



ÉCOLE NATIONALE
DES SCIENCES
GÉOGRAPHIQUES



RAPPORT PROJET JEE

31/01/2022

Théo Huard

Charles Laverdure

Plan de ce rapport :

- 1) Introduction / rappel des attentes
- 2) Description technique de l'outil
- 3) Guide d'installation et d'utilisation
- 4) Etat de l'outil (limites du travail)
- 5) Carnet de bord

1) Introduction / rappel des attentes

Ce rapport décrit dans le détail la réalisation du projet de JEE des TSI 2022 pour le groupe HUARD / LAVERDURE.

Pour rappel, le but de ce projet est de réaliser une application WEB en utilisant les technologies Java Enterprise Edition comme MVC, Spring, Hibenate ou encore SpringBoot.

Cette application est une application de gestion d'évènements dynamique. Elle doit permettre à un utilisateur / administrateur de gérer une liste d'évènements ainsi que les participants à ces évènements.

Pour les évènements, l'administrateur doit pouvoir créer, modifier ou supprimer un évènement et que cela soit pris en compte dans la base de données.

Liés à ces évènements, on trouve les participants. L'administrateur doit là aussi pouvoir créer, modifier ou supprimer des participants et voir le lien entre participant et évènement.

Nous allons aborder dans ce rapport comment nous avons mis en place cet outil. En passant par l'aspect technique de l'organisation et de la connexion entre eux des outils Java mais aussi comment le travail s'est organisé d'un point de vue pratique.

2) Description technique de l'outil

Dans cette partie, nous abordons l'organisation générale de l'application et comment nous avons choisi d'utiliser les outils imposés dans le cours. Pour rappel, les technologies qui nous étaient imposées dans le sujet étaient :

- **SPRING** et **HIBERNATE** pour la persistance des données.
- **SPRINGMVC** pour le contrôleur.
- **SPRINGBOOT** pour la configuration de Spring à (optionnel)
- **MAVEN** comme outil de Build.

Nous avons utilisé toutes les technologies citées ci-dessus et allons maintenant expliquer comment.

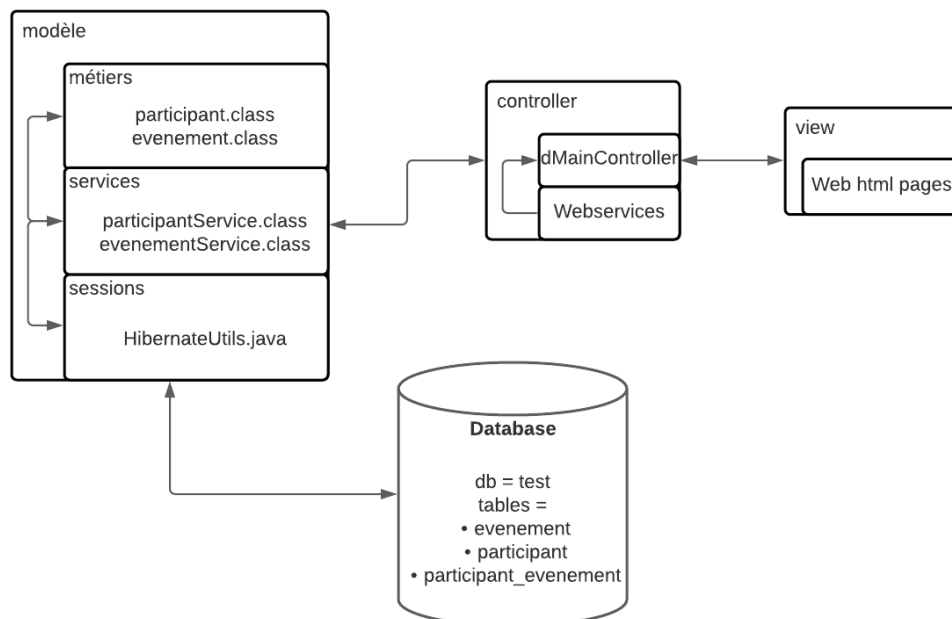
Notre application se sépare en trois parties (MVC) :

- La view : permet d'afficher la fenêtre graphique
- Le serveur : ou les données sont stockées
- Le contrôleur : permet la liaison entre l'interface graphique et le serveur

Le but est de pouvoir enregistrer et gérer une base de données à partir de pages WEB en passant par un contrôleur Java.

Pour la base de données, comme recommandé dans le TP4, nous utilisons PostgreSQL et avons choisi de télécharger pgAdmin afin de pouvoir visualiser cette même base.

Avant de rentrer dans le détail des technologies, voici un schéma de l'architecture du code afin de mieux se le représenter.



Pour la partie base de données en utilisant Spring et Hibernate pour la persistance des données, nous avons choisi de créer deux tables, une pour les participants et une autre pour répertorier les événements. Pour la liaison entre les deux, nous avons choisi de faire une relation bidirectionnelle manyToMany qui permet de lier un événement à plusieurs participants et aussi potentiellement un participant à plusieurs événements même si nous n'avons pas utilisé cette fonctionnalité dans l'application finale.

La création d'un événement se fait dans le fichier `evenement.java`. Celle d'un participant dans le fichier `participant.java`.

Le constructeur d'un événement prend en paramètres :

- num_enven
- Titre
- Theme
- Start_date
- Duree
- Max_participants
- Description
- Organisateur
- Type

Le constructeur d'un participant prend en paramètres :

- Num_pers
- Nom
- Prenom
- Email
- Date_naiss
- Organisation
- observations

Pour finir de décrire l'organisation de la BDD, la méthode manyToMany de hibernate permet de créer une table de liaison `participant_evenement` qui permet de lier les ids des événements avec ceux des participants qui assistent à l'événement.

Pour la partie MVC, la view contient les pages html qui permettent d'interagir avec l'application et dont le contenu est décrit dans la partie guide d'utilisation.

Notre Controller implémente des fonctionnalités permettant de réaliser toutes les opérations demandées dans le sujet en faisant le lien entre les services et les pages web.

3) Guide d'installation et d'utilisation

Installation

Pour l'installation de ce projet, un Readme.md est présent à la base de l'arborescence du projet. Il répertorie les différentes étapes afin de mettre en place et lancer le projet.

Utilisation

Une fois la fenêtre de démarrage ouverte, il est possible de gérer les événements et les participants. La première page affiche un tableau des différents événements créés par l'utilisateur. En haut de la page, l'utilisateur peut ajouter un nouvel événement.

Pour chaque événement des boutons sont situés à gauche du tableau :

- Supprimer : permet de supprimer l'événement dans la base de données
- Modifier : permet de changer les paramètres de l'événement
- Gérer les participants : ce bouton renvoie l'utilisateur sur la page d'administration des participants de l'événement choisi.

Une fois la page d'administration des participants chargée, l'utilisateur peut gérer les participants de l'événement choisi. En haut de la page, il peut ajouter un nouveau participant.

Pour chaque participant il peut :

- Supprimer : permet de supprimer le participant de la réunion
- Modifier : permet de modifier les paramètres du participant

4) Etat de l'outil (limites du travail)

Au moment du rendu (mercredi 02 février), nous avons pu mettre en place toutes les fonctionnalités demandées dans le sujet du projet sans avoir pu cependant réaliser les bonus.

Il persiste toutefois quelques problèmes au niveau de l'administration des participants. La suppression et la modification d'un participant n'ont pas été développées jusqu'au bout en raison des jointures entre les tables événements et participants qui ne permettent pas un accès total sur la base de données.

5) Carnet de bord

Pour l'organisation du travail, nous avons travaillé en binôme en utilisant les outils de partage de code de Github.

Nous étions à l'isolement durant la semaine de cours dû au Covid et avons donc pris du retard sur la réalisation des TPs. Nous avons donc commencé ce projet en finissant les TPs afin de pouvoir s'en servir.

Pour la répartition des tâches, Charles s'est occupé de la mise en place de hibernate, donc de la structure de base du projet avec la déclaration des classes participant et événement. Théo de son côté s'est occupé de la mise en place des liaisons MVC et du design de la View avec la création des pages Web.

Nous avons pu travailler en parallèle sans trop avoir à s'attendre sauf sur la fin du projet dans l'implémentation des dernières fonctionnalités du MVC notamment les fonctions de suppression et d'affiche des participants correspondants à un événement.

Nous avons pris du retard dans la rédaction du rapport et dans la mise en place des dernières fonctionnalités mais souhaitons rendre une application fonctionnelle même avec du retard plutôt qu'une application défaillante dans les temps.