

Rapport de Projet Génie Logiciel

Gabriel Resende Hugo Bernard

13/12/2024

Master – Classe de TI

INTRODUCTION

Notre projet est un programme de gestion de projet en équipe (comme Notion ou Asana). Il a été écrit en Java et Typescript à l'aide des frameworks Spring et Angular respectivement. Notre base de données est SQLite.

POUR LANCER LE PROJET

Pour lancer notre projet, placez-vous à la racine du projet et lancez le serveur Spring, puis déplacez-vous sur le dossier “frontend” et composez la commande suivante

```
>>"ng serve"
```

BASE DE DONNÉES

La base de données que nous avons choisie est SQLite. Elle nous fournit de la simplicité grâce à son aspect “imbriqué”. Nous avons estimé que ce choix a été le plus judicieux pour un équilibre entre fonctionnalité et simplicité.

La base est installée directement à travers les dépendances Spring. Cela facilite la mise en œuvre du projet sans avoir à démarrer une session de base de données externe.

BACKEND

Spring est notre framework de choix. Il fournit des interfaces puissantes qui nous permettent d’implémenter un serveur http complet avec un système Logique Métier complet pour la création de Projets, Utilisateurs et Tâches de Projet.

La relation entre Utilisateurs et Projets est Many-to-Many. Cela veut dire qu’un utilisateur peut travailler sur plusieurs projets ainsi qu’un projet pourrait avoir plusieurs participants.

-Répartition de Responsabilités

Pour avoir un code plus modulaire et une scalabilité accrue, nous avons opté pour une division des responsabilités en différents composants.

Nous avons les “modèles” des objets qui représentent la structure des entités de notre application tels que les utilisateurs ou les projets.

Nous avons les “repos”, ce sont les objets responsables pour le stockage et interfaçage avec la base de données pour chaque type de modèle.

Nous avons les “services”, ils gèrent tout l’aspect “logique métier” et “règles de business” de l’application tel que comment ajouter un participant à un projet ou attribuer une tâche à un utilisateur.

FRONTEND

Notre frontend a été écrit en Angular. Ses interfaces puissantes nous permettent de mettre en vitesse un prototype fonctionnel de notre vision de l'application.

Nous avons trois routes principales. La landing page (page d'accueil), la login page (page d'authentification) et le dashboard (tableau de bord).

Pour chaque objet du backend, les services récupèrent les données avec une requête HTTP GET, pour les stocker dans les classes correspondantes et ensuite les afficher dans les composants Angular.

Il y a ensuite les composants "create" pour chaque objet qui servent à implémenter les pages de création de chaque objet, qui les envoient également au backend dans la base de données avec des requêtes HTTP POST.

De la même manière, les composants "update" mettent à jour les objets dans la base de données avec des requêtes HTTP PUT.

Pour chaque composant, il y a un fichier HTML qui permet de configurer et afficher les pages selon les informations des objets.