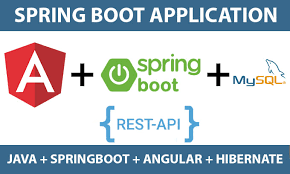
**Compte Rendu**

**APPLICATION DE GESTION E-BANK**

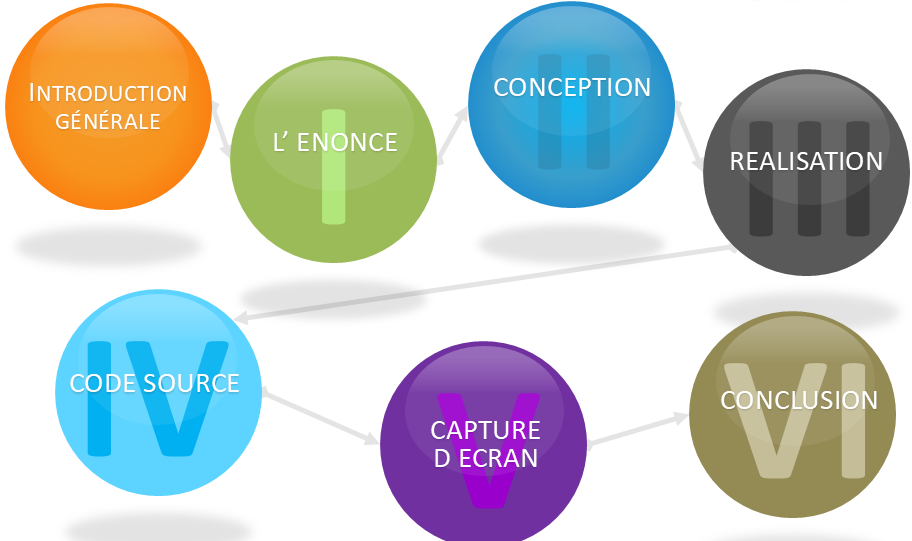
|  |
| --- |
| **Objectifs** **:**  Développer une application web basée sur Architecture microservices. |

****

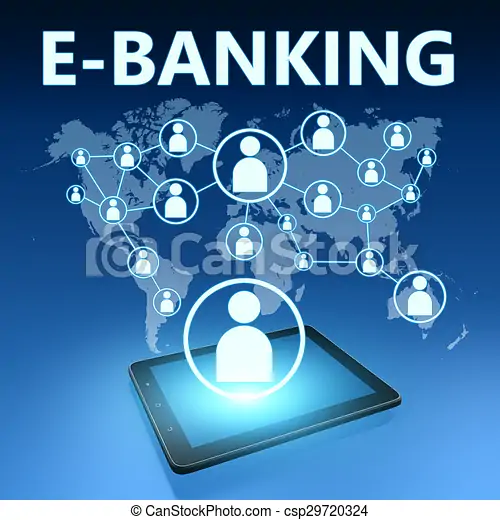
**ANNEE UNIVERSITAIRE : 2021-2022**

**ENCADRE PAR : M. MOHAMED YOUSSFI REALISE PAR : ELMEHDI FATHI**

**Plan de RAPPORT**

****

****

****

**E-banking Introduction**

La relation entre les banquiers et leurs clients connaît aujourd'hui une évolution majeure avec l'utilisation croissante de l'électronique.

Les différents révolutions de la technologie offrent de nouveaux modes de liaison électronique et permettent d'imaginer des nouveaux moyens de paiement par l’E-banking.

\* Définition : l’E-banking c’est l’utilisation de l'internet par une institution financière pour offrir à ses clients une gamme de services bancaires plus ou moins large, allant de la simple vitrine commerciale à la gestion à distance de transactions financières.

\*Les services de l’E-banking :

1. L’accès aux comptes : qui permet une vue synthétique des comptes avec solde global, d'observer les nouvelles transactions comme les chèques liquidés, les récents dépôts et retraits ;

2. Les virements d’un compte à un autre : Le E-banking offre la possibilité de transférer des fonds d'un compte à un autre d'une façon très simple et au temps réel ;

3. Le paiement des factures on line : les banques permettent à leurs clients de payer des factures online chaque mois et de transférer électroniquement des fonds dans les comptes des créditeurs.

4. D’autres services : tels que l'achat des certificats des dépôts, des applications de carte de crédit, les demandes de crédit comme l'auto-emprunt, le prêt personnel, l'éducationnel.

**ARCHITECTURE**

****

**Spring**

est un Framework de développement d'applications Java, qui apporte plusieurs fonctionnalités comme Spring Security, SpringMVC, Spring Batch, Spring Ioc, Spring Data, etc. Ces Frameworks ont pour objectif de faciliter la tâche aux développeurs. Malheureusement, leurs mises en œuvre deviennent très complexes à travers les fichiers de configuration XML qui ne cessent de grossir, et une gestion des dépendances fastidieuse. C'est pour répondre à cette inquiétude que le projet Spring Boot a vu le jour.

**Spring Boot**

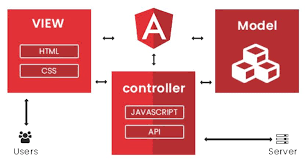
est un sous projet de Spring qui vise à rendre Spring plus facile d'utilisation en élimant plusieurs étapes de configuration. L'objectif de Spring Boot est de permettre aux développeurs de se concentrer sur des tâches techniques et non des tâches de configurations, de déploiements, etc. Ce qui a pour conséquences un gain de temps et de productivité (avec Spring Boot, il est très facile de démarrer un projet n-tiers).

**Les services REST**

représentent un style d'architecture pour développer des services web. Une API qui respecte les principes REST est appelée **API-RESTful**.

Les principes clés de REST impliquent la séparation de l'API en ressources logiques. Ce qui revient à penser à comment obtenir chaque ressource.

Une ressource est un objet ou une représentation d'objets contenant éventuellement des données. Exemple : un employé d'une société est une ressource. La manipulation de ces ressources repose sur le protocole HTTP à travers les méthodes d'actions **GET, POST, PUT, PATCH, DELETE**.

****

**Angular**

AngularJS est devenu l'un des plus dynamiques dans le développement web, c'est le framework préféré de tous les développeurs qui souhaitent développer des solutions riches en fonctionnalités. Cela est dû aux énormes avantages qu'il offre aux développeurs. Plusieurs grandes organisations préfèrent Angular pour créer des applications pour les processus internes ainsi que pour les utilisateurs externes. Angular étant un cadre exceptionnel, il est utilisé pour créer des applications Web d'entreprise, en particulier le développement frontal, en raison de la stabilité qu'il offre. Avec ses excellentes fonctionnalités et sa compatibilité, on peut sans aucun doute affirmer qu'Angular est un framework d'applications Web du futur.

**Avantages**

**Cohérence et réutilisabilité du code**

### Facile à apprendre, à utiliser et à tester

### Assistance Google et excellente assistance communautaire

### Fonctionnalités orientées SPA

### Liaison de données bidirectionnelle

### Interface utilisateur déclarative

### Intégration transparente et productivité haut de gamme

**Conception**

**UML**

Le **Langage de Modélisation Unifié**, de l'anglais *Unified Modeling Language* (**UML**), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orienté Object.

**UTILISATION**

UML est destiné à faciliter la conception des documents nécessaires au développement d’un logiciel orienté objet, comme standard de modélisation de l’architecture logicielle.

Les différents éléments représentables sont :

+ Activité d’un objet/logiciel

+ Acteurs

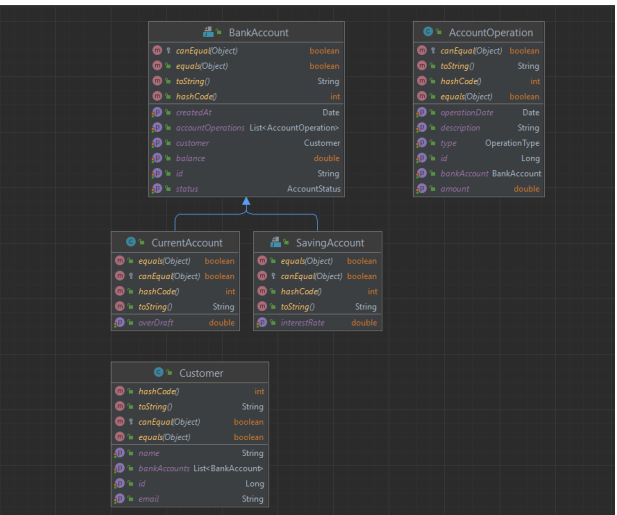
+ Processus

+ Schéma de base de données

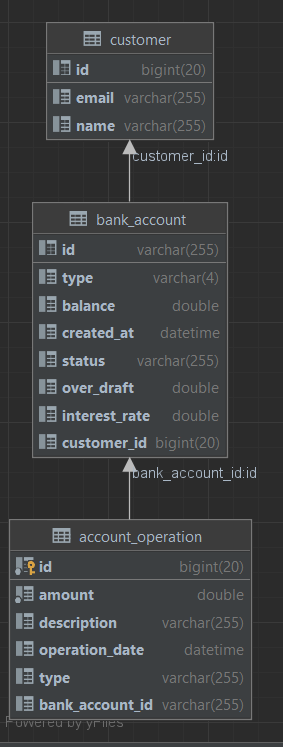
+ Composants logiciels

+ Réutilisation de composants.

[*Diagramme de classes*](https://fr.wikipedia.org/wiki/Diagramme_de_classes) (*class diagram*) : représentation des [classes](https://fr.wikipedia.org/wiki/Classe_(informatique)) intervenant dans le système

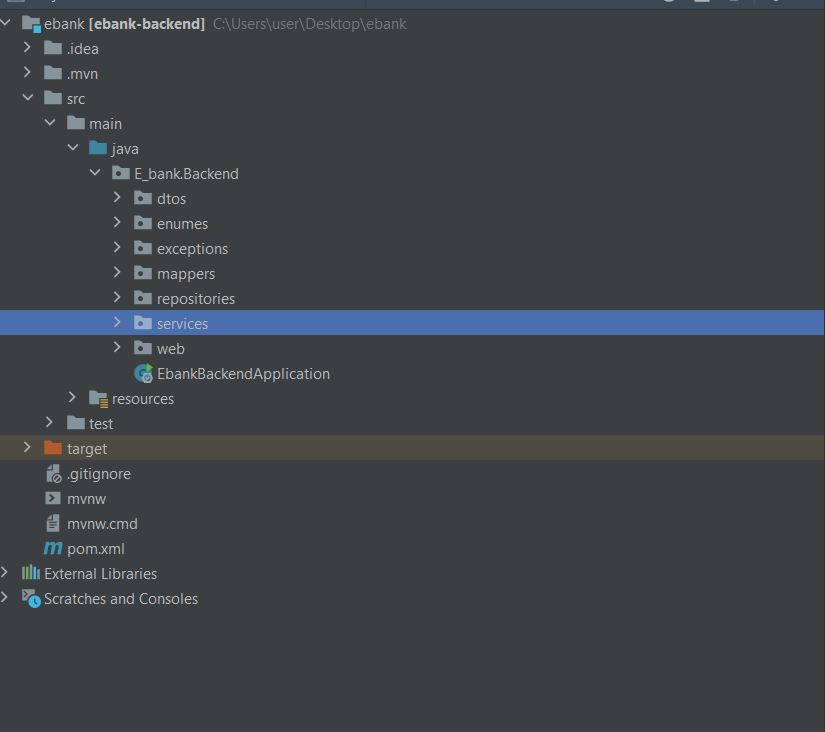
@Inheritance(strategy=InheritanceType.TABLE\_PER\_CLASS)****

@Inheritance strategy=InheritanceType.SINGLE\_TABLE)

****

**REALISATION**

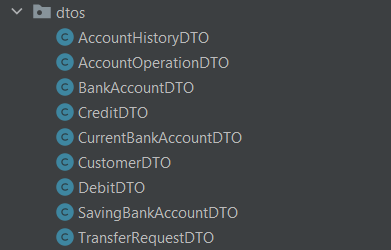
**Architecture Technique**

****

**COUCHE APPLICATION**

Cette couche contient la classe application qui est la classe de lancement d’une application Spring Boot L’annotation @SpringBootApplication est équivalente à @Configuration et @Beans ….

COUCHE DAO

Une couche DAO (Data Access Object) offrent plusieurs services :

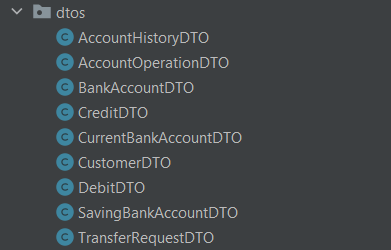
+ Centralisation des accès aux données

+ Simplification de l’accès aux données

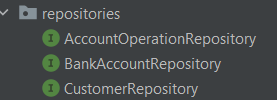
+ Abstraction du support de stockage

+ Travail sur les entités principales

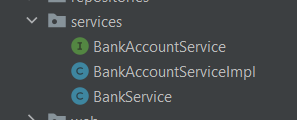
**COUCHE ENTITES**

****

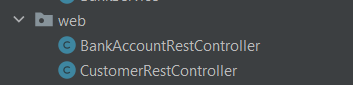
**COUCHE Reposotory**

****

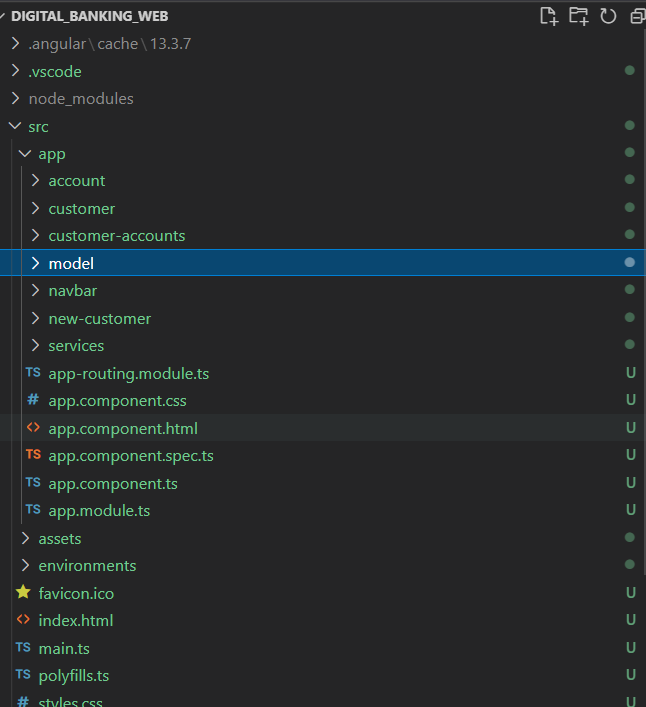
**COUCHE SERVICES**

****

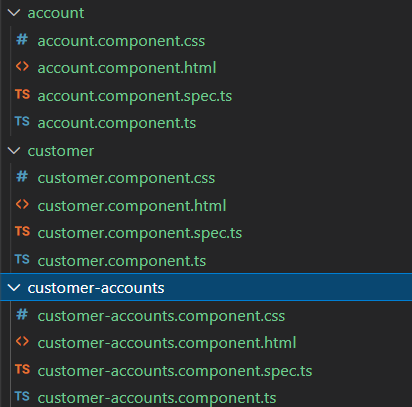
**Couche web**

****

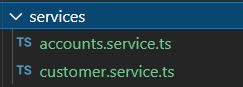
**Partie FRONTEND : ANGULAR**

****

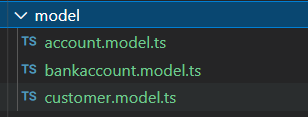
**COMPONENT**

****

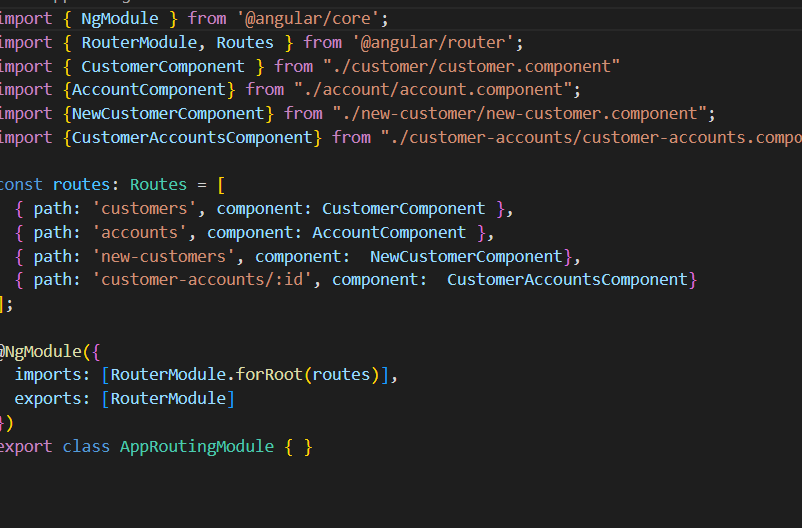
**SERVICES**

****

**MODELS**

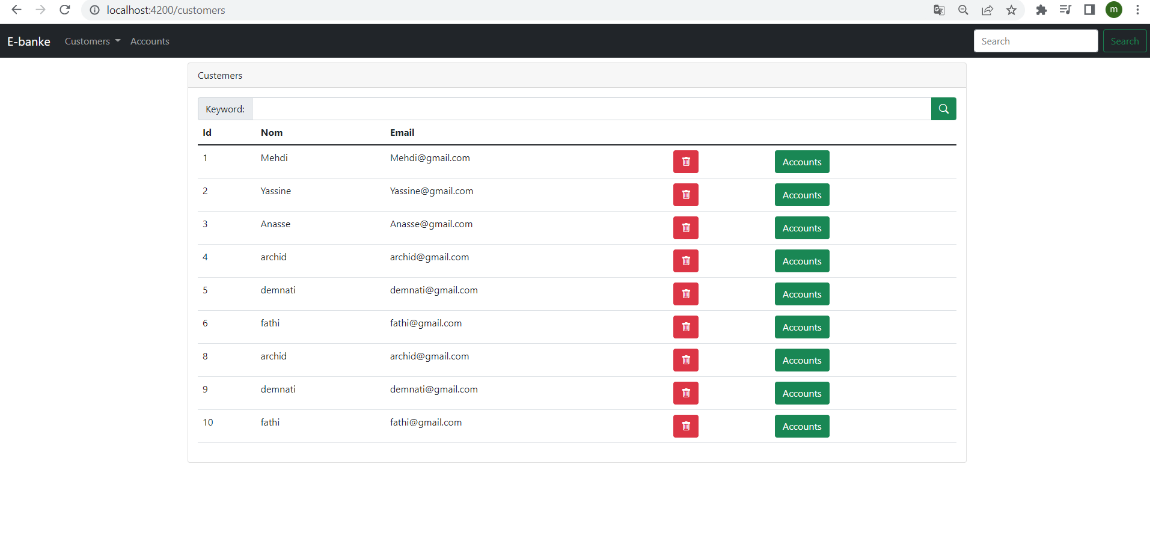
****

**NAVIGATION ENTRE LES COMPONENTS**

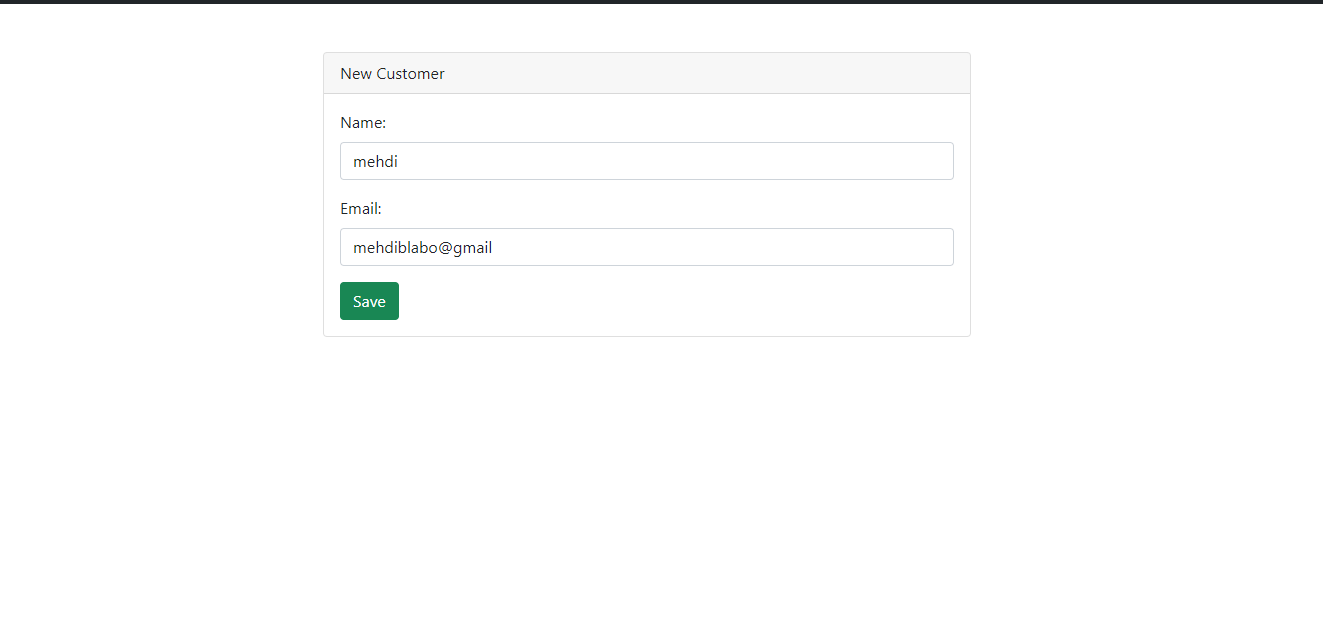
****

**DEMONSTATION**

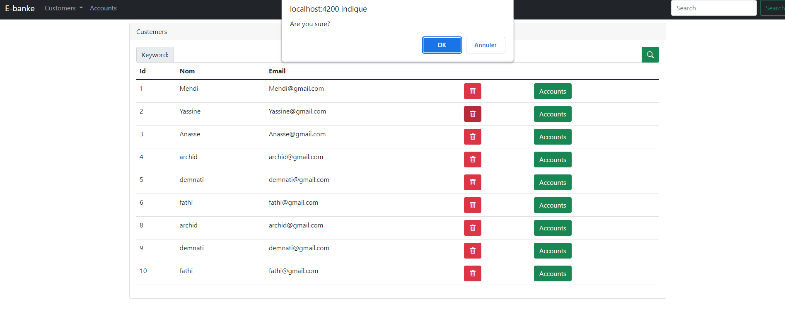
**page principale**

****

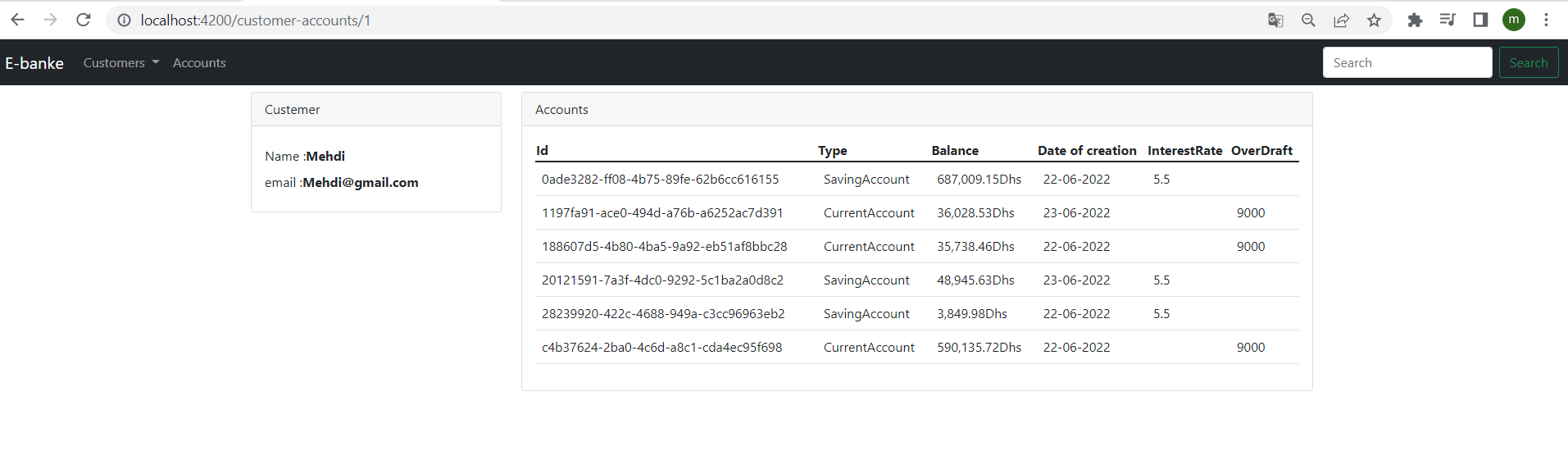
**Save User**

****

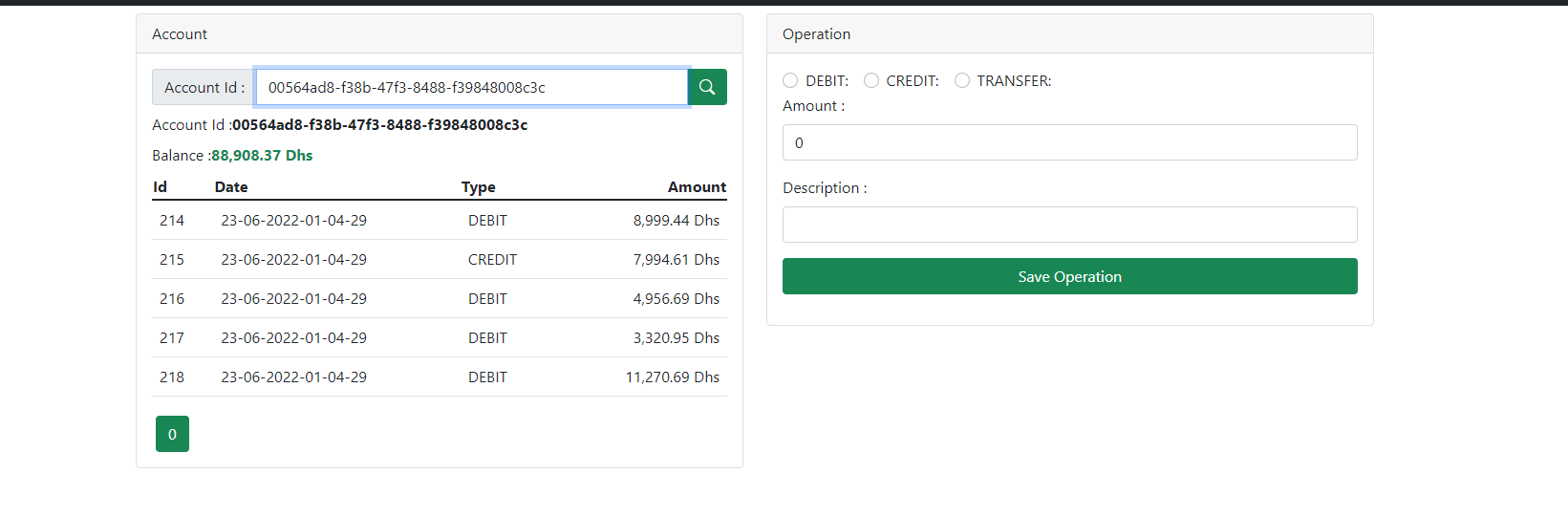
**DELETE**

****

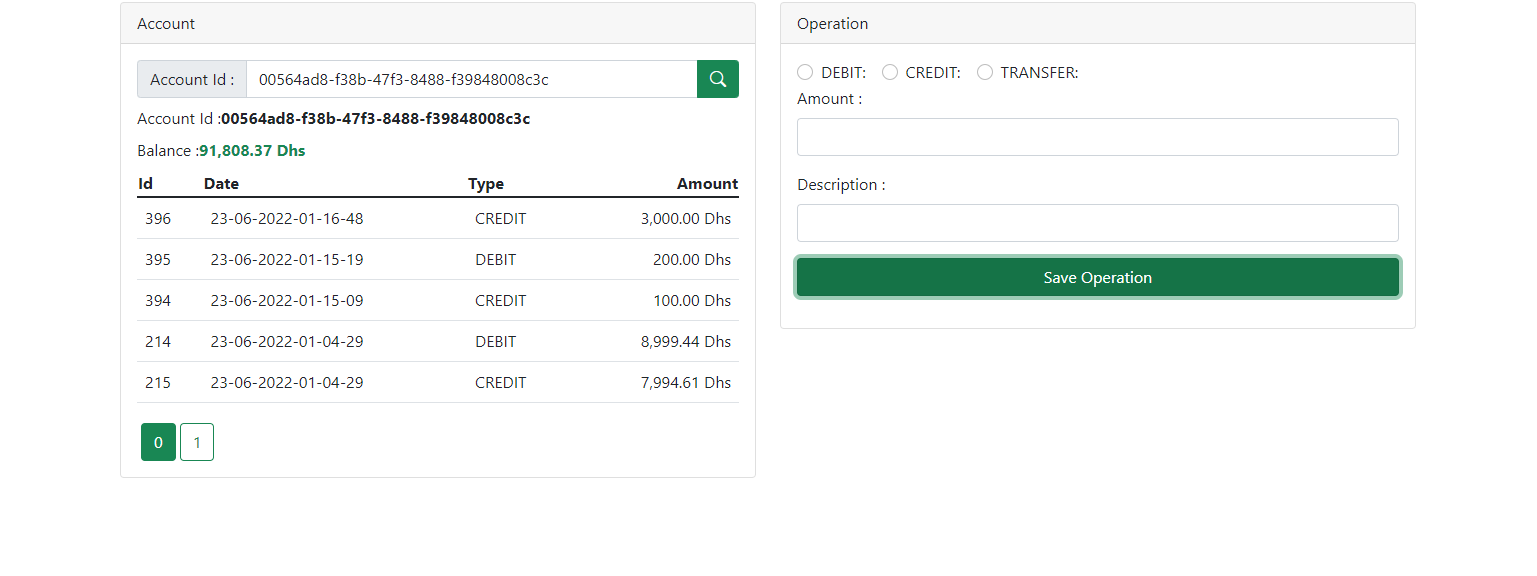
**Detail**

****

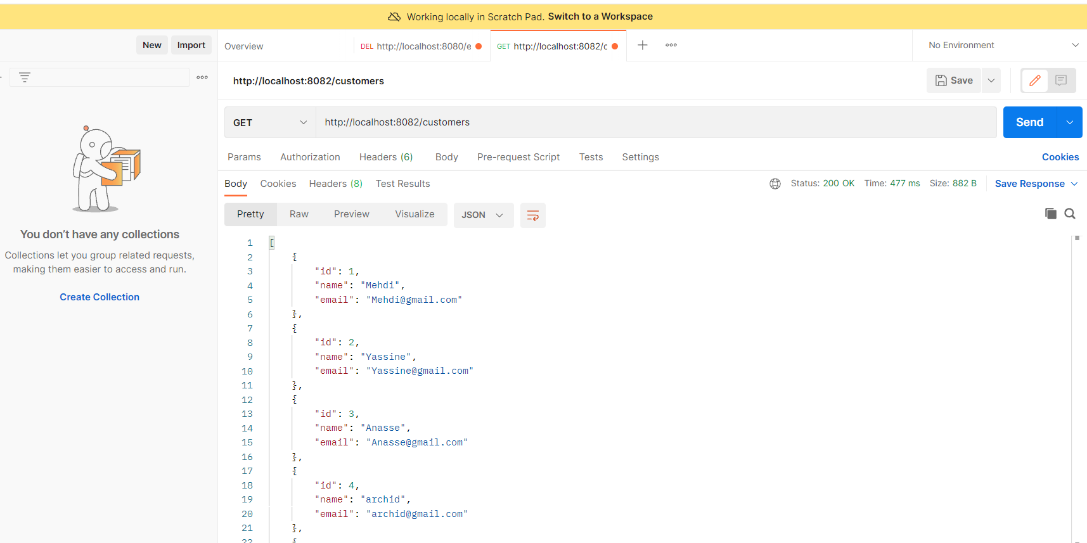
**Chercher un Account**

****

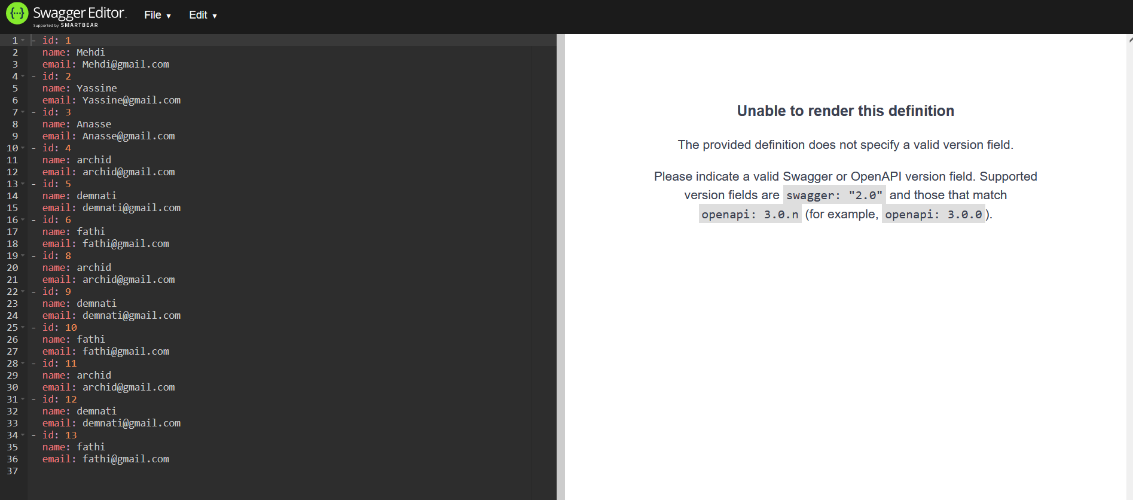
**Effectue tous les operation**

****

**POSTMAN**

****

**SWAGGER**

****

**Lien GITHUB**

**https://github.com/ELMEHDIFATHI/E\_banking/tree/master**

**Conclusion**

La mise en oeuvre de tout projet nécessite une parfaite Concentration pour comprendre

Les bonne pratiques de architecture micro services et reforce notre boite outils du spring boot ce processus n’est jamais complètement terminée, c’est une tâche itérative, à l’aide des expériences passées une fois le système est implémenté et functionne des améliorations sont nécessaires.​

Par exemple avoir la securite JWT,le style , .durant ce project qui n  a pas etait just un project pour moimais il etait aussi une ameliore pour notre competance soit en niveau de travaille personelle, l agilite , gestion de project ainsi que mon developpment personnelle.​

Au stade actuel le projet est passé en phase finale de réalisation. Mais nous visons de perspectives pour notre solution , d’autres fonctionnalités peuvent être rajoutées.​

**MERCI POUR VOTRE ENCADREMENT DURANT CE SEMESTRE**