UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

Rapport d'avancement

Projet DEGEL # 1.0

Conception d'un système informatique distribué
GIF600

Présenté à
M. Bernard Beaulieu
Pr Frédéric Mailhot

Présenté par l'équipe 2 :

BOLX2201	Xavier BOLDUC-MEILLEUR
DOSM2902	Mathieu DOSTIE
FUGE2701	Émile FUGULIN
GIRP2705	Philippe GIRARD
HIPT2501	Théo HIPAUT
LARJ2526	Julien LAROCHELLE
MARD1206	Donavan MARTIN

Sherbrooke – 7 juin 2018

1 Technique et gestion

1.1 Suivi de l'avancement

1.1.1 Backend

Projet de base La structure du projet *backend* a été choisie et le projet de base a été monté. Les frameworks utilisés tels que Hibernate, FlywayDB, Feign, Swagger et Spring ont été configurés.

Sécurité La première tentative d'authentification avec le CAS a été développée en utilisant la librairie Spring. Le service a été passé en HTTPS avec *Let's Encrypt* afin de protéger les informations.

Gitlab L'équipe utilise déjà l'intégration continue (CI) et le déploiement continu (CD) afin d'automatiser au maximum le roulement des tests et la mise à jour du serveur.

Docker Le projet roule entièrement dans Docker et peut être facilement déployé avec le *docker-compose.yml* sur toute machine possédant Docker.

1.1.2 Mobile

Application de base Une vue *calendrier* est fonctionnelle avec des données de test enregistrées dans le téléphone. Une vue *paramètres* possède des boutons de configuration.

Framework d'horaire L'équipe a eu plusieurs rencontres de conception pour décider si un *plugin* (et lequel) allait être utilisé ou non pour faire le calendrier.

Webview L'authentification avec le serveur CAS se déroulera via une *webview* directement dans l'application. L'application récupérera les données (tokens) directement dans la *webview*.

1.1.3 Gestion

GitLab La plateforme GitLab est abandonnée en ce qui concerne la gestion des tâches en raison des limitations de sa version gratuite.

JIRA La plateforme JIRA remplace GitLab pour la gestion des tâches. Elle offre exactement ce que l'équipe recherchait pour (entre autres) regrouper les tâches en catégories et en sprints.

Rapports GitLab demeure la plateforme de rédaction des rapports en LaTeX. Désormais, chaque personne peut écrire sa partie de rapport et la *pusher* dans GitLab sans avoir à compiler localement.

1.2 Écarts et révision de la planification

L'équipe fonctionne avec un barème de points (tableau 1-1). Le sprint 1 contient 53 points (104 à 141 heures). Plusieurs tâches reliées à Horarius ont dû être reportées, d'où la chute finale (figure 1-1).

TABLEAU 1-1 – Barème de points

Nombre de points	Durée (heures)
1	0,5 - 1
2	3 - 4
3	5 - 8
4	9 - 12
5	13+

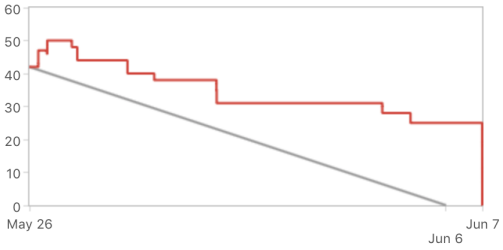

FIGURE 1-1 – Burndown Chart du sprint 1

1.3 Prévisions pour la prochaine itération

1.3.1 Backend

Authentification Posséder une méthode d'authentification avec OAuth2 fonctionnelle.

DIFF Être en mesure d'effectuer nous-mêmes un DIFF sur un fichier JSON enregistré localement.

API Concevoir un API pour recevoir les notifications de l'autre équipe.

1.3.2 Mobile

UI Corriger certains éléments incohérents de l'UI (titres dédoublés, barres de navigation).

Calendrier Produire la 2e itération du calendrier qui se connectera au serveur.

Authentification Commencer à intégrer l'authentification du *backend* dans l'application mobile.

Notification Développer des notifications de base sur l'application.

I18n Développer l'internationalisation sur l'application pour se faire approuver sur l'App Store.

1.3.3 Gestion

Pointage Peaufiner le pointage des tâches en ce qui a trait au temps requis pour chacune.

Git Établir des standards pour l'utilisation de Git.

1.4 Risques importants

1.4.1 Backend

L'accès au JSON d'Horarius s'avère plus complexe que prévu : l'iCal n'est pas présenté de la même manière, même si converti en JSON, ce qui nécessitera davantage de manipulation de données.

1.4.2 Mobile

L'offre de frameworks de calendriers en React Native avoisine la nullité et en concevoir un complètement s'avère tout un défi (*edge cases*, interface), donc on « étendra » le plus possible les *plugins*.

1.4.3 Gestion

L'équipe doit instaurer des normes à suivre sur Git. Chaque coéquipier possède sa propre expérience en entreprise de l'outil et définition de « bonnes pratiques », ce qui mène parfois à de petits débats.

2 Test et validation