





**PROJET DE QUALITE**

**2024 / 2025**

**CONCEPTION ET RÉALISATION D’UNE APPLICATION WEB DE GESTION DES TACHES PERSONNELLE**

**Ingénierie Informatique et Réseaux**

**Réalisé par:**

**Mme.Asma EL OUALI**

# Liste des figures

[Figure 4:Diagramme de cas d'utilisation de l'application 6](#_Toc155989758)

[Figure 7:Diagramme de classe 7](#_Toc155989759)

[Figure 8: Architecture de spring boot 13](#_Toc155989760)

[Figure 9:Boite d'authentification 14](file:///C:\Users\acer\Desktop\PROJET%20DE%20STAGE%20D.docx#_Toc155989761)

[Figure 10:Register 15](#_Toc155989762)

[Figure 11:Page d’Accueil 16](#_Toc155989763)

[Figure 12:La liste des notes 17](#_Toc155989764)

[Figure 13:Les Notes avec bottons supprimer et modifier et consulter 18](#_Toc155989765)

[Figure 14: Les Notes avec bottons supprimer et modifier et consulter 19](#_Toc155989766)

[Figure 15:L’ajout d’un note 20](#_Toc155989767)

# Table des matières

[Liste des figures i](#_Toc155990039)

[Table des matières ii](#_Toc155990040)

[Chapitre 1 : 1](#_Toc155990041)

[Contexte général du projet 1](#_Toc155990042)

[1.1. Présentation du projet 2](#_Toc155990043)

[1.1.1. Définition du projet 2](#_Toc155990044)

[1.1.2. Problématique 2](#_Toc155990045)

[1.1.3. Objectif du projet 2](#_Toc155990046)

[1.1.4. Etude fonctionnelle 2](#_Toc155990047)

[1.1.5. Etude non fonctionnelle 3](#_Toc155990048)

[1.2. Conclusion 3](#_Toc155990049)

[Chapitre 2 : Analyse et Conception 4](#_Toc155990050)

[2.1 Identification des acteurs 5](#_Toc155990052)

[2.2 Diagramme de cas d’utilisation 5](#_Toc155990053)

[2.2.1 Introduction 5](#_Toc155990054)

[2.2.2 Digramme de cas d’utilisation du projet 5](#_Toc155990055)

[2.2.2.1 Diagramme de classe 7](#_Toc155990056)

[2.3 Conclusion 7](#_Toc155990057)

[Chapitre 3 : Outils et Technologies utilisés 8](#_Toc155990058)

[3.1 Introduction 9](#_Toc155990060)

[3.2 Outils a utiliser 9](#_Toc155990061)

[3.2.1 Wamp Server 9](#_Toc155990062)

[3.2.2 Visual Studio Code 9](#_Toc155990063)

[3.3 Framework 9](#_Toc155990064)

[3.3.1 React js 9](#_Toc155990065)

[3.3.2 Spring Boot 9](#_Toc155990066)

[3.4 Base de donnee 10](#_Toc155990067)

[3.4.1 Mysql 10](#_Toc155990068)

[3.5 Langage utilises 10](#_Toc155990069)

[3.5.1 Java 10](#_Toc155990070)

[3.6 Outil de test 10](#_Toc155990071)

[3.6.1 Postman 10](#_Toc155990072)

[3.6.2 Selenium 10](#_Toc155990073)

[3.6.3 JUnit 11](#_Toc155990074)

[3.7 Conclusion 11](#_Toc155990075)

[Chapitre 4 : Réalisation 12](#_Toc155990076)

[4.1 Introduction 13](#_Toc155990078)

[4.2 Architecture de projet 13](#_Toc155990079)

[4.2.1 Architecture de l’application 13](#_Toc155990080)

[4.3 Présentation des maquettes figma 14](#_Toc155990081)

[4.3.1 Page de l’authentification 14](#_Toc155990082)

[4.3.2 Page d’accueil 15](#_Toc155990083)

[4.3.3 Liste des notes 17](#_Toc155990084)

[4.3.4 La liste des Notes avec modifier et supprimer et ajouter 18](#_Toc155990085)

[4.3.5 Note avec button supprimer et modifier et back to nots 19](#_Toc155990086)

[4.3.6 Page d’ajout de note 20](#_Toc155990087)

|  |
| --- |
| Chapitre 1 : |
| Contexte général du projet |

## Présentation du projet

## Définition du projet

Le sujet est une application de gestion des taches personnelles qui permet au individu d’organiser, de suivre et de gérer ses propres taches, activités et engagements.

## Problématique

La problématique de ce projet consiste à développer une solution informatique efficace pour simplifier la gestion des taches dans la conception d’une interface utilisateur intuitive et efficace qui permettra aux utilisateurs de saisir, organiser et suivre leurs taches de manière simple et pratique.

## Objectif du projet

L'objectif principal d'une application de gestion des tâches personnelles est d'aider les individus à organiser, suivre et gérer efficacement leurs tâches quotidiennes, leurs projets et leurs engagements. Elle vise à améliorer la productivité, à réduire le stress lié à la surcharge de travail et à permettre une meilleure gestion du temps*,* de fournir aux utilisateurs un outil pratique pour optimiser leur organisation personnelle et professionnelle*.*

## Etude fonctionnelle

La personne individu va profiter de cette application pour organiser et gérer des taches efficacement qui permet la facilitée de saisie les informations qui concernent les taches personnelles et professionnelle:

* Création de liste des taches
* Consultation de la liste des taches
* Suppression d’une tache
* Modification d’une tache

## Etude non fonctionnelle

Les besoins non fonctionnels concernent les exigences à prendre en considération afin de mettre en place une solution adéquate aux besoins de l’individu.

* Disponibilité : L’application doit assurer le bon fonctionnement, et en cas de maintenance l’application reste accessible.
* Performance : L'application doit être réactive et offrir une expérience utilisateur fluide, même avec un grand nombre de tâches et d'utilisateurs

## Conclusion

Dans ce premier chapitre, j'ai effectué une présentation de l'application. Ensuite, j'ai mis l'accent sur la problématique ainsi que sur les objectifs à atteindre. Dans le chapitre suivant, je vais aborder la phase d’analyse des besoins et leur application à la conception.

|  |
| --- |
| Chapitre 2 : |
| Analyse et Conception |

## Identification des acteurs

Avant d’entamer la conception, il est primordial de définir les acteurs participant dans l’application. Pour le présent projet, un seul acteur est identifié :

**Utilisateur d’application:** est l’acteur principal. L’objectif de notre application est de gérer des taches personnelles et professionnelle.

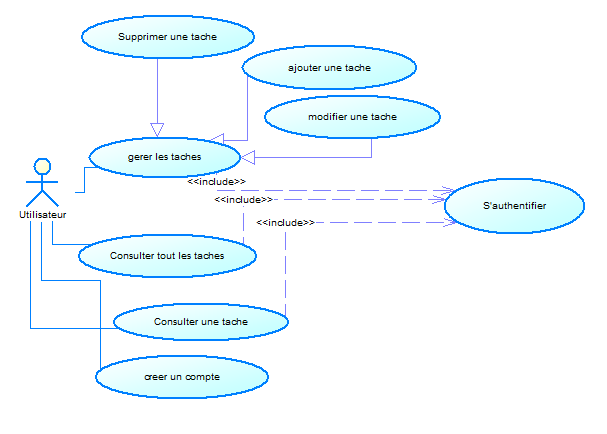
## Diagramme de cas d’utilisation

## Introduction

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

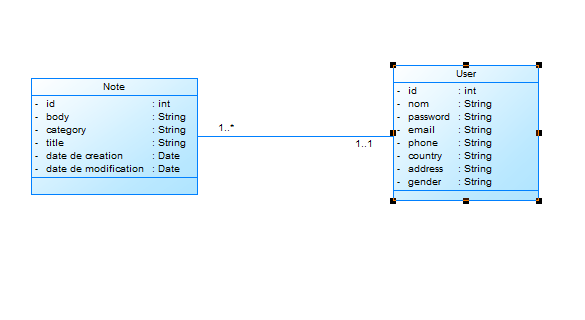
## Digramme de cas d’utilisation du projet

Le diagramme de cas d’utilisation ci-dessous illustre le rôle de chaque acteur dans notre Cas un seul acteur ainsi que l’étendue de ses responsabilités.



### Figure 1:Diagramme de cas d'utilisation de l'application

## Diagramme de classe



### Figure 2:Diagramme de classe

## Conclusion

Dans ce chapitre, j'ai présenté de manière globale les deux étapes essentielles du système élaboré pour l'analyse et la conception de mon application en suivant le processus de normalisation UML et les diagrammes, afin de faciliter la phase de réalisation. Le chapitre suivant sera consacré à la phase de développement de mon application.

|  |
| --- |
| Chapitre 3 : |
| Outils et Technologies utilisés |

## Introduction

Après avoir réalisé une conception qui répondait bien aux besoins de l'application, j'ai entamé la partie réalisation de l'application que je vais développer pour la gestion des taches en exposant les différents outils et langages de développement.

## Outils a utiliser

## Wamp Server

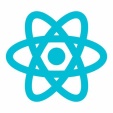
 Wamp est également connu sous le nom de Wamp Server, est un ensemble de logiciels open source qui permet de créer un environnement de développement web local sur un ordinateur Windows.

## Visual Studio Code

 Visual Studio Code est un éditeur de code source gratuit et open-source développé par Microsoft. Il est Conçu pour les développeurs de logiciels et offre un environnement de développement intégré (IDE) léger hautement personnalisable et extensible.

## Framework

## React js

 React Jsest un framework open source développé par Facebook qui permet de créer des applications web en utilisant JavaScript.

## Spring Boot

 SPRING BOOT est un framework open source basé sur le projet Spring de la fondation Apache. Il est spécialement conçu pour simplifier et accélérer le développement d'applications Java. Spring Boot facilite la création d'applications autonomes, prêtes à l'emploi, avec une configuration minimale et une prise en charge intégrée de nombreuses tâches courantes, telles que la gestion de la configuration, la gestion des dépendances, la création de microservices, et bien plus encore.

## Base de donnee

## Mysql

MYSQL est un système de gestion de bases de données relationnelle (SGBDR). Il est distribué sous une double licence GPL et propriétaire Il fait partie des logiciels de gestion de base de données les plus utilisés au monde.[9]



## Langage utilises

## Java

JAVA est un langage de programmation informatique de haut niveau, orienté objet et multiplateforme. Il a été développé par Sun Microsystems (maintenant une filiale d'Oracle Corporation) dans les années 1990. [9]

## Outil de test

## Postman

******POSTMAN est un logiciel permettant de créer et de tester des requêtes HTTP. Il est utilisé pour appeler et tester une API.[12]

## Selenium

 Sélénium est un framework de test automatisé gratuit (open source) utilisé pour valider les applications Web sur différents navigateurs et plates-formes

## JUnit

 JUnit est une bibliothèque Java qui facilite l’écriture et l’exécution des tests unitaires 1. Mockito est une bibliothèque Java qui permet de simuler des composants externes aux tests unitaires 23. En utilisant Mockito, vous pouvez simuler des objets qui ne sont pas encore implémentés ou qui ne sont pas encore disponibles pour les tests.

## Conclusion

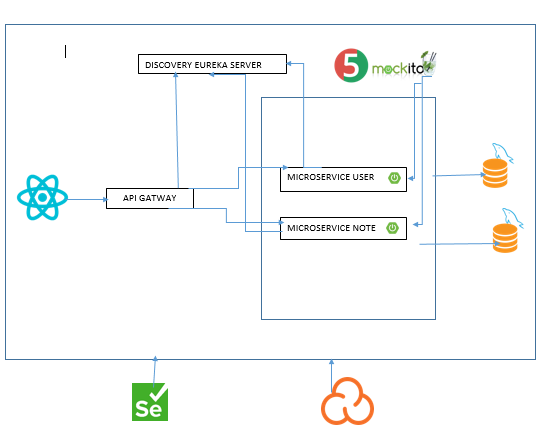
Dans ce chapitre, nous avons présenté l’environnement de développement ainsi que les différents outils et technologies a utiliser. Le chapitre suivant sera réservé à la présentation du produit final sous forme des maquettes figma.

|  |
| --- |
| Chapitre 4 : |
| Réalisation |

## Introduction

## Architecture de projet

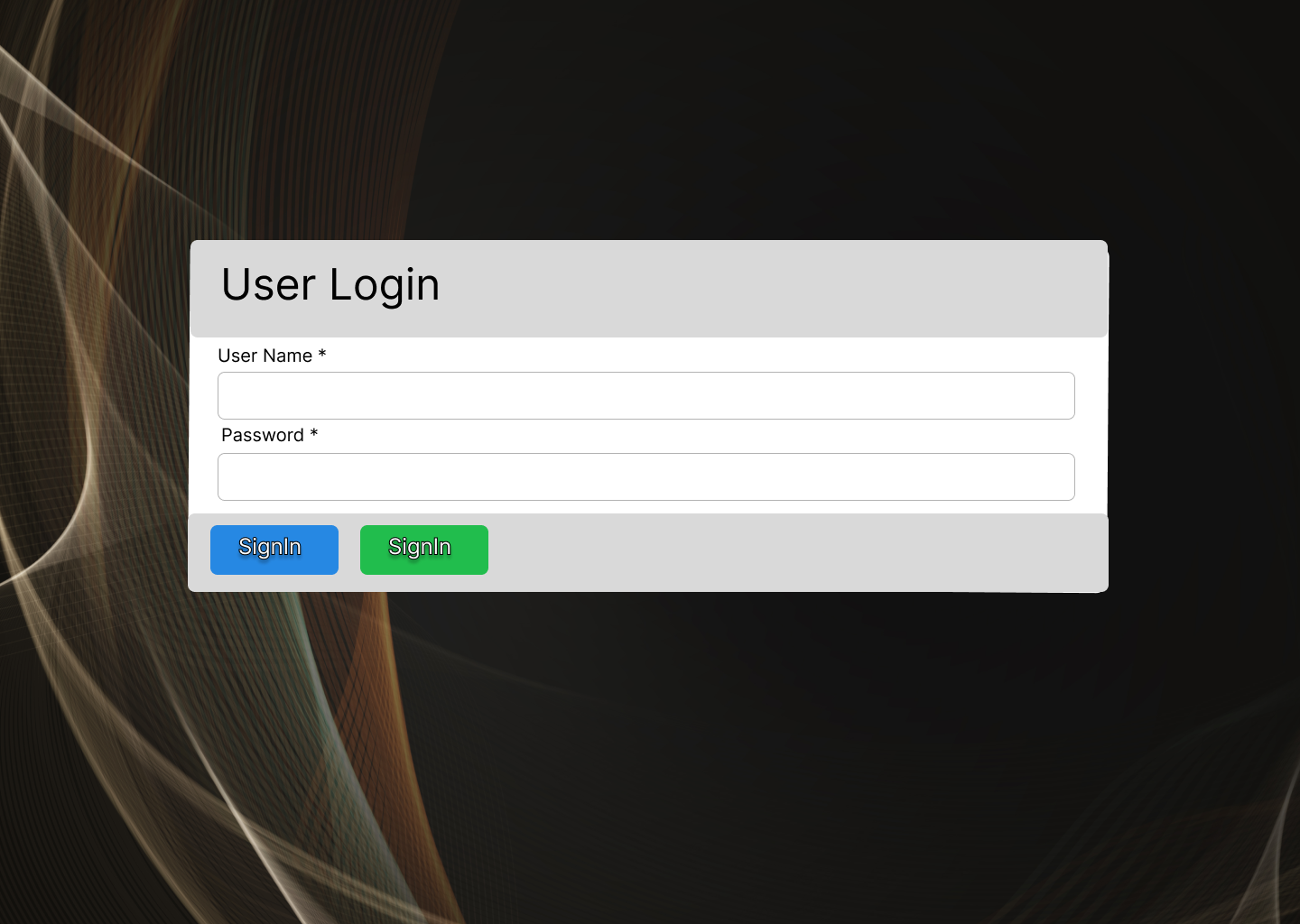
## Architecture de l’application



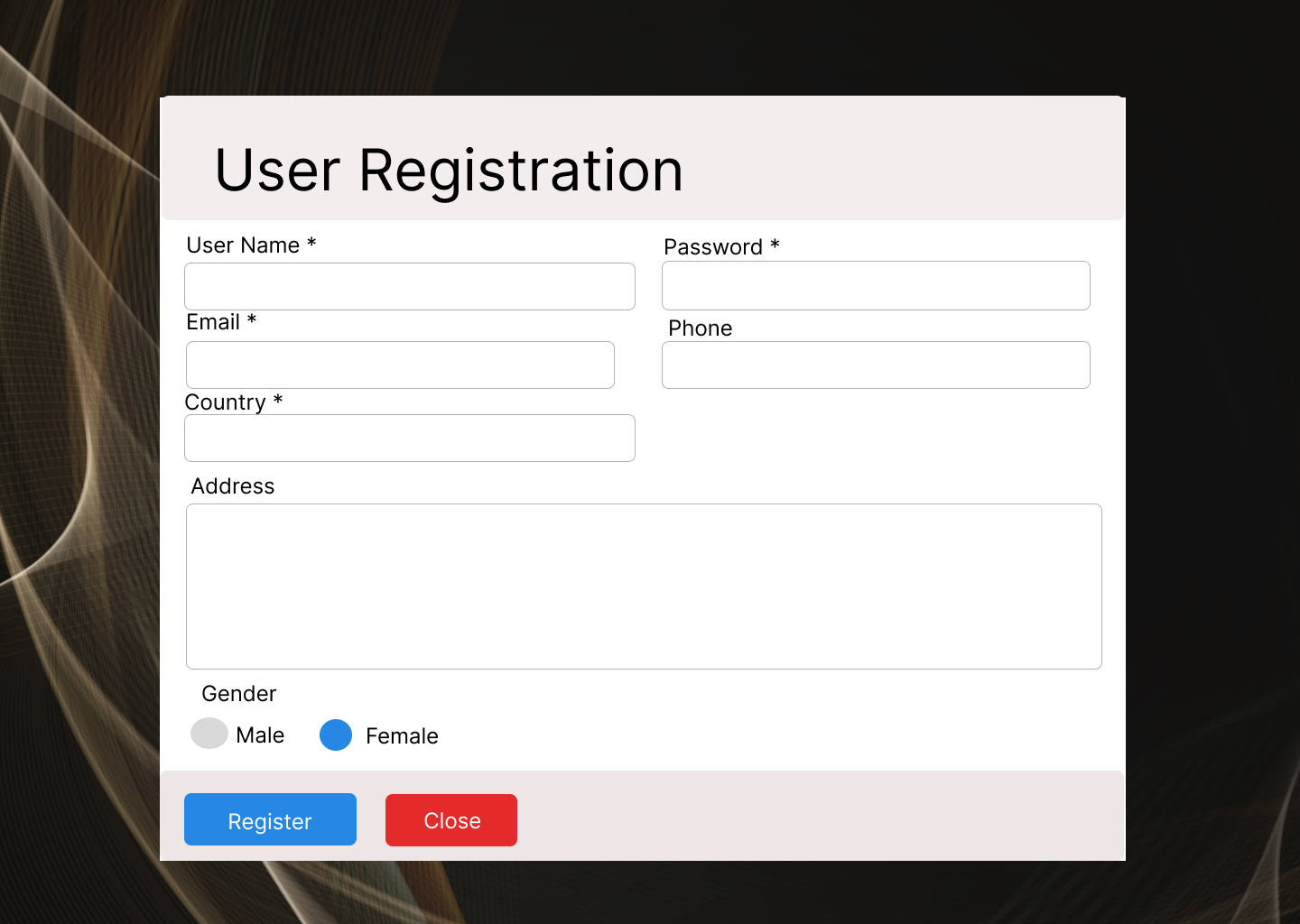
### Figure 3: Architecture de spring boot

## Présentation des maquettes figma

## Page de l’authentification

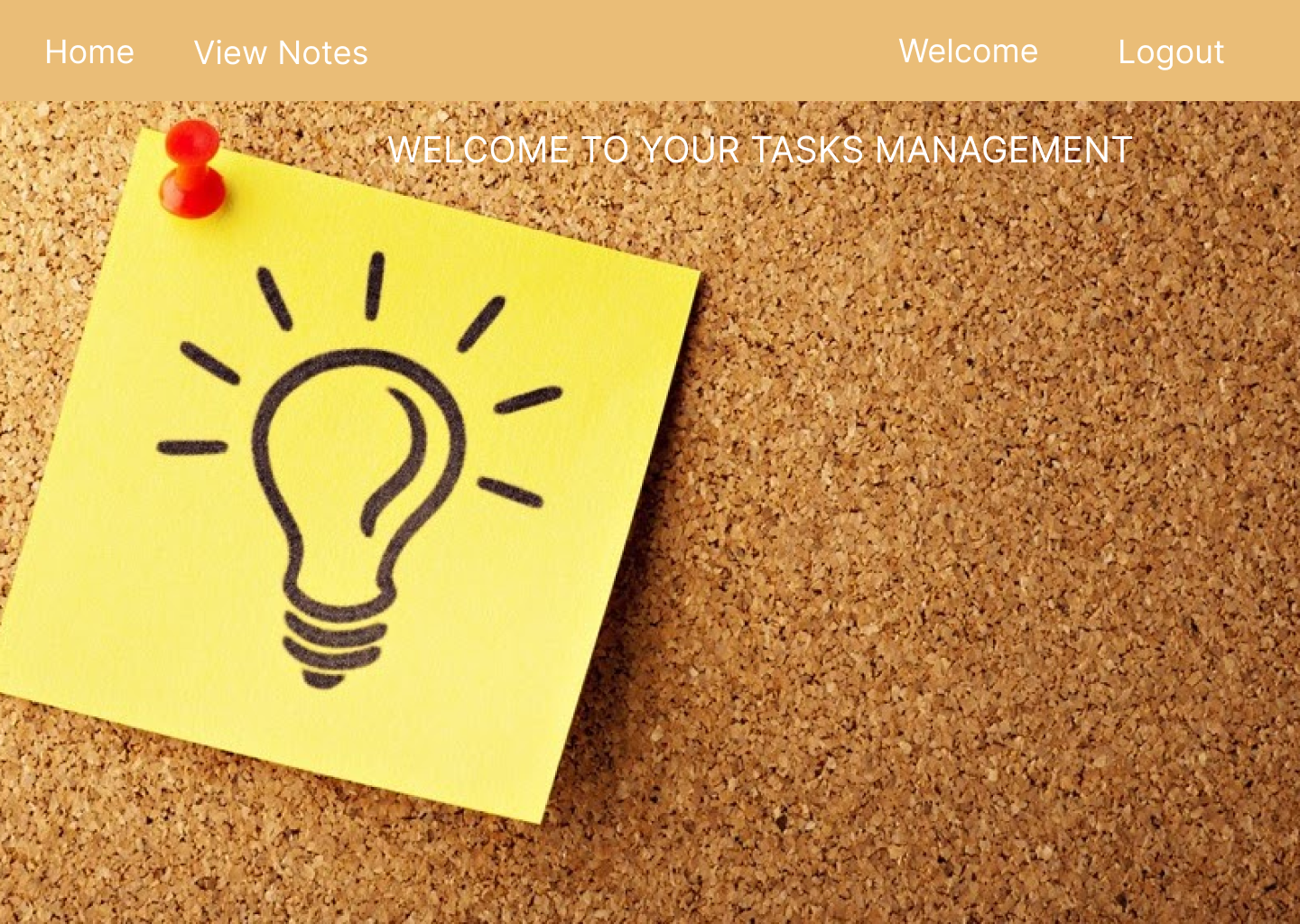


### Figure :Boite d'authentification



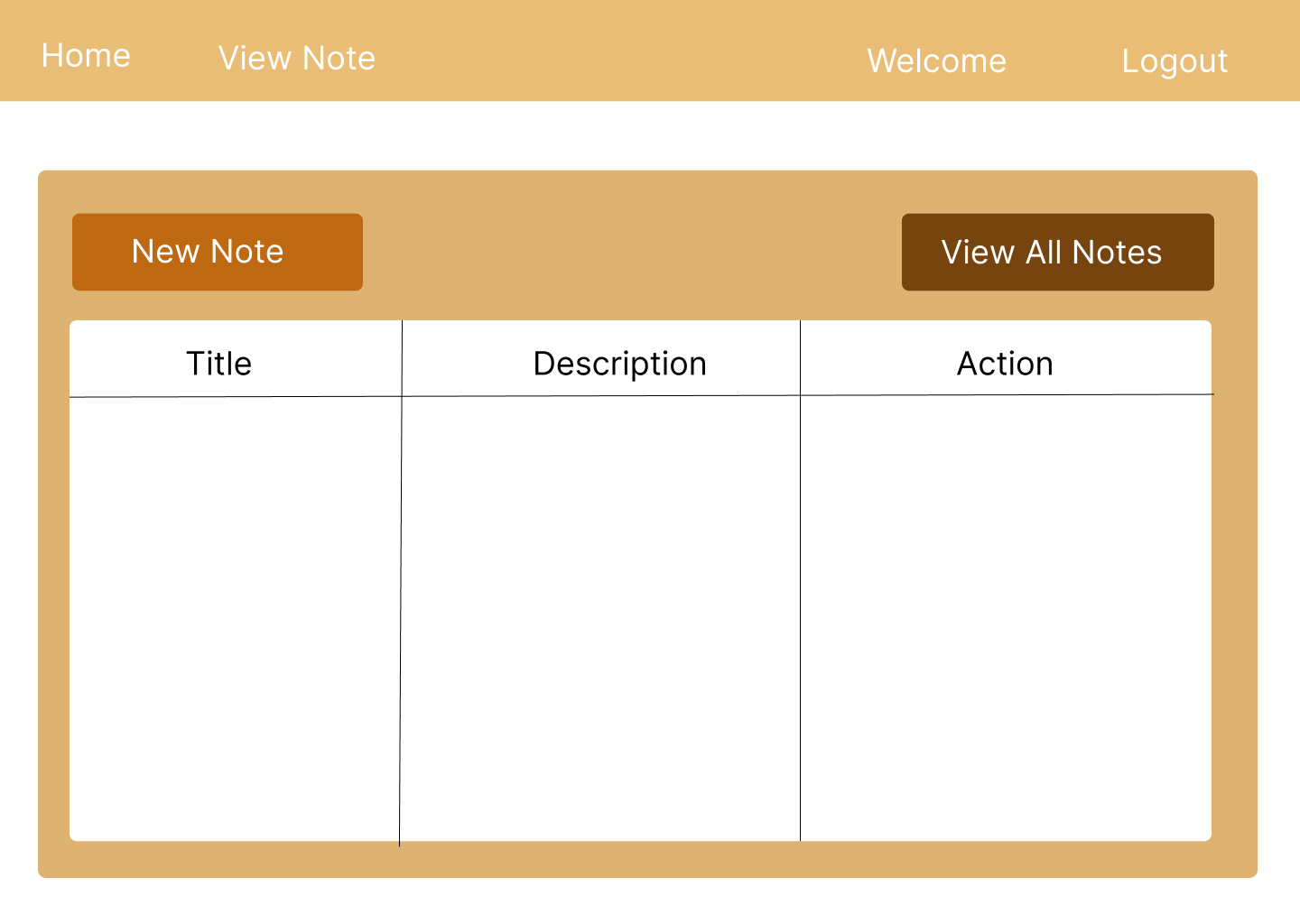
### Figure 5:Register

## Page d’accueil



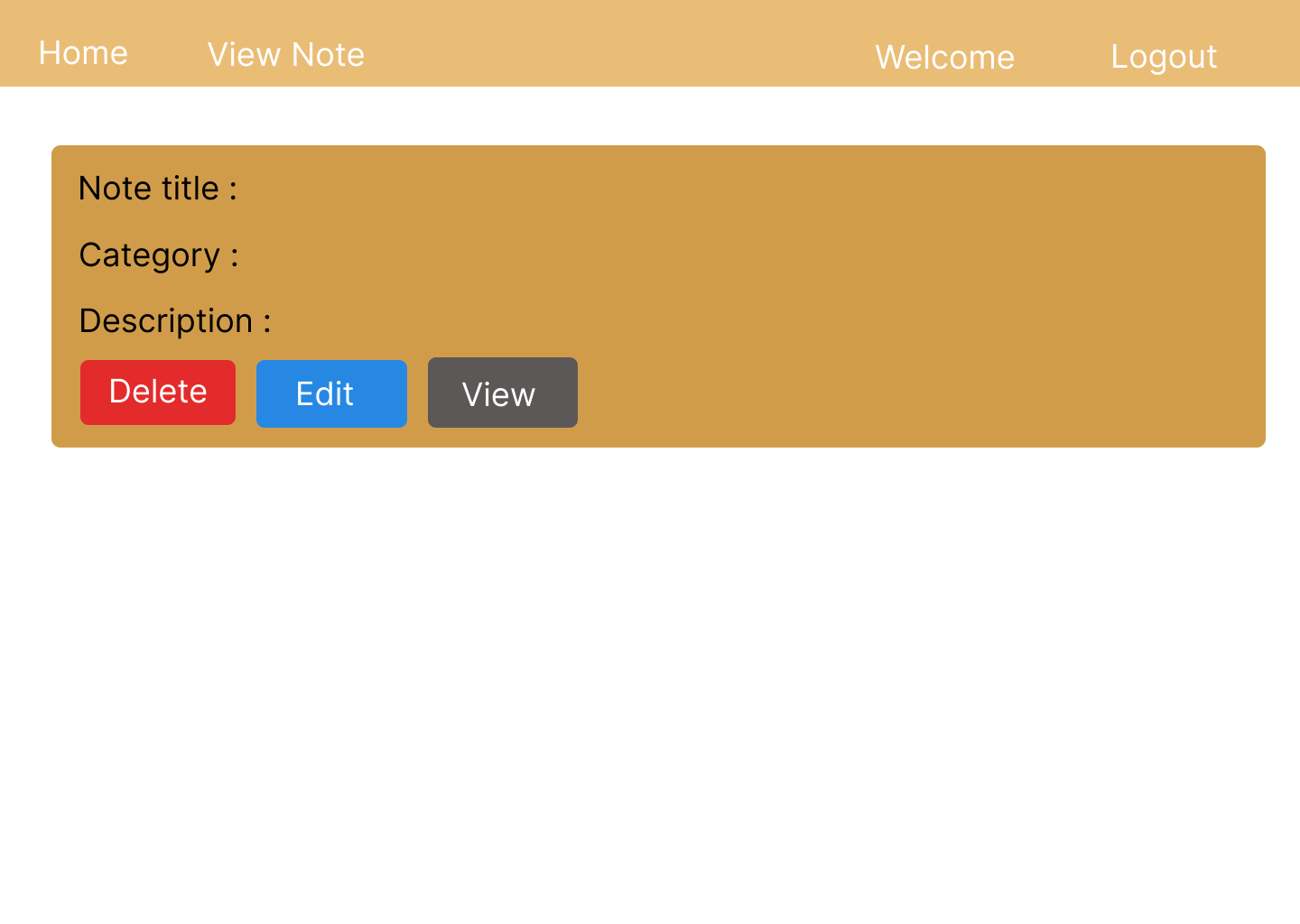
### Figure 6:Page d’Accueil

## Liste des notes



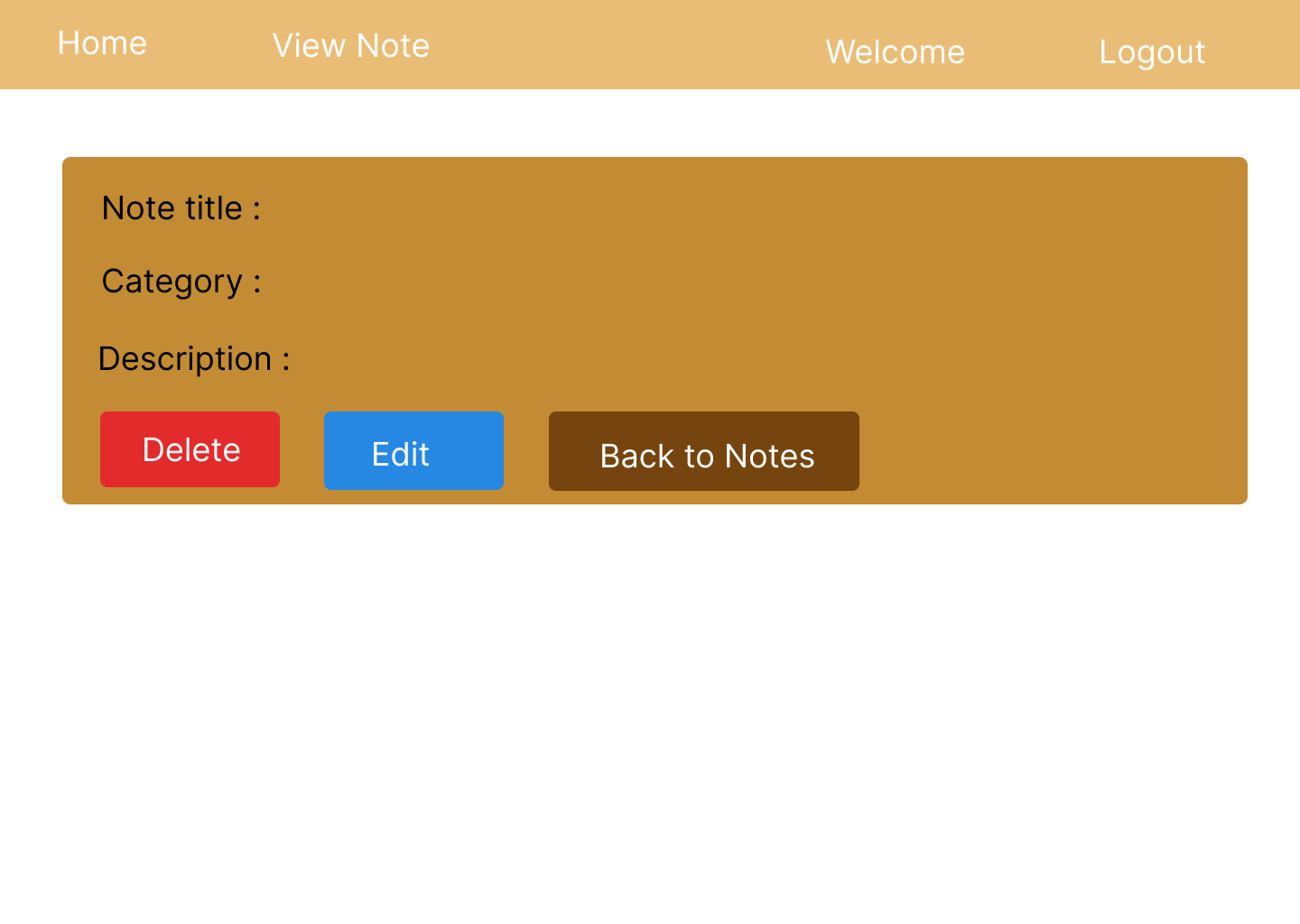
### Figure 7:La liste des notes

## La liste des Notes avec modifier et supprimer et ajouter



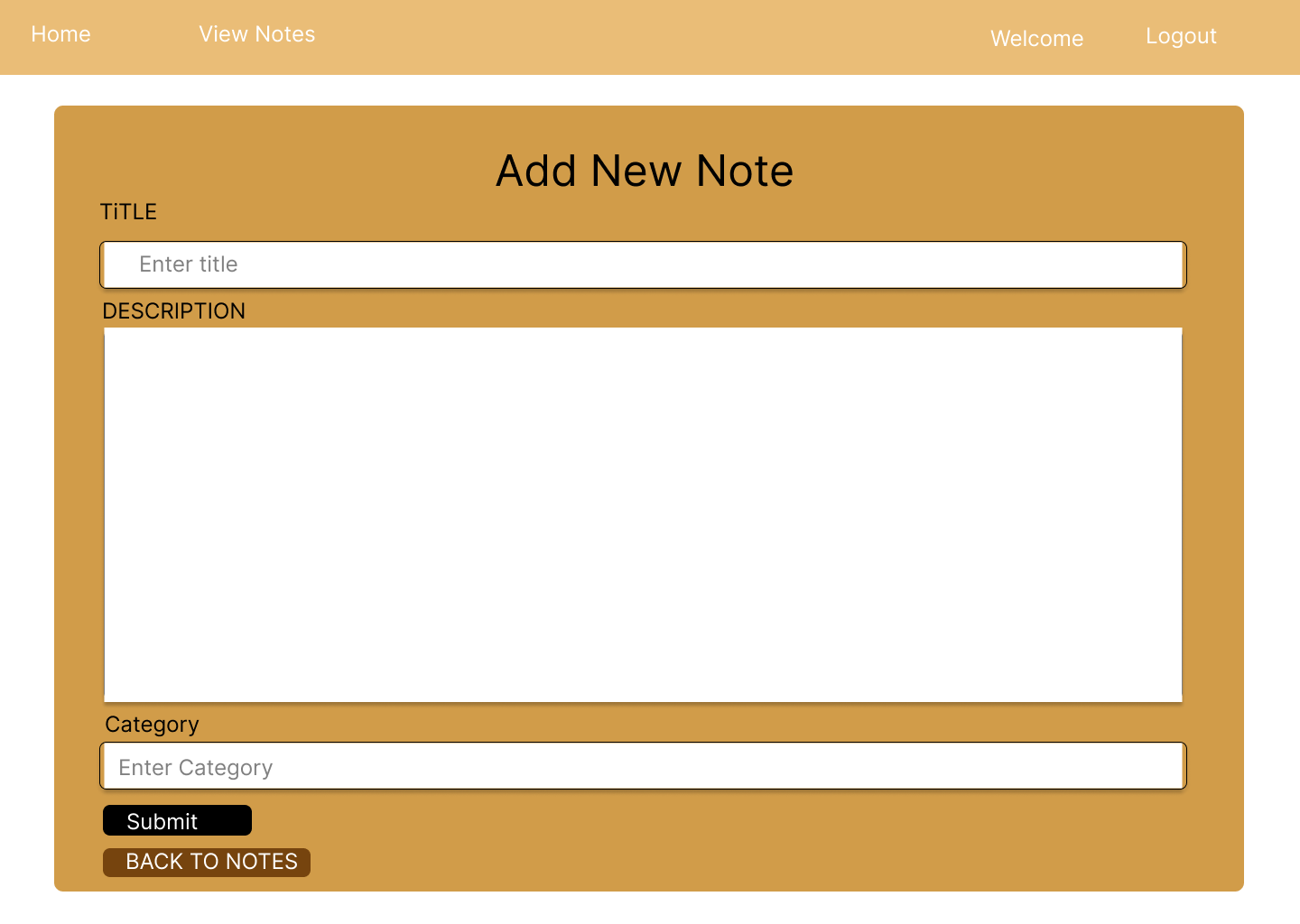
### Figure 8:Les Notes avec bottons supprimer et modifier et consulter

## Note avec button supprimer et modifier et back to nots



### Figure 9: Les Notes avec bottons supprimer et modifier et consulter

## Page d’ajout de note



### Figure 10:L’ajout d’un note

5. Docker – Docker – Compose :

version: '3'

services:

  # Service MySQL

  mysql:

    image: mysql:latest

    container\_name: mysql-container1

    environment:

      MYSQL\_ROOT\_PASSWORD: root

    ports:

      - "3307:3307"

    networks:

      - microservices-network

  # Service Consul

  consul:

    image: consul:1.15.4

    container\_name: consul-container

    ports:

      - "8500:8500"

    networks:

      - microservices-network

  # Service Gateway

  gateway-service:

    build:

      context: ./gateway

    ports:

      - "8888:8888"

    depends\_on:

      - mysql

      - consul

    networks:

      - microservices-network

    environment:

      SPRING\_CLOUD\_CONSUL\_HOST: consul

      SPRING\_CLOUD\_CONSUL\_PORT: 8500

      SPRING\_CLOUD\_CONSUL\_DISCOVERY\_ENABLED: 'true'

      SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/Micro\_ClientDB?createDatabaseIfNotExist=true

      SPRING\_DATASOURCE\_USERNAME: root

      SPRING\_DATASOURCE\_PASSWORD: root

    healthcheck:

      test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

      interval: 5s

      timeout: 2s

      retries: 100

  # Service Client

  client-service:

    build:

      context: ./client

    ports:

      - "8089:8089"

    depends\_on:

      - mysql

      - consul

      - gateway-service

    networks:

      - microservices-network

    environment:

      SPRING\_CLOUD\_CONSUL\_HOST: consul

      SPRING\_CLOUD\_CONSUL\_PORT: 8500

      SPRING\_CLOUD\_CONSUL\_DISCOVERY\_ENABLED: 'true'

      SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/Micro\_ClientDB?createDatabaseIfNotExist=true

      SPRING\_DATASOURCE\_USERNAME: root

      SPRING\_DATASOURCE\_PASSWORD: root

    healthcheck:

      test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

      interval: 5s

      timeout: 2s

      retries: 100

  # Service Note

  note-service:

    build:

      context: ./microservicenote

    ports:

      - "8088:8088"

    depends\_on:

      - mysql

      - consul

      - gateway-service

    networks:

      - microservices-network

    environment:

      SPRING\_CLOUD\_CONSUL\_HOST: consul

      SPRING\_CLOUD\_CONSUL\_PORT: 8500

      SPRING\_CLOUD\_CONSUL\_DISCOVERY\_ENABLED: 'true'

      SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/Micro\_NoteDB?createDatabaseIfNotExist=true

      SPRING\_DATASOURCE\_USERNAME: root

      SPRING\_DATASOURCE\_PASSWORD: root

    healthcheck:

      test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

      interval: 5s

      timeout: 2s

      retries: 100

  # Service phpMyAdmin

  phpmyadmin:

    image: phpmyadmin/phpmyadmin

    environment:

      PMA\_HOST: mysql

      PMA\_PORT: 3306

      MYSQL\_ROOT\_PASSWORD: root

    ports:

      - "8081:80"

    networks:

      - microservices-network

# Définition du réseau

networks:

  microservices-network:

    driver: bridge

 # Service Client

  client-service:

    build:

      context: ./client

    ports:

      - "8089:8089"

    depends\_on:

      - mysql

      - consul

      - gateway-service

    networks:

      - microservices-network

    environment:

      SPRING\_CLOUD\_CONSUL\_HOST: consul

      SPRING\_CLOUD\_CONSUL\_PORT: 8500

      SPRING\_CLOUD\_CONSUL\_DISCOVERY\_ENABLED: 'true'

      SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/Micro\_ClientDB?createDatabaseIfNotExist=true

      SPRING\_DATASOURCE\_USERNAME: root

      SPRING\_DATASOURCE\_PASSWORD: root

    healthcheck:

      test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

      interval: 5s

      timeout: 2s

      retries: 100

  # Service Note

  note-service:

    build:

      context: ./microservicenote

    ports:

      - "8088:8088"

    depends\_on:

      - mysql

      - consul

      - gateway-service

    networks:

      - microservices-network

    environment:

      SPRING\_CLOUD\_CONSUL\_HOST: consul

      SPRING\_CLOUD\_CONSUL\_PORT: 8500

      SPRING\_CLOUD\_CONSUL\_DISCOVERY\_ENABLED: 'true'

      SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/Micro\_NoteDB?createDatabaseIfNotExist=true

      SPRING\_DATASOURCE\_USERNAME: root

      SPRING\_DATASOURCE\_PASSWORD: root

    healthcheck:

      test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

      interval: 5s

      timeout: 2s

      retries: 100

  # Service phpMyAdmin

  phpmyadmin:

    image: phpmyadmin/phpmyadmin

    environment:

      PMA\_HOST: mysql

      PMA\_PORT: 3306

      MYSQL\_ROOT\_PASSWORD: root

    ports:

      - "8081:80"

    networks:

      - microservices-network

# Définition du réseau

networks:

  microservices-network:

    driver: bridge