



Gestion des projets

membres de groupe

Boudefoua Lina

Jonathan Gayraud

Lancement du projet

Ce projet utilise Docker pour déployer l'application web.

Pour lancer l'application:

- Compiler le projet avec Maven: `mvn clean package`
- Créer un fichier **Dockerfile** à la racine du projet
- Construire l'image Docker: `docker build -t demo.`
- Lancer le conteneur Docker :`sudo docker run -p 8091:8080 demo.`
- Vérifier que l'application fonctionne: <http://localhost:8091>.

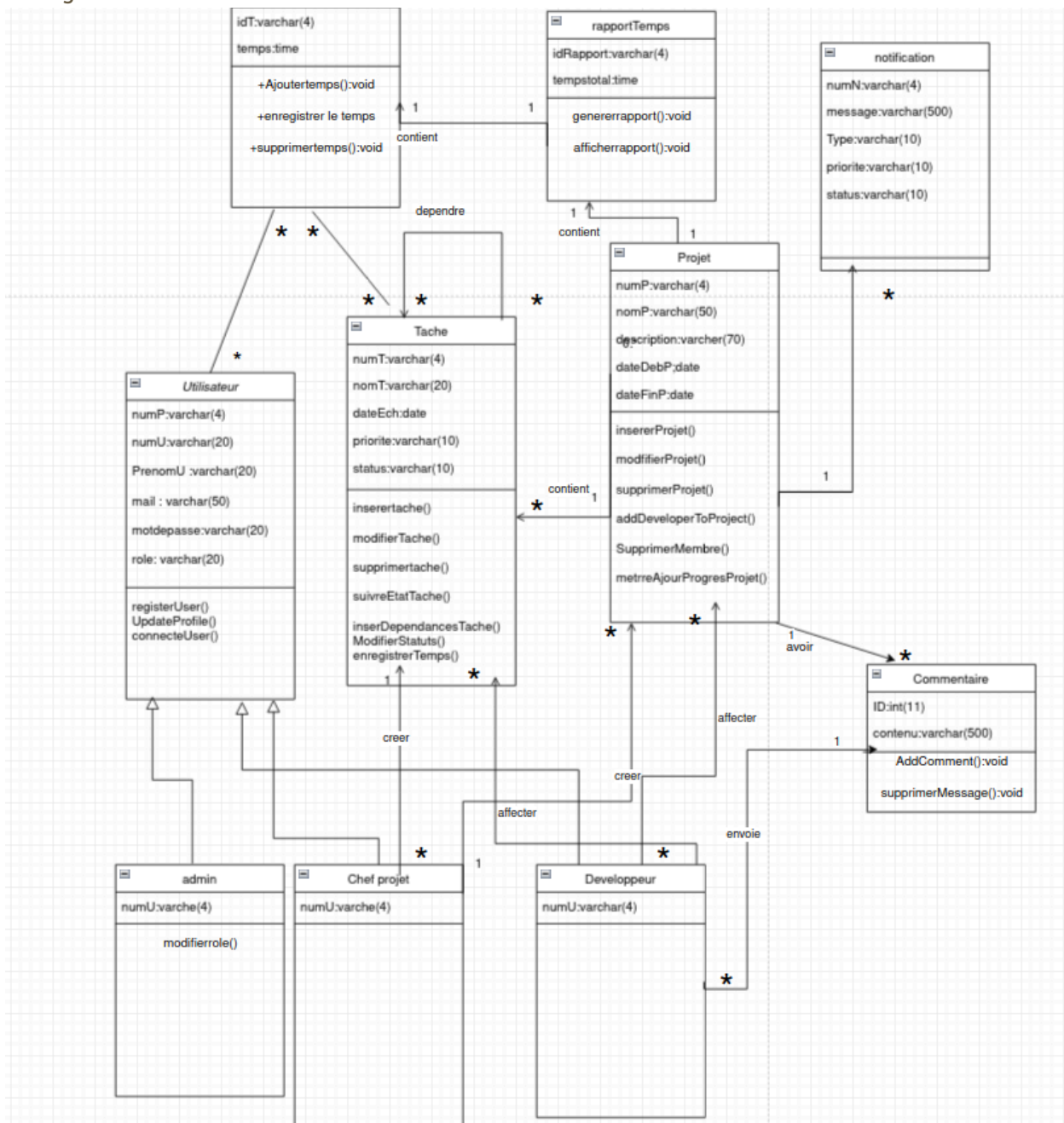
1-Collecte des Exigences

Document de spécification des exigences logicielles (SRS) : Rédiger un document détaillant les exigences fonctionnelles et non fonctionnelles.

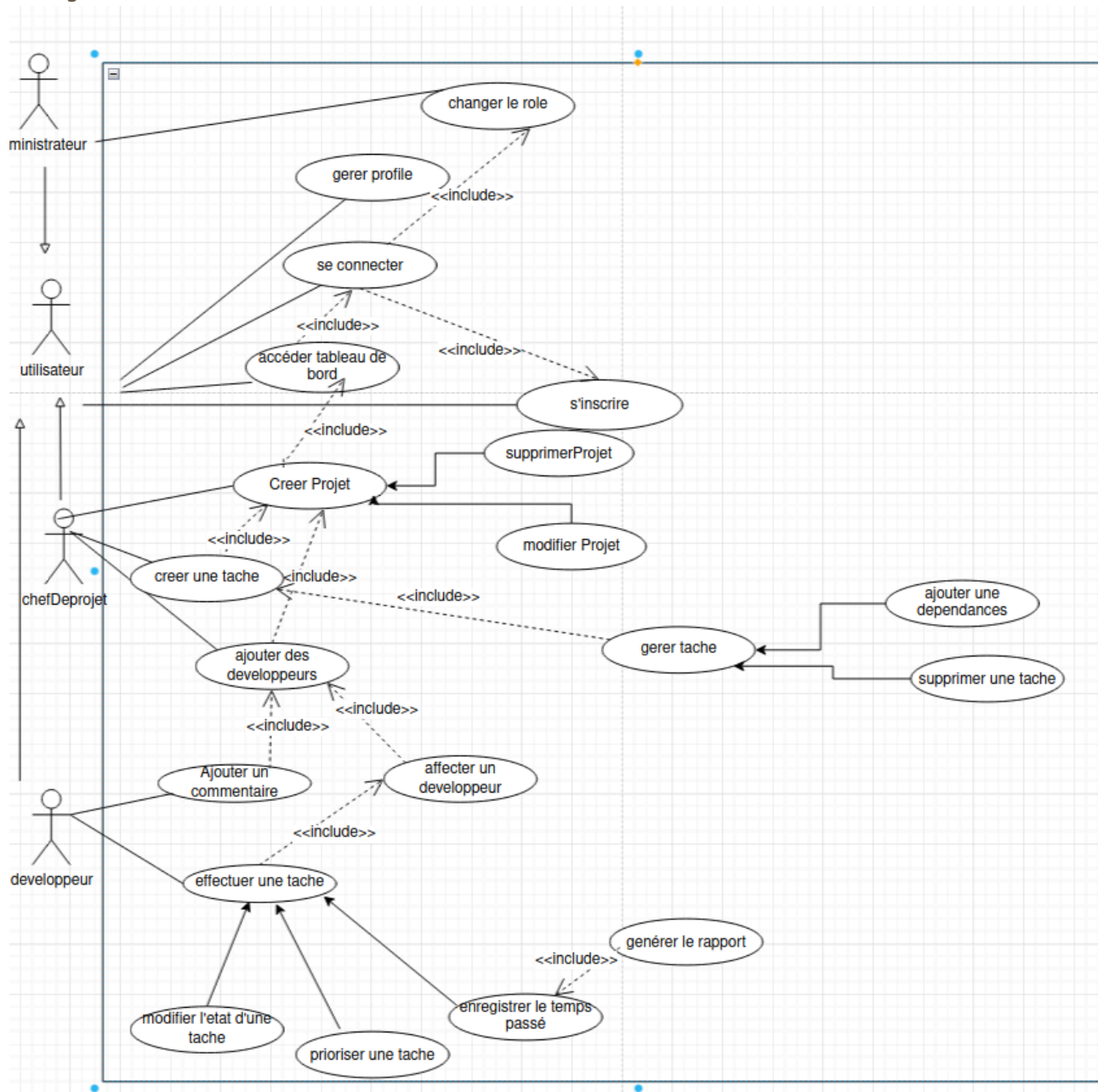
2-Outil de conception de l'application

2-1-Diagramme UML

1-Diagramme de Classe



2-Diagramme de Cas d'utilisation



3-Mise en oeuvre

L'application est principalement développée en java pour le backend et en Html/CSS/JavaScript pour le frontend.

Le frontend communique avec le backend via des **requêtes HTTP** (généralement avec la méthode POST)

L'application utilise également le framework Spring Boot qui permet de créer des applications prêtes à l'emploi facilement.

Chaque entité dans la base est implémentée avec une classe.

Ces classes contiennent les attributs de chaque table et sont généralement annotées avec les annotations de **JPA** (Java Persistence API) pour permettre une gestion efficace de la persistance dans la base de données.

Les **services** sont des classes qui contiennent la logique métier de l'application. Chaque entité a une classe **Service** correspondante qui est responsable de l'interaction avec la base de données .

Une classe Contrôleur pour appeler les méthodes de classe Service et faire la relation entre le backend et le front end en utilisant la méthode post ou get pour envoyer les données au contrôleur ,le contrôleur utilise les classes Services pour interagir avec la base et faire l'enregistrement ou la modification.

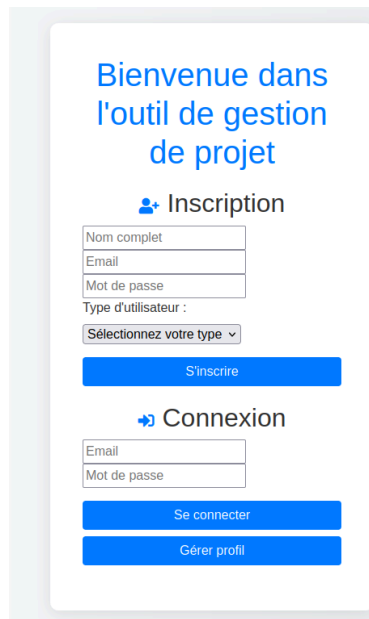
L'application utilise aussi une base de données en ligne phpMyAdmin qui permet de stocker toutes les données de l'application.

Le fichier pom.xml contient toutes les dépendances nécessaires au bon fonctionnement de l'application.

Le fichier application.properties contient toutes les configurations nécessaires au fonctionnement de l'application comme les informations concernant la base de données.

1-Page de connexion

La première fonctionnalité c'est l'inscription, elle permet à l'utilisateur de créer un compte. Une fois le compte de l'utilisateur créé, il peut se connecter à l'application ou gérer son profil.



The screenshot shows a web interface for a project management tool. At the top, it says "Bienvenue dans l'outil de gestion de projet" in blue. Below this, there are two main sections: "Inscription" (Registration) and "Connexion" (Login). The "Inscription" section includes input fields for "Nom complet", "Email", and "Mot de passe", a dropdown menu for "Type d'utilisateur" with the option "Sélectionnez votre type", and a blue "S'inscrire" button. The "Connexion" section includes input fields for "Email" and "Mot de passe", a blue "Se connecter" button, and a blue "Gérer profil" button.

2-Tableau de bord

Une fois l'utilisateur connecté, son dashboard s'affiche en fonction du rôle auquel il appartient.

Le dashboard du Manager permet à un manager d'effectuer plusieurs tâches tel que:

- Créer des projets
- Ajouter des développeurs au projet
- Afficher les développeurs du projet
- Créer des tâches pour chaque projet
- Créer des dépendances entre les tâches
- Afficher les tâches de chaque projet

Tableau de Bord Chef de Projet

Aperçu des Projets

projettest
sfsfsfsgsvs

Début: 2024-12-12

Fin: 2024-12-21

Progress: 5

Afficher les DéveloppeursAfficher les tâchesAjouter des utilisateurs

Ajouter un Projet

Liste des tâches

Ajouter une Tâche

Nom de la tâche	Priorité	Date d'échéance	Numéro utilisateur	Status	Numéro projet	
sgqvev	élevée	2024-12-19	projettest	testuser@gmail.com	unstarted	<div><div>Afficher Dépendances</div><div>Nom d'une tâche dépendante</div><div>Ajouter Dépendance</div><div><div>Éditer</div><div>Supprimer</div></div></div>

Le dashboard du Développeur permet à un développeur de suivre l'avancée du projet auquel il appartient. Il peut notamment:

- Voir les projet auquel il est affecté
- Ajouter des commentaire dans un projet
- Afficher les commentaire sur un projet
- Voir les tâche qui doit réaliser
- Modifier le statut d'une tâche

ID de l'utilisateur connecté :

projettest

Ajouter Commentaire

Afficher les commentaires

Commentaires

Aucun commentaire pour ce projet.

projetCap

Ajouter Commentaire

Afficher les commentaires

Commentaires

Aucun commentaire pour ce projet.

Aperçu des Projets

Tasks

Task Name	Priority	Status	Due Date	Project			
sgqvev	élevée	unstarted	2024-12-19	projettest	begin	Update Status	Heures

Enregistrer le temps

Le dashboard du l'administrateur permet modifier le rôle de l'utilisateur

Attribuer Rôle Manager

Adresse e-mail :

Rôle :

4-Test

Pour garantir le bon fonctionnement de l'application, on fait des tests Unitaires et d'Intégration pour chaque classe. Pour ce fait, on utilise Junit qui permet de réaliser ces tests.

5-Déploiement et DevOps

L'application est déployée localement en utilisant Docker.

pipeline CI/CD

Dans le workflow GitHub Actions , les tests et la création d'une **image Docker** à partir de fichier Dockerfile sont exécutés **lors d'un push sur la branche main**.

Crée un répertoire `.github/workflows` .

Crée un fichier `main.yml` dans le répertoire `.github/workflows/`.

```
name: Java CI with Maven

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      # Checkout le code
      - name: Checkout code
        uses: actions/checkout@v3

      # Installer JDK 17
      - name: Set up JDK 17
        uses: actions/setup-java@v3
        with:
          java-version: '17'
          distribution: 'temurin'

      # Installer Maven via APT
      - name: Install Maven
        run: |
          sudo apt update
          sudo apt install -y maven

      # Build et exécution des tests
      - name: Build and run tests with Maven
        run: mvn clean install

      - name: Build Docker image
        run: docker build -t demo .

      # Étape 4: Exécuter les tests avec Maven
      - name: Run tests with Maven
        run: mvn test # Exécuter les tests après la compilation

      # Étape 5: Deploy to server
      - name: Deploy to server
        run: |
          docker run -d -p 8091:8080 demo
```

