis d’accord sur les technologies que nous allions utiliser pour le développement de l’application.

Les données que nous utilisons nous proviennent de l’open data de la métropole Lilloise.

Nous avons choisi Spring et Spring Boot pour développer le back de l’application.



Nous avons utilisé Insomnia Core pour tester nos services d’accès vers l’api afin de pouvoir interroger les fichiers Json.

Pour la partie front nous avons choisi Angular qui nous a permis de développer la partie front et de la séparer complètement de la partie back-end de l’application. Et de nous permettre d’adopter plus facilement une architecture REST.

Afin d’intégrer une carte interactive des parkings, nous avons utilisé la librairie Leaflet pour permettre un développement plus efficace de l’application.

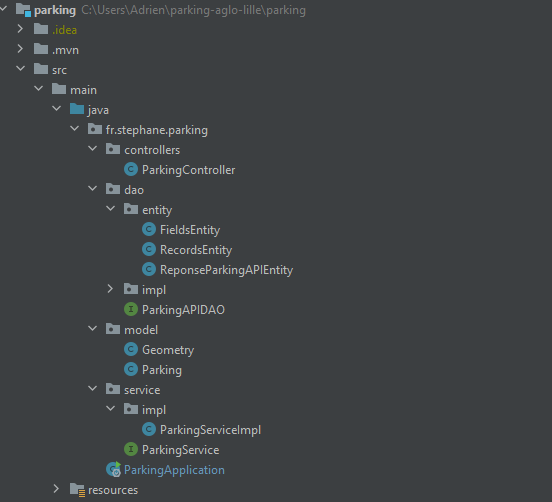
PHASE DE DEVELOPPEMENT

Pour permettre un développement efficace de l’application et pour éviter des problèmes de compatibilité, nous avons utilisé l’outil de versioning GIT et plus particulièrement GITHUB.

Partie Back

Nous avons initialisé le projet à l’aide du site Spring Inializer ce qui nous a permis de générer très facilement l’architecture du projet.



L’architecture du projet est organisée selon le model « Data Acces Object » (DAO).

Les entités sont stockées dans un package entity, nous avons utilisé principalement trois entités.

Une entité Records qui correspond à l’objet que l’on va récupérer par la requête au fichier Json.

Une entité Fields qui correspond aux détails de l’objet Records.

Une entité ReponseParkingApiEntity qui va correspondre à la liste de tous les objets que nous récupérons via la consommation du fichier Json.

Nous avons ensuite créé un modèle de parking, pour définir les données que nous allons garder pour un parking.

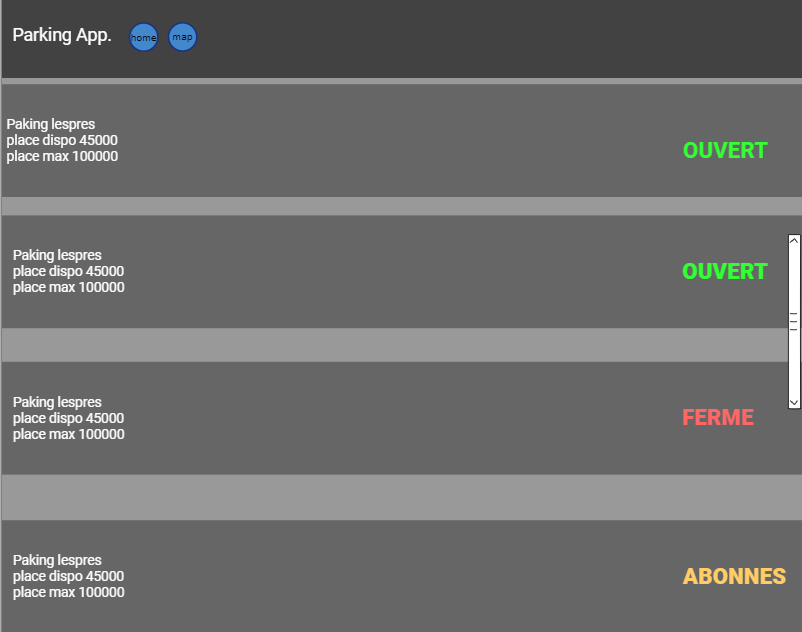
Par la suite nous avons créé un service parking comportant une fonction qui nous permet d’associer les données que nous recevons à un parking.

Parking que l’on ajoutera ensuite à une liste de parkings et que l’on retournera pour permettre un affichage et une lecture global des données.



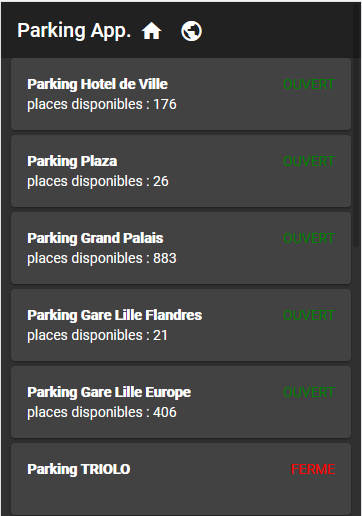
Partie Front

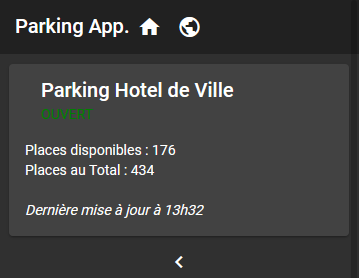
Premièrement, nous avons maquetté la page d’accueil de l’application à l’aide de Pencil.



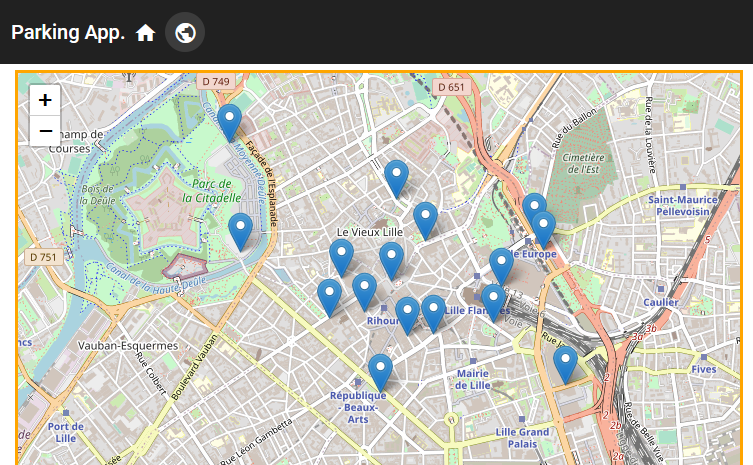
Par la suite, nous avons utilisé NodeJs pour initialiser le projet en lignes de commandes.

L’application se compose de trois composants principaux :

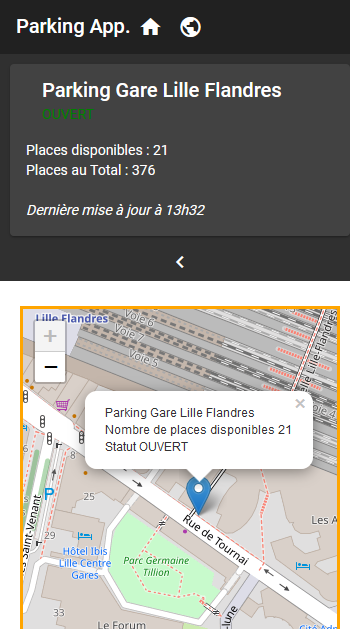
*  le composant parking qui nous permet de récupérer toutes les informations de la requête http effectuée à l’aide de notre application Spring et de gérer son affichage,
* le composant Parking-detail qui va nous permettre de gérer la vue d’un seul parking,



* le composant Map qui va nous permettre d’initialiser une carte comprenant la globalité des parkings, avec l’aide de la librairie Leaflet.



Pour le bon fonctionnement de l’application nous avons créé les services suivants :

* un service map-detail qui nous permet d’initialiser une map pour notre page Détail Parking,
* un service marker qui va nous permettre de créer à l’aide de deux fonctions, la globalité des markers de la map pour les parkings, ou le marker d’un seul parking,
* un service pop-up dans lequel nous allons retrouver la fonction de création des pop-ups des différents markers des parkings.

CONCLUSION

Ce projet a été pour nous un véritable défi, nous nous sommes confrontés à des technologies que nous ne connaissons pas, cela nous a permis de solidifier nos compétences et en apprendre des nouvelles.

Nous avons connu quelques imprévus :

* Le départ d’un membre de l’équipe de développement
* Un bug de l’open data
* L’intégration de Leaflet

Ces imprévus ont causé quelques dépassements dans les délais par rapport aux prévisions initiales

Toutes les fonctionnalités demandées par le « client » ont été développées, nous avons même réfléchis à d’éventuelles nouvelles fonctionnalités (par exemple une gestion d’itinérances) que nous pourrions développer par la suite.