

# L3 Informatique - Tours

Application d'entreprise

## Rapport projet spring

## Sommaire

<b>1 Initialisation</b>	<b>2</b>
<b>2 Structure</b>	<b>2</b>
2.1 Généralité	2
2.2 Explication	2
2.3 Composition finale	3
<b>3 Technologies</b>	<b>3</b>
<b>4 Organisation</b>	<b>4</b>
<b>5 Amélioration</b>	<b>4</b>
<b>6 Synthèse</b>	<b>4</b>
6.1 Andreas	4
6.2 Lou	5

# 1 Initialisation

Pour que l'application puisse démarrer il est nécessaire de configurer le fichier `application.properties`, par défaut la base de donnée est nommée "payaya", l'application peut créer elle-même la table. Lors de la première connexion, l'application vérifiera si la table des activités est vide, si c'est le cas, elle liera le fichier `le.json` fourni pour remplir la base de données. Par conséquent un fichier dump de la base de données n'est pas nécessaire néanmoins un sera fourni.

## 2 Structure

### 2.1 Généralité

La structure de l'application est la suivante:

ressources		java				
static	templates	utility	persistant	object	repository	controller

### 2.2 Explication

La partie ressources contient les fichiers html, css et javascript.

La partie static contient les fichiers CSS et javascript.

La partie templates contient les fichiers HTML.

La partie java contient tous les fichiers en java de l'application.

La partie utility contient les classes `ActivityReader`, `Common` et `Hasher`.

`ActivityReader` est utilisé pour lire le fichier json contenant toutes les activités si la base est vide.

La classe `Common` est appelée par tout le programme pour contrôler qu'un utilisateur est connecté.

La classe `Hasher` a pour objectif de chiffrer les mots de passe, en utilisant sha256.

Le package persistant a pour rôle de représenter les entités avec le lien dans la base de données.

Les entités sont les suivantes:

`Activities`: cette entité représente une activité, ayant un identifiant, une adresse, une description, une latitude, une longitude, un nom, une pathologie et un url.

Groupactivities: cette entité représenté un groupe d'activité, elle possède un identifiant, un nom et un utilisateur

Groupandactivities: cette entité représente la liaison n-n entre une activité et un groupe d'activité, car une activité peut être contenu dans plusieurs groupes et un groupe peut contenir plusieurs fois la même activité. Elle contient aussi la potentielle note que donnera l'utilisateur

User: cette entité représente un utilisateur, elle contient un pseudonyme, un mot de passe chiffré, l'âge, le genre et les pathologies du client.

Le package objet ne contient que la classe Activity, elle permet de faire des manipulations en front sans affecter directement l'entrée mère.

Le package repository contient les interfaces chargées de faire les requêtes sur les différentes tables de la base de données, chacune liée à une classe.

Le package controller contient les classes chargés de gérer le mapping du site, la classe ActivityController doit gérer les différentes actions liés aux activités, la classe GroupController celles liées aux groupes, la classe UserController s'occupe des traitements liés à l'utilisateur tandis que le general controller s'occupe de l'index et des erreurs.

Les groupes et les activités n'ont pas de notes propres, en revanche nous utilisons la moyenne des notes des GroupAndActivities pour calculer leur note.

Je tiens à signaler qu'il se peut que dans le dump il existe des résidus de base de données que Hibernate n'a pas supprimer telle qu'une table Book ou des attributs qui ne sont plus présents dans la base de données.

## 2.3 Composition finale

Le projet est composé de 68,4% de Java, 27,8% d'HTML, 2,3% de CSS et de 0,6% de Javascript.

## 3 Technologies

L'application a été faite en Java 19, avec le spring-initializer d'intliji, notre gestionnaire de projet est Maven.

Pour lire les fichiers JSON, nous avons décidé de prendre la bibliothèque org.json, car c'est celle utilisée nativement par Android Studio et que par conséquent nous l'avons déjà utilisée.

Pour la base de donnée nous sommes parties sur une base de donnée MySQL classique car c'était le plus simple à configurer, avec JPA pour lier à la base, Hibernate nous sert à générer automatiquement les tables aux besoins.

Pour relier le serveur à l'interface nous avons utilisé thymeleaf, qui permet de récupérer les variables passés dans le modèle, et renvoyer vers des URL paramétriques pour ne pas perdre des informations lors de certaines action tel que la création d'un groupe depuis une activité. De plus, thymeleaf nous permet aussi la gestion des fichiers HTML.

Pour la génération des cartes, nous avons utilisé leaflet car Andreas c'en était déjà servi et qu'il n'est pas nécessaire d'installer quoique ce soit.

## 4 Organisation

Pour se répartir le travail, il a été décidé d'appliquer un kanban, qui est une méthode agile via le site de Trello. Pour le versionning et la gestion du travail parallèle, github a été utilisé. Je dois aussi signaler que suite aux concours d'école d'ingénieur, aux stages et aux autres activités, le projet n'a pas été conçu en beaucoup de temps et la méthode agile n'as pas vraiment eu le temps d'être plus bénéfique qu'une méthode en V ou en cascade.

## 5 Amélioration

Dans les pistes d'améliorations il y aurait une refonte visuelle, nous avons décidés de prioriser la programmation serveur car c'était l'objet du cours, néanmoins il serait nécessaire dans le cas d'un projet pour une entreprise de mettre plus d'énergie dans la présentation, ce n'est pas un travail très difficile, le CSS et l'HTML nécessaire est à l'a porté d'un étudiant de première année mais cela prend du temps qui se faisait rare avec les concours d'école d'ingénieurs et les stages.

## 6 Synthèse

### 6.1 Andreas

Ce projet m'as permis d'être plus à l'aise avec le web de manière générale, en effet durant les autres cours je devais toujours apprendre un langage, et ses particularités en même temps qu'un framework et le principe de programmation serveur, cette fois-ci ayant déjà une bonne compréhension de java, j'ai pu me concentrer sur la programmation web serveur car l'apprentissage du framework spring a été rapide.

De plus, j'ai aussi appris à lier javascript, java et html avec aisance.

Enfin ceci est une remarque plus au niveau de la matière que du projet, mais les problèmes de configurations à répétition font que je suis maintenant habitué et que je ne suis plus autant perdu que ce soit avec IntelliJ ou avec Maven.

## 6.2 Lou

Ce projet m'a permis de mieux assimiler les concepts vus lors de ce semestre, notamment les servlets et les beans. De plus il m'a permis de me remémorer un peu de CSS. Cependant je n'ai pas été très présent dans le projet, l'essentiel du crédit revient donc à Andreas.

D'autre part, ce projet m'a permis de mieux cerner l'utilité de JEE qui me paraissait être une solution bien complexe comparée à d'autres méthodes de développement web. C'était une bonne amorce à mon stage qui utilise des technologies web, bien que très différentes (langage différent: JavaScript).