

**Équipe de développeurs :**

**El Bakkali Tamara Soufian, Konstantopoulos Alexios, Sanou Pap, Bernard Loïc, Haal Benoît, Nguyen Khanh, Manzer Ali, Piekarski Maciej**

**Client : Dallant Justin**

## **Planification du projet**

### **Itération 4 :**

Ce document récapitule tout ce qui a été réalisé lors de la quatrième et dernière itération de ce projet. Voici la liste de ce qui a été convenu durant la réunion avec le client le 27 avril :

- Histoire 8 : Afficher les statistiques en jours et heures, 3 points
- Histoire 6 : Permettre à un utilisateur de se retirer d'une collaboration, 3 points
- Histoire 7 : Importer et exporter un projet, 23 points
- Histoire 10 : Notifier l'utilisateur des tâches qui lui sont assignées et qui se terminent bientôt, lors de la connexion, 6 points
- Débogage de certaines fonctionnalités, 4 points
- Refactoring du code, 20 points

L'équipe s'est vraiment concentré sur l'amélioration globale du code et s'est assurée non seulement de la qualité des fonctionnalités nouvelles mais également des fonctionnalités précédentes afin de remettre un produit de qualité.

Ainsi vous pouvez lire ci-dessous la liste de toutes les fonctionnalités disponibles dans l'application :

#### **Itération 4 :**

- L'utilisateur a la possibilité d'exporter et importer ses projets
  - Le projet est sauvegardé dans un fichier compressé.
- L'utilisateur peut se retirer d'une collaboration
- Les statistiques sont affichées en jours
- L'utilisateur a une notification si une tâche assignée se termine bientôt

### Itération 3 :

- Un tab Calendrier :
  - L'utilisateur peut sélectionner un ou plusieurs de ses projets pour voir leurs tâches sur le calendrier.
  - L'utilisateur peut naviguer la vue par mois.
- Dans le tab Statistiques :
  - Une nouvelle statistique « durée initiale du projet » a été ajoutée.
  - La fonctionnalité de l'exportation du projet a été mis à jour.
- L'amélioration de l'affichage :
  - Les éléments de l'affichage ont été centralisés et s'ajustent correctement avec la taille de la fenêtre.
  - Une limite minimale a été appliqué à la taille de la fenêtre de l'application
- Collaboration de projet :
  - L'utilisateur peut maintenant ajouter un ou plusieurs collaborateurs à ses projets
  - Les projets sur lesquels il collabore sont visible dans l'accueil
  - L'utilisateur peut attribuer des tâches aux collaborateurs.

### Itération 2 :

- Un utilisateur peut maintenant créer et modifier des tâches pour ses projets, et il peut également sélectionner une date de début et/ou une date de fin. Les tâches peuvent être triées en fonction du temps
- Un nouveau tab affichant les statistiques des projets :
  - L'utilisateur peut voir le temps restant à chacun des projets ainsi que la durée totale de celui-ci.
  - Le nombre de tâches du projet ainsi qu'un diagramme circulaire qui montre les tâches terminées et celles qui sont toujours en cours.
  - Une option pour regarder les statistiques globales : voir le temps total cumulé de tous les projets et les tâches terminées/ouvertes...
- L'utilisateur peut exporter ces statistiques dans un fichier .csv
  - Celui-ci a le choix entre exporter les statistiques d'un projet spécifique ou bien de tous ses projets.

### Itération 1 :

- La possibilité de s'inscrire, de se connecter (avec l'implémentation d'une base de données dans le projet).
- De modifier les différents champs de l'utilisateur notamment :
  - Le nom de l'utilisateur
  - L'adresse mail de l'utilisateur
  - Le mot de passe de l'utilisateur
  - Le prénom et nom de famille de l'utilisateur

Dans la fenêtre de modification, il y a également un champ permettant de revérifier le nouveau mot de passe tapé, afin de valider qu'ils sont bel et bien identiques. Une fois que la modification a été validée, une fenêtre apparaît afin de demander à l'utilisateur une confirmation de ses nouvelles modifications.

- La création et modification de projets. Il est possible de modifier :
  - Le titre du projet
  - La description
  - La date de fin du projet

## **Motivation des choix pris lors de la conception :**

- Model-View-Controller (MVC) : Architecture classique pour la conception de projets visuels utilisant JavaFx. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet et améliore la clarté de l'architecture.
- SQLite : Base de données locales afin de stocker les utilisateurs ainsi que leurs projets. Les raisons principales sont que cette librairie est simple à comprendre et nous n'avons pas besoin de serveur distant.
- Singleton design pattern : Le singleton est un patron de conception qui permet de restreindre le nombre d'instanciations d'une classe, nous l'utilisons pour réduire les connexions à la base de données. Il permet aussi de réduire les ressources utilisées en instanciant un seul et unique objet utilisé à travers tout le code. Il est aussi possible, grâce au singleton, d'utiliser l'héritage car les méthodes ne sont plus globales ("static").
- Observer design pattern : Celui-ci permet d'établir un lien entre des objets qui dépendent d'un « sujet ». Ce sujet peut par la suite tous les notifier d'éventuels changements qu'il subit. Nous avons plusieurs composants d'interface différents qui dépendent des données du même modèle. Afin de limiter le couplage entre les différentes vues, tous les observateurs sont informés à partir du modèle qui leur passe l'objet qui a été modifié.
- Facade design pattern : Ce patron de conception nous permet de fusionner facilement différentes parties de notre programme en une façade qui elle délègue les tâches aux sous-systèmes respectifs. De cette manière, la complexité du modèle est cachée et tous les contrôleurs passent désormais par cette façade.

