**Collège de Rosemont**

**PROJET  
Le Cahier des Charges  
Sprint 3**

**Par  
Arielle Cissy Loé  
Maria Riabtchik  
Martin Bryson  
Martin Fortier  
Maxime Specogna-Roger  
Nicolas Jacques  
gr. 00802**

**Travail remis à M. Sylvain Labranche dans le cadre du cours  
DÉVELOPPEMENT DES APPLICATIONS INFORMATIQUES - PROJET  
420-B34-RO**

**Le 6 octobre 2022**

Table des matières

[Présentation du sprint 3 3](#_Toc115941011)

[Explication des fonctionnalités souhaitées 3](#_Toc115941012)

[Répartition des tâches à chacun des membres de l’équipe 4](#_Toc115941013)

[Product owner 4](#_Toc115941014)

[Scrum master 4](#_Toc115941015)

[Équipe de développement 4](#_Toc115941016)

[Backend 5](#_Toc115941017)

[Frontend 5](#_Toc115941018)

[Technologies choisies et utilisées 6](#_Toc115941019)

[Échéancier du sprint 7](#_Toc115941020)

[Quand seront les mêlées quotidiennes? 8](#_Toc115941021)

[Quand devront être implémentées les fonctionnalités? 8](#_Toc115941022)

# Présentation du sprint 3

## Explication des fonctionnalités souhaitées

* Les utilisateurs devront pouvoir acheter et vendre des actions;
  + L’information sur les actions disponibles du marché devra être reçue à travers d’un API qui fournira les informations en temps réel à notre application.
  + Dans la page du marché boursier, l’utilisateur pourra chercher les actions qu’il désire, voir leur prix et la quantité disponible, puis acheter les actions désirées, ou s’il en possède déjà, il pourra en vendre.
* Les utilisateurs devront pouvoir mettre de l’argent dans leur compte afin de pouvoir faire des transactions sur le marché boursier;
  + Afin de pouvoir acheter des actions du marché, l’utilisateur pourra verser de l’argent dans son compte. La vente des actions enverra donc l’argent dans le compte de l’utilisateur ou pourra l’utiliser pour acheter d’autres actions.
* Les utilisateurs devront pouvoir voir leur solde dans leurs informations personnelles dans l’onglet solde, ainsi que dans leur portfolio et dans le marché;
  + Afin de savoir exactement l’argent qu’ils ont de disponible pour leurs transactions, il est important qu’ils puissent voir l’évolution actif de leur solde peu importe où ils se trouvent dans le site. Afin de rendre le tout plus disponible, ils doivent y avoir accès quand ils ajoutent de l’argent dans leur compte (donc dans la page de leurs informations personnelles), lorsqu’ils achètent et vendent des actions (donc dans la page du marché), ainsi que lorsqu’ils regardent leur portfolio (donc dans la page du portfolio).
* Les utilisateurs devront pouvoir se connecter et se déconnecter sécuritairement;
  + Il y aura une sécurité solide pour empêcher l’accès aux gens non-autorisés à gérer le compte des utilisateurs. Un nom d’utilisateur et un mot de passe sécuritaire sont requis pour se connecter. À la page d’accueil on pourra procéder à l’inscription d’un nouvel utilisateur et mot de passe pour accéder au système.
* Les utilisateurs devront pouvoir supprimer leur compte si celui-ci n’a plus d’actions ni d’argent de disponibles;
  + L’option doit être disponible aux utilisateurs de pouvoir supprimer leur compte. Ceci dit, celui-ci ne doit pas avoir d’argent restant dans le solde ni d’actions dans le portfolio pour que celles-ci ne tombent pas dans le vide, et donc reste dans le marché boursier.
* Les utilisateurs devront pouvoir suivre l’évolution de leur compte, c’est-à-dire voir la variation des différentes actions se trouvant dans leur compte, afin de pouvoir faire des choix plus judicieux pour leurs transactions futures;
  + Dans la page portfolio, on veut afficher le meilleur titre et le pire titre du portefeuille de l’utilisateur.
  + On veut afficher une charte à secteurs (pie chart) représentant le pourcentage du portefeuille que représentent chaque lot d’actions par titre.
  + On veut afficher la liste des actions détenues par l’utilisateur.
  + On veut montrer le bilan des pertes et profits pour le portefeuille de l’utilisateur.

# Répartition des tâches à chacun des membres de l’équipe

## Product owner

Nicolas est toujours le *Product owner*. C’est un candidat idéal, parce qu’il veille à ce que tout notre cahier de charge soit impeccable depuis le début du projet.

Selon les principes du Scrum, Nicolas est responsable du produit et de l’élaboration du *Backlog*. Comme lors du sprint précédent, nous avons décidé de le faire en équipe.

Les communications avec le client sont aussi maintenues par Nicolas puis transmises à toute l’équipe à l’aide de notre groupe Discord.

## Scrum master

Arielle est la Scrum Master. Elle fait la gestion du projet et guide l’équipe dans le processus Scrum. Elle planifie aussi les réunions Scrum à chaque cours (les lundis et jeudis) et communique avec certains petits groupes ou individu pour s’assurer de l’avancement du projet.

Après le *backlog*, Arielle s’est chargée de la gestion du compte Jira. Bien que nous ayons mis certaines balises de temps en place, chacun de nous avions la responsabilité d’inscrire des échéanciers liés à leur tâche respective sur Jira. De cette manière, nous sommes en mesure d’évaluer le temps qui aura été nécessaire pour terminer chaque partie du projet.

La Scrum Master s’est assurée de suivre le déroulement du projet et de noter les problèmes qui ont été rencontrés. Cette fois-ci les problèmes survenus étaient surtout liés à l’incorporation de l’API et à l’assemblage des différentes parties de la page portfolio (graphique, tableau, données etc.)

## Équipe de développement

Pour ce sprint, nous avons encore divisé l’équipe en deux. Un groupe qui s’occupe du *Frontend* et l’autre du *Backend*. Cette manière de fonctionner nous permet de mieux subdiviser les tâches et d’ainsi mieux suivre l’évolution de notre application. Cette fois-ci, nous avons été heurtés à plusieurs contraintes. Entre autres, la communication et le respect des rôles de chacun. Sans toutefois être un problème, ces contraintes ne nous ont pas permis de respecter notre planification initiale. Nous avons tout de même réussi à implémenter la plupart de nos fonctionnalités et avons dû en laisser tomber quelques-unes.

### Backend

|  |  |
| --- | --- |
| Maria | Martin Fortier |
| Utiliser l’API pour récupérer les *quotes* | Support technique sur les tâches des autres |
| Méthode de recherche inversée | Testeur de qualité |

### Frontend

|  |
| --- |
| Martin Bryson |
| Créer et finaliser le Wireframe |
| Ajouter le graphique à la page portfolio |
| Coder et implémenter le graphique |

|  |
| --- |
| Arielle |
| Créer la page portfolio |
| Créer une page portfolio sans solde |
| Créer une barre de recherche dynamique |

|  |
| --- |
| Maxime |
| Ajouter le logo au header du template (signe de dollar vert) |
| Faire l’ajout et le retrait du solde sur la page utilisateur |

|  |
| --- |
| Nicolas |
| Créer un bouton pour supprimer un compte utilisateur |
| Faire les vérifications avant la suppression du compte |

Même si nous avions chacun nos tâches respectives, chacun de nous a contribuer au débogage et à l’optimisation de toutes les pages en équipe.

# Technologies choisies et utilisées

Le backend continue d’utiliser la technologie de PHP (MVC) pour la programmation, WAMPserver pour héberger notre serveur, MySQL pour la base de données, ainsi que GitHub pour les mises à jour du code et JIRA pour la gestion du projet. Le frontend continue d’utiliser HTML/CSS/Javascript avec le Framework Bootstrap. De plus, nous avons intégré un API qui va chercher en temps réel les informations des actions boursières. Sur la page portfolio, nous avons intégré une charte à secteurs (pie chart) de Google, afin d’avoir une vue rapide de notre portefeuille.

Nous avons continué d’utiliser la collection d’outils de Bootstrap (v5.0.2 et 5.2.1), qui contient des codes HTML et CSS, des éléments interactifs et une extension Javascript.

Étant donné que nous utilisons PHP, la plateforme de développement Web permettant de faire fonctionner ce genre de scripts est WampServer. Cet environnement comprend trois serveurs, un interpréteur de script et un administrateur phpMyAdmin (un gestionnaire de bases de données à travers MySQL). Nous avons aussi décidé d’implémenter les fonctions Session pour les utilisateurs en PHP.

Pour la soumission et le partage des versions de notre projet, nous utilisons Git et GitHub. Cet outil de collaboration est utile pour le travail à distance et en équipe.

La gestion du projet se fait à l’aide de Jira. Un système de suivi de bogues, de gestion des incidents et de gestion de projet. Cet outil est très utile pour gérer un projet d’après les principes Agile et Scrum.

* PHP (MVC)
  + C’est un langage avec lequel nous sommes familiers et il nous permet d’utiliser le modèle MVC facilement
* Bootstrap
  + Inclus HTML, CSS et Javascript
  + Les versions 5.02 (compatible avec MDB) et 5.2.1
    - <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
  + Librairie MDB
    - <https://mdbootstrap.com/>
* WAMPSERVER
  + Le serveur que nous utilisons pour stocker les données. Simple et facile à utiliser
* MySQL
  + Utilisé pour créer la base de données
* GitHub
* Imposé par le client : Jira
  + Utile pour la gestion du projet. Nous l’avons connecté avec GitHub et Figma
* API IEX CLOUD: <https://iexcloud.io/docs/api/>
* Googlepiechart <https://developers.google.com/chart/interactive/docs/gallery/piechart>

# Échéancier du sprint

Nous avons tenté de retrouver la même cadence qu’au premier Sprint, mais malheureusement, ça n’a pas été possible. La gestion du temps reste un défi pour tous les membres de l’équipe. En plus des stages de certains et de la charge de travail des autres cours, s’ajoute le niveau de connaissance qui varie d’un membre de l’équipe à l’autre.

**19 septembre**

Création du *Backlog*. Supervisé par Nicolas à l’aide de Jira. Les brouillons de Wireframe sont faits.

**20 septembre**

Précision des détails qu’on incorpore sur la page Portfolio. Modification du Wireframe.

**21 septembre**

La page Marché est refactorisée. Le graphique est terminé.

**25 septembre**

Le bouton « Supprimer le compte » est ajouté. Réglage de bogues.

**26 septembre**

Création de la page Portfolio. L’API fonctionne. Il faut l’intégrer au reste du programme. Demande d’extension pour la remise du projet.

**27 octobre**

Les fonctions « Ajout » et « retrait du solde » fonctionnent.

**2 octobre**

Tentative et abandon de l’idée d’une barre de recherche dynamique. Création de la méthode de recherche inversée. Réglage de bogues.

**3 octobre**

Démonstration de la barre de recherche inversée. Réglage de bogues.

**4 octobre**

Réglage de bogues relatif à l’affichage des informations provenant de l’API.

## Quand seront les mêlées quotidiennes?

Pour commencer le projet, nous avons décidé d’avoir des mêlées au minimum à chaque période de cours du projet, soit les lundis et jeudis à 8h00 afin de se tenir au courant (l’heure a été changée afin d’accommoder Maria qui est en stage). Nous avons aussi établi d’avoir une mêlée par écrit le samedi où les gens détiennent la journée au complet pour mettre à jour le tout, surtout pour palper le pouls. Ces mêlées sont celles établies de base, mais on s’est entendu que pour les sprints plus intensifs, ces mêlées peuvent devenir plus fréquentes, pour s’assurer du développement adéquat du projet. C’est au « *product owner*» de décider si une semaine a besoin de plus de mêlées ou non afin d’atteindre nos objectifs pour chaque sprint.

La raison principale pourquoi nous n’effectuons pas plus de mêlées quotidiennes est que la plupart d’entre nous avons des horaires qui ne concordent pas bien et donc c’est difficile de dédier des journées spécifiques outre les périodes de cours du projet. Par conséquent, nous travaillons quand nous pouvons et nous tenons au courant par écrit sur Discord, avec un sommaire obligatoire les samedis.

## Quand devront être implémentées les fonctionnalités?

Les fonctionnalités sont implémentées à chaque sprint, à la fin du sprint. On se donne une journée d’avance afin de s’assurer que l’implémentation se fait adéquatement, avec une journée pour pallier aux problèmes si jamais il y a des problèmes à l’implémentation de nos fonctionnalités (quand on merge les branches du projet).

