

16 Avril 2018



Projet Génie Logiciel
-
**Application de calcul d'itinéraire
dans le métro parisien**

Sprint n°1
11 au 16 Avril 2018

Equipe :
Rodolphe Aubry, Laurene Cladt, Charlotte Fernandes, Benjamin Rath

Enseignant:
Olivier Perrin

Sommaire

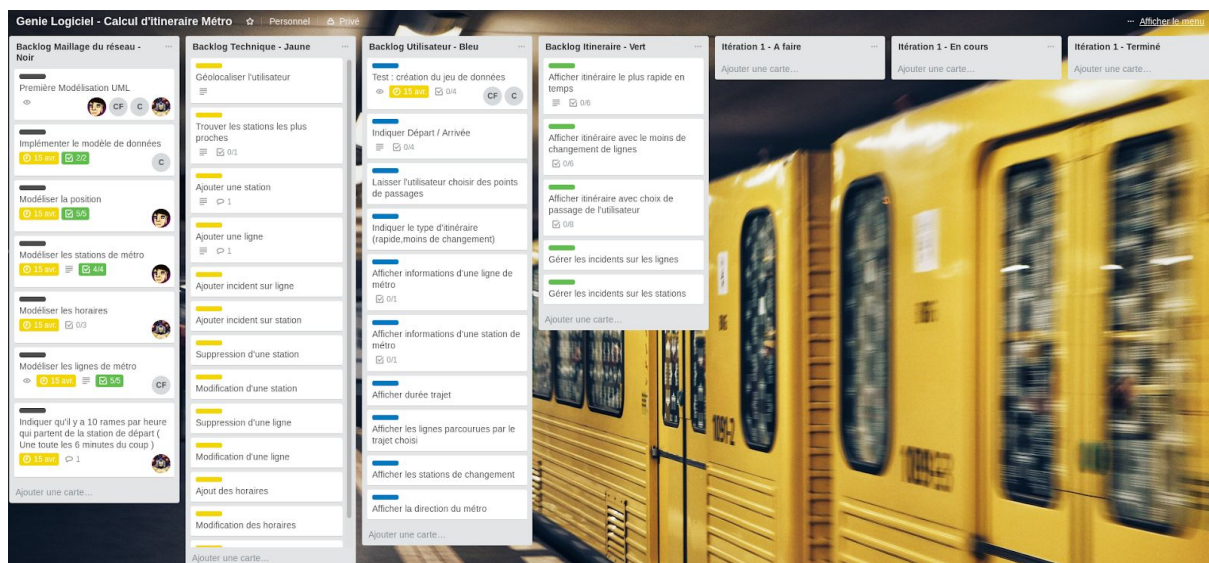
Sommaire	1
Réunion pre-sprint	2
Sprint	4
Burndown chart	6
Réunion post-sprint	7

Réunion pre-sprint

Notre projet a commencé le mercredi 11 avril 2018. Nous avons eu une réunion, après laquelle nous avons pu commencer notre premier sprint.

Notre première mission a été d'étudier le sujet que nous avons à traiter : créer une application permettant de calculer un itinéraire dans le métro parisien. Cela a permis à chacun de préparer la première réunion pendant laquelle nous allons tous présenter nos idées.

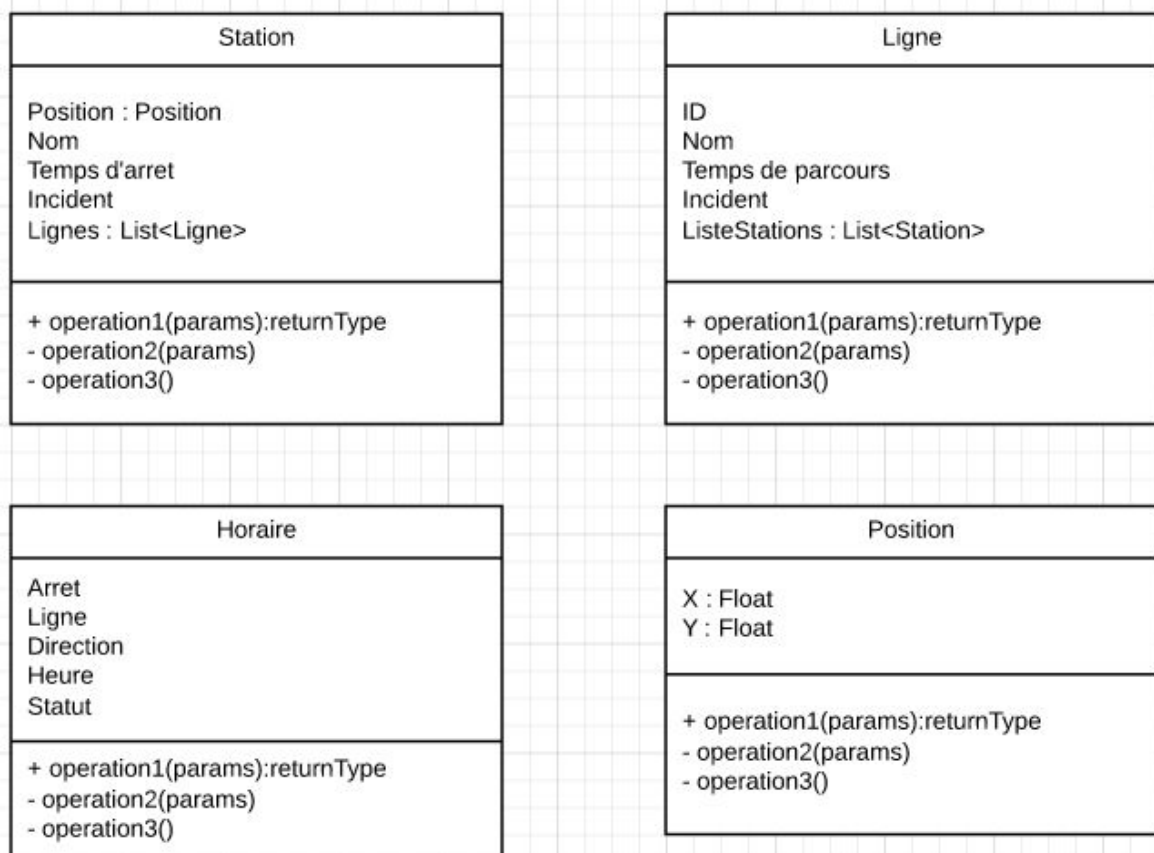
Lors de notre première réunion, le 11 avril 2018, nous avons commencé par un brainstorming pour récolter les idées de chacun. Après cela, nous avons mis en place un tableau Trello pour pouvoir établir nos backlogs.



Nous avons décidé de diviser nos tâches en plusieurs catégories

- Backlog Modélisation
 - Création du modèle UML, des classes de l'application et des tests.
- Backlog Technique
 - Géolocalisation, ajout d'une ligne/station, suppression d'une ligne/station, etc.
- Backlog Utilisateur
 - Affichage de l'itinéraire, propositions d'options, etc.
- Backlog Itinéraire
 - Itinéraire le plus rapide, sans incident, etc.

Durant cette réunion, nous avons décidé d'établir le modèle UML de notre application ensemble.



Cela fait, nous avons pu organiser notre premier sprint et nous répartir différentes tâches.

Sprint

Notre premier sprint avait pour objectif de remplir une mission simple : mettre en place la base de notre application. Nous avons décidé d'utiliser un modèle Modèle Vue Contrôleur.

Liste des tâches

Tâches	Estimation	Développeurs	Terminée	Temps réel
Implémenter modèle de données		Laurene	✓	
Modéliser la position	0.5j	Benjamin	✓	0.5j
Modéliser les stations de métro	0.5j	Benjamin	✓	0.25j
Modéliser les lignes de métro	0.5j	Charlotte	✓	0.25j
Modéliser les horaires	0.5j	Rodolphe	✓	0.25j
Création jeu de données	0.25j	Laurene	✓	0.5j

Implémentation modèle de données

L'objectif était de mettre en place le github et de créer la base MVC du projet.

Modéliser la position

Afin de pouvoir représenter l'emplacement de l'utilisateur et des stations de métro, il était nécessaire de créer une classe Position. Grâce à cette classe, nous avons pu placer les différents éléments nécessitant des coordonnées. Nous avons également délimité les coordonnées de Paris en utilisant des valeurs finales en longitude et

latitude. De cette façon il est possible de simuler une position aléatoire de l'utilisateur dans Paris et aussi de choisir des coordonnées pour les stations.

Modéliser les stations

La classe Station représente une station de métro de Paris. Les informations importantes sont son nom, son temps d'arrêt, s'il y a un incident ou non ainsi qu'une position. Nous avons également défini un temps minimum et un temps maximum d'arrêt. Il est possible d'indiquer un incident, de changer son nom, son temps d'arrêt mais aussi de changer sa position en indiquant la latitude et la longitude.

Modéliser les lignes de métro

Calculer un itinéraire de métro nécessite également des lignes de métro. Pour cela, nous avons une classe qui permet de créer une ligne et de lui attribuer un identifiant, un nom, un temps de parcours, un incident et une liste des stations qui sont présentes sur cette ligne.

Dans cette classe, trois méthodes ont été créées:

- trouverStation : elle permet à l'aide du nom d'une station de déterminer si elle se trouve ou non sur une ligne.
- getTerminus : elle permet de retourner la toute dernière station d'une ligne.
- getDepart : elle permet de retourner la première station d'une ligne.

Dans cette modélisation de lignes, plusieurs tests ont été développés pour déterminer si tout marchait comme il le fallait.

Modéliser les horaires

Dans le cadre de l'utilisation d'un métro pour un utilisateur, il est nécessaire d'en connaître les horaires afin de savoir quand la prochaine rame arrivera à un point donné.

La classe Horaire permet de créer les horaires des rames pour un arrêt donné, une ligne donnée et dans une certaine direction. On peut aussi y préciser le statut tel que "Annulé", "En service", etc.

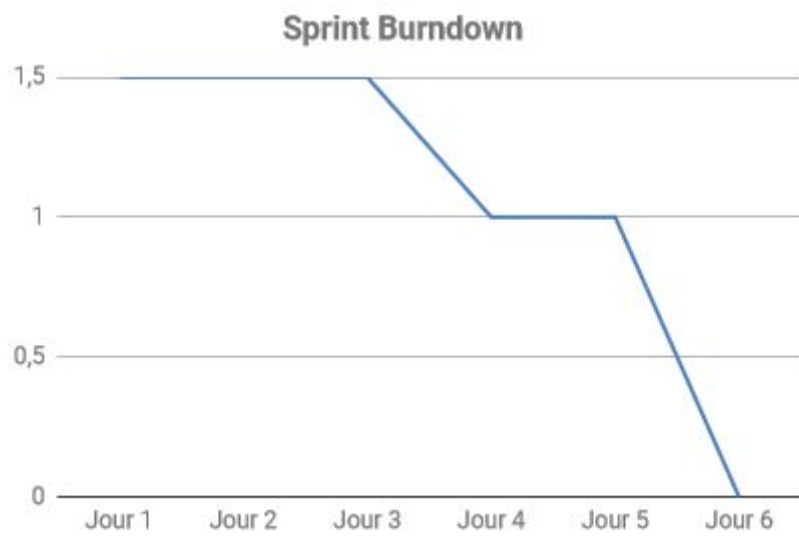
Différentes fonctions et méthodes ont été implémentées telles que :

- `getProchainPassage` : permet de savoir la date de prochain passage en fonction du nombre de minutes entre les rames et la date actuelle.
- `minutesEntreRames` : permet de savoir au moyen d'une constante (`NOMBRE_RAMES_HEURE`) le nombre de minutes qui passeront entre deux rames.

Création du jeu de données

La création du jeu de données consistait à modéliser les lignes et stations de métros existantes, afin de pouvoir tester les fonctionnalités de notre application. Nous avons choisis de modéliser les quatre premières lignes du métro parisien, ainsi que les stations associées et leurs positions respectives.

Burndown chart



Réunion post-sprint

Après avoir terminé notre première itération, nous avons discuté ensemble du résultat de notre application afin de vérifier que tout était conforme aux choix que nous avons fait lors de la première réunion. Il n'y a eu aucun problème, ni difficulté durant ce premier sprint. Nous avons ensuite discuté des fonctionnalités que nous allons implémenter lors de la seconde itération qui débutera le 17 avril 2017.

