



**ECOLE MAROCAINE DES  
SCIENCES DE L'INGENIEUR**  
*Membre de* **HONORIS UNITED UNIVERSITIES**

## Rapport sur la réalisation de l'application web Digital-Banking

« Réalisée avec Spring Boot, Spring Security et Angular.js »

Encadré par: **Mr. Mohamed Youssfi.**

Réalisé par: **Mohamed El Masnaoui.**

## Sommaire

I.	Introduction :	3
II.	Conception générale :	5
III.	Captures d'écran des interfaces de l'application :	6
1)	Interface 1 :	6
2)	Exemple de l'interface 1 :	6
3)	Interface 2 :	7
4)	Exemples de l'interface 2 :	7
5)	Interface 3 :	8
IV.	Outils et technologies utilisées :	8
6)	Intellij :	8
7)	Spring boot :	9
8)	Spring security :	10
9)	Angular :	10
V.	Conclusion :	12

## I. Introduction :

Bienvenue dans l'introduction à notre application de gestion de comptes bancaires ! Dans cet aperçu, nous allons explorer les fonctionnalités clés de notre application, qui permettra aux utilisateurs de gérer efficacement leurs comptes bancaires. Nous avons choisi d'utiliser les frameworks Spring Boot, Spring Security et Angular pour développer cette application robuste et conviviale.

Notre application vise à offrir une plateforme complète et sécurisée pour les opérations bancaires en ligne. Chaque compte bancaire appartient à un client spécifique, ce qui permet une gestion personnalisée des comptes et une expérience utilisateur optimale.

L'un des aspects essentiels de notre application est la gestion des opérations bancaires. Les utilisateurs pourront effectuer des opérations de débit et de crédit sur leurs comptes respectifs. Cela permettra aux clients de suivre leurs transactions, d'analyser leurs dépenses et de gérer leurs finances de manière plus efficace. De plus, notre système enregistrera toutes les opérations effectuées, ce qui garantit la transparence et l'intégrité des données.

Nous avons identifié deux types de comptes principaux : les comptes courants et les comptes épargnes. Les comptes courants sont conçus pour les transactions quotidiennes, tandis que les comptes épargnes sont destinés à la constitution d'économies et à la gestion des intérêts. Chaque type de compte aura ses propres fonctionnalités spécifiques et sera adapté aux besoins financiers individuels des clients.

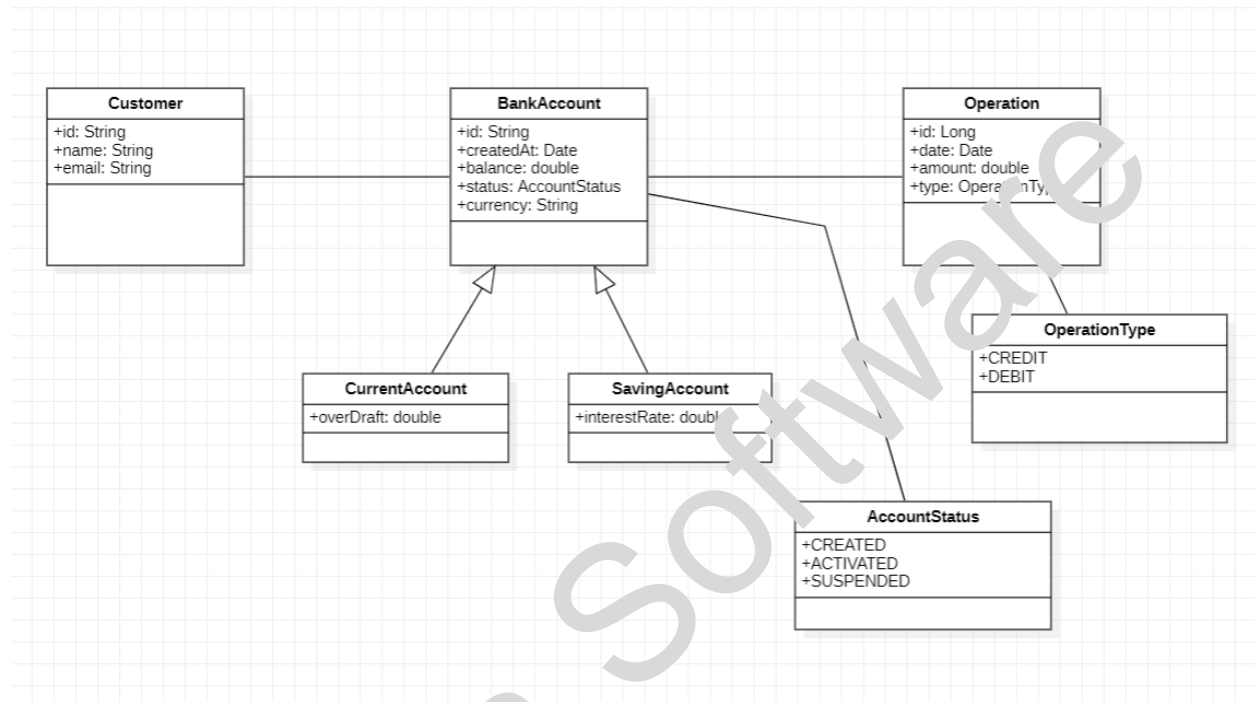
Pour garantir la sécurité des données sensibles des utilisateurs, nous avons choisi d'implémenter Spring Security, un puissant framework de sécurité pour les applications Java. Spring Security offre des fonctionnalités telles que l'authentification, l'autorisation et la protection contre les attaques courantes telles que les injections SQL et les attaques de session. Ainsi, nos utilisateurs peuvent avoir la tranquillité d'esprit quant à la confidentialité de leurs informations financières.

En ce qui concerne le développement de l'interface utilisateur, nous utiliserons Angular, un framework JavaScript populaire pour la création d'applications Web. Angular offre une structure solide pour développer des interfaces utilisateur réactives et dynamiques. Grâce à ses fonctionnalités de liaison de données en temps réel, nous pourrions offrir une expérience utilisateur fluide et interactive.

Dans l'ensemble, notre application de gestion de comptes bancaires vise à simplifier la vie financière de nos utilisateurs. Grâce à des fonctionnalités complètes de gestion de comptes, de suivi des transactions et de sécurité renforcée, nous espérons offrir une solution pratique pour les opérations bancaires quotidiennes.

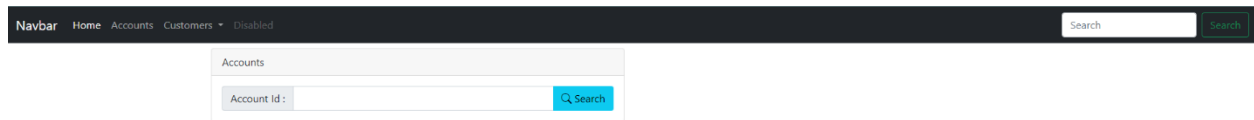
## II. Conception générale :

Pour mieux comprendre le concept du projet on a essayé de réaliser le diagramme de classe ci-dessous qui décrit les différentes entités de ce projet.



### III. Captures d'écran des interfaces de l'application :

#### 1) Interface 1 :

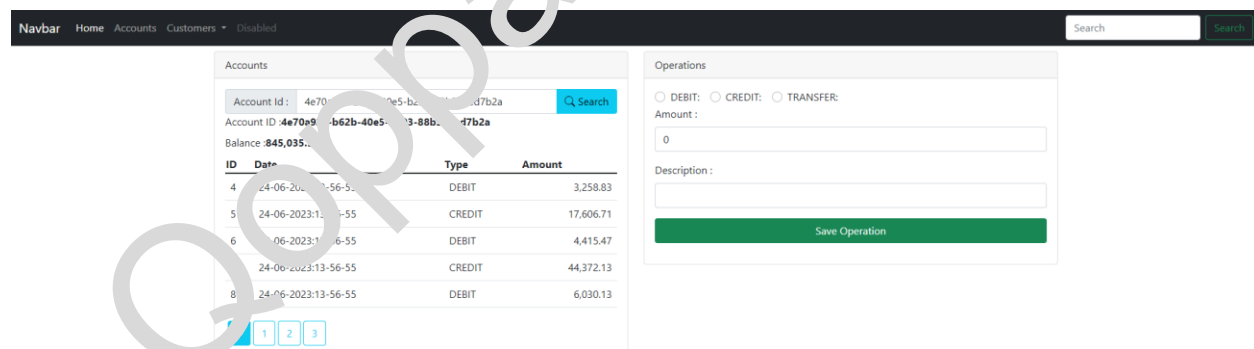


Accounts

Account Id :

Dans cette interface on trouve une case à remplir avec l'id du compte pour voir les informations sur ce dernier ainsi que la possibilité de réaliser des opérations Débit, Crédit ou Transfer.

#### 2) Exemple de l'interface 1 :



Accounts

Account ID :

Account ID : 4e70e9b62b40e51388b17b2a

Balance : 845,035...

ID	Date	Type	Amount
4	24-06-2023:13-56-55	DEBIT	3,258.83
5	24-06-2023:13-56-55	CREDIT	17,606.71
6	24-06-2023:13-56-55	DEBIT	4,415.47
7	24-06-2023:13-56-55	CREDIT	44,372.13
8	24-06-2023:13-56-55	DEBIT	6,030.13





Operations

☐ DEBIT: ☐ CREDIT: ☐ TRANSFER:

Amount :



Description :

### 3) Interface 2 :

Customers				
Keyword :				
ID	Name	Email		
1	Hassan	Hassan@gmail.com		<a href="#">Accounts</a>
2	Imane	Imane@gmail.com		<a href="#">Accounts</a>
3	Mohamed	Mohamed@gmail.com		<a href="#">Accounts</a>
4	Amine	Amine@gmail.com		<a href="#">Accounts</a>

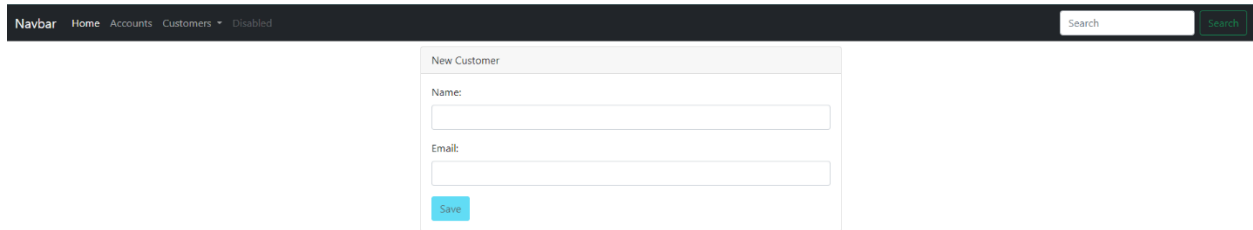
Cette interface est dédiée aux clients et divisée en deux parties : la première pour entrer un mot clé ou bien le nom complet du client pour pouvoir voir son ID, nom et l'email ainsi que la possibilité de le supprimer et voir les comptes concernant ce client.

### 4) Exemples de l'interface 2 :

Customers				
Keyword : Am				
ID	Name	Email		
3	Mohamed	Mohamed@gmail.com		<a href="#">Accounts</a>
4	Amine	Amine@gmail.com		<a href="#">Accounts</a>

```
4
{"id": 4, "name": "Amine", "email": "Amine@gmail.com"}
```

## 5) Interface 3 :



The screenshot shows a web application interface. At the top, there is a dark navigation bar with the text 'Navbar Home Accounts Customers Disabled' and a search bar on the right with the placeholder 'Search' and a 'Search' button. Below the navigation bar, the main content area features a form titled 'New Customer'. The form has two input fields: 'Name:' and 'Email:', each followed by a text input box. At the bottom of the form is a blue 'Save' button.

Cette interface est dédiée à l'ajout de nouveaux clients. On entrant le nom et l'email puis enregistrer les informations en cliquant sur le bouton Save.

## IV. Outils et technologies utilisées :

### 1. IntelliJ :



IntelliJ IDEA est un environnement de développement intégré (IDE) puissant et polyvalent, largement utilisé par les développeurs Java. Il offre une gamme complète de fonctionnalités pour faciliter le développement logiciel, avec un éditeur de code avancé, une vérification des erreurs en temps réel, des outils de gestion de version, le débogage et bien plus encore. L'IDE prend également en charge de nombreux langages de programmation et offre une excellente intégration avec les frameworks populaires tels que Spring et Maven. Avec des fonctionnalités



axées sur la productivité et une communauté active, IntelliJ IDEA permet aux développeurs de travailler efficacement et de créer des applications de haute qualité.

## 2. Spring boot :



Spring Boot facilite la création d'applications Java en fournissant un ensemble complet de fonctionnalités prêtes à l'emploi. Il offre une configuration automatique, ce qui signifie que les développeurs n'ont pas besoin de passer du temps à configurer manuellement l'infrastructure de leur application. Spring Boot intègre également un serveur web embarqué, ce qui permet de déployer facilement des applications sans avoir à configurer un serveur externe. De plus, il propose une gestion simplifiée des dépendances grâce à l'utilisation de starters, qui fournissent toutes les bibliothèques nécessaires pour différents scénarios de développement. Spring Boot est également compatible avec les technologies Spring existantes, ce qui permet aux développeurs de bénéficier de la richesse de l'écosystème Spring tout en simplifiant le processus de développement. En résumé, Spring Boot offre une approche pratique et efficace pour développer des applications Java en réduisant la complexité et en favorisant la productivité des développeurs.

### 3. Spring security :



Spring Security fournit une couche de sécurité robuste pour les applications Java, offrant une protection contre les attaques et la gestion de l'authentification et de l'autorisation. Il offre une configuration simple et flexible pour sécuriser les points d'entrée de l'application, les URL, les méthodes et les ressources. Spring Security prend en charge plusieurs mécanismes d'authentification tels que l'authentification basée sur les formulaires, l'authentification basée sur les jetons, l'authentification basée sur les certificats et l'authentification par LDAP. De plus, il fournit des fonctionnalités d'autorisation granulaires, permettant de définir des rôles et des permissions pour contrôler l'accès aux ressources. Spring Security intègre également des fonctionnalités avancées telles que la protection contre les attaques de sécurité courantes, la gestion des sessions, la détection des tentatives d'intrusion et les fonctionnalités de gestion des mots de passe. En résumé, Spring Security offre une solution complète pour sécuriser les applications Java, en offrant des mécanismes d'authentification et d'autorisation flexibles ainsi que des fonctionnalités de protection contre les attaques, ce qui permet aux développeurs de créer des applications sécurisées de manière efficace.

### 4. Angular :



Angular simplifie la création d'applications web dynamiques en utilisant une approche basée sur les composants. Il offre une structure robuste pour développer des interfaces utilisateur interactives et réactives. Angular utilise le langage TypeScript pour permettre une écriture de code plus robuste et offre des fonctionnalités avancées telles que le binding de données bidirectionnel, la gestion des événements, la manipulation du DOM et la gestion de la navigation. Il intègre également un système de modules qui permet de découper l'application en fonctionnalités réutilisables. Angular inclut une vaste bibliothèque de composants prêts à l'emploi, ainsi qu'un outil de ligne de commande (Angular CLI) pour faciliter la création, la compilation et le déploiement de projets. Il offre également une prise en charge complète des tests unitaires et d'intégration, ce qui facilite la validation et la maintenance du code. En résumé, Angular est un framework puissant et complet pour le développement d'applications web modernes, offrant des fonctionnalités avancées, une structure modulaire et une productivité accrue pour les développeurs.

## V. Conclusion :

En conclusion, notre application de gestion de comptes bancaires basée sur les frameworks Spring Boot, Spring Security et Angular offre une solution complète et sécurisée pour répondre aux besoins financiers de nos utilisateurs. En combinant la puissance de Spring Boot et Spring Security, nous garantissons des fonctionnalités de gestion de comptes avancées, une protection solide des données et une expérience utilisateur transparente. Grâce à Angular, nous offrons une interface utilisateur réactive et intuitive pour une navigation fluide et une gestion aisée des opérations bancaires. Notre objectif est de simplifier la vie financière de nos clients en leur offrant une plateforme fiable et conviviale pour gérer leurs comptes bancaires, suivre leurs transactions et prendre le contrôle de leurs finances. Avec notre application, nous visons à offrir une expérience bancaire en ligne sécurisée, transparente et efficace pour répondre aux besoins changeants de nos utilisateurs.