

Alexandre Pominville (1392978)
Julien Gélneau Poirier (0766919)
Émile Tremblay (1647637)
Éric Marquis (9171475)
Pleshko Oleksandr Stepanovitch (1508300)

26 septembre 2022

Cahier de charges : Sprint 2
remis dans le cadre du cours

420-B34-RO – Développement des applications informatiques

Explication du projet et fonctionnalités souhaitées

Dans cette section, je vais détailler l'interprétation d'équipe quant à la définition des besoins énoncés par le client. Ce dernier nous a demandé de réaliser un logiciel qui permettra une validation informatisée des feuilles de temps des employés afin de s'assurer que celles-ci respectent bien les règles en place dans l'entreprise. Actuellement, chaque semaine, tous les employés doivent remplir une feuille de temps donc en faire la validation est une tâche de longue haleine.

Dans le premier sprint nous avons réalisé un prototype permettant de valider des fichiers json en ligne de commandes et d'obtenir des fichiers de sortie qui nous informe si les règles ont été respectées. Maintenant, notre client désire que notre application fournisse une interface graphique qui permet aux employés d'y entrer les informations de leur feuille de temps et d'obtenir les messages d'erreurs appropriés s'il déroge à une règle. À ceci s'ajoute une interface pour le gestionnaire où ce dernier pourrait consulter les feuilles de temps remises pour les approuver.

En plus des deux exigences fonctionnelles présentées dans le cahier de charges du sprint 1 s'ajoutent :

EF3: Les employés ont la possibilité d'avoir des congés de maladie, lors d'un congé de maladie l'employé doit charger 420 minutes dans le code de projet 999. Il ne peut employer ce code la fin de semaine et ne peut ajouter d'autres entrées de temps pour la journée. Les 420 minutes sont considérées comme du temps de présence au bureau.

EF4: Les employés vont pouvoir profiter de congés fériés. Lors d'un congé férié un employé doit charger 420 minutes dans le code 998. Contrairement au congé de maladie, lors d'un congé férié il est permis de faire du télétravail en plus durant la journée. Les 420 minutes sont considérées comme du temps de présence au bureau et il n'est pas permis de les utiliser la fin de semaine.

EF5: Le concept d'employé 'régulier' n'existe plus – ces derniers font partie de deux groupes distincts soit des employés de production (numéro est supérieur ou égal à 1000 mais inférieur à 2000) et des employés d'exploitation (Le numéro est supérieur ou égal à 2000). Les mêmes règles s'appliquent qu'aux employés réguliers pour ces deux groupes.

EF6: Les employés de l'administration doivent travailler au moins 36 heures au bureau par semaine (excluant le télétravail).

EF7: S'assurer que le temps d'un employé pour une journée ne dépasse pas 24 heures.

Explication du fonctionnement du logiciel

Afin d'utiliser notre application qui a été réalisée à l'aide du framework Spring Boot il faut une base de données Oracle 11 ou plus. Vous devrez ensuite modifier le fichier `application.yml` contenu dans : *Dossier_du_repo/projetfinal/src/main/resources* afin qu'il reflète la configuration de votre base de données.

```
spring:
  jpa:
    database-platform: org.hibernate.dialect.Oracle9Dialect
    hibernate:
      ddl-auto: create

  datasource:
    url: jdbc:oracle:thin:@//localhost:1539/EXERCICESPDB
    username: HR
    password: HR
    driver-class-name: oracle.jdbc.driver.OracleDriver

  logging:
    level:
      '[org.hibernate.SQL]': DEBUG
      '[org.hibernate.type.descriptor.sql.BasicBinder]': TRACE
```

La propriété `spring.jpa.hibernate.ddl-auto: create` permet la génération du modèle relationnel sur notre database à l'aide du framework hibernate contenu dans Spring Data JPA. Ce dernier nous permettra de faciliter la gestion des objets qui y seront persistés. Nous recommandons également d'installer le pack d'extensions VS Code suivant :

- Spring Boot Extension Pack (Spring Boot Tools, Spring Initializr Java Support, Spring Boot Dashboard)

Vous pourrez ensuite rouler le projet en cliquant sur Run dans le Spring Boot Dashboard:

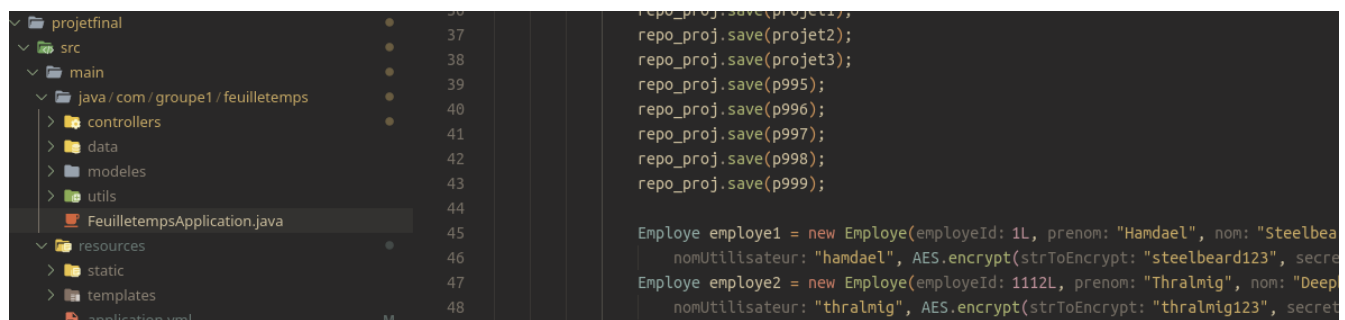


Après avoir démarré l'application ouvrez votre navigateur à la page <http://localhost:8080>

Cette dernière vous permettra de vous connecter à notre application soit en tant que gestionnaire :

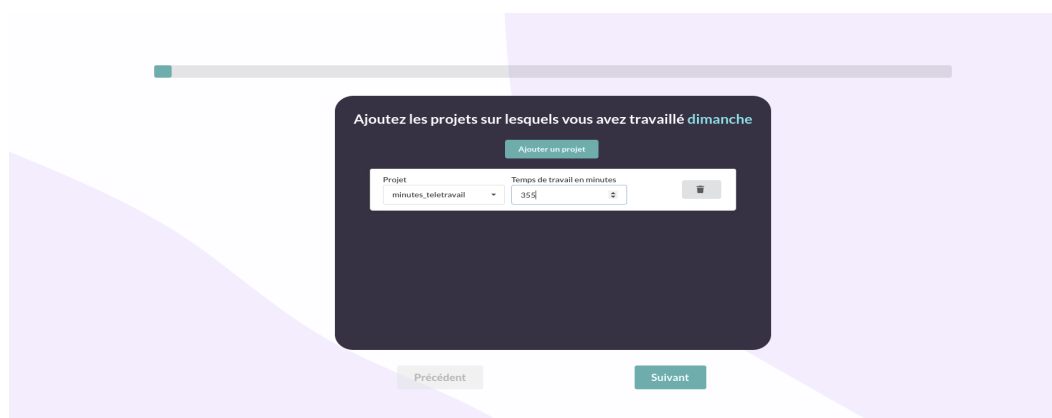
user: gestionnaire / password: gestionnaire (ce compte est hardcodé)

Ou à l'aide d'un des utilisateurs 'employés' de base générés dans le fichier contenant le main (FeuilletempsApplication.java)

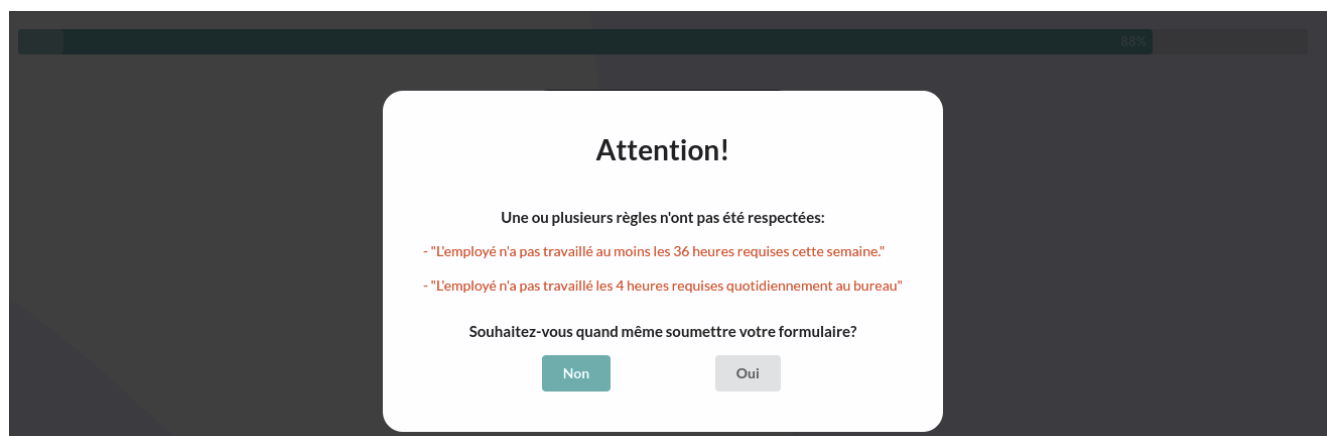


Page Employé

Lorsque vous vous connectez à la page employé l'application vous demandera d'entrer pour chaque journée de la dernière semaine (dimanche au samedi) le nombre d'heures travaillées pour chaque projet.



À la fin de la semaine on demande à l'employé de soumettre son formulaire et on lui affiche les erreurs si son formulaire ne respecte pas les règles de l'entreprise:



The image shows a dark-themed web interface with a central white dialog box. The dialog box has a title 'Attention!' in bold. Below the title, it states 'Une ou plusieurs règles n'ont pas été respectées:' followed by two red error messages: '- "L'employé n'a pas travaillé au moins les 36 heures requises cette semaine."' and '- "L'employé n'a pas travaillé les 4 heures requises quotidiennement au bureau"'. At the bottom of the dialog box, it asks 'Souhaitez-vous quand même soumettre votre formulaire?' with two buttons: 'Non' (highlighted in teal) and 'Oui' (grey).

Nous avons déterminé qu'un employé doit être capable de soumettre son formulaire malgré le fait que celui-ci ne respecte pas les règles. Ce choix a été fait pour éviter une trop grande rigidité du système dans le cas d'exceptions pouvant arriver dans la vie de tous les jours. S'il désire aller corriger les erreurs il peut cliquer sur non et le soumettre à nouveau. Dans le cas d'une soumission forcée, il sera indiqué dans l'objet feuille de temps persisté que cette feuille n'est pas validée par le système et elle sera affichée différemment dans l'interface du gestionnaire. Il est à noter que c'est l'approbation du gestionnaire qui rend la feuille de temps valide et non la validation, la validation elle lui permet d'utiliser les avantages d'un système informatique pour vérifier le respect des règles et ainsi l'approuver plus rapidement.

Page gestionnaire

Lorsqu'on se connecte en tant que gestionnaire nous avons une page qui nous permet de choisir la semaine pour laquelle nous désirons voir les feuilles de temps des employés. Ces dernières sont donc affichées en tenant compte de leur état de validation. Sur cette page, le gestionnaire peut ensuite les approuver ou envoyer à l'utilisateur un courriel lui demandant de remplir sa feuille de temps à nouveau dans le cas d'une feuille de temps non-validée.

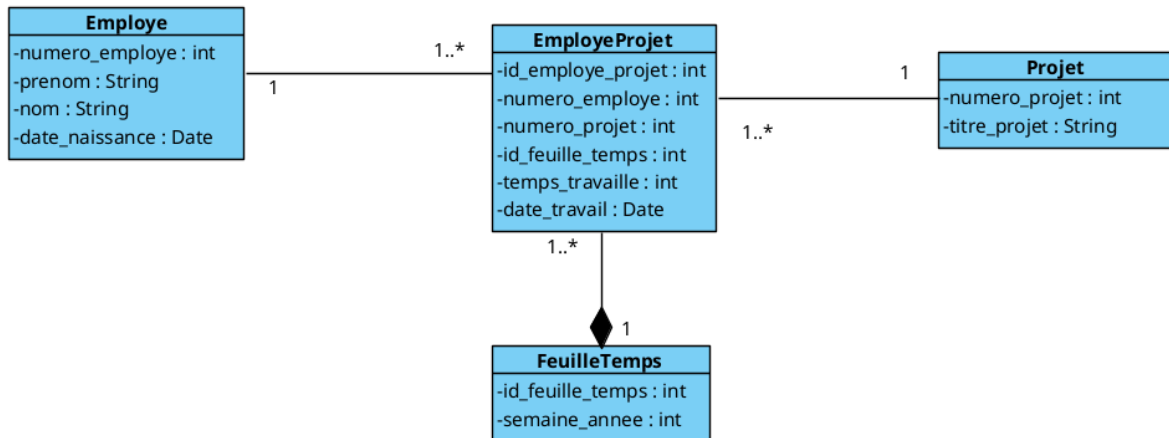
Tous	Approuvés	Non-approuvés	Non-envoyés	Semaine du 18 au 24 septembre 2022 ▾
Emile Tremblay ID: 1001 ✓ Formulaire approuvé	Eric Marquis ID: 1021 ! Formulaire à approuver	Julien Poirier ID: 1061 ! Formulaire à approuver	Alex Pom ID: 1051 *** En attente du formulaire	

Division du travail en équipe

La division du travail était plus facile cette fois-ci en raison de la charge de travail. Nous avons confié à Émile la conception des vues. Éric a été chargé de la refactorisation du module de traitement dans lequel sont validées les feuilles de temps. Il a également intégré les nouvelles exigences fonctionnelles demandées par le client. Julien et Alexandre se sont quant à eux davantage chargés des contrôleurs Spring et de la mise en place de la traduction du modèle relationnel en se divisant le travail chacun de leur côté. Oleksandr quant à lui sera en charge de tester le fonctionnement des règles lors de la validation et l’affichage des messages d’erreurs en conséquences (en raison de leur évolution et du refactoring que nous y avons fait).

Annexe

Diagramme de classe UML



Modèle de données

