

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 État du sprint

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 *Letsencrypt* afin de protéger les informations des utilisateurs.

Gitlab Une installation pleinement fonctionnelle de GitLab a été mise en place. 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.

1.2 Mobile

Application de base L'équipe a développé une itération fonctionnelle de l'application capable de démontrer la faisabilité de la plupart des fonctionnalités finales. D'abord une vue calendrier est fonctionnelle avec des données de test enregistrés dans le téléphone. Ensuite, une vue paramètre possède des boutons glissières et poussoir. En addition, il est possible d'alterner entre les deux vues à l'aide d'onglets au bas de l'écran.

Framework d'horaire L'équipe a discuté sur le choix du framework d'horaire, notamment sur le choix entre un framework *custom* ou non. S'il est *custom*, comment le faire ? Sinon, lequel choisir ?

Webview L'authentification avec le serveur CAS se déroulera via une webview directement dans l'application. L'équipe s'est donc penchée sur l'intégration et les interactions possibles avec ce qu'il y a derrière une webview quant à l'authentification (récupération de tokens, de cookies, etc.).

Réglages Une première page de réglages est développée sur l'application et prête pour la démonstration. Il s'agit encore d'une version préliminaire.

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 (une seule personne assignée, un seul réviseur, graphes manquants).

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 L^AT_EX. Désormais, chaque personne peut écrire sa partie de rapport et la pusher dans GitLab sans avoir à compiler localement. GitLab s’occupe de générer le PDF.

2 Problèmes rencontrés

2.1 Backend

Horarius L’accès au JSON d’horarius s’avère plus complexe que prévu. La version iCal n’est pas présentée de la même manière, même si convertie en JSON, ce qui nécessiterait davantage de manipulation de données.

2.2 Mobile

L’offre de frameworks de calendriers en React Native avoisine la nullité. Dans notre cas, il faut offrir une vue par heure, par jour, par semaine et peut-être par mois. Il n’est pas impossible de concevoir nous-mêmes le CSS du calendrier, mais c’est malgré tout un défi qui se comptera en dizaines d’heures de travail.

2.3 Gestion

L’équipe aura avantage à instaurer des normes à suivre sur Git. Se rapprochant de la fin du baccalauréat, chaque membre de l’équipe possède sa propre expérience en entreprise de l’outil et sa propre définition de « bonnes pratiques », ce qui mène parfois à de petits débats.

3 Aperçu de l’avenir

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.

3.2 Mobile

UI Corriger certains éléments incohérents de l’UI.

Calendrier Produire la 2^e itération du calendrier.

Backend Intégrer ce qui existera prochainement du backend.

Notification Développer les notifications sur l’application.

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.