

Développement d'une Application Web pour la Gestion Bancaire

Réalisé par :

➤ **Glouib Chaima**

Encadré par :

➤ **Hafidi Imad**

Année Universitaire : 2018-2019

Remerciements

Je tiens tout d'abord à remercier Dieu le tout puissant qui m'a donné la force et la patience et la volonté d'accomplir ce Modeste travail. Je tiens à saisir cette occasion et adresser mes profonds remerciements à : Mr Hafidi Imad, mon encadrant de mon PFA. je tiens aussi très honnêtement, à lui exprimer mes reconnaissances pour avoir énormément appris sur les applications client/serveur dans le domaine de gestion bancaire.

Introduction générale

L'évolution des langages de programmation a amené de nouveaux outils aidant à la conception des applications en informatique. En effet, l'événement de l'orienté objet facilite l'abstraction du problème à résoudre en fonction des données du problème lui-même (par l'utilisation des classes et d'objets). C'est ainsi, que la réalisation de logiciels de gestion demeure une activité professionnelle difficile. Malgré les progrès apportés par le génie logiciel, les développements d'application répondant aux besoins exprimés se rationalisent lentement. En revanche, l'offre d'outils de développement ne cesse de croître et l'importance des langages de programmation est toujours prépondérante. De plus en plus d'applications utilisent même plusieurs langages de programmation dans le cadre d'un projet unique. Nous constatons enfin, un intérêt grandissant pour issues des technologies objets. L'objectif de notre PFE est faire la gestion d'une banque en se basant sur une architecture de type client/serveur et en utilisant le mécanisme JAVA RMI. Notre rapport est organisé comme suit :

La suite du document est organisée en 4 chapitres : Le chapitre 1 Etude du projet, le chapitre 2 Analyse et conception, le dernier chapitre est consacrée à la présentation de l'application.

Résumé

Dans le but de créer d'un outil performant, capable de surmonter le défi, nous avons trouvé qu'il sera nécessaire de travailler avec un Framework, notre choix est tombé sur Spring Boot. Ce choix est justifié par la large et la puissante documentations que propose Spring que ce soit en anglais qu'en français, ainsi que sa simplicité qui permet sa maîtrise en un temps limite. La solution proposée consiste à créer une application web qui propose une simple application web de gestion des comptes bancaires implémenter par les différents Frameworks que nous expliquerons dans notre outil pédagogique.

Chapitre 1 : Etude du projet

1) Introduction

Ce chapitre est consacré à la présentation du cadre du projet et à la spécification des besoins. Nous décrivons dans ce chapitre le sujet de notre projet, ses différentes parties et son but.

2) Cahier des charges:

Le but du projet est de réaliser un outil pédagogique d'enseignement des différentes Frameworks Spring Boot, cet outil doit permettre à l'utilisateur :

- ✓ Comprendre l'architecture en couche d'une application web.
- ✓ Consulter un exemple d'une application web réalisée par les différents Frameworks
 - La couche présentation est implémentée par : Thymeleaf, Spring Mvc.
 - La couche métier par : Spring core.
 - La couche persistance par : Spring Data .

Un système de gestion de compte bancaire est l'application web que nous utiliserons dans notre outil pédagogique, c'est un système qui supporte la gestion automatique des clients et leurs comptes. Les différentes fonctionnalités de l'application sont données ci-après.

Après analyse d'un document décrivant les tâches relatives à la gestion bancaire, les opérations élémentaires suivantes ont été recensées :

- ✓ Création de compte pour un nouveau client (ou existant),
- ✓ Modification d'un compte existant,
- ✓ Suppression d'un client et de ses comptes,
- ✓ Modification des informations d'un client,
- ✓ Dépôt et retrait de liquidité sur un compte,
- ✓ Virement d'argent d'un compte client vers un autre

Quant aux droits d'accès, il y a trois niveaux définis en fonction du statut de l'utilisateur. Le niveau le plus élevé possède des droits les plus importants et correspond à celui de l'administrateur. Les

trois niveaux de droits définis sont les suivants:

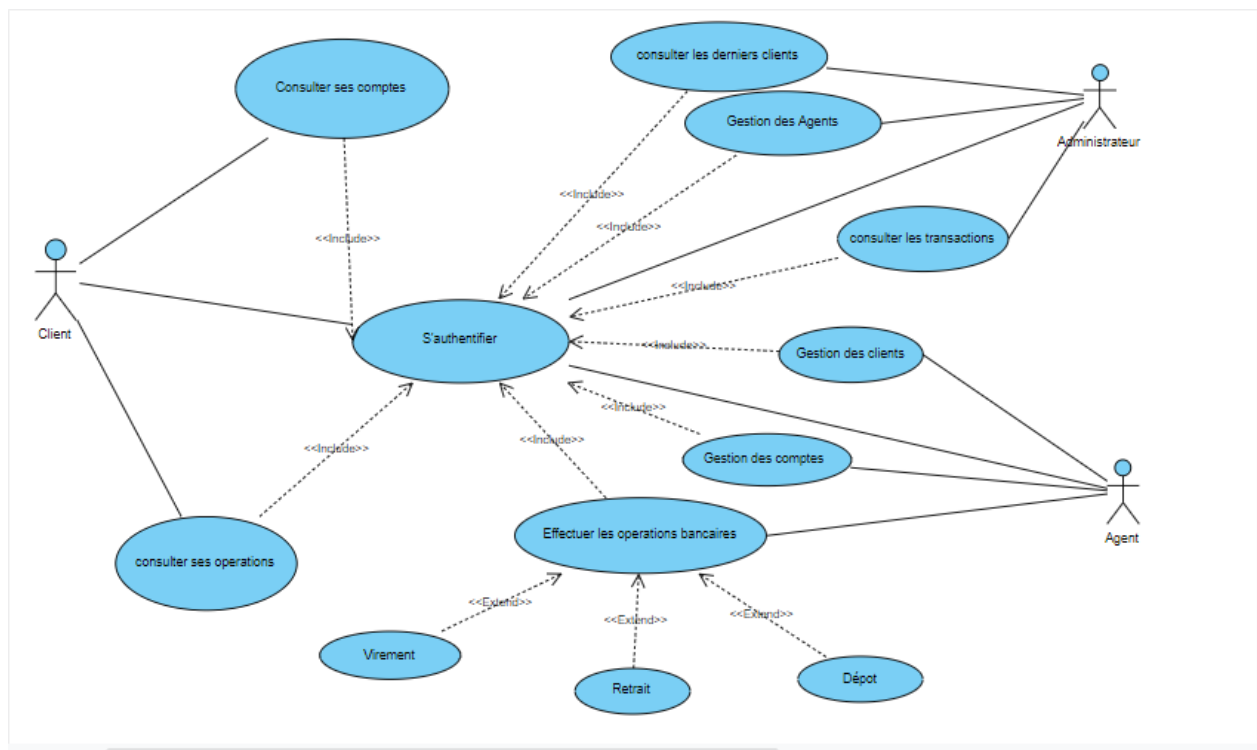
- ✓ Niveau 1 : Statut Administrateur, le plus élevé
- ✓ Niveau 2 : Statut Agent (Employé de la banque)
- ✓ Niveau 3 : Statut Client, le plus faible.

3) Les fonctionnalités :

➤ Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont composés d'**acteurs** et de **cas d'utilisation**. Un **acteur** est un utilisateur, humain ou non, du système qui est doté d'un nom qui correspond à son rôle. Un **cas d'utilisation** est une manière spécifique d'utiliser le système. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Le diagramme 1 représente les différents cas d'utilisation de notre système. Il a été réalisé avec l'outil ArgoUML.



4) Les Scénarios possibles:

Scénario de cas d'utilisation « authentification »

1. L'utilisateur renseigne le login et le mot de passe	2. Le système vérifie l'identité de l'utilisateur
	3. Le système affiche la page d'accueil

Scénario de cas d'utilisation « gestion de comptes »

a. Création de compte

1. L'employé renseigne son login et son mot de passe	2. Le système vérifie l'identité de L'employé
	3. Le système affiche la page d'accueil
4. L'employé demande la page de création de compte	5. Le système affiche la page de création de compte
6. L'employé renseigner les informations sur le compte ainsi que le propriétaire du compte	7. Le système vérification des données saisies

b. Modification

1. L'employé renseigne son login et son mot de passe	2. Le système vérifie l'identité de L'employé
	3. Le système affiche la page d'accueil
4. L'employé recherche le compte à modifier	5. Le système affiche la page de modification du compte
6. L'employé modifie les champs souhaités	

Scénario de cas d'utilisation « effectuer dépôt »

1. L'employé renseigne le login et mot de passe	2. Le système vérifie l'identité de L'employé
	3. Le système affiche la page d'accueil
4. L'employé identifie le compte pour effectuer un dépôt	5. Le système vérifie l'identité du compte
	6. Le système affiche le compte
7. L'employé renseigne le montant à déposer	8. Le système confirme le dépôt

Scénario de cas d'utilisation « effectuer retrait»

1. L'employé renseigne le login et le mot de passe	2. Le système vérifie l'identité de L'employé
	3. Le système affiche la page d'accueil
4. L'employé identifie le compte pour effectuer un retrait	5. Le système vérifie l'identité du compte
	6. Le système affiche l'état du compte
7. L'employé renseigne le montant à retirer	8. Le système vérifie si le montant est disponible

Scénario de cas d'utilisation « effectuer virement»

1. L'employé enseigne le login et le mot de passe	2. Le système vérifie l'identité de L'employé
	3. Le système affichage de la page d'accueil
4. L'employé identifie le compte émetteur	5. Le système vérifie l'identité du compte émetteur
7. L'employé identifie le compte destinataire	6. Le système affiche le compte émetteur
	8. Le système vérifie l'identité du compte destinataire
10. L'employé renseigne le montant à virer	9. Le système Affiche le compte destinataire
	11. Le système confirme le virement

Scénario de cas d'utilisation « consulter son solde »

1. Le client s'identifie en renseignant son login et son mot de passe	2. Le système vérifie l'identité du client
	3. Le système affiche la page d'accueil
4. Le client demande la page de consultation du solde ou de l'historique	5. Le système affiche le solde du compte ou l'historique

5) Diagrammes de séquence

Un diagramme de séquence indique l'interaction entre plusieurs acteurs. Les schémas suivants représentent dans chaque cas les diagrammes de séquences. Ils ont été réalisés avec l'outil Visual Paradigme [15].

II.3.1 Diagramme de séquence du scénario « création de compte »

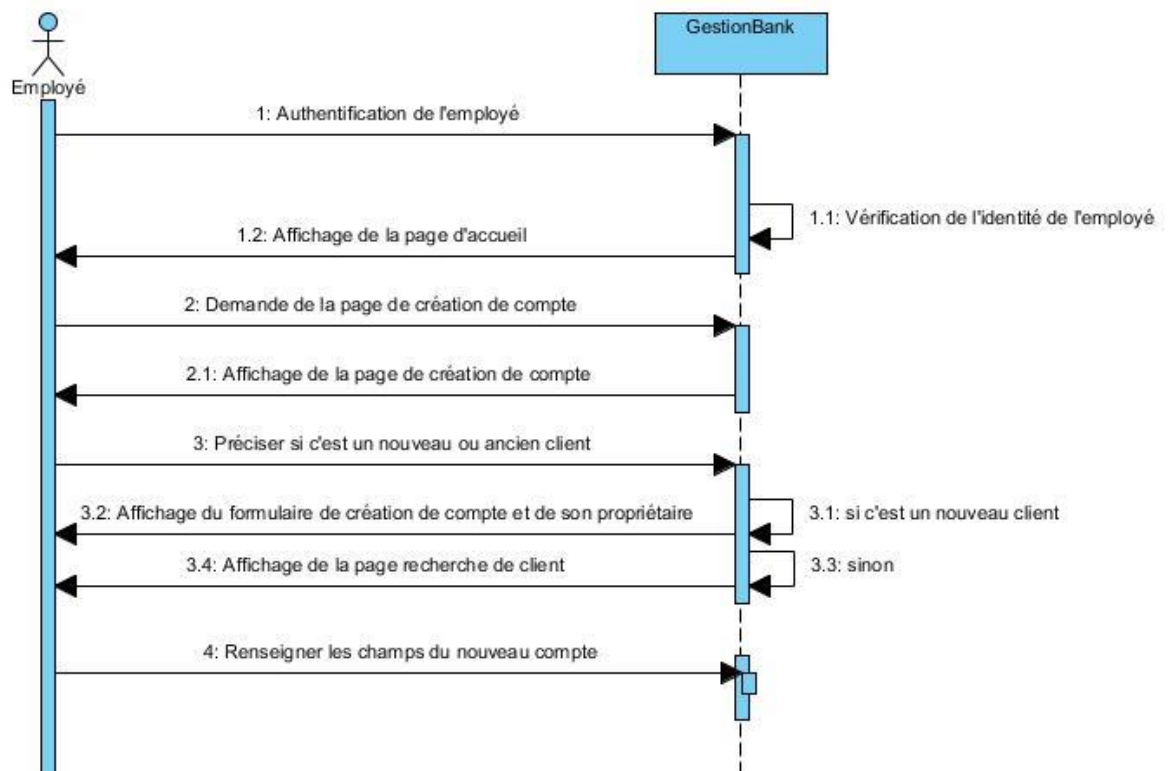
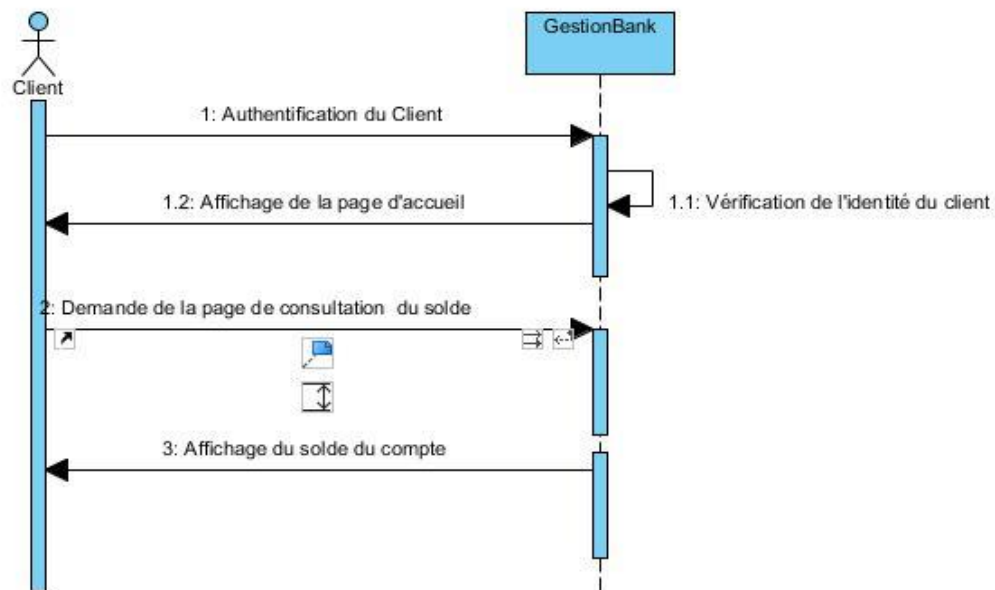


Diagramme de séquence du scénario « consulter son solde »



II.3.3 Diagramme de séquence du scénario « effectuer un dépôt »

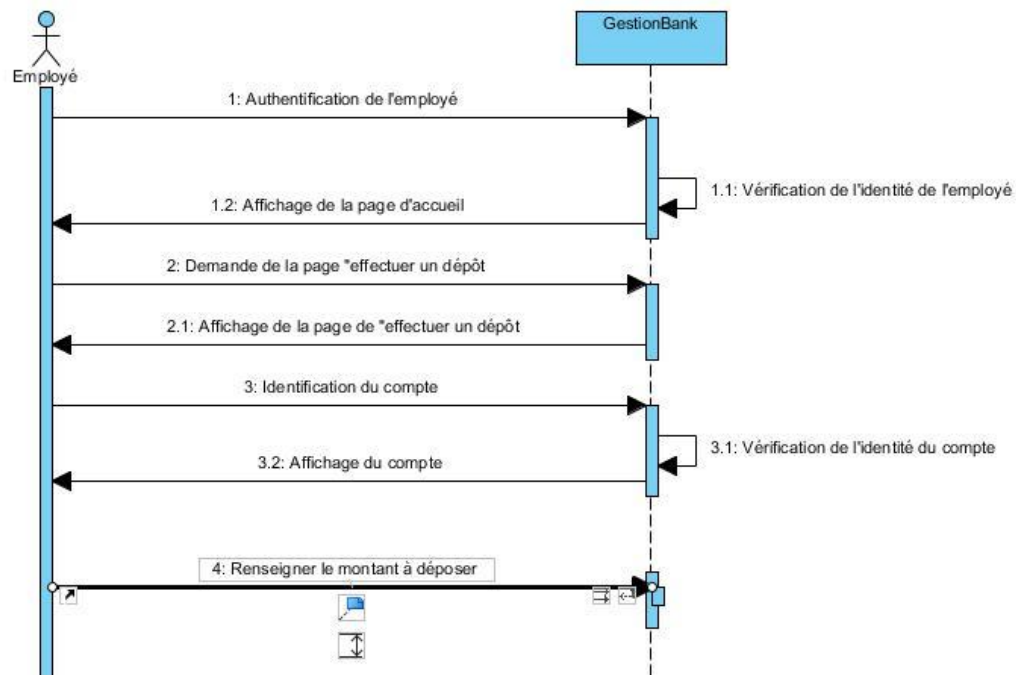
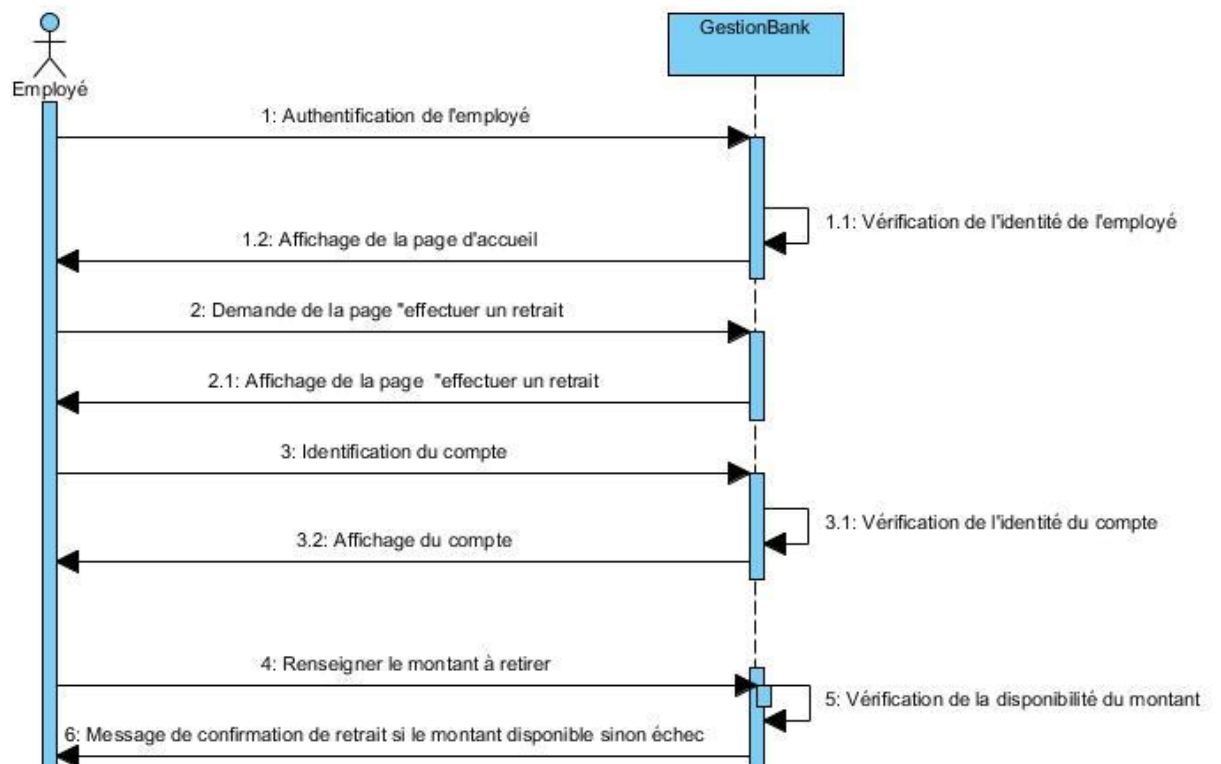


Diagramme de séquence du scénario « effectuer un retrait »



Conclusion:

Ce chapitre a été dédié à la présentation du projet et à la spécification de ses différents besoins que nous avons détaillés à travers les diagrammes de cas d'utilisation. Ceci nous a permis de bien comprendre le sujet et d'avoir une vue d'ensemble sur les fonctionnalités fondamentales de notre application.

Cette vue globale est le pilier et le support de base de notre conception que nous exposerons dans les deux chapitres : Conception et Implémentation.

Chapitre 2 : Analyse et conception

I. Introduction

Dans ce chapitre, nous présentons l'architecture que nous avons utilisés pour élaborer cet outil. Les diagrammes de classes qui décrit les intervenants sur l'application, ainsi que leurs relations avec les autres classes.

- Les données sont stockées dans une base de données MySQL
- L'application se compose de trois couches :
 - La couche DAO qui est basée sur Spring Data, JPA, Hibernate et JDBC
 - La couche Métier (Interfaces et implémentations)
 - La couche web :
 - Les contrôleurs Spring MVC
 - Les Vue basée sur Thymeleaf
- Sécuriser l'application en utilisant un système d'authentification basé sur Spring Security

Architecture technique :

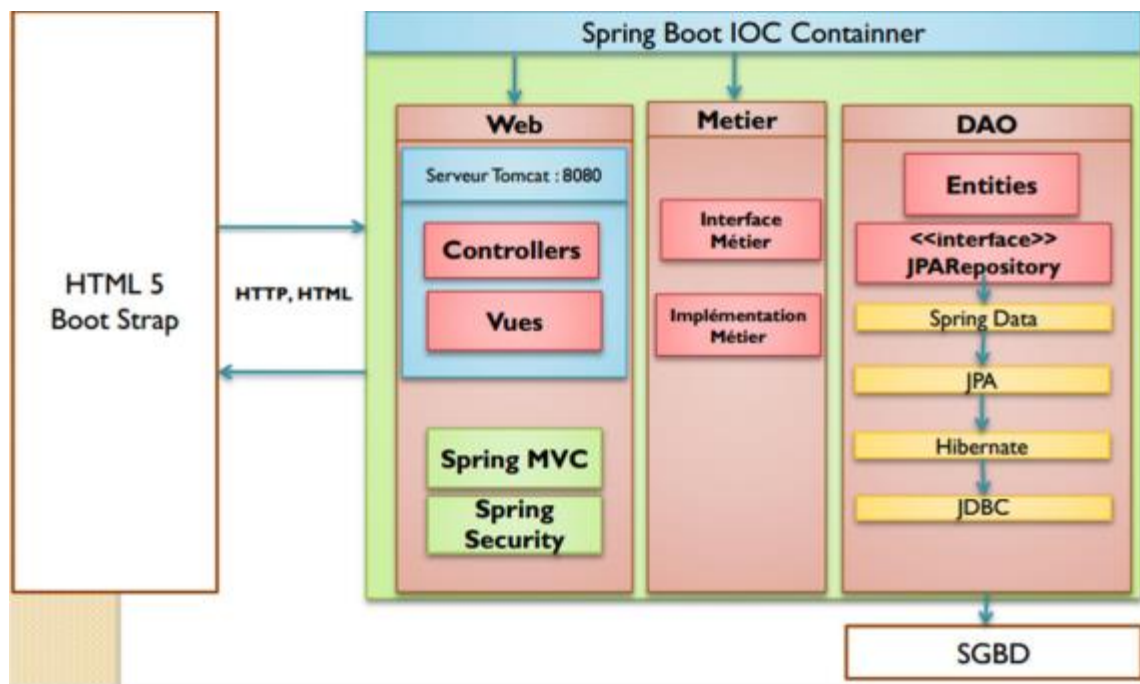
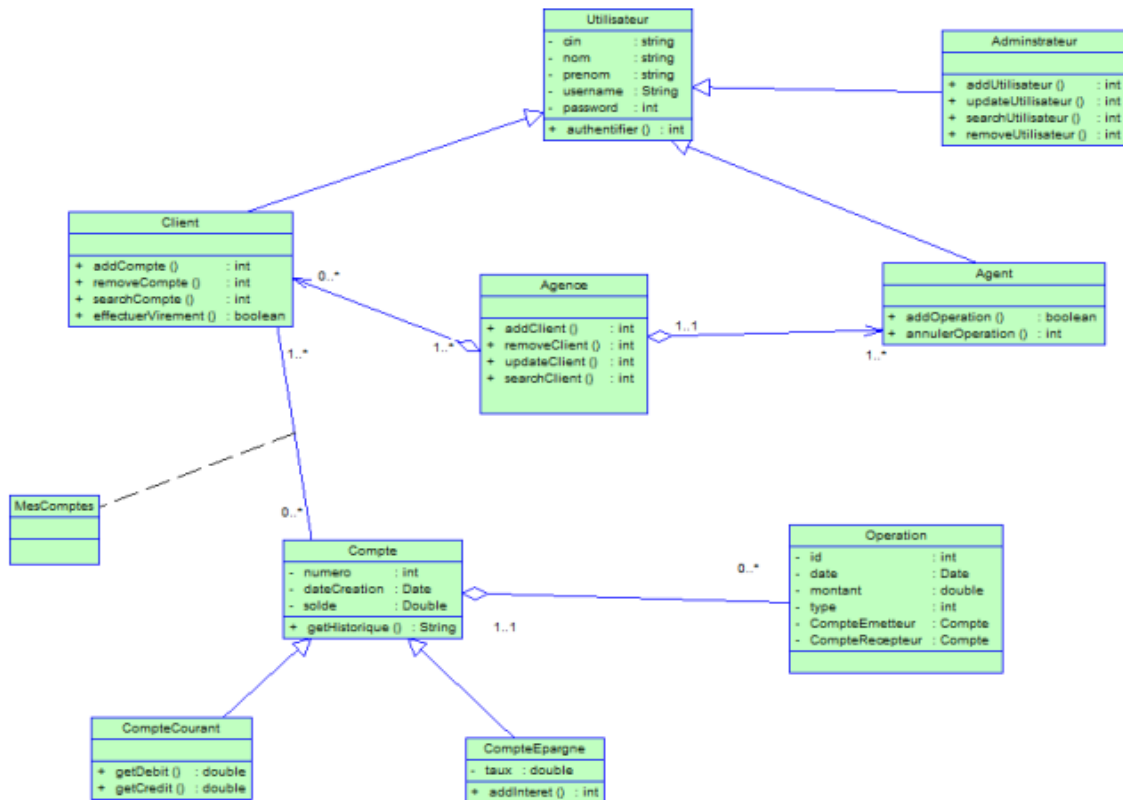
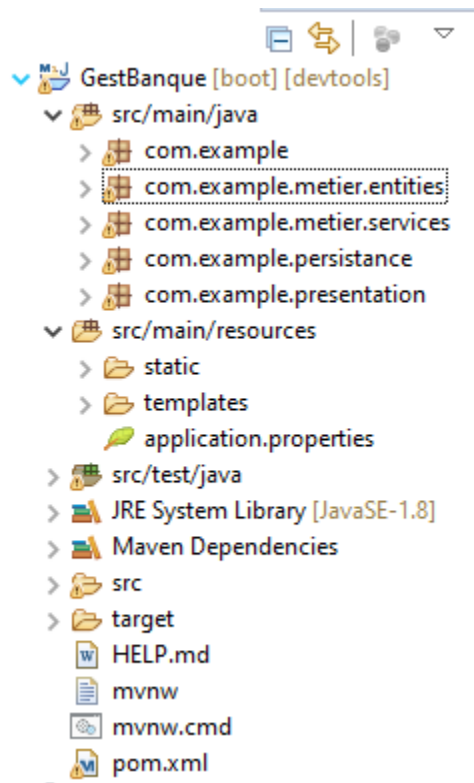


Diagramme de classes des entités



- ✓ Utilisateur : qui représente un utilisateur simple de l'application.
- ✓ Administrateur : Classe spéciale de Personne, qui représente l'Administrateur de l'application.
- ✓ Client : Cette classe hérite de la classe Utilisateur, et elle représente le client de notre application.
- ✓ Agent : La classe qui représente le rôle Agent pour notre application
- ✓ Agence : Cette classe qui représente une agence bancaire contenant l'ensemble des clients et ses propres agents.
- ✓ Compte : La classe générale qui représente un compte bancaire.
 - CompteCourant : Le compte de type courant.
 - CompteEpargne : Le compte de type épargne.
- ✓ Operation : Elle représente les différents opération effectuée sur les compte bancaires (Retrait, versement et virement).

Structure de projet :



Conclusion :

Ce chapitre est un bref récapitulatif de tous les diagrammes que nous avons performé afin de réussir notre application, ainsi que nous avons montré l'architecture en couche d'une application web Spring Boot. Une fois la conception est faite, nous arrivons à la phase de développement et de réalisation qui doit respecter les directives de la conception.

Chapitre 3: Implémentation

1) Introduction:

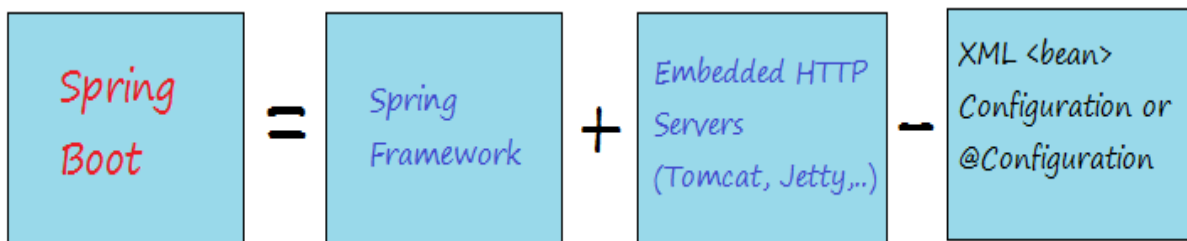
Dans ce chapitre, nous introduisons toutes les technologies liées au contexte de notre travail. On définit chaque technologie par une présentation bref ainsi que nous verrons l'implémentation d'outil pédagogique.

2) Technologies Utilisées:

Spring Boot

Spring Boot est une étape avancée qui simplifier le démarrage et le développement de nouvelles applications Spring. Avec Spring Boot, des configurations de Spring sont atténuées. Spring Boot soutient des conteneurs embarqués (Embedded containers). Cela permet des application web d'exécuter indépendamment sans déploiement sur Web Server.

Spring Boot peut s'expliquer simplement par l'illustration ci-dessous:



Les avantages du Spring Boot sont:

- ✓ Il est facile de développer l'application basées sur Spring avec Java ou Groovy.
- ✓ Il diminue énormément du temps et augmente la productivité.
- ✓ Il évite d'écrire plusieurs de codes d'expressions standard ,des Annotations et des configurations XML.
- ✓ Il est très facile d'intégrer des applications Spring Boot avec ses écosystème de Spring comme Spring JDBC, Spring ORM, Spring Data, Spring Security etc.
- ✓ Il suis l'approche "Configuration par défaut" afin de diminuer le temps et l'effort de développement.
- ✓ Il fournit des serveurs intégrés (Embedded HTTP servers) comme Tomcat, Jetty .. afin de développer et de tester des applications web à la manière la plus facilement.

JPA

La persistance des objets en Java a longtemps donné lieu à des manipulations fastidieuses. Aujourd'hui, la Java Persistence API apporte une solution performante et simple à utiliser. S'appuyant sur JDBC, JPA propose une abstraction suffisante pour que le développeur n'ait, dans la majorité des cas, pas à se préoccuper du fonctionnement de JDBC.

Hibernate

La persistance est la notion qui traite de l'écriture de données sur un support informatique. Pour sa part, le mapping objet-relationnel désigne l'interaction transparente entre le cœur d'une application, modélisé en conception orientée objet, et une base de données relationnelle.

Le rôle d'un système ORM est de convertir automatiquement, à la demande, la base de données sous forme d'un graphe d'objet. L'ORM s'appuie pour cela sur une configuration associant les classes du modèle fonctionnel et le schéma de la base de données. L'ORM génère des requêtes SQL qui permettent de matérialiser ce graphe ou une partie de ce graphe en fonction des besoins.

Thymeleaf

Thymeleaf est un moteur de Template Java moderne côté serveur pour les environnements Web et autonomes.

L'objectif principal de Thymeleaf est d'intégrer des modèles naturels élégants à votre flux de travail de développement : un code HTML pouvant être correctement affiché dans les navigateurs et fonctionnant également en tant que prototypes statiques, ce qui permet de renforcer la collaboration au sein des équipes de développement.

Avec des modules pour Spring Framework, une multitude d'intégrations avec vos outils favoris et la possibilité d'intégrer vos propres fonctionnalités, Thymeleaf est idéal pour le développement Web JVM HTML5 moderne - bien qu'il puisse faire beaucoup plus.

Bootstrap :

Bootstrap est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

MySQL :

Un système de gestion de bases de données relationnelles (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.

XML :

Un métalangage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG...

3) Les dépendance MAVEN

Le répertoire **src** ne doit contenir que des fichiers sources apportés au projet. Dans cette idée, des bibliothèques externes utilisées par le projet ne doivent être que des liens vers d'autres artefacts Maven et surtout pas copiées dans le répertoire **src** du projet.

Maven propose de le définir par configuration dans le pom.xml. C'est ensuite le plugin Maven de gestion de dépendances qui ira télécharger sur les repositories distants les fichiers jar indiqués comme dépendances, s'ils ne se trouvent pas dans le répertoire local.

Tous les artifacts gérés par Maven sont identifiés par 4 informations :

groupId : Définit l'organisation ou groupe qui est à l'origine du projet. Il est formulé sous la forme d'un package Java (org.archetypejs par exemple)

artifactId : Définit le nom unique dans le groupe pour le projet.

version : Définit la version du projet. Les numéros de version sont souvent utilisés pour des comparaisons et des mises à jour.

type : Type de projet. On trouvera le plus souvent : jar, war, maven-plugin, pom...

La configuration à ajouter dans le fichier pom.xml est la suivante :

✓

Spring framework

```
<!-- Spring -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${org.springframework-version}</version>
  <exclusions>
    <!-- Exclude Commons Logging in favor of SLF4j -->
    <exclusion>
      <groupId>commons-logging</groupId>
      <artifactId>commons-logging</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
```

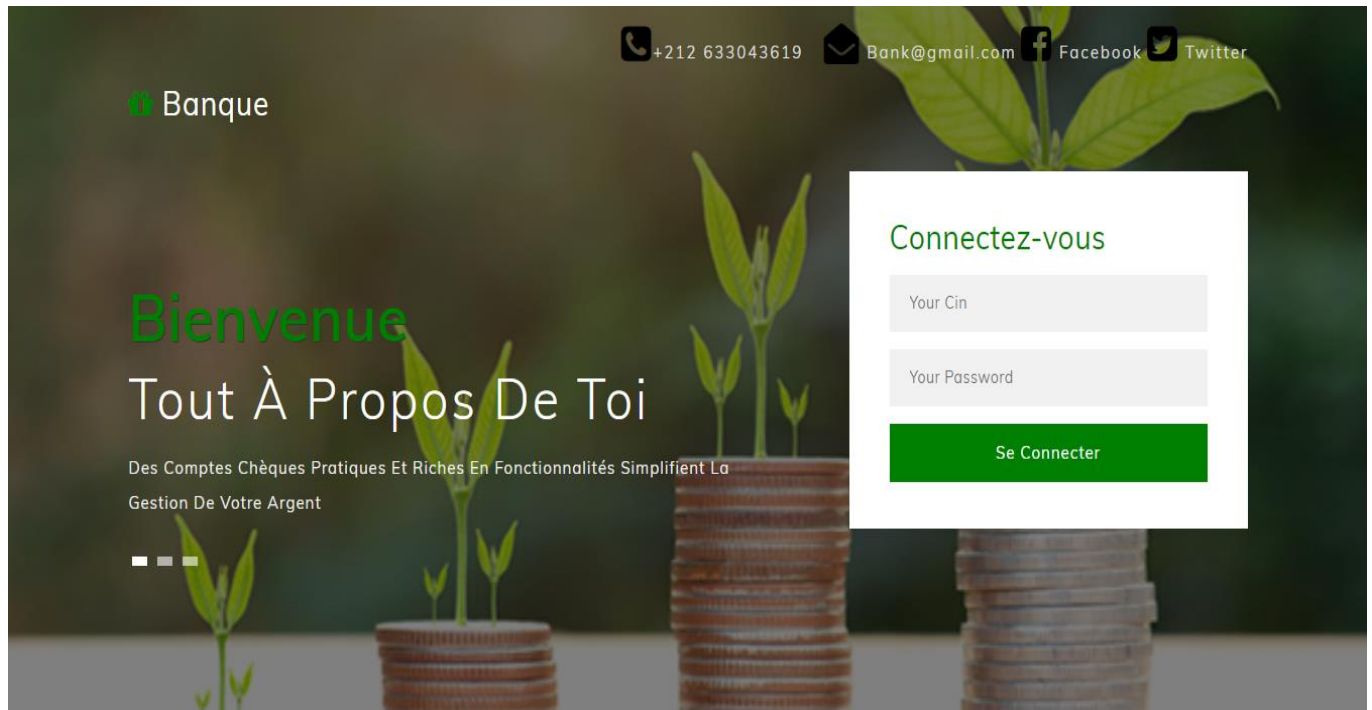
✓

MySQL Connector

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.6</version>
</dependency>
```

4) Implémentation de l'application de gestion des comptes bancaires :

a. Authentification :



L'interface graphique de Login permet à l'utilisateur de s'authentifier à l'application en fournissant son Login et Mot de passe. En cliquant sur le bouton «Se Connecter », l'application cherche un utilisateur dans la base de données avec ce login et ce mot de passe, ; s'il existe il sera redirigé vers la page d'accueil.

Pour Admin :



➤ la page d'accueil

Derniers Clients Inscrits

CIN	NOM	PRENOM	THELEPHONE	DateInscrit
Q347543	glouib	chaima		2019-10-12
chamchouma	glouib	chaima	0633043619	2019-09-07
clien1	glouib	fdfgd	8676	2019-08-13

localhost:8083

La page d'accueil contient quelques statistiques tel que le nombre des opérations de type virement effectuées dans le jour courant, en gauche nous avons un menu qui va nous permet de naviguer dans les différents pages l'application.

➤ Nouvelle Agence et La liste des Agences :

CODE	NOM	ADRESSE	CONTACT	EMAIL	ACTION
jj	hfhfh		hh	hh@gmail.com	Supprimer Editer

- ✓ Le formulaire d'ajout : Permet d'ajouter une agence.
- ✓ Une liste : Consulter la liste des agences avec la possibilité de les supprimer ou les modifier.

➤ Nouveau Agent et La liste des Agents :

CIN	NOM	PRENOM	AGENCE	ADRESSE	DATE_DE_RECRUTEMENT	ACTION
agen	s	ss	hfhfh	s	2019-08-06	Supprimer Editer
glouiba	glouib	chaiaa	hfhfh	335 EL WIAAM khouribga	2019-11-01	Supprimer

- ✓ Le formulaire d'ajout : Permet d'ajouter un agent.
- ✓ Une liste : Consulter la liste des agents avec la possibilité de les supprimer ou les modifier.

➤ Transactions :

AGENCE	GENRE	DATE DU :	DATE AU :	ACTION
hfhfh	Depots	10/05/2019	13/09/2019	<button>Afficher</button>

CODE	MONTANT	DATE_OPERATION	TYPE
25	22.0	2019-09-06 01:00:00.0	versement
25	22.0	2019-09-06 01:00:00.0	versement
25	22.0	2019-09-06 01:00:00.0	versement

L'admin peut consulter les Operations effectuées dans une période précis.

Pour Agent :

➤ Nouveau Client et La liste des Clients :

GestionBanqueAgent

Setting

Menu

Clients

Comptes

Operations

Nouveau Client/Liste Des Clients

CIN Du Client

Nom Du Client

Prenom Du Client

Agence Du Client

Adresse Du Client

Date D'INSCRIT

Password Du Client

.....

Action

Enregistrer

Liste Des Clients

Mot Cle

Rechercher

CIN	NOM	PRENOM	AGENCE	ADRESSE	Date_D'INSCRIT	ACTION
clien	gloui	fdfgd	hfhhh	gh	2019-08-05	Supprimer Editer
clien1	gloub	fdfgd	hfhhh	gh	2019-08-13	Supprimer Editer

- ✓ formulaire d'ajout : Permet d'ajouter un client
- ✓ Une liste : Consulter la liste des clients avec la possibilité de les supprimer ou les modifier.

➤ Nouveau Comptes et La liste des Comptes :

GestionBanqueAgent

Setting

Menu

Clients

Comptes

Operations

TYPE

Compte_Courant

DATE_CREATION

jj/mm/aaaa

CLIENTS

clien
clien1
Q347543
chamchouuma

ACTIVE

0

SOLDEMAX

0,0

DECOUVERT

0,0

TAUX

0,0

Action

Enregistrer

Liste Des Comptes

Mot Cle

Rechercher

CODE	TYPE	ACTIVE	SOLD	DATE_CREATION	AGENT	CLIENTS	ACTION
402881896d032cf5016d0334dc130001	compteCourant	true	134.0	2019-08-08	agen	clien	Supprimer Editer
402881896d033bec016d0343ef360004	Compte_Courant	false	88.0	2019-09-13	agen	clien clien1 chamchouuma	Supprimer Editer

- ✓ formulaire d'ajout : Permet d'ajouter un client
- ✓ Une liste : Consulter la liste des clients avec la possibilité de les supprimer ou les modifier.

➤ Nouvelle Opération et La liste des Operations :

The screenshot shows the 'GestionBanqueAgent' application interface. On the left is a dark sidebar menu with 'Clients', 'Comptes', and 'Operations'. The main area is titled 'Nouvelle Operation/Liste Des Operations'. It contains three forms for different transaction types: 'Retrait' (highlighted with an orange border), 'VERESSEMENT' (highlighted with a green border), and 'VIREMENT' (highlighted with a red border). Each form has fields for 'Code_Compte_Emetteur', 'Code_Compte_Recepteur', and 'Montant' (set to 0,0), with a 'Valider' button at the bottom. Below these forms is a table titled 'Liste Des Operations'.

ID	TYPE	MONTANT	DATE_OPERATION	CODE_COMPTTE_EMETTEUR	CODE_COMPTTE_RECEPTEUR
25	versement	22.0	2019-09-06 01:00:00.0	---	402881896d033bec016d0343ef360004
24	retirement	22.0	2019-09-06 01:00:00.0	402881896d032cf5016d0334dc130001	---
1	retirement	34.0	2019-07-10 01:00:00.0	402881896d032cf5016d0334dc130001	---

- ✓ Retrait : Effectuer un retrait en fournissant le code du compte et le montant.
- ✓ Versement : Effectuer un versement en donnant les informations nécessaires.
- ✓ Virement : Il permet à l'agent de saisir le code du compte émetteur et le code du compte récepteur ainsi que le montant.
- ✓ Liste Des opérations : Contient la liste des opérations déjà effectuées.

Pour Client :



➤ Liste Des Comptes Du Client Courant :

This screenshot shows the 'Mes Comptes' section of the 'GestionBanqueClient' application. The sidebar menu is identical to the previous screenshot. The main content area has a light blue header with the title 'Mes Comptes'. Below this is a table with 8 columns: CODE, TYPE, SOLD, ACTIVE, DATE_DE_CREATION, DECOUVERT, SOLDMAX, and TAUX. The table contains four rows of account data.

CODE	TYPE	SOLD	ACTIVE	DATE_DE_CREATION	DECOUVERT	SOLDMAX	TAUX
402881896d032cf5016d0334dc130001	compteCourant	134.0	true	2019-08-08	222.0	333.0	---
402881896d033bec016d0343ef360004	Compte_Courant	88.0	false	2019-09-13	---	---	---
402881896d033bec016d03891f1e0009	Compte_Epargne	0.0	false	2019-09-11	---	---	---
402881896d033bec016d03896ff0000a	Compte_Epargne	0.0	false	2019-09-10	---	---	---

➤ Liste Des Operations effectuées par le Client Courant :

GestionBanqueClient						Setting
Menu						
Mes Comptes						
Mes Operations						
Mes Operations						
id	TYPE	MONTANT	DATE_OPERATION	CODE_COMPTE_EMETTEUR	CODE_COMPTE_RECEPTEUR	
24	retirement	22.0	2019-09-06 01:00:00.0	402881896d032cf5016d0334dc130001	---	
25	versement	22.0	2019-09-06 01:00:00.0	---	402881896d033bec016d0343ef360004	

Conclusion :

A travers ce dernier chapitre, nous avons décrit, en premier lieu, les outils et les technologies utilisés. La deuxième partie de notre réalisation a été dédiée à la présentation de notre travail, ainsi achevé, moyennant des captures d'écrans et une description des fonctionnalités offertes par l'application.

Conclusion et perspectives

Conclusion et perspectives Au terme de ce rapport, nous dressons le bilan complet de notre travail qui se situe dans le cadre de stage de 4ème année Génie Informatique à l'Ecole Nationale des Sciences Appliquées Khouribga. Notre travail a consisté à implémenter un outil pédagogique des différents Frameworks JEE principaux à savoir Spring Boot, Hibernate, Thymeleaf, nous avons développé une simple application de gestion des comptes bancaires en utilisant ces Frameworks, pour que l'utilisateur puisse voir un exemple concret d'une application web. Ce rapport présente les différentes étapes lors de l'élaboration du projet, nous avons d'abord présenté le contexte général., par la suite, nous avons entamé la conception générale et détaillée de notre solution. Notre projet est clôturé par une phase de réalisation effectuée, cette phase comporte le développement de l'application de gestion des comptes bancaires et l'outil pédagogique. Nous pensons que ce travail n'est pas encore fini, reste à faire évoluer l'outil pédagogique, élargir les possibilités des applications web en rajoutant d'autres architectures et technologies de développement web. Pour conclure, ce stage nous a été bénéfique, dans la mesure où il nous a permis de découvrir le domaine de développement web et ses défis persistants et de mettre en œuvre nos connaissances théoriques et notre créativité pour trouver une solution innovatrice et performante qui facilite l'apprentissage.

Bibliographie

Mohamed Youssfi tutorials:

<https://www.youtube.com/watch?v=GJU7zLW-Xi8>
<https://www.youtube.com/watch?v=AooDDD3UcsA>
<https://www.youtube.com/watch?v=rQOLlremOAK>
<https://www.youtube.com/watch?v=RKgSU-NW9I0>
<https://www.youtube.com/watch?v=u2z-oTMAVKY>