24 Avril 2018

UNIVERSITÉ DE LORRAINE

Projet Génie Logiciel

-

Application de calcul d'itinéraire dans le métro parisien

Sprint n°2

17 au 24 Avril 2018

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

Enseignant:
Olivier Perrin

Sommaire

Sprint

Liste des tâches

Tâches	Estimation	Développeurs	Terminée	Temps réel
Sérialisation	0.25	Laurene	✔	0.5
Menu utilisateur	0.25	Laurene	✔	0.25
Afficher les informations d'une ligne et d'une station	0.25	Charlotte	✔	0.25
Ajouter et supprimer une ligne	0.5	Charlotte	✔	0.25
Refactoring	0.5	Charlotte	✔	0.25
Géolocaliser l'utilisateur	0.25	Benjamin	✔	0.25
Ajouter et supprimer une station	0.5	Benjamin	✔	0.25

Indiquer Départ / Arrivée	0.25	Benjamin	✔	0.5
Trouver les stations les plus proches	0.5	Rodolphe	✔	0.5
Afficher le temps de parcours entre 2 stations	1	Rodolphe	✘	

Sérialisation

La sérialisation permet de sauvegarder les données de test et de garantir la persistance des données de l'application en implémentant le chargement et la sauvegarde des données. Nous avions décidé de sérialiser chaque objet "Ligne" dans un fichier JSON, avant de nous rendre compte que cela nous causait quelques problèmes, notamment au niveau de la gestion des stations. Nous avons donc changé la méthode de sauvegarde en utilisant des HashMap plutôt que des ArrayList, et sérialisé les listes de stations et lignes dans leurs fichiers .txt respectifs. Les fichiers contenant les données de test ont été ajoutés au GitHub afin de pouvoir retrouver ces données en cas de besoin.

Menu utilisateur

Le menu utilisateur est l'interface d'utilisation de l'application. C'est par ce biais que l'utilisateur choisit les interactions qu'il désire effectuer. C'était une fonctionnalité facile à implémenter : il suffit de récupérer les choix de l'utilisateur et de définir les actions associées.

Afficher les informations d'une ligne et d'une station

A partir du menu, l'utilisateur aura la possibilité d'afficher les informations d'une ligne ou d'une station qu'il aura saisie. Pour écrire ces fonctions, nous avons créé les classes LigneController et StationController. Ces fonctions n'ont posé aucun problème, il suffit simplement de récupérer le nom de la ligne / station souhaitée, de vérifier qu'elle est présente dans la HashMap correspondante et d'afficher ses informations.

Ajouter et supprimer une ligne

L'administrateur aura la possibilité d'ajouter et de supprimer les lignes qu'il souhaite. L'ajout d'une ligne a été complexe à réaliser car il faut vérifier de nombreux éléments :

- Le nom de la ligne ne doit pas exister dans la HashMap comportant toutes les lignes pour éviter les doublons.
- Le temps de parcours d'une ligne doit être strictement supérieur à 0.
- La liste des stations doit contenir plus de deux stations. Il faut également vérifier que ces stations existent dans la HashMap comportant toutes les stations.

Pour supprimer une ligne existante, il suffit de vérifier que la ligne saisie existe et de la supprimer de la HashMap comportant toute les lignes.

Géolocaliser l'utilisateur

Afin de placer fictivement l'utilisateur lors de son utilisation de l'application, il nous a fallu initialiser sa position dans Paris, pour ce faire nous avons décidé de lui affecter une position aléatoire dans les coordonnées de Paris au début du lancement de l'application.

Ajouter ou supprimer une station

L'administrateur aura la possibilité d'ajouter et supprimer des stations dans l'application. L'application remplace la station si celle-ci existe déjà et vérifie que les coordonnées sont dans Paris pour l'ajouter. Pour la suppression l'application vérifie que la station existe.

Indiquer Départ / Arrivée

L'utilisateur doit pouvoir indiquer son départ et son arrivée, pour cela on lui affiche les deux stations les plus proches, mais celui-ci peut choisir n'importe quel station. Il doit cependant indiquer une station existante, différente entre le départ et l'arrivée, mais également qui n'ait pas d'incident.

Refactoring

Nous avons décidé de réécrire une partie de notre programme car il ne respectait pas le modèle MVC. Ce modèle a pour objectif d'organiser correctement un programme, or dans notre vue (Main) nous avions les fonctions concernant la récupération et la sauvegarde de nos fichiers lignes.txt et stations.txt.
Pour ce faire, nous avons mis toutes ces fonctions dans les contrôleurs correspondants. Notre code est donc mieux organisé. La vue ne comporte que de l'affichage, et nos contrôleurs les fonctions appelées par la vue et les fonctions faisant le lien entre le modèle et la vue.

Trouver les stations les plus proches

Pour un meilleur confort utilisateur, nous avons décidé d'afficher à l'utilisateur les 2 stations les plus proches de sa position actuelle. Les stations les plus proches sont à la fois forcément différentes et ne sont listées que celles n'ayant pas d'incident.

Afficher le temps de parcours entre 2 stations

Après avoir commencé à envisager l'implémentation d'une nouvelle classe afin de pallier à nos besoins, nous nous sommes rendus compte que le temps de parcours entre deux stations et plus globalement, la gestion des horaires, devraient être sujet à une discussion, nous avons donc décidé de réfléchir à cette tâche dans la prochaine itération et de la diviser en sous-tâches.

Burndown chart

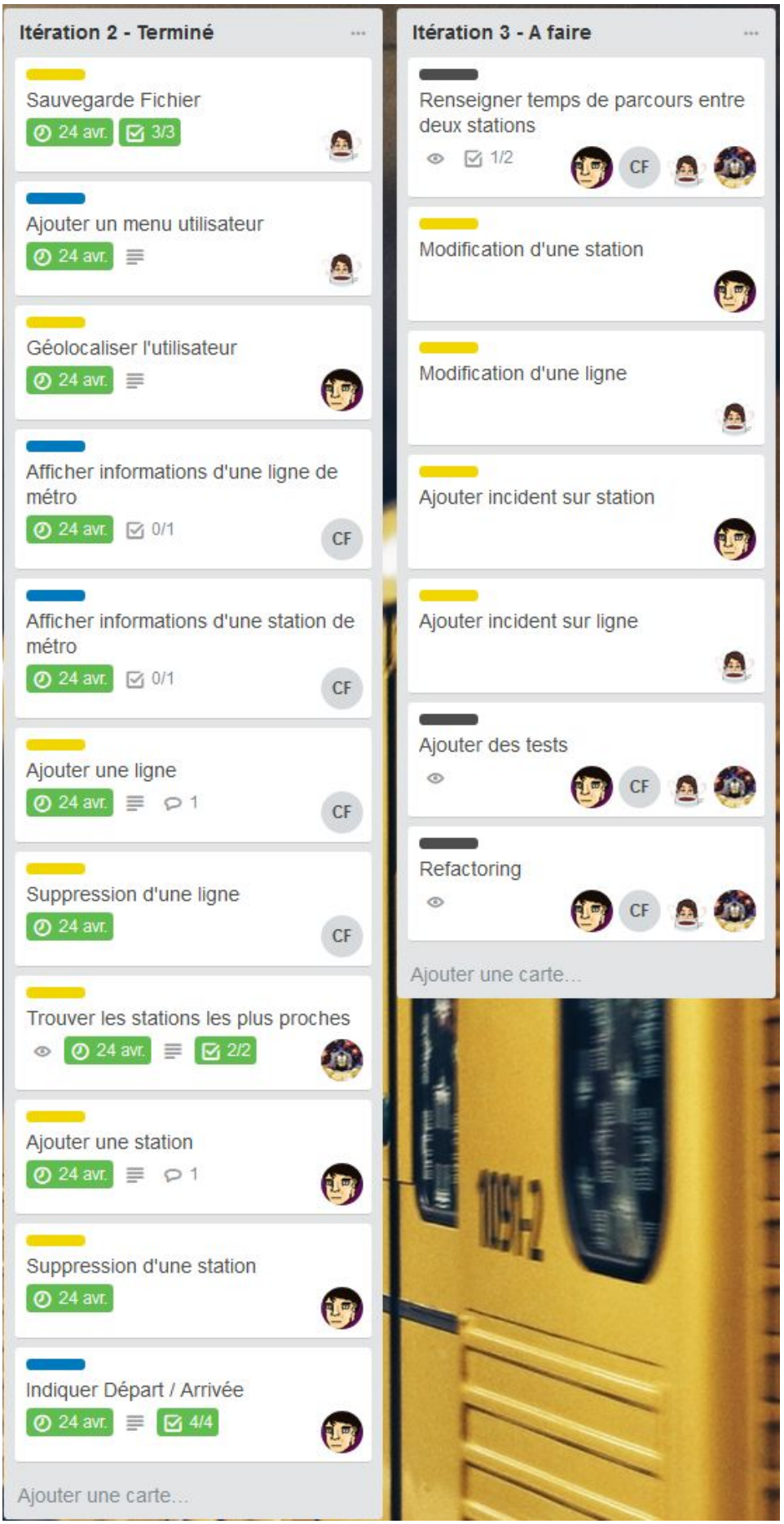
Légende :

Optimal

Actuel

Réunion post-sprint

A la fin de cette seconde itération, nous avons remarqué que la tâche “Afficher le temps de parcours entre 2 stations” était à revoir car la modélisation demandait plus de réflexion. Nous avons donc décidé de l’aborder durant la troisième itération. Nous avons également dû gérer le fait que deux personnes ne seront pas disponibles tout le long de cette itération. Nous avons alors décidé que la prochaine itération sera plus légère en terme de nouvelles fonctionnalités. Nous allons également ajouter plus de tests et revoir les classes déjà existantes.


Itération 2 - Terminé
Sauvegarde Fichier
24 avr.
3/3
Ajouter un menu utilisateur
24 avr.
Géolocaliser l'utilisateur
24 avr.
Afficher informations d'une ligne de métro
24 avr.
0/1
CF
Afficher informations d'une station de métro
24 avr.
0/1
CF
Ajouter une ligne
24 avr.
1
CF
Suppression d'une ligne
24 avr.
CF
Trouver les stations les plus proches
24 avr.
2/2
Ajouter une station
24 avr.
1
Suppression d'une station
24 avr.
Indiquer Départ / Arrivée
24 avr.
4/4
Ajouter une carte...
Itération 3 - A faire
Renseigner temps de parcours entre deux stations
1/2
CF
Modification d'une station
Modification d'une ligne
Ajouter incident sur station
Ajouter incident sur ligne
Ajouter des tests
CF
Refactoring
CF
Ajouter une carte...