

Rapport de Projet de Fin d'Etudes

Pour l'obtention du

Diplôme National en Licence Appliquée

Titre : Développement d'un projet E-Banking en JEE

Organisme d'accueil :

BFI



Soutenu le : 28/06/2018

Auteur : Missaoui Sadek

INSAT, LA3GL

Tuteur professionnel : Bassi Selim

Directeur d'intégration

Tuteur académique : Abderrazak Jemai

Maître de conférences en informatique

Rapport de Projet de Fin d'Etudes
Pour l'obtention du
Diplôme National en Licence Appliquée

Titre : Développement d'un projet E-Banking en JEE

Auteur : Missaoui Sadok
INSAT, LA3GL

AUTORISATIONS DE DEPOT

<p>Tuteur professionnel : Bassi Selim</p> <p>Date __/__/__</p> <p>Cachet & Signature obligatoires</p>	<p>Tuteur académique : Jemai Abderrazak</p> <p>Date __/__/__</p> <p>Signature obligatoire</p>
---	---

Remerciements

Je tiens, tout d'abord, à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

Je dédie ce modeste travail et ma profonde gratitude à mon père et ma mère pour leurs patiences et encouragements.

Au terme de mon projet de fin d'études, qu'il me soit permis d'exprimer mes sentiments de gratitude aux différentes personnes qui m'ont encadrés et soutenues tout au long de notre travail rendant ainsi possible l'accomplissement du présent projet.

Je tiens tout particulièrement à exprimer ma profonde gratitude et respect à Monsieur **BASSI Selim**, de m'avoir offert l'opportunité d'effectuer mon stage au sein de BFI.

Notre profonde gratitude s'adresse tout particulièrement à Monsieur **JEMAI Abderrazak**, qui en dépit de ses multiples occupations et sollicitations, a accepté de diriger, de lire et d'apporter des correctifs à ce travail sans oublier ses conseils et suggestions éclairées.

Mes remerciements s'adressent aussi à toutes l'équipe du département informatique de BFI qui m'a accompagné et aidé d'une manière et d'une autre à la réalisation du présent travail.

Table des matières

Contenu

Introduction Générale.....	1
Chapitre I : Présentation de l'entreprise et étude de l'existant	3
I.1. Introduction :	3
I.2. Présentation de l'organisme d'accueil :	3
I.3. Problématique :	3
I.4. Analyse de l'existant :	4
I.4.1. Solution Existante :	4
I.4.2. Critiques :	4
I.5. La solution proposée:	4
I.5.1. Les spécifications fonctionnelles :	5
I.5.2. Spécifications non-fonctionnelles :	5
I.6. Conclusion :	6
Chapitre II : Technologies et Frameworks : état de l'art	7
II.1. Introduction :	7
II.2. Technologies et outils de développements :	7
II.2.1. Outils de conception :	7
II.2.2. Langage de modélisation :	7
II.2.3. Environnement technique du travail :	7
II.2.4. Frameworks de développement :	8
II.2.5. Outils de développements :	9
II.2.6. Langage de programmation :	9
II.3. Conclusion :	9
Chapitre III : Conception	10
III.1. Introduction :	10
III.2. Capture des besoins:	10

III.2.1	Identification des acteurs :.....	10
III.2.2	Description des acteurs :.....	10
III.2.3	Diagrammes de cas d'utilisation :	11
III.3.	Backlog du produit :	13
III.4.	Phase d'analyse (diagramme de classes d'analyse):.....	15
III.5.	Phase de conception :	16
III.5.1.	Cas d'utilisation « S'authentifier »:.....	16
i.	Description textuelle :.....	16
ii.	Diagramme de classes de cas d'utilisation :	17
iii.	Diagramme de séquences :	17
III.5.2.	Cas d'utilisation « Commander un chéquier »:	18
i.	Description textuelle :.....	19
ii.	Diagramme de classes de cas d'utilisation :	19
iii.	Diagramme de séquences :	19
III.5.3.	Cas d'utilisation « Demander un virement intrabancaire »:	20
i.	Description textuelle :.....	20
ii.	Diagramme de classes de cas d'utilisation :	21
iii.	Diagramme de séquences :	21
III.5.4.	Cas d'utilisation « Demander un virement interbancaire »:	22
i.	Description textuelle :.....	22
ii.	Diagramme de classes de cas d'utilisation :	23
iii.	Diagramme de séquences :	24
III.5.5.	Cas d'utilisation « Consulter les comptes »:	25
i.	Description textuelle :.....	25
ii.	Diagramme de classes de cas d'utilisation :	25
iii.	Diagramme de séquences :	26
III.5.6.	Cas d'utilisation « Consulter les commandes de chéquier »:	26
i.	Description textuelle :.....	27
ii.	Diagramme de classes de cas d'utilisation :	27
iii.	Diagramme de séquences :	27
III.5.7.	Cas d'utilisation « Mettre à jour les coordonnées du client »:	28
i.	Description textuelle :.....	28

ii.	Diagramme de classes de cas d'utilisation :	28
iii.	Diagramme de séquences :	28
III.5.8.	Cas d'utilisation « Mettre à jour le mot de passe »:	29
i.	Description textuelle :	29
ii.	Diagramme de classes de cas d'utilisation :	30
iii.	Diagramme de séquences :	30
III.5.9.	Cas d'utilisation « Consulter les abonnés »:	31
i.	Description textuelle :	31
ii.	Diagramme de classes de cas d'utilisation :	31
iii.	Diagramme de séquences :	32
III.5.10.	Cas d'utilisation « Consulter les virements »:	33
i.	Description textuelle :	33
ii.	Diagramme de classes de cas d'utilisation :	33
iii.	Diagramme de séquences :	34
III.6.	Conclusion :	34
Chapitre IV :	Codage et déploiement	35
IV.1.	Introduction :	35
IV.2.	Codage :	35
IV.3.	Déploiement :	35
IV.4.	Conclusion :	36
Chapitre V :	Tests	37
V.1.	Introduction :	37
V.2.	Tests :	37
V.2.1.	Cas d'utilisation « S'authentifier »:	37
V.2.2.	Cas d'utilisation « Commander un chéquier »:	37
V.2.3.	Cas d'utilisation « Demander un virement interbancaire »:	37
V.2.4.	Cas d'utilisation « Consulter les comptes »:	38
V.2.5.	Cas d'utilisation « Mettre à jour les coordonnées du client »:	38
V.2.6.	Cas d'utilisation « Consulter les abonnés »:	39
V.3.	Conclusion :	39
Conclusion		40

Liste des figures

Figure 1 : Architecture générale d'application web	8
Figure 2 : Héritage entre les acteurs	10
Figure 3 Diagramme de cas d'utilisation de la gestion de données personnelles par l'abonné	11
Figure 4 : Diagramme de cas d'utilisation de l'administrateur	12
Figure 5 : Diagramme de cas d'utilisation de la consultation de données financières par l'abonné	12
Figure 6 : Diagramme de cas d'utilisation de la gestion des opérations par l'abonné	13
Figure 7 : Diagramme de classes de domaine SI	15
Figure 8 : Diagramme de classes de domaine Persistantes	16
Figure 9 : Diagramme de classes de cas d'utilisation « s'authentifier »	17
Figure 10 : Diagramme de séquences de cas d'utilisation « s'authentifier »	18
Figure 11 : Diagramme de classes de cas d'utilisation « Commander un chéquier »	19
Figure 12 : Diagramme de séquences de cas d'utilisation « Commander un chéquier »	20
Figure 13 : Diagramme de classes de cas d'utilisation « Demander un virement intrabancaire »	21
Figure 14 : Diagramme de séquences de cas d'utilisation « Demander un virement intrabancaire »	22
Figure 15 : Diagramme de classes de cas d'utilisation « Demander un virement interbancaire »	23
Figure 16 : Diagramme de séquences de cas d'utilisation « Demander un virement interbancaire »	24
Figure 17 : Diagramme de classes de cas d'utilisation « Consulter les clients »	25
Figure 18 : Diagramme de séquences de cas d'utilisation « Consulter les comptes »	26
Figure 19 : Diagramme de séquences de cas d'utilisation « Consulter les opérations »	26
Figure 20 : Diagramme de classes de cas d'utilisation « Consulter les commandes de chéquier »	27
Figure 21 : Diagramme de séquences de cas d'utilisation « Consulter les commandes de chéquier »	27
Figure 22 : Diagramme de classes de cas d'utilisation « Mettre à jour les coordonnées du client »	28
Figure 23 : Diagramme de séquences de cas d'utilisation « Mettre à jour les coordonnées du client » ...	29
Figure 24 : Diagramme de classes de cas d'utilisation « Mettre à jour le mot de passe de l'abonné »	30
Figure 25 : Diagramme de séquences de cas d'utilisation « Mettre à jour le mot de passe de l'abonné »	30
Figure 26 : Diagramme de classes de cas d'utilisation « Consulter les abonnés »	31
Figure 27 : Diagramme de séquences de cas d'utilisation « Consulter les abonnés »	32
Figure 28 : Diagramme de séquences de cas d'utilisation « gérer les bénéficiaires »	33
Figure 29 : Diagramme de classes de cas d'utilisation « Consulter les virements »	34
Figure 30 : Diagramme de séquences de cas d'utilisation « Consulter les virements »	34
Figure 31 : Diagramme de composants de l'application E-Banking	35
Figure 32 : Diagramme de déploiement de l'application E-Banking	36
Figure 33 : interface d'identification	37
Figure 34 : interface d'identification message d'erreur	37

Figure 35 : interface de commande chéquier	37
Figure 36 : interface de création d'un virement interbancaire	38
Figure 37 : interface de vérification du code de validation.....	38
Figure 38 : interface de consultation des comptes	38
Figure 39 : interface de consultation des opérations.....	38
Figure 40 : interface de mise à jour des coordonnées du client	39
Figure 41 : les messages d'erreur.....	39
Figure 42 : interface de consultation des abonnés	39
Figure 43 : l'option d'ajouter/modifier un abonné	39

Liste des tables

Table 1 : Environnement technique du travail.....	7
Table 2 : Description détaillée des acteurs	11
Table 3 : Backlog du produit.....	13
Table 4 : Description textuelle du cas d'utilisation « s'authentifier ».....	16
Table 5 : Description textuelle du cas d'utilisation « Commander un chéquier »	19
Table 6 : Description textuelle du cas d'utilisation « Demander un virement intrabancaire »	20
Table 7 : Description textuelle du cas d'utilisation « Demander un virement interbancaire »	22
Table 8 : Description textuelle du cas d'utilisation « Consulter les comptes ».....	25
Table 9 : Description textuelle du cas d'utilisation « Consulter les commandes de chéquier ».....	27
Table 10 : Description textuelle du cas d'utilisation « Mettre à jour les coordonnées du client »	28
Table 11 : Description textuelle du cas d'utilisation « Mettre à jour le mot de passe ».....	29
Table 12 : Description textuelle du cas d'utilisation « Consulter les abonnés ».....	31
Table 13 : Description textuelle du cas d'utilisation « Consulter les comptes ».....	33

Les acronymes

J2EE: Java Enterprise Edition

DAO: Data Access Object

UML: Unified Modeling Language

HTTP: Hyper Text Transfer Protocol

JSF: Java Server Faces

JSON: JavaScript Object Notation

MVC: Model-View-Controller

JDBC: Java Data Base Connectivity

JSP: Java Server Page

SGBDR: Système de Gestion de Bases de Données Relationnelles

SPA: Single Page Application

Introduction Générale

Pressés par la modernisation croissante, par l'invasion des nouvelles technologies et par l'émergence des nouveaux acteurs sur le marché des services financiers, les banques cherchent à adapter des solutions non seulement pour asseoir leurs spécificités, mais aussi pour se bâtir une identité propre.

En fait, et afin de fidéliser ses clients et attirer une clientèle potentielle, la banque est amenée à approcher différemment ses clients en se différenciant, soit dans ses moyens de traitement de l'information et de la communication bancaire, soit dans l'exécution des différentes fonctions liées directement à la banque. On parle ainsi de la révolution technologique en matière d'information et de communication qui a permis de nouvelles formes de distribution de services bancaires telle que la banque électronique.

Cette entité est reconnue la plupart du temps par son appellation commerciale d'origine anglais: « E- Banking ». D'ailleurs, ce terme englobe le processus par lequel un consommateur peut gérer ses transactions bancaires électroniquement sans être obligé de visiter une succursale physique.

En outre, l'adoption de la banque électronique devient une nécessité pour la banque qui souhaite maintenir sa part de marché, ainsi que retenir ses clients. Ce concept est entré dans les mœurs de plusieurs institutions financières pour le traitement de leurs opérations courantes, dans la mesure où il répond aux attentes des utilisateurs, notamment les clients, les fournisseurs et les partenaires institutionnelles.

Notons toutes fois que l'E-Banking du point de vue de ses acteurs tel que la banque, les clients, n'est ni une technologie, ni une finalité en soi mais bien une pratique d'affaire et plus précisément une pratique d'affaire innovante.

Dans ce contexte, ce projet de fin d'études a pour but de concevoir et développer une application d'E-Banking comprenant 2 parties :

Introduction Générale

- Une application web du système d'informations (SI) qui utilise les services web REST pour transférer les données stockées par la banque (client, compte, ...) sous format JSON.
- Une application Web Principale Persistante qui consiste d'une interface web Abonne, pour les utilisateurs, qui sert à consulter les données financières et gérer les virements et les opérations et d'une interface web Administrateur, pour les employés de la banque, qui sert à manipuler les abonnées et leurs virements/bénéficiaires.

Notre travail s'intitulant « Développement d'un projet E-Banking en JEE » est composé de cinq chapitres :

- Dans le premier chapitre, on présente l'entreprise et étudiera l'existant.
- Le deuxième chapitre présentera état de l'art : Technologies+Frameworks.
- Le troisième chapitre portera sur les étapes de conception de l'application.
- Le quatrième chapitre sera dédié à la partie déploiement et codage de ce projet où on va détailler l'architecture des deux applications et le schéma de la base de données.
- Et le dernier chapitre présentera les tests de validation.

On finira par une conclusion générale ainsi que la proposition de quelques perspectives.

Chapitre I : Présentation de l'entreprise et étude de l'existant

I.1. Introduction :

Dans ce chapitre, nous abordons l'environnement du stage en présentant l'entreprise d'accueil BFI. Puis, nous présentons la problématique et l'analyse de l'existant tel que la solution existante et ses critiques. Nous clôturons ce chapitre par la description du plan de notre projet en précisant les spécifications fonctionnelles et non- fonctionnelles.

I.2. Présentation de l'organisme d'accueil :

BFI est spécialisée depuis 1994 dans l'édition et l'intégration de solutions logicielles destinées aux banques et institutions financières. Bien implantée sur le marché africain, **BFI** a réussi à se forger une réputation de très haut niveau, en équipant plus de 200 clients dans 22 pays différents notamment avec des institutions de renommée mondiale. Ainsi, **BFI** offre à ses clients une gamme de solutions couvrant la globalité des besoins d'une banque en matière de systèmes d'information front et back-office ainsi que tous les métiers de la banque : gestion des clients, gestion des risques, contrôle des engagements, des crédits, trésorerie, marché des capitaux, moyens de paiement, comptabilité... leur permettant ainsi de disposer d'un système d'information à la hauteur des enjeux stratégiques. [1]

I.3. Problématique :

A l'heure actuelle, les entreprises ont besoin de suivre de près les changements sur leurs systèmes d'information pour pouvoir répondre aux exigences des différents utilisateurs. Ce besoin est apparu dès qu'il y a eu une prise de conscience de la situation des entreprises à cause de la mauvaise administration de la gestion des changements.

Cela a engendré de graves conséquences, entre autres, la non-conformité aux termes et aux conditions prédéfinies ainsi que le non-respect des délais. C'est dans ce cadre que BFI a constaté qu'il est urgent de développer une application de gestion des changements pour répondre à son propre besoin de la gestion de nouveaux projets ou de la modification des applications déjà réalisées.

Pour cette raison, nous avons intégré le département informatique au sein de BFI pour accomplir cette mission. Notre projet consiste alors à réaliser une application permettant à la banque en question de contrôler, de communiquer et de répondre plus efficacement à différentes opérations par les utilisateurs.

I.4. Analyse de l'existant :

BFI allie la maîtrise des nouvelles technologies de l'information à la connaissance du métier de ses clients. Ses équipes pluridisciplinaires maîtrisent un large éventail de compétences, toutes au service d'un même objectif : optimiser la croissance de ses clients.

I.4.1. Solution Existante :

Carthago est une solution logicielle globale bancaire couvrant l'ensemble des métiers de la banque de détail notamment la gestion des clients, des dépôts à vue et de l'épargne, de la comptabilité et du contrôle budgétaire. [2]

Les solutions Carthago sont développées autour de notre Framework transactionnel garantissant évolutivité et indépendance. Architecture fonctionnelle modulaire basée sur un socle technique conforme à l'état de l'art des technologies JEE, garantissant interopérabilité, évolutivité, pérennité et montée en charge. [3]

I.4.2. Critiques :

- L'accessibilité est la disponibilité et le caractère utilisable d'une ressource pour des groupes spécifiques.
- Les risques opérationnels et de sécurité sont liés à des insuffisances dans la fiabilité ou l'intégrité du système. Les considérations de sécurité jouent un rôle prépondérant car les systèmes d'information peuvent faire l'objet d'attaques autant internes qu'externes.

I.5. La solution proposée:

Le travail à faire est de concevoir et développer une application web « E-Banking ». Une application Spring MVC qui sera installée sur un terminal web développé en langage de programmation JAVA dédiée à la fois aux clients de la banque et aux administrateurs. Cette application est divisée sous forme de deux applications ; l'une constitue le système d'informations fourni normalement par la banque et communiquant avec l'interface en utilisant les Web Services REST et l'autre présente l'application persistante qui présente l'application principale qui manipule les données.

I.5.1. Les spécifications fonctionnelles :

L'application permet aux utilisateurs de:

- S'authentifier : Pour bénéficier des fonctionnalités offertes de l'application, l'utilisateur s'authentifier en saisissant son code abonné/nom utilisateur et son mot de passe.
- Accéder aux services clients :
 - Consulter les comptes et leurs opérations
 - Vérifier les bénéficiaires valides.
 - Demander et/ou consulter un chéquier.
 - Demander et/ou consulter une carte bancaire.
 - Effectuer des virements sécurisés simples intra-bancaires.
 - Effectuer des virements sécurisés simples inter-bancaires.
- Accéder aux données personnelles :
 - Changer le mot de passe.
 - Mettre à jour les coordonnées du client dans la base de données de la banque.
- Accéder aux services administrateurs :
 - Consulter la liste des abonnés.
 - Ajouter, manipuler et supprimer des abonnés.
 - Consulter les bénéficiaires de chaque abonné.
 - Ajouter, manipuler et supprimer des bénéficiaires.
 - Consulter les virements intrabancaires et interbancaires.
 - Valider ou invalider les virements bancaires.

I.5.2. Spécifications non-fonctionnelles :

Outre que les besoins fonctionnels cités précédemment, pour bien répondre aux exigences des utilisateurs, le système devra pouvoir assurer les besoins non fonctionnels suivants :

- L'ergonomie et la convivialité :
 - L'application fournira une interface conviviale, simple et homogène à utiliser et qui ne requiert aucun pré requis, donc elle pourra être exploitable par tout type d'utilisateurs (même les non informaticiens).
 - L'utilisateur doit être guidé lors de la saisie de certaines informations, afin de respecter les formats des champs de notre base de données, dans le cas d'une faute, l'application affiche un message d'erreur.

- La navigation entre les interfaces de notre future application web doit être légère et fluide.
- La sécurité : L'accès aux informations n'est possible qu'après vérification des privilèges et des droits d'accès. Ainsi tout utilisateur passera par une phase d'authentification pour pouvoir consulter les services offerts par l'application : (nom utilisateur + mot de passe numérique) récupérés lors de l'ajout d'un nouveau abonné. Les abonnés et les administrateurs ne peuvent pas accéder qu'aux leurs options relatives.
- L'extensibilité : L'architecture de l'application permettra l'évolution et la maintenance (ajout ou suppression ou mise à jour) au niveau de ses différents modules d'une manière flexible.
- Contraintes humaines : Notre solution doit tenir compte des besoins de chaque acteur participant dans la gestion des changements. Elle doit leur faciliter la tâche et assurer leur activité dans les bonnes conditions. Donc, notre application doit être facile à utiliser en présentant des interfaces ergonomiques.
- Contraintes techniques :
 - Toutes les applications web nécessitent une authentification par certificat qui sera établit en mettant en place le Framework Spring Security. Pour notre cas, on a dû suivre le protocole interne et intégrer ce Framework d'authentification et de contrôle d'accès.
 - Lors d'un virement bancaire le client est invité à rentrer son mot de passe pour vérifier la transaction. Dans le cas d'un virement interbancaire, un code de validation est envoyé à son e-mail.
 - Les mots de passes des abonnés doivent êtres cryptés « BCrypt » au niveau de la base de données afin de garder sécurisé l'accès à l'application.
 - La saisie du mot de passe doit être masquée et seulement possible par une pad numérique pour éviter le vol des mots de passe.

I.6. Conclusion :

Dans ce premier chapitre, nous avons commencé par présenter notre organisme d'accueil BFI. Ensuite, nous avons indique la solution existante et ses déficiences. Enfin, après avoir décrire le contexte du projet et la solution proposée, nous nous dirigerons naturellement vers la planification de l'application.

Chapitre II : Technologies et Frameworks : état de l'art

II.1. Introduction :

Dans ce chapitre, le planisme sera présenté. Il s'agit en fait d'une phase de planification et architecture qui présente l'état de l'art dans cette période conduits à construire une bonne vision du produit: préparer l'environnement de développement et indiquer les outils de conception/développement.

II.2. Technologies et outils de développements :

II.2.1. Outils de conception :

- Enterprise Architect :

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne « Sparx Systems ». Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus. [4]

II.2.2. Langage de modélisation :

UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de représentations graphiques. Il est apparu dans le monde du génie logiciel, dans le cadre de la «conception orientée objet». [5]

II.2.3. Environnement technique du travail :

La table 1 détaille la configuration matérielle des machines de développement et de déploiement utilisées ainsi que les différents logiciels qui y sont installés :

Table 1 : Environnement technique du travail

Machine	Machine1
Système d'exploitation	Windows 8.1 Entreprise (x64)
Logiciels installées	Eclipse - Apache Tomcat Server - Enterprise architect Maven - WampServer – MySQL - JDK

II.2.4. Frameworks de développement :

Tout système d'information nécessite la réalisation de trois groupes de fonctions: le stockage des données, la logique applicative et la présentation (MVC). Ces trois parties sont indépendantes les unes des autres. D'où notre choix de l'architecture 3 tiers pour l'application Web.

En effet, cette architecture est un modèle logique d'architecture applicative qui vise à séparer, très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et à présenter cette application comme un empilement de trois couches, étages, ou niveaux qui sont les suivantes : Couche Présentation, Couche Métier / Business et Couche Accès aux données. La figure 1 schématise l'architecture générale de notre application web. [6]

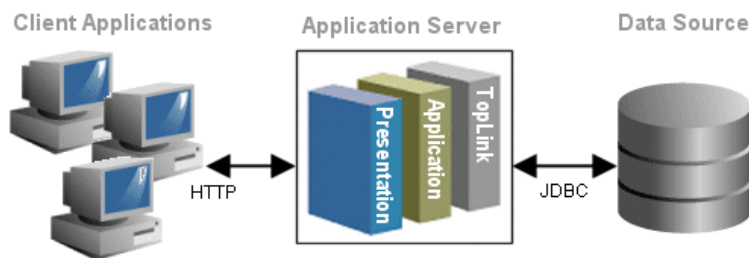


Figure 1 : Architecture générale d'application web

Parlons maintenant de la mise en place de la plate-forme J2EE : Cette opération se constitue de deux étapes fondamentales : la mise en place des couches et celle du serveur web.

- Mise en place des couches

Nous avons choisi de coder l'application en se basant sur les couches Hibernate, Spring et Spring Security afin de faciliter l'effort de l'implémentation et de favoriser l'aspect sécurité.

- La couche Spring

Spring est un Framework libre pour construire et définir l'infrastructure d'une application java, dont il facilite le développement et les tests. Il est effectivement un conteneur dit «léger». Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre eux. [7]

Pour l'authentification, on a dû intégrer le Framework Spring Security. C'est un cadre qui se concentre sur la fourniture de l'authentification et l'autorisation aux applications Java. [8]

- La couche Hibernate

C'est une couche qui sert à la persistance des objets dans les bases de données relationnelles. Elle présente un avantage remarquable contre l'accès classique aux données manifestés par les JDBC, vu la rapidité de traitement via des appels à des méthodes objet de haut niveau.

- Mise en place du serveur Web : Apache Tomcat

Apache Tomcat est un serveur d'application Java permettant d'exécuter des servlets et des pages JSP. Il peut être utilisé ou couplé avec un serveur Web, et porté sur n'importe quel système sur lequel une machine virtuelle Java est installée. [9]

II.2.5. Outils de développements :

- Eclipse Oxygen 4.6 (La nouvelle version de l'IDE)

Eclipse est un environnement de développement open source, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible et universel. [10]

- MySQL

MySQL est 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, autant par le grand public que par des professionnels. [11]

- Maven

Apache Maven est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier, géré par Apache Software Foundation. [12]

II.2.6. Langage de programmation :

- Java

Java est un langage de programmation orienté objet, développé par Sun Microsystems permettant de créer des logiciels compatibles avec des nombreux systèmes d'exploitation et donne aussi la possibilité de développer des programmes pour téléphones portables. [13]

- AngularJS 4

AngularJS est une Framework qui facilite la création d'applications sur le Web. AngularJS combine des modèles déclaratifs, des injections de dépendances, des outils de bout en bout et des meilleures pratiques intégrées pour résoudre les problèmes de développement. [14]

II.3. Conclusion :

Au long de ce chapitre, nous avons présenté l'état de l'art, c'est en fait une phase de planification pendant laquelle nous avons défini les technologies et les environnements de développement ainsi que l'architecture globale de l'application.

Chapitre III : Conception

III.1. Introduction :

Dans ce chapitre, nous présentons la conception de l'application. C'est une étape primaire et essentielle dédiée à la mise au point et à l'écriture d'un projet. Nous allons alors identifier les rôles des utilisateurs, indiquer les cas d'utilisation, déterminer un backlog initial, formuler les classes en précisant l'architecture MVC et préparer la démarche de l'emploi de l'application.

III.2. Capture des besoins:

Nous procéderons par l'identification des acteurs puis la description des cas d'utilisation de la future application.

III.2.1 Identification des acteurs :

Les deux acteurs principaux sont :

- Abonne
- Administrateur

En tenant compte des fonctionnalités offertes en commun pour les deux acteurs, il existe un autre plus général « Utilisateur » avec lequel ils constituent une relation d'héritage comme il est présenté dans la figure 2.

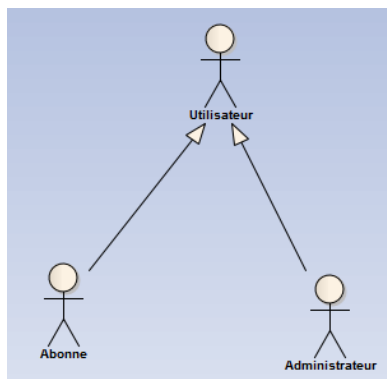


Figure 2 : Héritage entre les acteurs

III.2.2 Description des acteurs :

La table 2 définit les acteurs de l'application ainsi que leurs rôles.

Table 2 : Description détaillée des acteurs

Acteur	Définition	Rôle
Utilisateur	Toute personne utilisant l'application E-Banking.	Il a un accès à un espace public (l'index) ou il pourra accéder aux d'autres fonctionnalités de site sauf celles d'E-Banking. Il peut s'authentifier pour accéder à ces fonctionnalités.
Abonne	Tout client de la banque possédant un login et son mot de passe afin d'utiliser l'application.	Il s'authentifie afin qu'il puisse accéder à son espace client où il bénéficiera d'un ensemble de fonctionnalités (consultation comptes, demande carte ou chéquier, virements ...). Il peut aussi accéder à son profile afin de le mettre à jour.
Administrateur	Il s'agit de la personne surveillant les abonnées et leurs opérations.	Il est responsable de la gestion des abonnés E-Banking, la gestion des bénéficiaires et la mise à jour des demandes de virements (valider ou les invalider).

III.2.3 Diagrammes de cas d'utilisation :

Un cas d'utilisation est la description d'un ensemble de séquences d'actions qu'un système effectue pour produire un résultat observable à un acteur. Un cas d'utilisation représente une exigence fonctionnelle de votre système dans son ensemble. L'accent est mis sur ce qu'un système fait, plutôt que sur la façon dont il le fait. Les diagrammes de cas d'utilisation sont présentés par les figures 3, 4, 5 et 6.

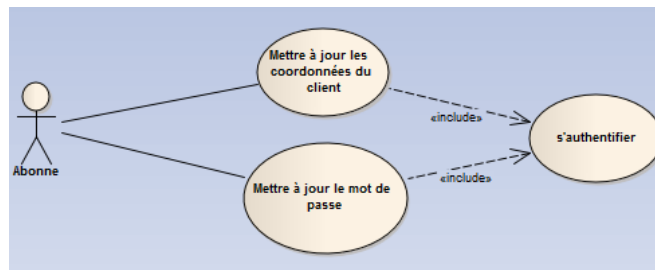


Figure 3 Diagramme de cas d'utilisation de la gestion de données personnelles par l'abonné

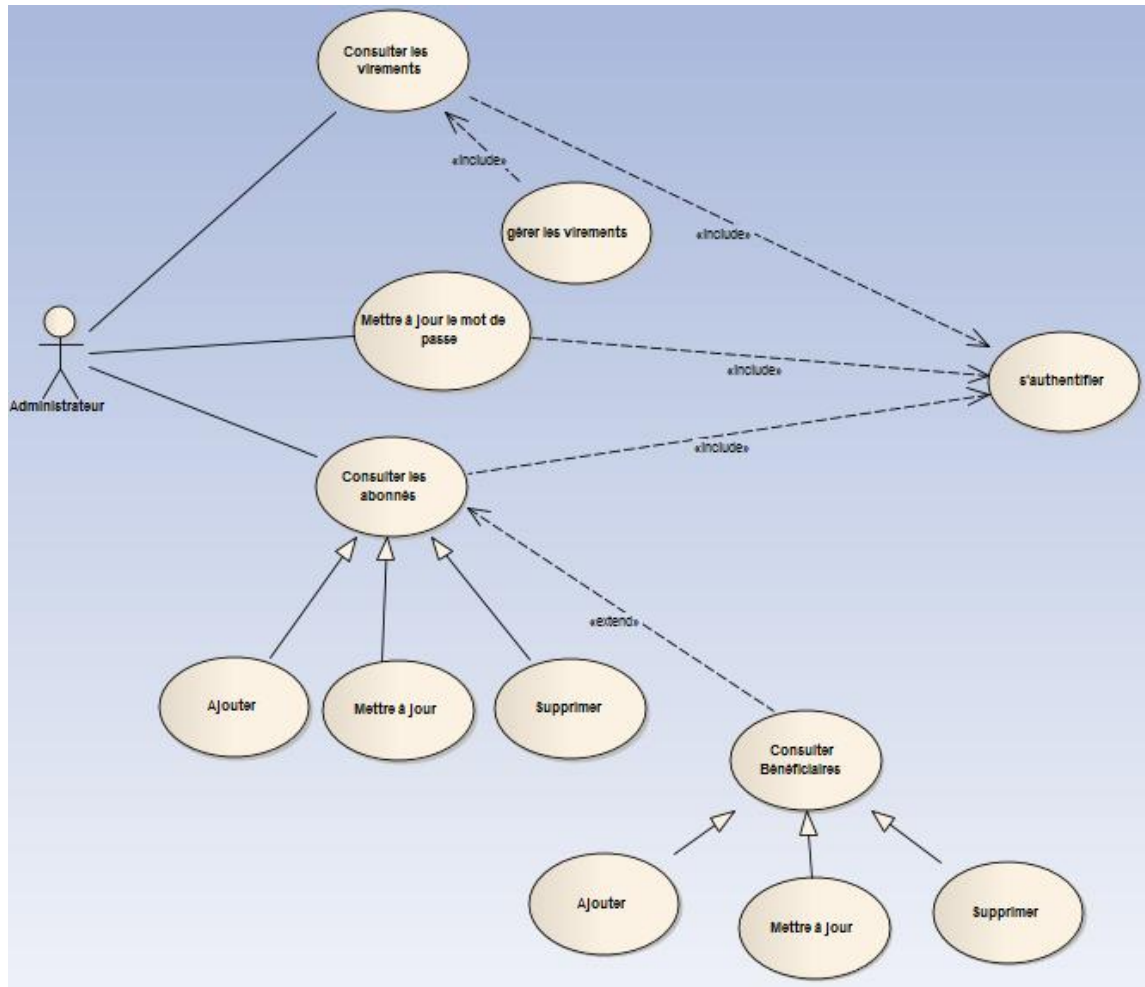


Figure 4 : Diagramme de cas d'utilisation de l'administrateur

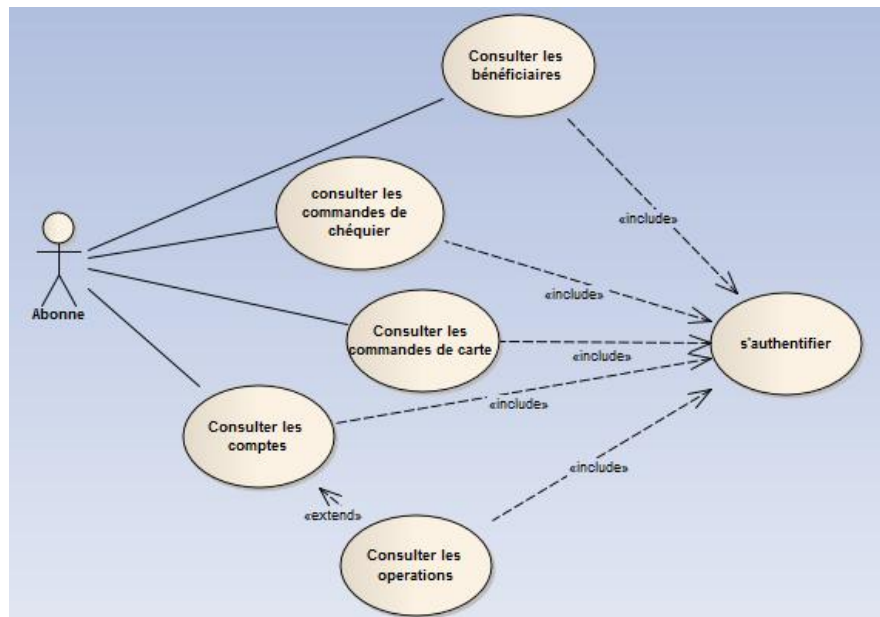


Figure 5 : Diagramme de cas d'utilisation de la consultation de données financières par l'abonné

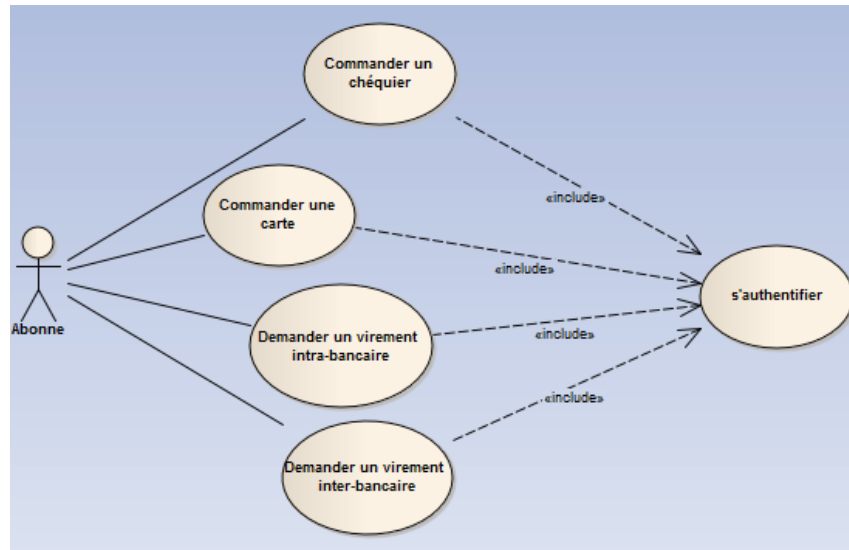


Figure 6 : Diagramme de cas d'utilisation de la gestion des opérations par l'abonné

III.3. Backlog du produit :

Pour chaque cas d'utilisation on identifie la priorité, Le risque et la description présenté dans la table 3.

- Priorité: Elle permet de classer les cas d'utilisation par ordre de priorité dans le bon fonctionnement de l'application.
- Estimation : Sert à estimer l'effort nécessaire pour implémenter une fonctionnalité.
- Description : Elle permet de décrire les cas d'utilisation de sorte à indiquer leurs rôles.

Table 3 : Backlog du produit

ID	Nom	Priorité	Estimation	Description
1	S'authentifier	Haute	Moyenne	En tant qu'un utilisateur de l'application E-Banking, je veux m'authentifier afin de bénéficier des fonctionnalités de l'application.
2	Commander un chéquier	Haute	Moyenne	En tant qu'abonné, je veux demander un nouveau chéquier afin de faciliter mes transactions quotidiennes.
3	Commander une carte	Haute	Moyenne	En tant qu'abonné, je veux demander une nouvelle carte afin de faciliter mes transactions quotidiennes.

4	Demander un virement intra-bancaire	Haute	Haute	En tant qu'abonné, je veux transférer de l'argent entre mes comptes domiciles au sein de la banque.
5	Demander un virement inter-bancaire	Haute	Haute	En tant qu'abonné, je veux transférer de l'argent entre mes comptes et mes bénéficiaires.
6	Consulter les comptes	Haute	Faible	En tant qu'abonné, je veux consulter mes comptes et leurs soldes.
7	Consulter les bénéficiaires	Haute	Faible	En tant qu'abonné, je veux consulter mes bénéficiaires valides.
8	Consulter les opérations	Haute	Faible	En tant qu'abonné, je veux consulter mes opérations et de prendre un extrait en utilisant la date et la pagination.
9	Consulter les commandes de chéquier	Haute	Faible	En tant qu'abonné, je veux consulter l'état des commandes de chéquier.
10	Consulter les commandes de carte	Haute	Faible	En tant qu'abonné, je veux consulter l'état des commandes de carte.
11	Mettre à jour les coordonnées du client	Faible	Moyenne	En tant qu'abonné, je veux voir et/ou mettre à jour mes informations.
12	Mettre à jour le mot de passe de l'abonné	Haute	Moyenne	En tant qu'abonné, je veux changer mon mot de passe.
13	Consulter les abonnés	Haute	Faible	En tant qu'administrateur, je veux consulter la liste des abonnés.
14	Ajouter un abonné	Haute	Moyenne	En tant qu'administrateur, je veux ajouter un abonné.
15	Mettre à jour un abonné	Haute	Moyenne	En tant qu'administrateur, je veux mettre à jour les coordonnées des abonnés afin de leur faciliter l'accès.
16	Supprimer un abonné	Haute	Faible	En tant qu'administrateur, je veux supprimer un abonné.

17	Consulter les bénéficiaires	Haute	Faible	En tant qu'administrateur, je veux consulter la liste des bénéficiaires de chaque abonné.
18	Ajouter un bénéficiaire	Haute	Moyenne	En tant qu'administrateur, je veux ajouter un bénéficiaire à l'abonné.
19	Mettre à jour un bénéficiaire	Haute	Moyenne	En tant qu'administrateur, je veux mettre à jour les coordonnées des bénéficiaires (valides ou non, nom..).
20	Supprimer un bénéficiaire	Haute	Faible	En tant qu'admin, je veux supprimer un bénéficiaire d'un abonné.
21	Consulter les virements	Moyenne	Faible	En tant qu'administrateur, je veux consulter la liste des virements.
22	Gérer les virements	Moyenne	Faible	En tant qu'administrateur, je veux gérer les virements en les validant/invalidant.
23	Mettre à jour le mot de passe de l'administrateur	Haute	Moyenne	En tant qu'administrateur, je veux changer mon mot de passe.

III.4. Phase d'analyse (diagramme de classes d'analyse):

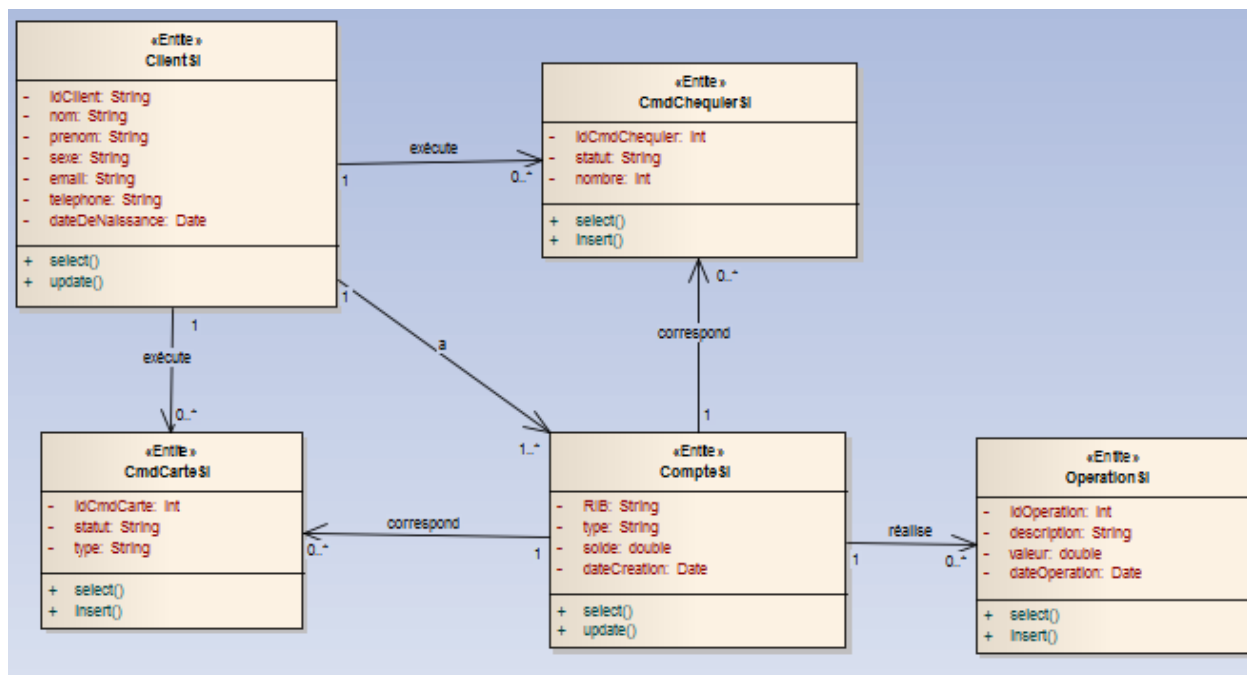


Figure 7 : Diagramme de classes de domaine SI

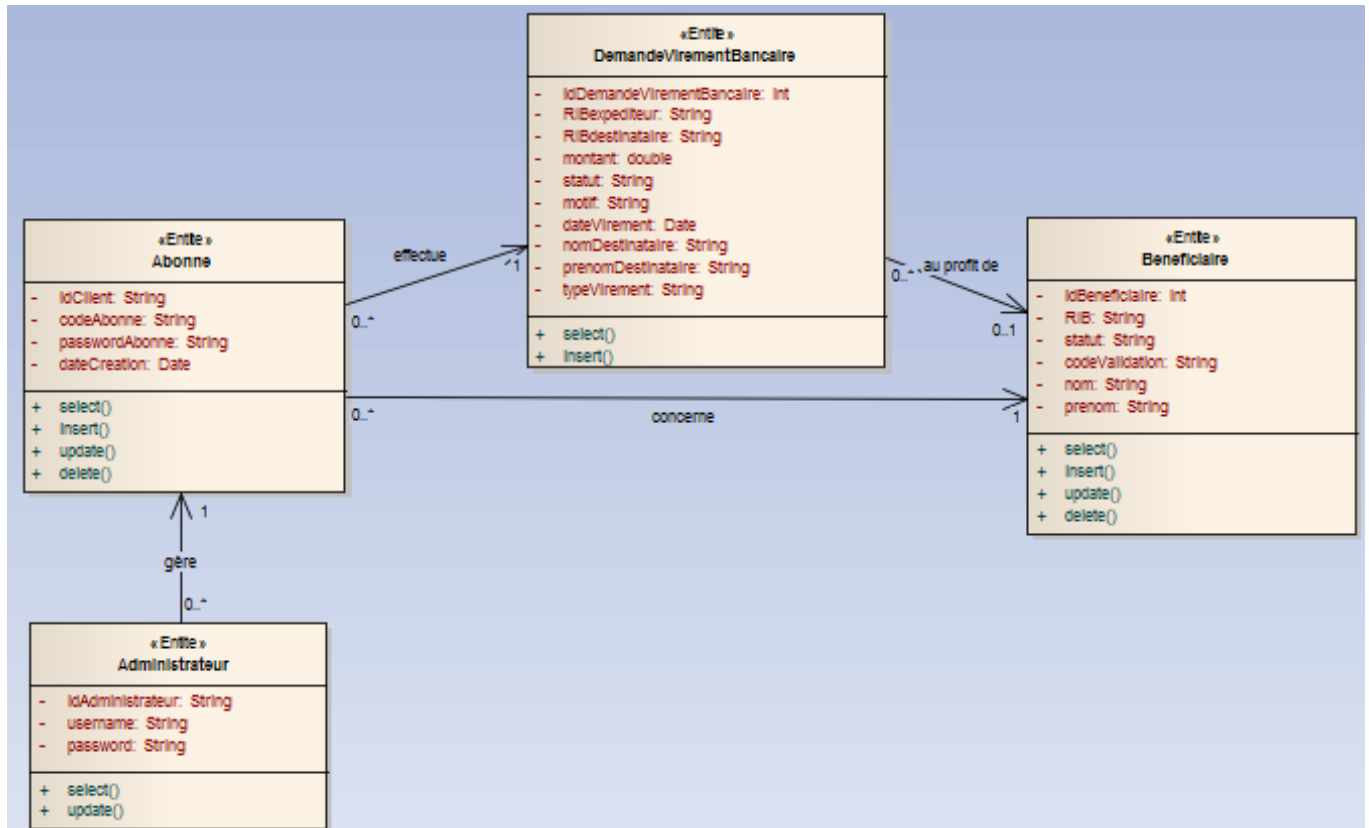


Figure 8 : Diagramme de classes de domaine Persistantes

La phase d'analyse a comme but l'élaboration du diagramme de classe de domaine. Le diagramme de domaine est la modélisation des entités et concepts présentes dans le domaine de l'application. Il s'agit dans ce diagramme de classes métiers qui ne contiennent que des données (figures 7 et 8).

III.5. Phase de conception :

III.5.1. Cas d'utilisation « S'authentifier »:

i. Description textuelle :

La table 4 décrit textuellement le cas d'utilisation « s'authentifier ».

Table 4 : Description textuelle du cas d'utilisation « s'authentifier »

Cas d'utilisation	S'authentifier
Acteur	Utilisateur
Pré condition	L'utilisateur entre le site de l'application.
Post condition	L'utilisateur E-Banking identifié comme étant un abonné ou un administrateur.
Description du scénario principal	- L'utilisateur remplit les champs d'authentification et clique sur le bouton «Login».

	- Le système vérifie les informations saisies par l'utilisateur et affiche l'interface suivante.
Exception	Si un des champs est invalide, le système affiche un message d'erreur et réinitialise les champs.

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « s'authentifier » (figure 9).

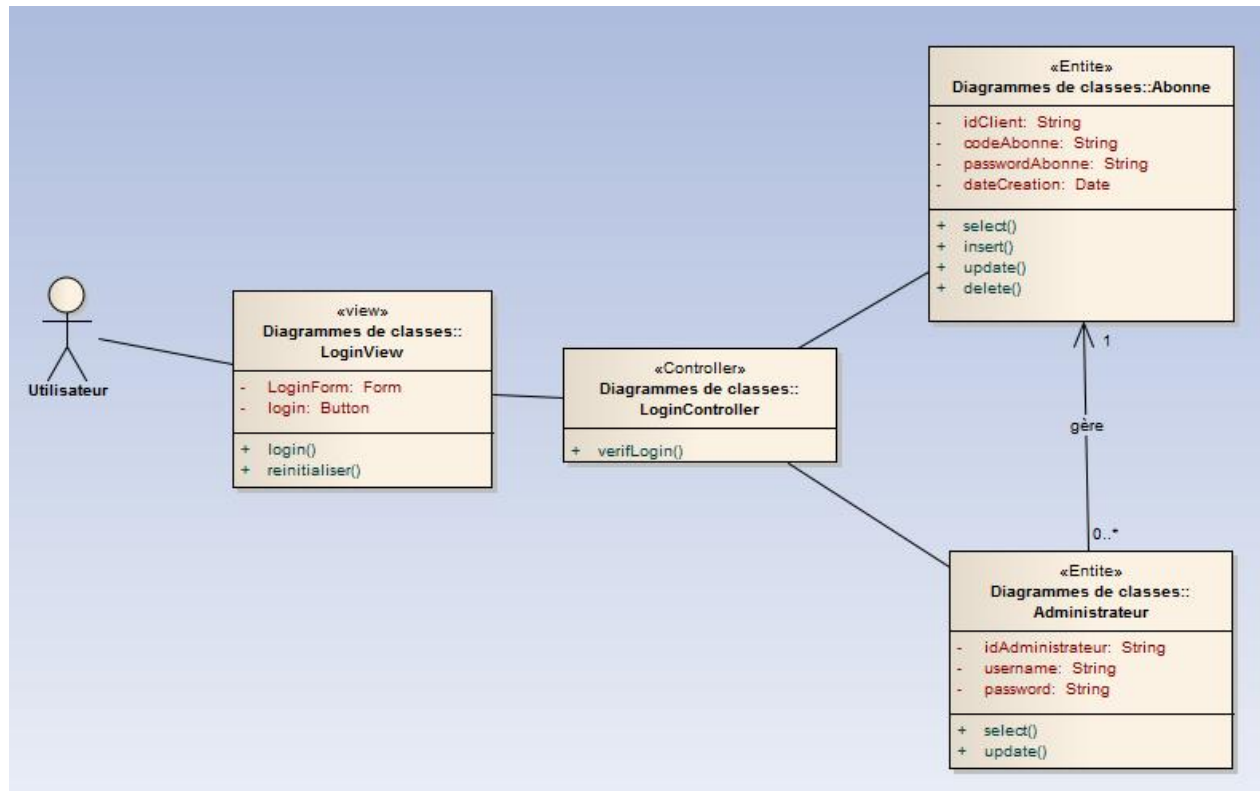


Figure 9 : Diagramme de classes de cas d'utilisation « s'authentifier »

iii. Diagramme de séquences :

A travers ce diagramme, nous décrivons le scénario du même cas d'utilisation. Dans un premier lieu, l'utilisateur E-Banking remplit le formulaire d'authentification et clique sur le bouton «Login». Ensuite le système vérifie s'il s'agit bien d'un abonné ou non puis s'il s'agit d'un administrateur ou non. Dans le cas du succès, l'utilisateur va être connecté et réorienter vers la page destinataire. Dans le cas d'échec, il doit remplir à nouveau le formulaire.

La figure 10 illustre le diagramme de séquences du cas d'utilisation «S'authentifier».

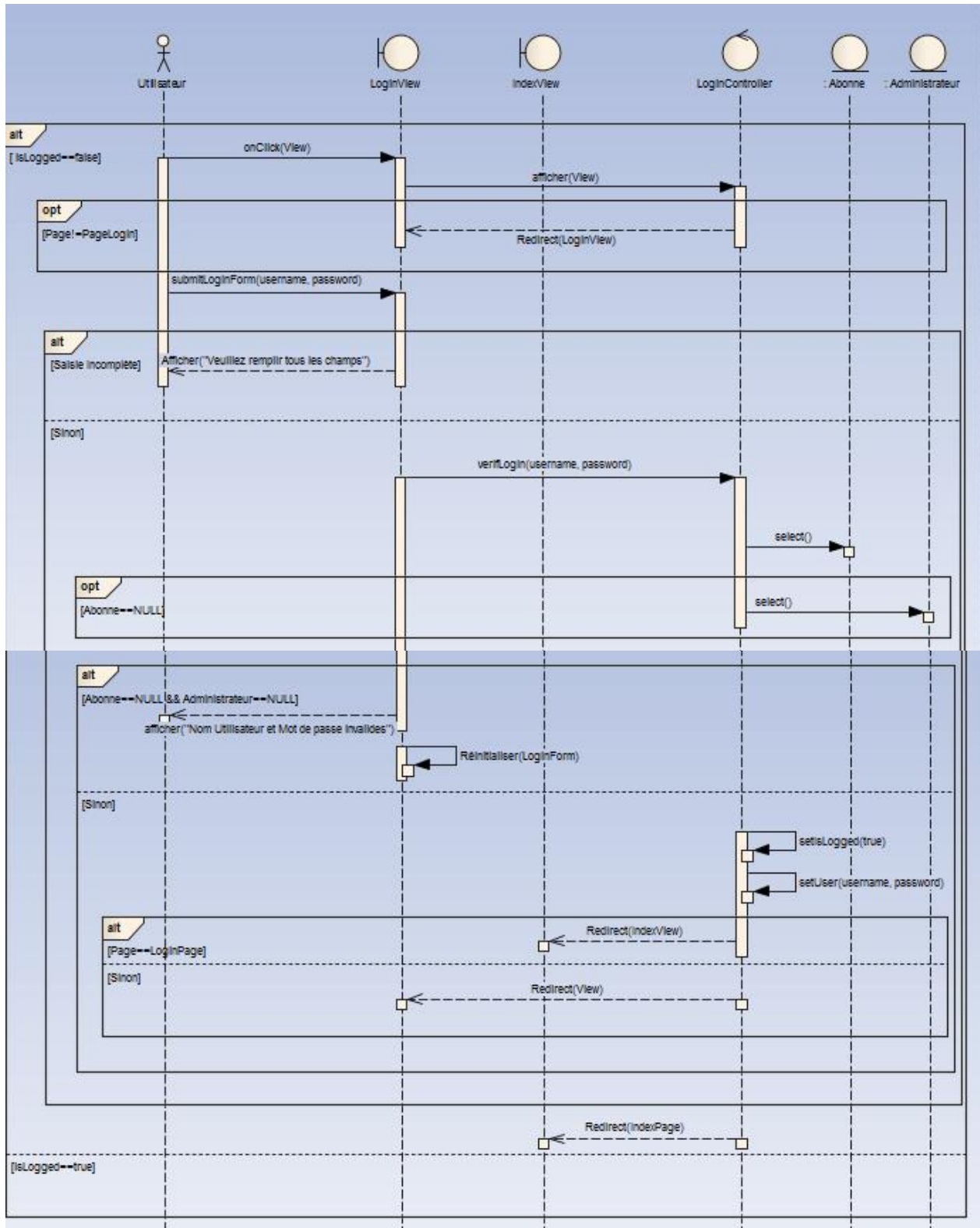


Figure 10 : Diagramme de séquences de cas d'utilisation « s'authentifier »

III.5.2. Cas d'utilisation « Commander un chéquier »:

Ce cas d'utilisation est identique au cas d'utilisation « Commander une carte ».

i. Description textuelle :

La table 5 décrit textuellement le cas d'utilisation « Commander un chéquier »

Table 5 : Description textuelle du cas d'utilisation « Commander un chéquier »

Cas d'utilisation		S'authentifier
Acteur	Abonné	
Pré condition	L'utilisateur E-Banking est identifié comme étant un abonné.	
Post condition	La demande du chéquier est enregistrée.	
Description du scénario principal	<ul style="list-style-type: none"> - A travers un menu, l'abonné choisit « Commander un chéquier ». - Le système affiche l'interface appropriée. - L'abonné remplit les champs nécessaires et clique sur le bouton « Submit ». - Le système vérifie les informations saisies par l'abonné, enregistre la demande et affiche un message de succès. 	
Exception	Si un des champs est vide, le système affiche un message d'erreur.	

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « Commander un chéquier » se traduit par les différentes classes participantes à la réalisation de ce dernier (figure 11).

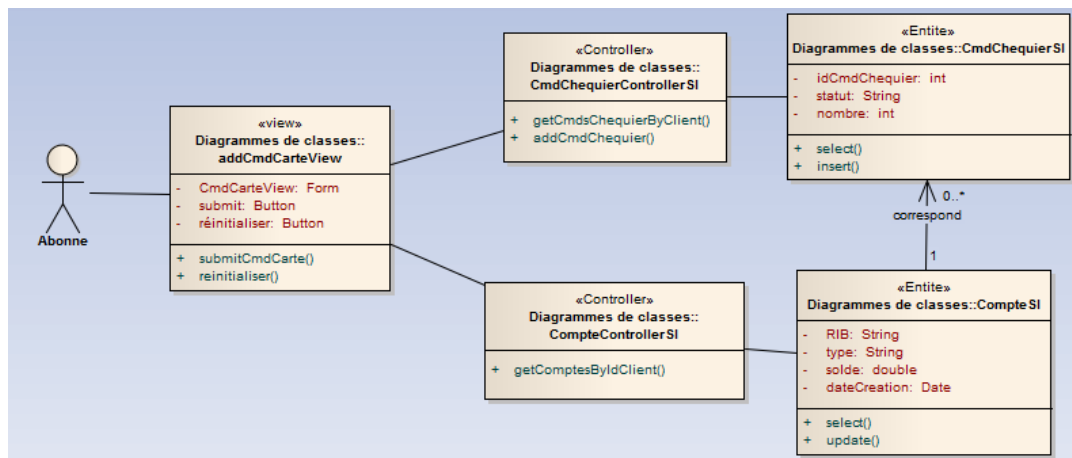


Figure 11 : Diagramme de classes de cas d'utilisation « Commander un chéquier »

iii. Diagramme de séquences :

A travers ce diagramme, nous allons décrire le cas d'utilisation « Commander un chéquier ». L'abonné remplit le formulaire. Ensuite le système vérifie la saisie et enregistre la demande.

La figure 12 illustre le diagramme de séquences du cas d'utilisation «Commander un chéquier»

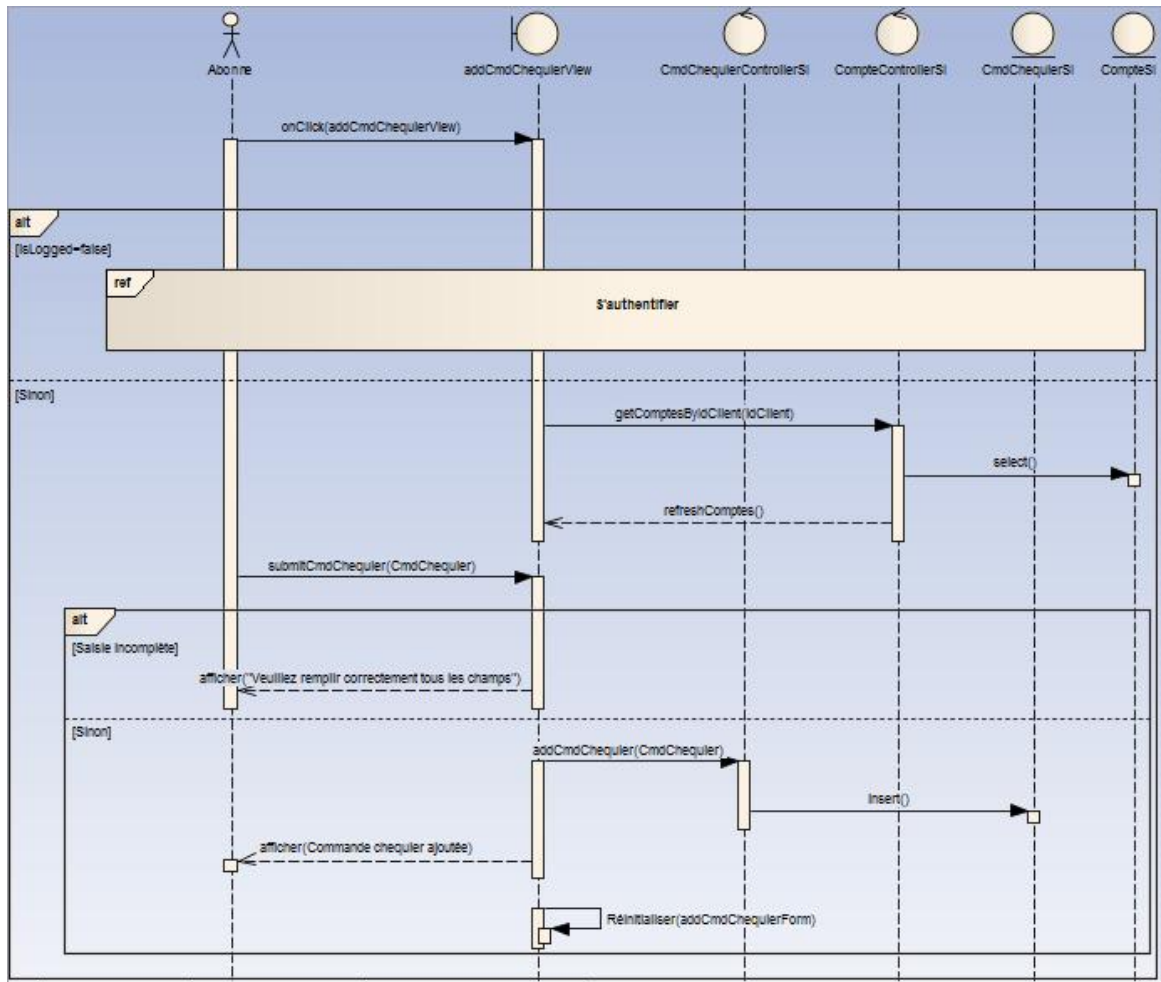


Figure 12 : Diagramme de séquences de cas d'utilisation « Commander un chéquier »

III.5.3. Cas d'utilisation « Demander un virement intrabancaire »:

i. Description textuelle :

La table 6 décrit textuellement le cas d'utilisation « Demander un virement intrabancaire »

Table 6 : Description textuelle du cas d'utilisation « Demander un virement intrabancaire »

Cas d'utilisation	S'authentifier
Acteur	Abonné
Pré condition	L'utilisateur E-Banking est identifié comme étant un abonné.
Post condition	La demande du virement intrabancaire est enregistrée.
Description du scénario principal	- A travers menu, l'abonné choisit «Demander un virement intrabancaire». - Le système affiche l'interface appropriée.

	<ul style="list-style-type: none"> - L'abonné remplit les champs nécessaires en choisissant les 2 comptes du transfert et clique sur le bouton « Submit ». - L'abonné entre son mot de passe. - Le système vérifie les informations saisies par l'abonné, enregistre la demande et affiche un message de succès. - Les soldes des comptes utilisés sont altérés.
Exception	<ul style="list-style-type: none"> - Si un des champs est vide, l'application affiche un message d'erreur. - Si le mot de passe est incorrect, l'application annule le transfert.

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « Demander un virement intrabancaire » se traduit par les différentes classes participantes à la réalisation de ce dernier (figure 13).

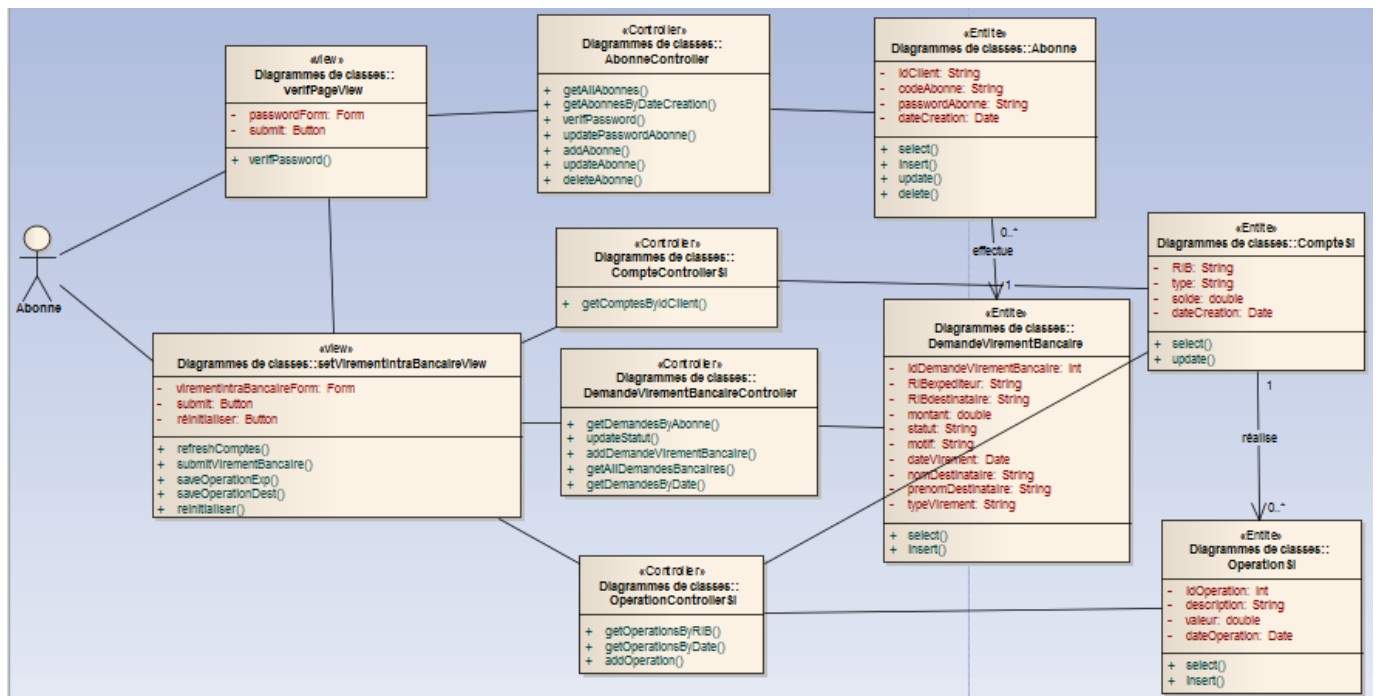


Figure 13 : Diagramme de classes de cas d'utilisation « Demander un virement intrabancaire »

iii. Diagramme de séquences :

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Demander un virement intrabancaire ». D'abord, l'abonné remplit le formulaire en indiquant le compte expéditeur et le compte destinataire. Ensuite le système demande le mot de passe, vérifie la saisie et enregistre la demande en mettant à jour les soldes.

La figure 14 illustre le diagramme de séquences du cas d'utilisation « Demander un virement intrabancaire »

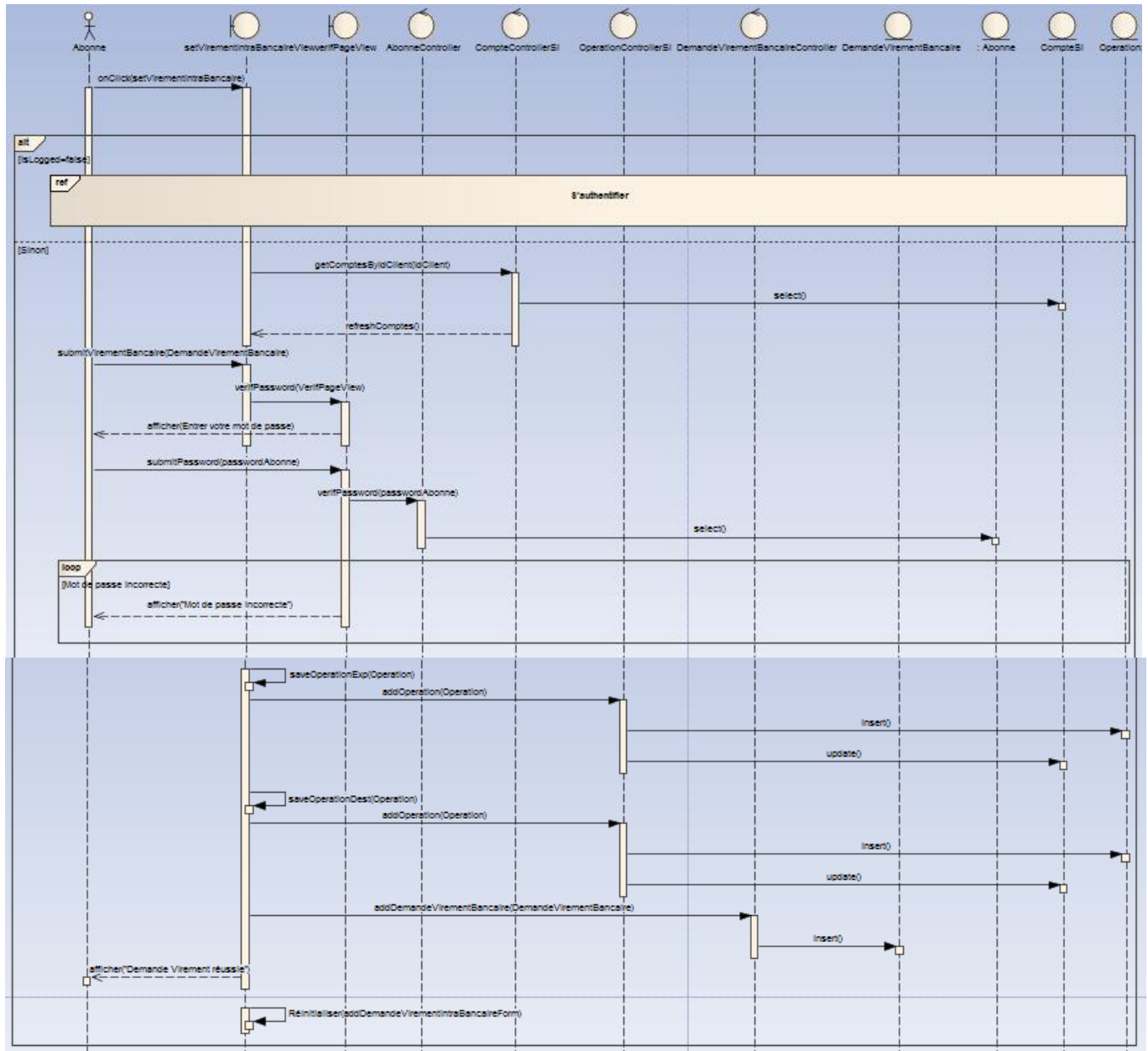


Figure 14 : Diagramme de séquences de cas d'utilisation « Demander un virement intrabancaire »

III.5.4. Cas d'utilisation « Demander un virement interbancaire »:

i. Description textuelle :

La table 7 décrit textuellement le cas d'utilisation « Demander un virement interbancaire »

Table 7 : Description textuelle du cas d'utilisation « Demander un virement interbancaire »

Cas d'utilisation	S'authentifier
-------------------	----------------

Acteur	Abonné
Pré condition	L'utilisateur E-Banking est identifié comme étant un abonné.
Post condition	La demande du virement interbancaire est enregistrée.
Description du scénario principal	<ul style="list-style-type: none"> - A travers menu, l'abonné choisit «Demander un virement interbancaire». - Le système affiche l'interface appropriée. - L'abonné remplit les champs nécessaires en choisissant le compte et le bénéficiaire et clique sur le bouton « Submit ». - L'abonné entre son mot de passe et un code de validation (envoyé par mail). - Le système vérifie les informations saisies par l'abonné, enregistre la demande et affiche un message de succès. - Les soldes des comptes utilisés sont altérés.
Exception	<ul style="list-style-type: none"> - Si un des champs est vide, l'application affiche un message d'erreur. - Si le mot de passe est incorrect, l'application annule le transfert. - Si le code de validation est incorrect, l'application annule le transfert.

ii. Diagramme de classes de cas d'utilisation :

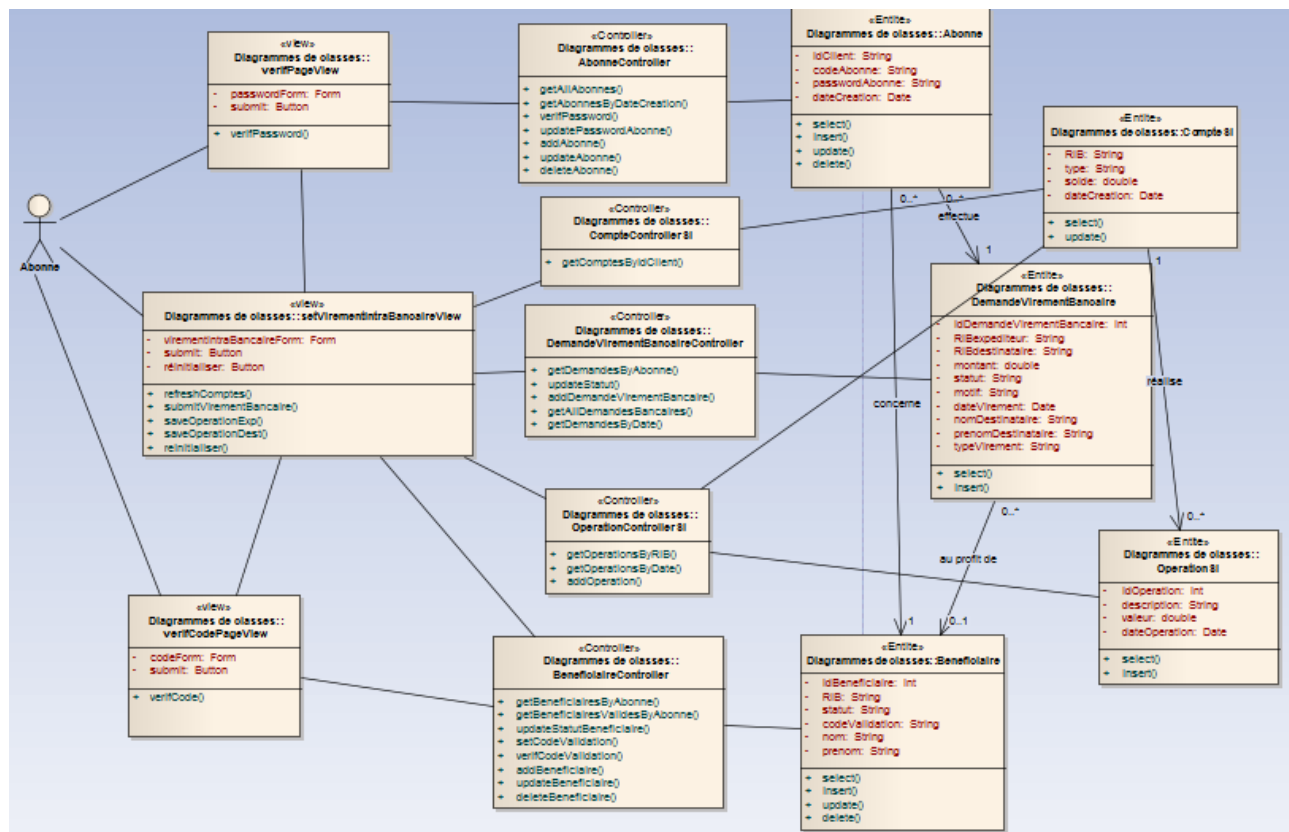


Figure 15 : Diagramme de classes de cas d'utilisation « Demander un virement interbancaire »

Le diagramme de classes du cas d'utilisation « Demander un virement interbancaire » se traduit par les différentes classes participantes à la réalisation de ce dernier (figure 15).

iii. Diagramme de séquences :

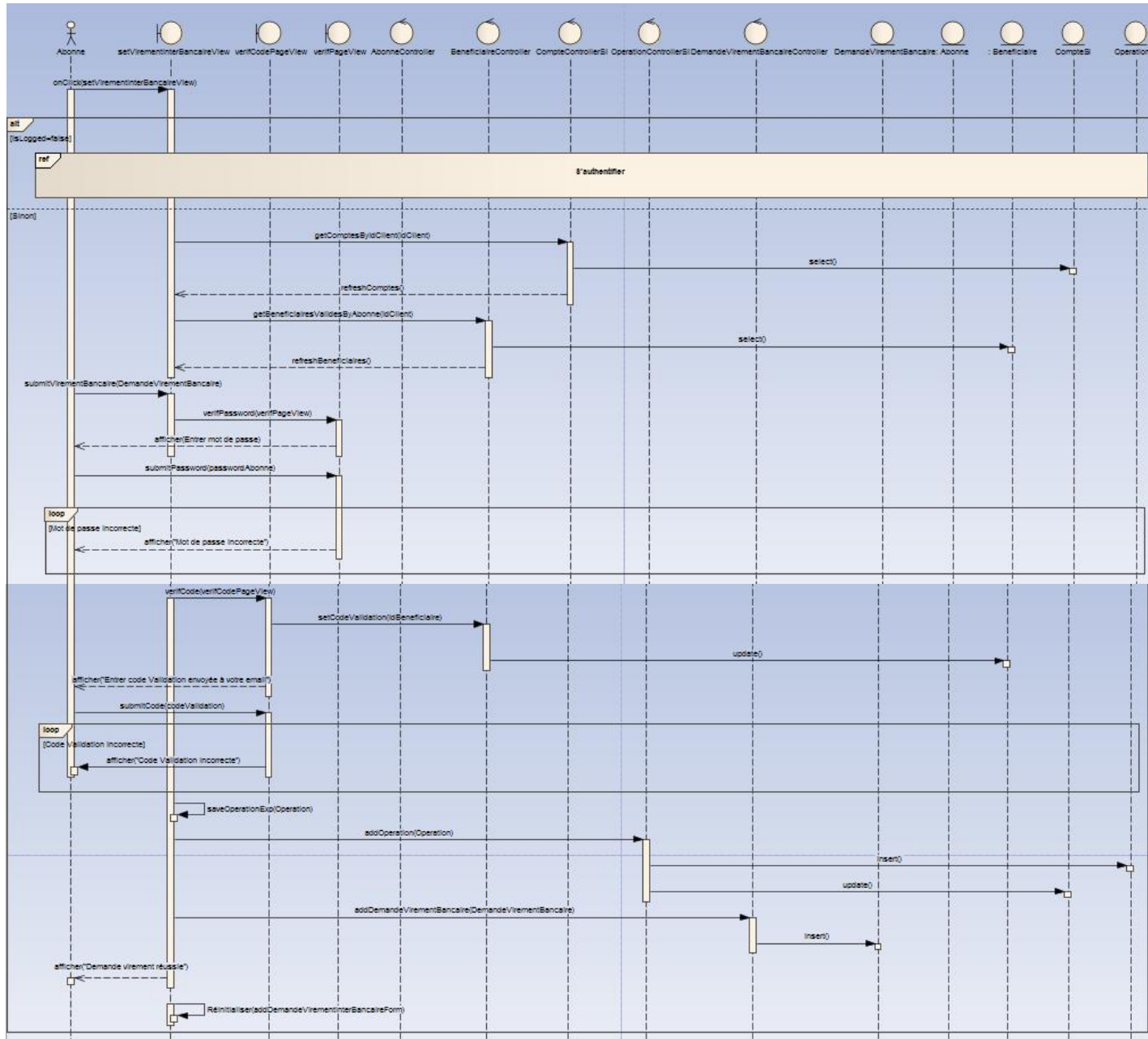


Figure 16 : Diagramme de séquences de cas d'utilisation « Demander un virement interbancaire »

A travers ce diagramme, nous allons décrire le cas d'utilisation « Demander un virement interbancaire ». D'abord, l'abonné remplit le formulaire en indiquant le compte expéditeur et le bénéficiaire destinataire. Ensuite le système demande le mot de passe et la code de validation

envoyé par email, vérifie la saisie et enregistre la demande en mettant à jour les soldes. La figure 16 illustre le diagramme de séquences du cas d'utilisation «Demander un virement interbancaire»

III.5.5. Cas d'utilisation « Consulter les comptes »:

i. Description textuelle :

La table 8 décrit textuellement le cas d'utilisation « Consulter les comptes »

Table 8 : Description textuelle du cas d'utilisation « Consulter les comptes »

Cas d'utilisation		S'authentifier
Acteur	Abonné	
Pré condition	L'utilisateur E-Banking est identifié comme étant un abonné.	
Post condition	L'abonné consulte ses comptes avec succès.	
Description du scénario principal	<ul style="list-style-type: none"> - A travers un menu, l'abonné choisit « Consulter les comptes ». - Le système affiche l'interface appropriée. - L'application affiche les comptes et ses attributs. 	
Point d'extension	<ul style="list-style-type: none"> - L'abonné clique sur le RIB du compte pour afficher ses opérations. 	
Consulter les opérations	<ul style="list-style-type: none"> - Le système affiche l'interface appropriée. - L'application affiche les opérations et ses attributs. - L'abonné peut rechercher les opérations par date. 	

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « Consulter les comptes » et son extension « Consulter les opérations » (figure 17).

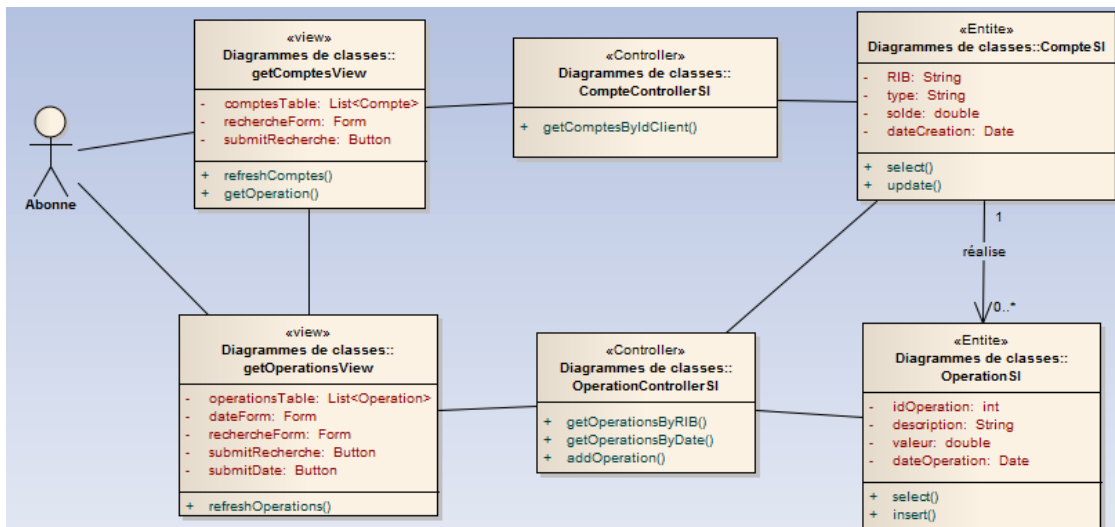


Figure 17 : Diagramme de classes de cas d'utilisation « Consulter les clients »

iii. Diagramme de séquences :

D'abord, l'abonné consulte la liste des comptes puis il peut choisir de consulter les opérations de l'un de ses comptes. La figure 18 illustre le diagramme de séquences du cas d'utilisation « Consulter les comptes » et la figure 19 illustre « Consulter les opérations ».

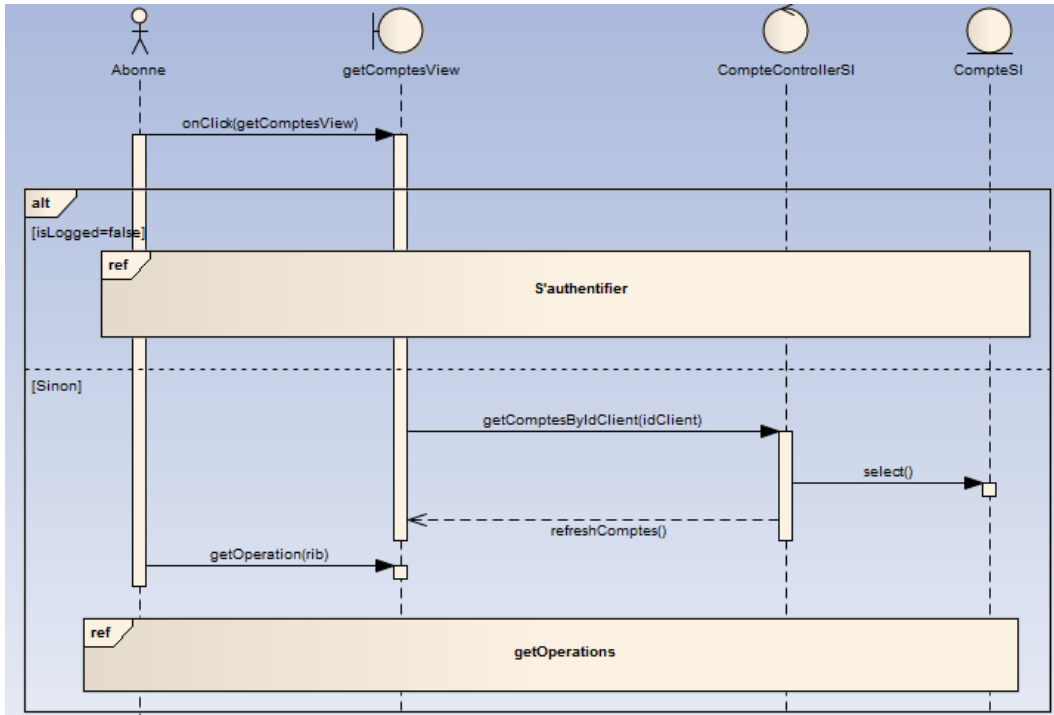


Figure 18 : Diagramme de séquences de cas d'utilisation « Consulter les comptes »

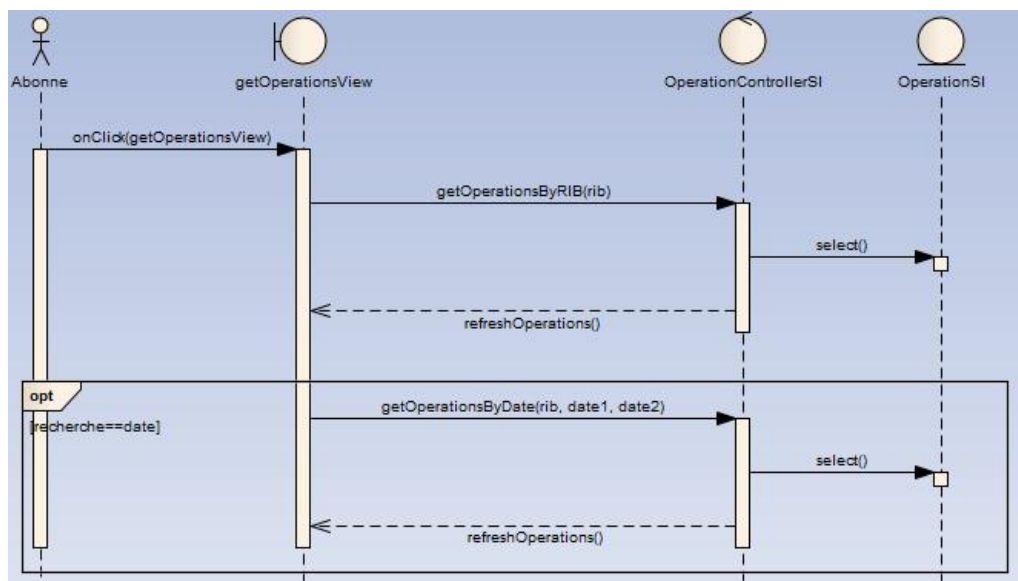


Figure 19 : Diagramme de séquences de cas d'utilisation « Consulter les opérations »

III.5.6. Cas d'utilisation « Consulter les commandes de chéquier »:

Ce cas d'utilisation est identique au cas d'utilisation « Consulter les commandes de carte ».

i. Description textuelle :

La table 9 décrit textuellement le cas d'utilisation « Consulter les commandes de chéquier »

Table 9 : Description textuelle du cas d'utilisation « Consulter les commandes de chéquier »

Cas d'utilisation	S'authentifier
Acteur	Abonné
Pré condition	L'utilisateur E-Banking est identifié comme étant un abonné.
Post condition	L'abonné consulte ses commandes de chéquier avec succès.
Description du scénario principal	<ul style="list-style-type: none"> - A travers un menu, l'abonné choisit « Consulter les commandes de chéquier ». - Le système affiche l'interface appropriée. - L'application affiche les commandes de chéquier.

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « Consulter les commandes de chéquier » se traduit par les différentes classes participantes à la réalisation de ce dernier (figure 20).

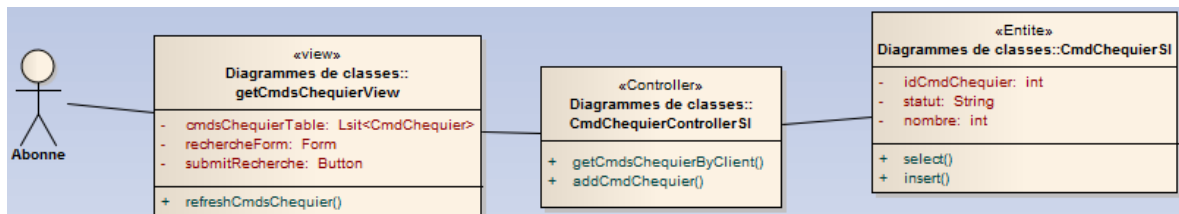


Figure 20 : Diagramme de classes de cas d'utilisation « Consulter les commandes de chéquier »

iii. Diagramme de séquences :

L'abonné consulte la liste des commandes de chéquier (La figure 21).

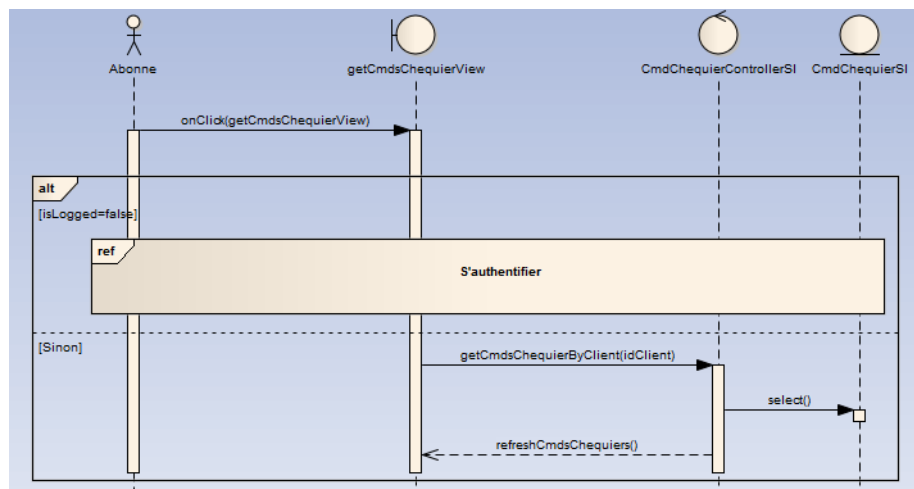


Figure 21 : Diagramme de séquences de cas d'utilisation « Consulter les commandes de chéquier »

III.5.7. Cas d'utilisation « Mettre à jour les coordonnées du client »:

i. Description textuelle :

La table 10 décrit textuellement le cas d'utilisation « Mettre à jour les coordonnées du client »

Table 10 : Description textuelle du cas d'utilisation « Mettre à jour les coordonnées du client »

Cas d'utilisation	S'authentifier
Acteur	Abonné
Pré condition	L'utilisateur E-Banking est identifié comme étant un abonné.
Post condition	L'abonné met à jour ses coordonnées.
Description du scénario principal	<ul style="list-style-type: none"> - A travers un menu, l'abonné choisit « Mettre à jour les coordonnées ». - Le système affiche l'interface appropriée. - L'application affiche les coordonnées actuelles sous forme d'un formulaire. - L'utilisateur modifie les coordonnées. - Le système vérifie les informations saisies par l'abonné, enregistre les coordonnées et affiche un message de succès.
Exception	- Si un des champs est vide, l'application affiche un message d'erreur.

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « Mettre à jour les coordonnées du client » se traduit par les différentes classes participantes à la réalisation de ce dernier (figure 22).

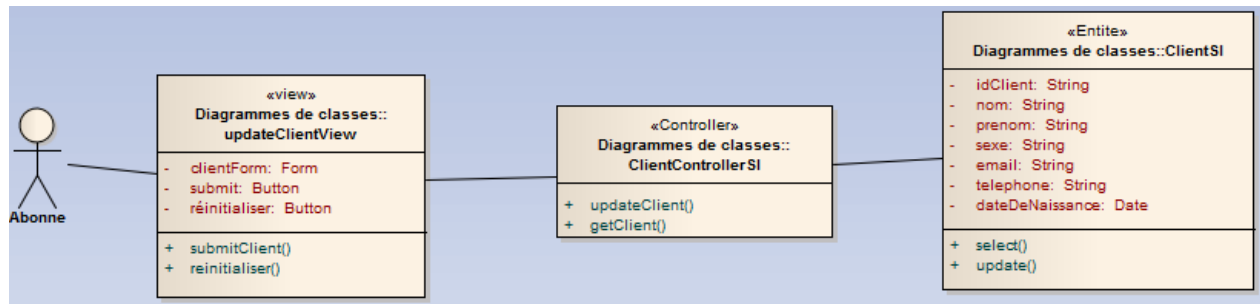


Figure 22 : Diagramme de classes de cas d'utilisation « Mettre à jour les coordonnées du client »

iii. Diagramme de séquences :

L'abonné consulte un formulaire contenant ses coordonnées personnelles ou il peut les changer. Puis, l'application vérifie les changements et les enregistre.

La figure 23 illustre le diagramme de séquences du cas d'utilisation «Mettre à jour les coordonnées du client ».

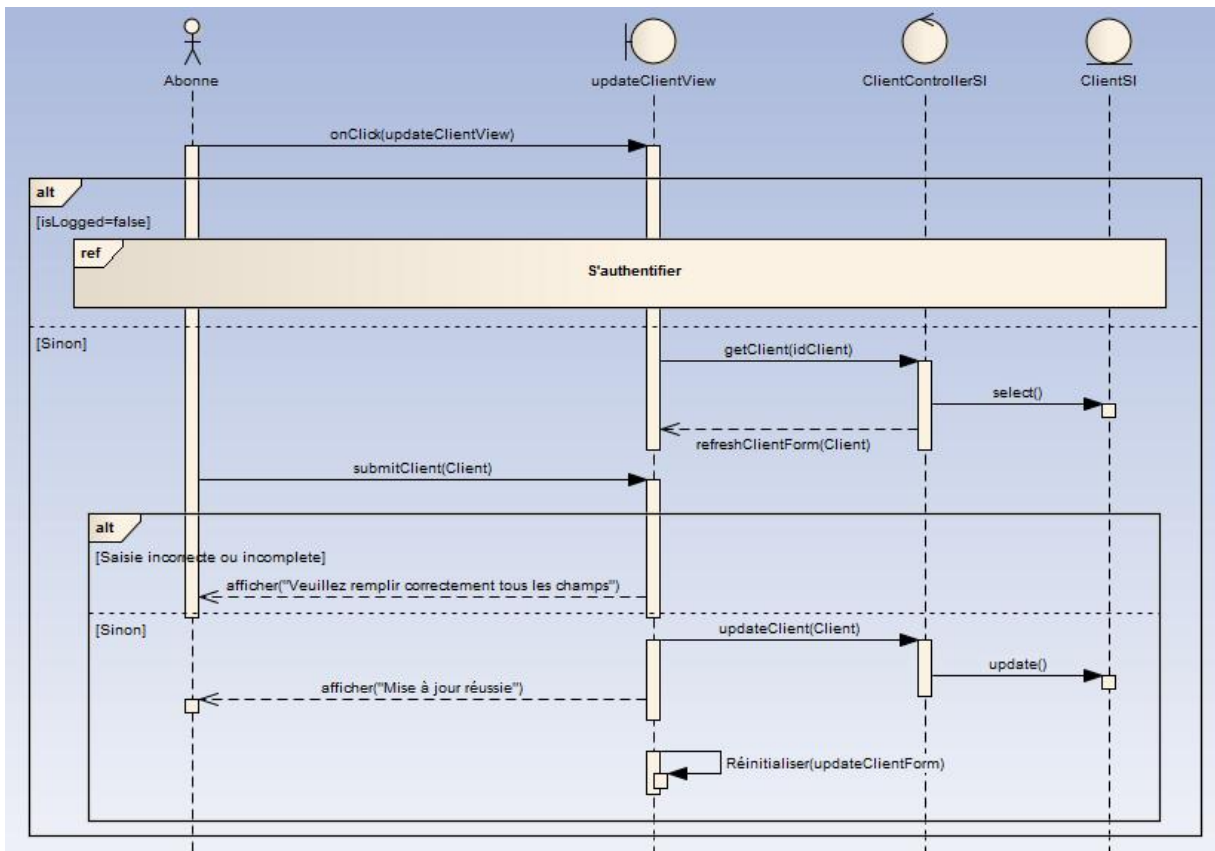


Figure 23 : Diagramme de séquences de cas d'utilisation « Mettre à jour les coordonnées du client »

III.5.8. Cas d'utilisation « Mettre à jour le mot de passe »:

i. Description textuelle :

La table 11 décrit textuellement le cas d'utilisation « Mettre à jour le mot de passe »

Table 11 : Description textuelle du cas d'utilisation « Mettre à jour le mot de passe »

Cas d'utilisation	S'authentifier
Acteur	Utilisateur
Pré condition	L'utilisateur E-Banking est identifié.
Post condition	L'abonné/administrateur met à jour ses coordonnées.
Description du scénario principal	<ul style="list-style-type: none"> - A travers un menu, l'utilisateur choisit « Mettre à jour le mot de passe ». - Le système affiche l'interface appropriée. - L'utilisateur remplit les champs nécessaires et clique sur le bouton « Submit ». - Le système vérifie les informations saisies par l'utilisateur, enregistre la demande et affiche un message de succès.
Exception	- Si un des champs est vide, l'application affiche un message d'erreur.

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « Mettre à jour le mot de passe de l'abonné » se traduit par les différentes classes participantes à la réalisation de ce dernier (figure 24). Il est important de clarifier que le diagramme est identique dans le cas d'utilisation « Mettre à jour le mot de passe de l'administrateur ».

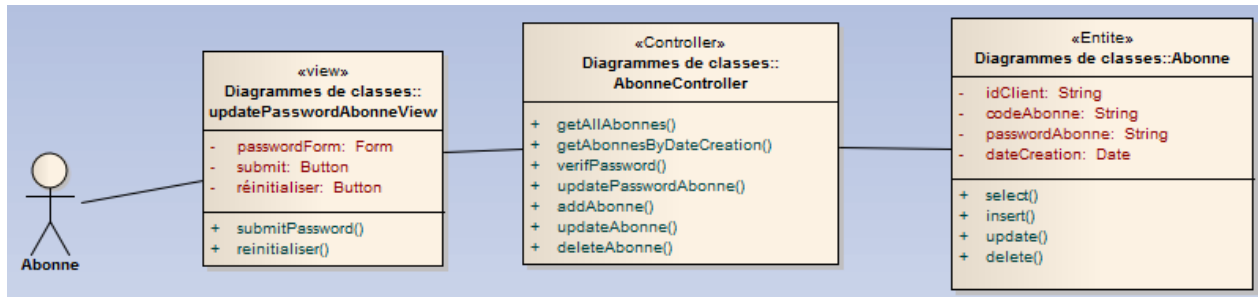


Figure 24 : Diagramme de classes de cas d'utilisation « Mettre à jour le mot de passe de l'abonné »

iii. Diagramme de séquences :

L'abonné remplit le formulaire par le mot de passe ancien et le nouveau mot de passe deux fois pour vérifier la requête. Dans le cas du succès, l'application change le mot de passe. Dans le cas d'échec, le système réinitialise le formulaire et l'utilisateur doit le remplir à nouveau.

La figure 25 illustre le diagramme de séquences du cas d'utilisation «Mettre à jour le mot de passe de l'abonné ».

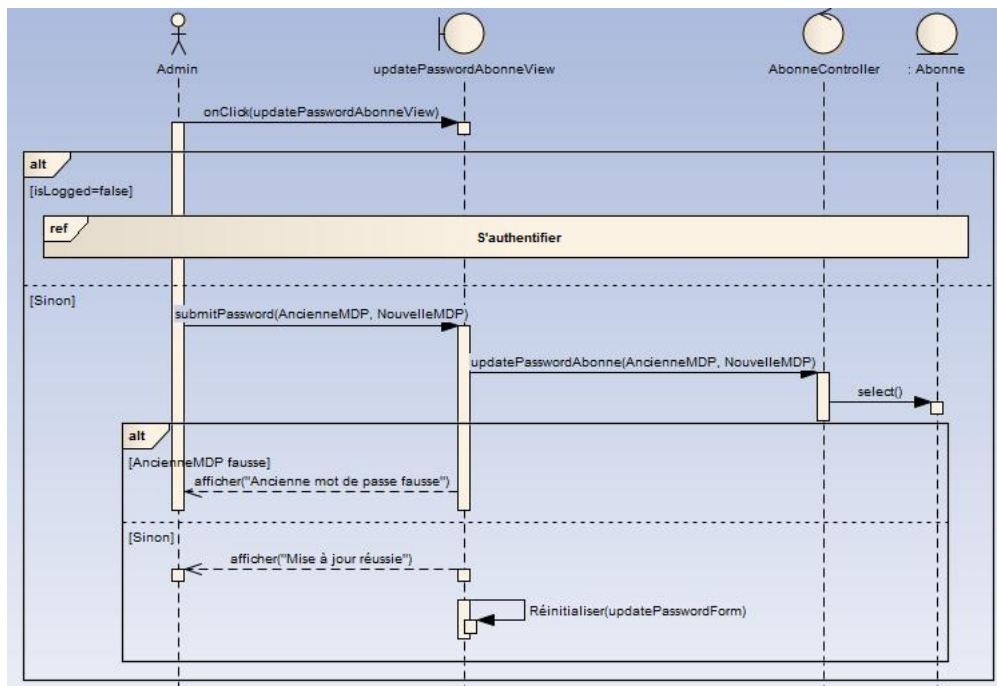


Figure 25 : Diagramme de séquences de cas d'utilisation « Mettre à jour le mot de passe de l'abonné »

III.5.9. Cas d'utilisation « Consulter les abonnés »:

i. Description textuelle :

La table 12 décrit textuellement le cas d'utilisation « Consulter les abonnés »

Table 12 : Description textuelle du cas d'utilisation « Consulter les abonnés »

Cas d'utilisation	S'authentifier
Acteur	Administrateur
Pré condition	L'utilisateur E-Banking est identifié comme étant un administrateur.
Post condition	L'administrateur consulte les abonnés avec succès.
Description du scénario principal	<ul style="list-style-type: none"> - A travers un menu, l'administrateur choisit « Consulter les abonnés ». - L'application affiche les abonnés avec l'option de les modifier, supprimer ou manipuler les bénéficiaires, l'admin peut ajouter d'autres abonnés. - L'admin peut rechercher les abonnés par date de création.
Point d'extension	<ul style="list-style-type: none"> - A travers la table des abonnés, l'administrateur clique sur un bouton pour gérer les bénéficiaires - afficher ses bénéficiaires. - Le système affiche l'interface appropriée. - L'application affiche les bénéficiaires avec l'option de les modifier ou supprimer. L'abonné peut ajouter d'autres bénéficiaires.

ii. Diagramme de classes de cas d'utilisation :

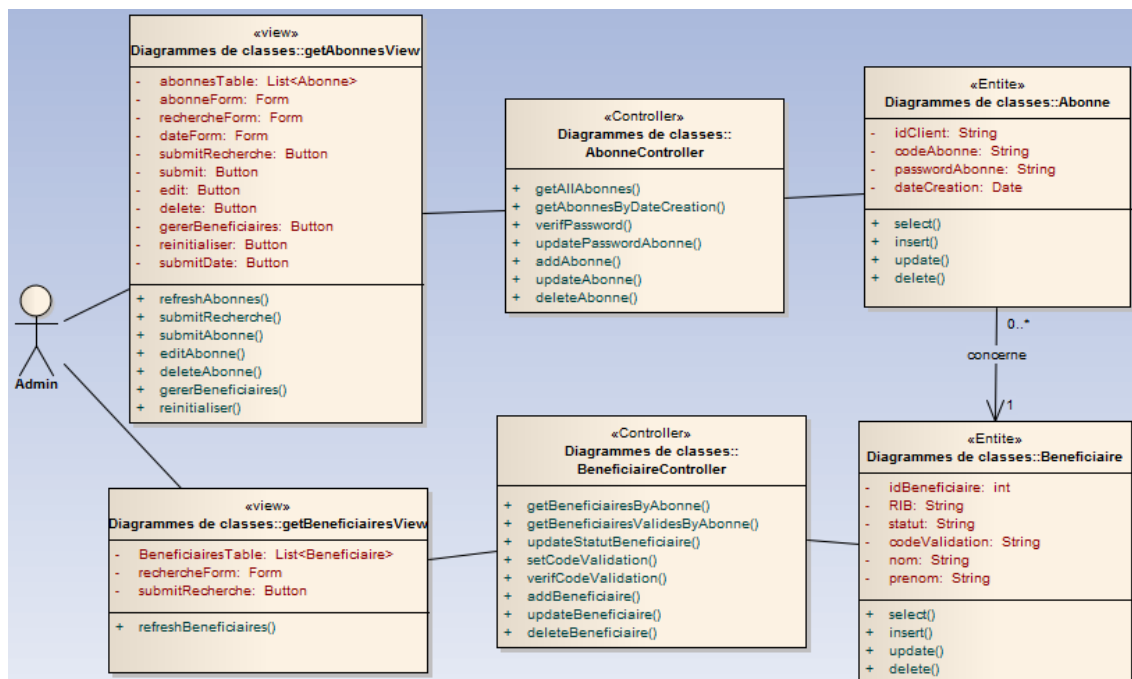


Figure 26 : Diagramme de classes de cas d'utilisation « Consulter les abonnés »

Le diagramme de classes du cas d'utilisation « Consulter les abonnés » et son extension « gérer les bénéficiaires » (figure 26).

iii. Diagramme de séquences :

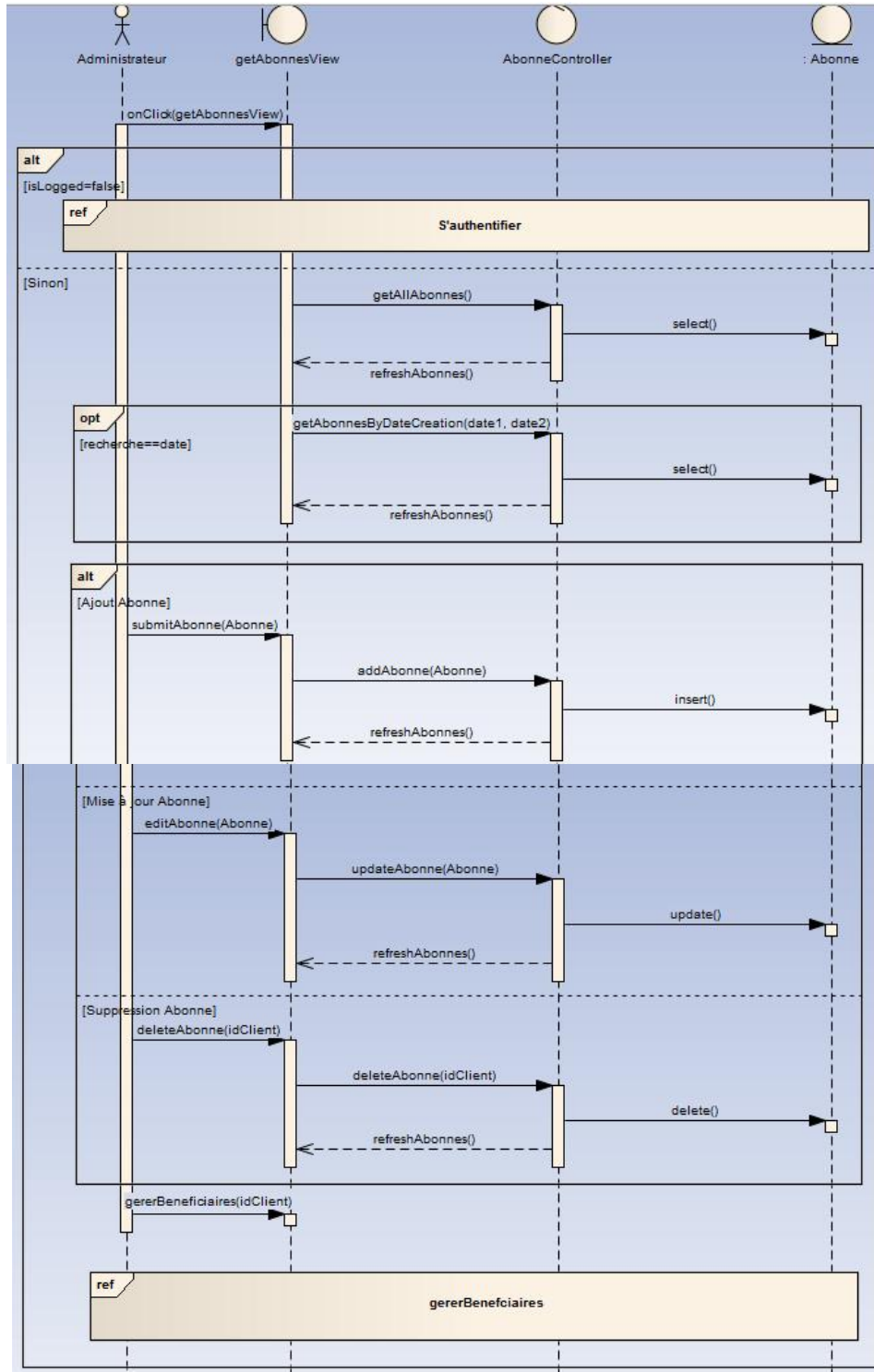


Figure 27 : Diagramme de séquences de cas d'utilisation « Consulter les abonnés »

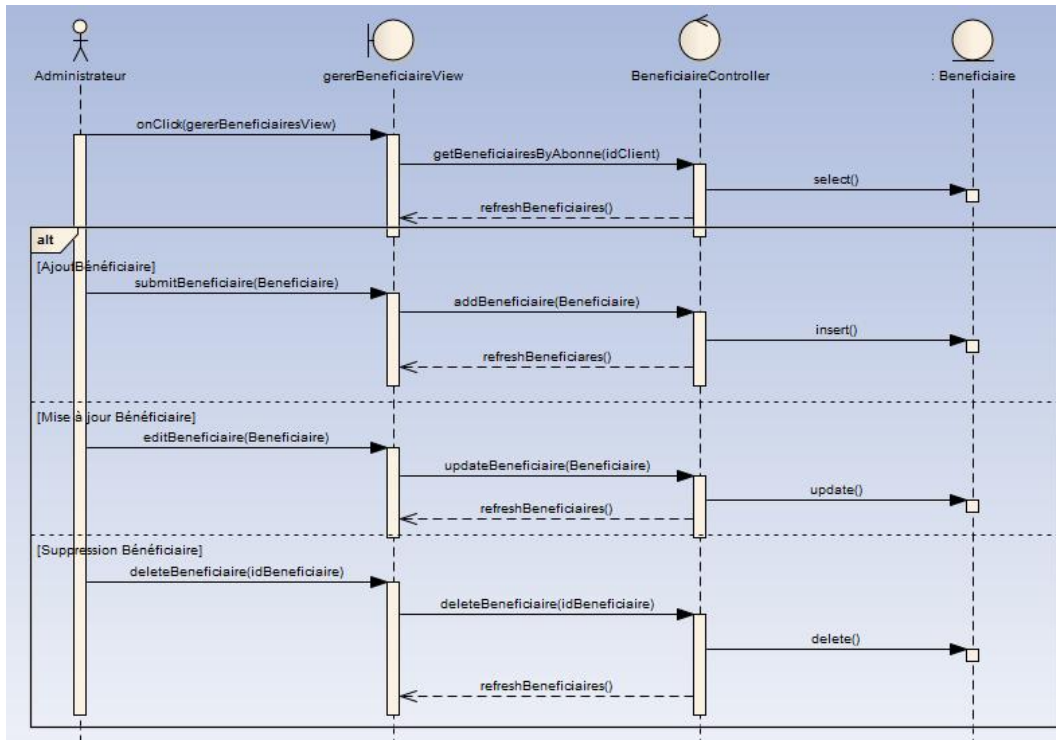


Figure 28 : Diagramme de séquences de cas d'utilisation « gérer les bénéficiaires »

D'abord, l'administrateur consulte la liste des abonnés puis il peut choisir de consulter leurs bénéficiaires. L'admin peut ajouter, éditer ou supprimer les abonnés et les bénéficiaires. La figure 27 illustre le diagramme de séquences du cas d'utilisation « Consulter les comptes » et la figure 28 illustre « Consulter les opérations ».

III.5.10. Cas d'utilisation « Consulter les virements »:

i. Description textuelle :

La table 13 décrit textuellement le cas d'utilisation « Consulter les virements »

Table 13 : Description textuelle du cas d'utilisation « Consulter les comptes »

Cas d'utilisation	S'authentifier
Acteur	Administrateur
Pré condition	L'utilisateur E-Banking est identifié comme étant un admin.
Post condition	L'admin consulte ses virements avec succès.
Description du scénario principal	<ul style="list-style-type: none"> - A travers un menu, l'abonné choisit « Consulter les virements ». - L'application affiche les virements. - L'admin peut valider/invalider les virements.

ii. Diagramme de classes de cas d'utilisation :

Le diagramme de classes du cas d'utilisation « Consulter les virements » (figure 29).

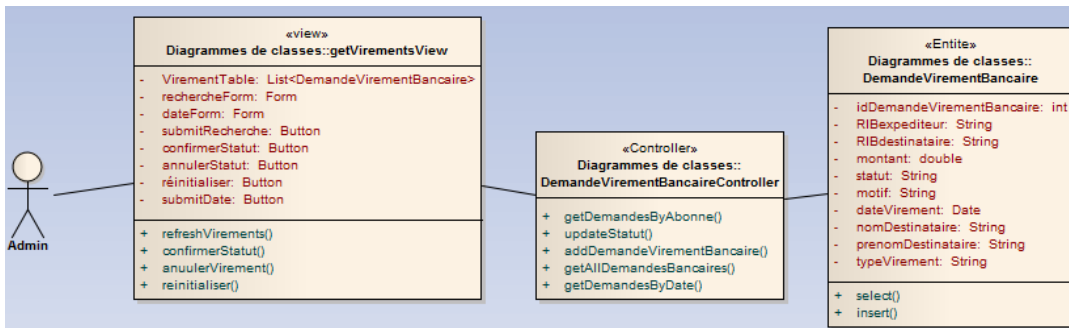


Figure 29 : Diagramme de classes de cas d'utilisation « Consulter les virements »

iii. Diagramme de séquences :

D'abord, l'admin consulte la liste des virements puis il peut choisir de les valider ou invalider.

La figure 30 illustre le diagramme de séquences du cas d'utilisation « Consulter les comptes ».

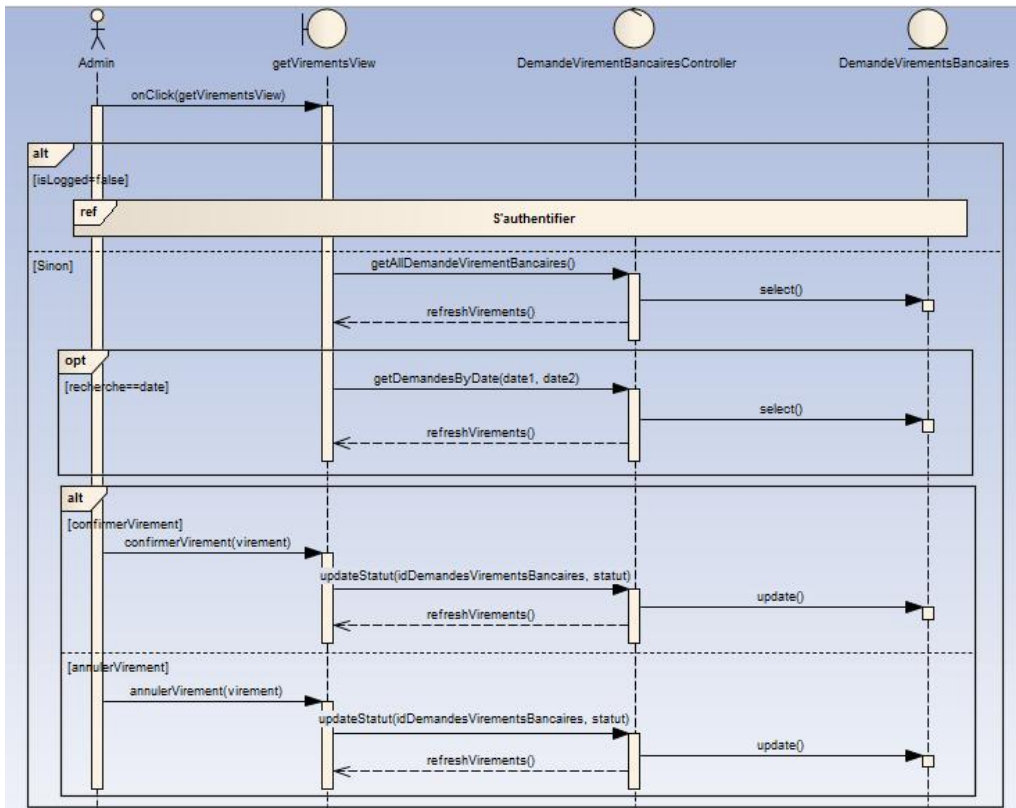


Figure 30 : Diagramme de séquences de cas d'utilisation « Consulter les virements »

III.6. Conclusion :

Au long de ce chapitre, nous avons présenté la conception de l'application ; en identifiant les utilisateurs et leurs rôles, les cas d'utilisations, les classes participantes et le fonctionnement de l'application. Dans le prochain chapitre, nous allons développer et coder l'application.

Chapitre IV : Codage et déploiement

IV.1. Introduction :

Dans ce chapitre, nous donnerons un aperçu sur le travail accompli au niveau de codage et déploiement des l'application défini dans les chapitres précédents. Plus précisément, les travaux menés dans cette étape résument tout simplement dans l'implémentation et la réalisation des cas d'utilisation analysés lors des étapes précédentes, donc nous allons présenter les diagrammes de composant suivi par le diagramme de déploiement.

IV.2. Codage :

Pour notre cas, le codage se traduit par un diagramme de composants (Figure 31).

Nous remarquons l'architecture MVC (Model, Controller, View), de plus la base de données présentés par une table SQL et les DAO (Data Access Object).

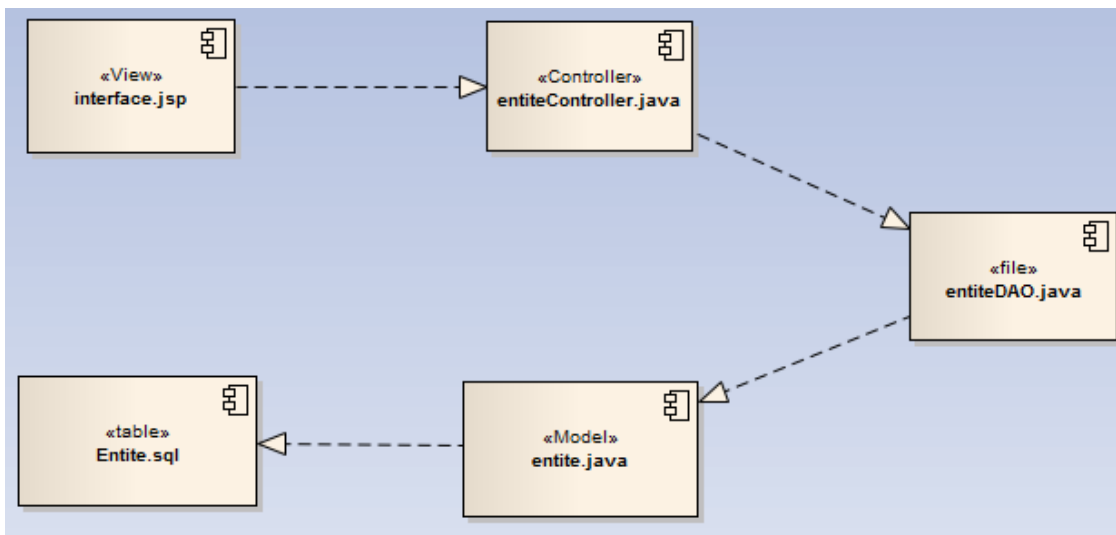


Figure 31 : Diagramme de composants de l'application E-Banking

IV.3. Déploiement :

Pour notre cas, le déploiement se traduit par un diagramme de déploiement qui permet de représenter la répartition géographique des composants matériels de notre système sous forme de nœuds. (Figure 32).

L'utilisateur demande une ressource auprès de son ordinateur. Ceci se traduit par une requête HTTP qui sera envoyée à l'un des serveurs de l'application. Ce dernier fournit la ressource, mais en faisant appel à un autre serveur qui est le serveur de la base de données.

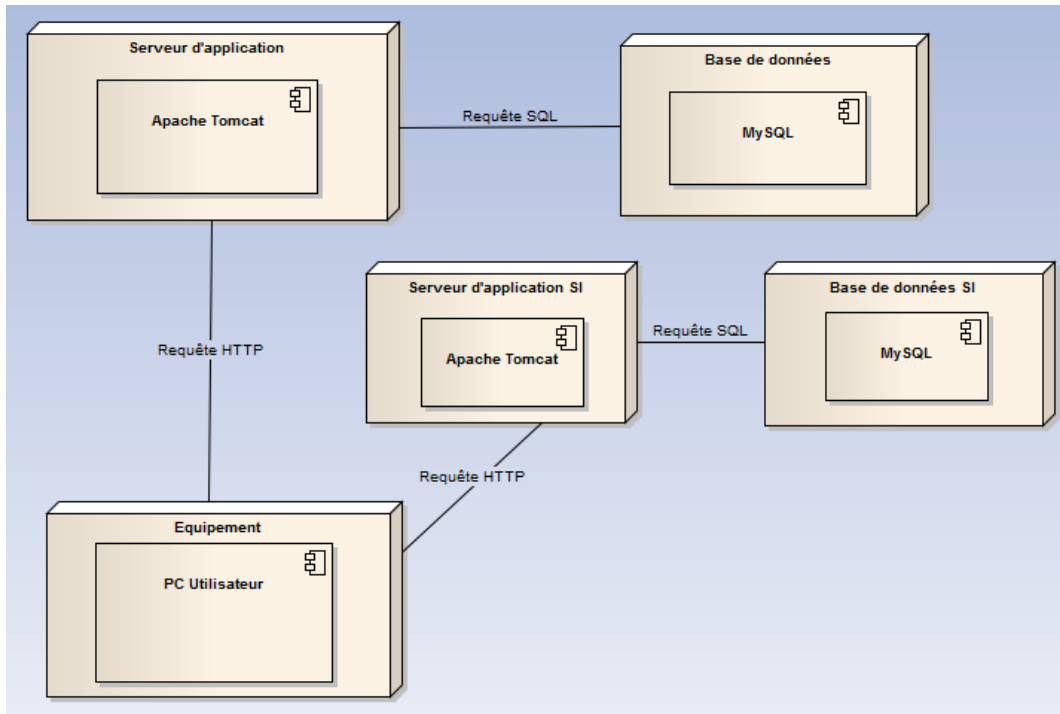


Figure 32 : Diagramme de déploiement de l'application E-Banking

IV.4. Framework JavaScript : AngularJS

Le Framework AngularJS a été imposé par l'entreprise BFI comme il est compatible avec les fonctionnalités E-Banking. De plus, il fournit une manière structurée de construire des applications JavaScript complètes, utilise une architecture MVC, idéal pour SPA et fournit un moyen plus facile de charger dynamiquement des morceaux de code et des vues modulaires dans notre application.

IV.5. Conclusion :

Au cours de ce chapitre, nous avons essayé de donner une meilleure visibilité du projet sur le plan technique, présenter diagramme de déploiement et diagramme de composants de l'application globale.

Chapitre V : Tests

V.1. Introduction :

Dans ce chapitre, nous illustrons les captures d'écran qui représentent quelques tests qui assurent le bon fonctionnement de l'application « E-Banking ».

V.2. Tests :

V.2.1. Cas d'utilisation « S'authentifier »:

La figure 33 représente l'interface d'authentification. La figure 34 représente le message d'erreur.

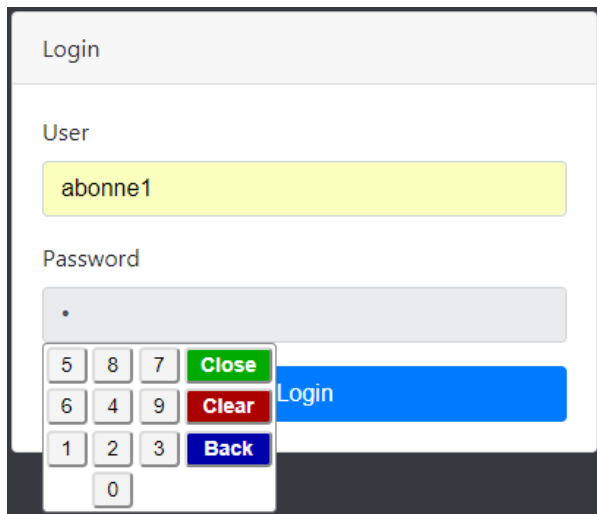


Figure 33 : interface d'identification

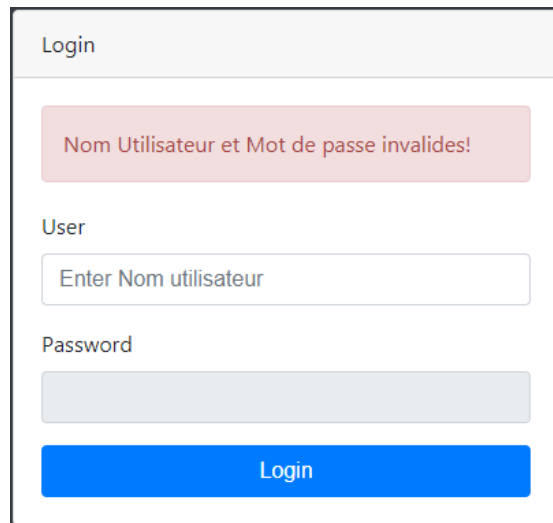


Figure 34 : interface d'identification message d'erreur

V.2.2. Cas d'utilisation « Commander un chéquier »:

La figure 35 représente l'interface de la création d'une commande chéquier.

Créer une commande Chéquier

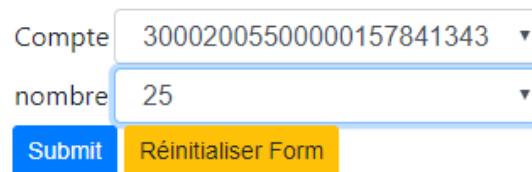


Figure 33 : interface de commande chéquier

V.2.3. Cas d'utilisation « Demander un virement interbancaire »:

Chapitre V : Tests

La figure 36 représente l'interface de la création d'un virement interbancaire. La figure 37 présente un autre prompt qui demande de l'abonné d'entrer la code de validation envoyée.

Créer un virement interbancaire

RIB Expéditeur 30002005500000157841343 Solde :1611 DT

RIB Destinataire(Beneficiaire) 30002005500000016736801

nom destinataire Chbil

prenom destinataire Hamza

montant 250,500

motif Paiement

Figure 36 : interface de création d'un virement interbancaire

Créer un virement interbancaire

RIB Expéditeur 30002005500000157841343 Solde :1611 DT

RIB Destinataire(Beneficiaire) 30002005500000016736801

nom destinataire

prenom destinataire

montant

motif Paiement

Code Validation

Figure 37 : interface de vérification du code de validation

V.2.4. Cas d'utilisation « Consulter les comptes »:

La figure 38 représente l'interface de la liste des comptes de l'abonné. La figure 39 représente les opérations d'un compte particulier.

Comptes

Recherche

10 ▾

RIB	Type	Solde	DateCreation
30002005500000157841343	épargne	1611Dt	2018-04-05
30004000031234567890143	courant	1187Dt	2018-04-05

Figure 34 : interface de consultation des comptes

5 ▾

Description	Valeur	Date
Virement interBancaire a 30002005520000156739587	-100Dt	2018-05-11
Virement intraBancaire de 30004000031234567890143	100Dt	2018-05-11
Virement intraBancaire a 30004000031234567890143	-350Dt	2018-05-11
Virement intraBancaire a 30004000031234567890143	-300Dt	2018-05-11
Virement intraBancaire a 30004000031234567890143	-100Dt	2018-05-07

« < 1 2 3 > »

Figure 35 : interface de consultation des opérations

V.2.5. Cas d'utilisation « Mettre à jour les coordonnées du client »:

La figure 40 représente l'interface de mise à jour des coordonnées du client. La figure 41 représente les messages d'erreur.

Chapitre V : Tests

Mise à jour du client

Sexe

Nom

Prenom

Email

DateDeNaissance

Telephone

Figure 40 : interface de mise à jour des coordonnées du client

Mise à jour du client

Sexe

Nom

*Nom ne doit pas être vide

Prenom

Email

*Email doit être valide

DateDeNaissance

*Date de Naissance doit être sous format : yyyy-MM-dd

Telephone

Figure 41 : les messages d'erreur

V.2.6. Cas d'utilisation « Consulter les abonnés »:

La figure 42 représente l'interface de la consultation des abonnés. La figure 43 représente l'option d'ajouter/modifier un abonné.

Id Client	Code Abonne	Password	Date de creation	Options
BF112345	abonne1	\$2a\$04\$DfK5sGj7FjdFkM4P/IKpWOG.gzg4wPbKQDz6NKL4Ev1c68OcN0DwK	2018-04-07	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
BF134567	abonne3	\$2a\$04\$JzgWGVxltLZMKZ7pD/6Sr.vu9l6Pnorie/PdSFHTv6jmkKbtN5zi	2018-03-17	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
BF156789	abonne2	\$2a\$04\$ITT6Fluqlm/cBhrYyish.8Pnh9lLrIgDsOpMox1Bb13MrgurF5W	2018-04-01	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
BF172534	abonne4	\$2a\$04\$RWegMlvZm2A94m4CAupD.hiCaWXX/KQ9XVck8d68Nyo8pdsGPbbRu	2018-04-22	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Figure 42 : interface de consultation des abonnés

Id Client

Code Abonne

Password (Cryptage auto)

Date de Creation

Figure 43 : l'option d'ajouter/modifier un abonné

V.3. Conclusion :

Nous avons assuré le bon fonctionnement de l'application en démontrant les tests faits sur les différents cas d'utilisation. Cela présente la fin du développement de l'application « E-Banking ».

Conclusion

Les banques en ligne sont devenues les accompagnantes indispensables de l'homme. Ceci est dû principalement à la mise au point de multitude d'applications web qui couvrent beaucoup de secteurs, et parviennent même à satisfaire quelqu'un de nos opérations bancaires.

C'est en considération de ce cadre qu'il nous a été confié de réaliser, au sein de BFI, une application web qui constitue d'un système d'informations qui représente la base de données E-BankingSI d'une banque, et une base de données persistante E-Banking qui représente les utilisateurs. Les études conceptuelles et techniques, ainsi que les différentes étapes qui ont abouti à la réalisation de notre solution, ont fait l'objet des chapitres de ce rapport.

Ce stage représente pour nous une opportunité irremplaçable, qui nous a permis de participer à un projet d'envergure, de faire face aux contraintes réelles de l'entreprise, et de vivre les différentes phases clés du développement et la production d'un logiciel.

Pour accomplir ce travail, UML est désormais notre langage de modélisation en utilisant le logiciel Enterprise Architect. L'application est développée en langage JAVA (Spring MVC) sous Eclipse Oxygen. D'autre part, nous avons développé des web services REST permettant de gérer la connexion avec le serveur de base de données. Afin de créer la base de données, un système de gestion de base de données MySQL a été exploité.

Ces choix nous ont permis de réaliser une application complète répondant aux attentes de l'entreprise pour laquelle nous avons œuvré.

À l'issue de ce projet, nous pouvons envisager les perspectives suivantes :

- Multilinguisme de l'application.
- Réaliser une version mobile.
- Augmenter le niveau de sécurité de l'application.
- Améliorer le design de l'application.

Bibliographie

[1] **Présentation de BFI. Edition et Intégration de solutions logicielles bancaire, marché bancaire : BFI Tunisie** [en ligne]. [Consulté le 12 Mai 2018]. Disponible à l'adresse: <http://www.bfigroupe.com/Fr/presentation-de-bfi> 11 5

[2] **Global Banking. Edition et Intégration de solutions logicielles bancaire, marché bancaire : BFI Tunisie** [en ligne]. [Consulté le 12 Mai 2018]. Disponible à l'adresse: <http://www.bfigroupe.com/Fr/global-banking> 59 9

[3] **Technologie. Edition et Intégration de solutions logicielles bancaire, marché bancaire : BFI Tunisie** [en ligne]. [Consulté le 12 Mai 2018]. Disponible à l'adresse: <http://www.bfigroupe.com/Fr/technologie> 11 41

[4] **Enterprise Architect. Enterprise Architect - Use Case Model** [en ligne]. [Consulté le 17 Mai 2018]. Disponible à l'adresse: <http://www.sparxsystems.com/products/ea/index.html> (Anglais)

[5] **Site additeam.com** [en ligne]. [Consulté le 17 Mai 2018]. Disponible à l'adresse: <http://www.additeam.com/SSII/uml/>

[6] **Site SlidePlayer.com** [en ligne]. [Consulté le 17 Mai 2018]. Disponible à l'adresse: <http://slideplayer.com/slide/8737567/>

[7] **GARDEUX, Vincent. Spring. Bienvenue - JEE Framework** [en ligne]. [Consulté le 17 Mai 2018]. Disponible à l'adresse: http://gardeux-vincent.eu/Documents/ProjetJEE/BACSWW_Hibernate_Jaxb_Spring/content/spring.html

[8] **Spring Security. Spring Data REST** [en ligne]. [Consulté le 17 Mai 2018]. Disponible à l'adresse : <http://projects.spring.io/spring-security/>

[9] **Business : actualités et tendances sur JDN. JDN : E-business, FinTech, Big Data, IoT, tendances média, décideurs...** [en ligne]. [Consulté le 19 Mai 2018]. Disponible à l'adresse : <http://www.journaldunet.com/encyclopedie/definition/972/34/20/tomcat.shtml>

[10] **Wiki du FabLab Robert Houdin. tuto_installation_de_eclipse [Wiki du FabLab Robert Houdin]** [en ligne]. 31 October 2013. [Consulté le 19 Mai 2018]. Disponible à l'adresse : http://fablab-robert-houdin.org/wiki/doku.php?id=tuto_installation_de_eclipse

[11] **Pourquoi MySQL? MySQL** [en ligne]. [Consulté le 19 Mai 2018]. Disponible à l'adresse : <https://www.mysql.com/fr/why-mysql/>

[12] **PORTER, Brett, ZYL, Jason van and LAMY, Olivier. Maven – Welcome to Apache Maven. Maven – Installing Apache Maven** [en ligne]. [Consulté le 19 Mai 2018]. Disponible à l'adresse : <https://maven.apache.org/>

[13] **FUTURA. Java. Futura** [en ligne]. [Consulté le 19 Mai 2018]. Disponible à l'adresse : <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-java-485/>

[14] **AngularJS** [en ligne]. [Consulté le 19 Mai 2018]. Disponible à l'adresse : <https://docs.angularjs.org/misc/faq>

Annexes

Schéma de la base de données :

Pour construire le schéma de base de données de notre application, nous devons appliquer certaines règles pour passer d'un schéma entité association (diagramme de classe) vers un schéma relationnel.

La déduction du schéma relationnel se base sur un ensemble de règles qui sont présentées comme suit :

- Règle 1 : Toute entité devient une table.
- Règle 2 : Chaque propriété de l'entité devient un attribut de la table.
- Règle 3 : L'identifiant de l'entité devient une clé primaire de la table
- Règle 4 : Chaque association « un à plusieurs » est représentée par une clé étrangère dans la table fille.
- Règle 5 : Chaque association devient une relation, si les cardinalités maximales sont «n..n»
- Règle 6 : Chaque association «1..1» est représentée par l'intégration d'une clé étrangère dans la table la moins récente.
- Règle 7 : Chaque classe association entre deux classes est représentée par une table qui prend pour clé primaire la concaténation des clés primaires des deux classes.

Dans ce qui suit, nous présentons les tables de la base de données EBanking.

Table : Administrateur		
Champ	Type	Contraints
idAdministrateur	Varchar(8)	Clé primaire
Username	Varchar(40)	---
Password	Varchar(60)	---

Table : Abonné		
Champ	Type	Contraints
idClient	Varchar(8)	Clé primaire
CodeAbonne	Varchar(40)	Unique
PasswordAbonne	Varchar(60)	---
dateCreation	Date	---

Table : Bénéficiaire		
Champ	Type	Contraints
idBeneficiaire	int	Clé primaire, auto-incrément
RIB	Varchar(23)	---
Nom	Varchar(255)	---
Prenom	Varchar(255)	---
Statut	Varchar(255)	---
CodeValidation	Varchar(255)	---
idClient	Varchar(8)	Clé étrangère

Table : DemandeVirementBancaire		
Champ	Type	Contraints
idDemandeVirementBancaire	int	Clé primaire, auto-incrément
RIBexpediteur	Varchar(23)	---
RIBdestinataire	Varchar(23)	---
Montant	Double	---
Motif	Varchar(255)	---
Statut	Varchar(255)	---
DateVirement	Date	---
TypeVirement	Varchar(255)	---
idClient	Varchar(8)	Clé étrangère
idBeneficiaire	Varchar(8)	Clé étrangère

Dans ce qui suit, nous présentons les tables de la base de données EBankingSI.

Table : ClientSI

Champ	Type	Constraints
idClient	Varchar(8)	Clé primaire
Sexe	Varchar(255)	---
Nom	Varchar(255)	---
Prenom	Varchar(255)	---
Email	Varchar(255)	---
Telephone	Int	---
DateNaissance	Date	---

Table : CompteSI

Champ	Type	Constraints
RIB	Varchar(23)	Clé primaire
Type	Varchar(255)	---
Solde	Double	---
dateCreation	Date	---
idClient	Varchar(8)	Clé étrangère

Table : OperationSI

Champ	Type	Constraints
idOperation	Int	Clé primaire, auto-incrément
Description	Varchar(255)	---
Valeur	Double	---
dateOperation	Date	---
RIB	Varchar(23)	Clé étrangère

Table : CmdChequierSI		
Champ	Type	Constraints
idCmdChequier	Int	Clé primaire, auto-incrément
Statut	Varchar(255)	---
Nombre	Int	---
idClient	Varchar(8)	Clé étrangère
RIB	Varchar(23)	Clé étrangère

Table : CmdCarteSI		
Champ	Type	Constraints
idCmdCarte	Int	Clé primaire, auto-incrément
Statut	Varchar(255)	---
Type	Varchar(8)	---
idClient	Varchar(8)	Clé étrangère
RIB	Varchar(23)	Clé étrangère

ملخص: تنفيذ حل الوصول الآمن للبنوك الإلكترونية عن طريق إنشاء قاعدة بيانات بحيث يمكن للمستخدمين استخدام ميزات الحساب والبحث عن المعاملات وتنفيذ التحويلات داخل وبين البنوك مع مجموعة محددة من المستخدمين. من ناحية أخرى، يستخدم المشرف هذا الحل لإضافة المستخدمين وتعديلهم والتعامل مع المستخدمين منه.

المراجع: JEE, Spring MVC, Hibernate, AngularJS, REST, MySQL: الخدمات المصرفية الإلكترونية

Titre : Développement d'un projet E-Banking en JEE

Résumé : Implémenter une solution d'accès sécurisé E-Banking en créant une base de données MySQL de sorte que les abonnés puissent utiliser les fonctionnalités de consultation du compte et des opérations et l'exécution des virements intra et interbancaire avec un ensemble spécifique des bénéficiaires. D'autre part, l'admin utilise cette solution pour ajouter et modifier les abonnés et manipuler leurs bénéficiaires.

Mots-clés: EBanking, JEE, Spring MVC, Hibernate, AngularJS, REST, MySQL

Title: Development of an E-Banking project with JEE

Abstract: Implement an E-Banking solution with secured access by creating a MySQL database so that subscribers can use their accounts' and transactions' lookup features and the execution of intra and interbank transfers with a specific set of beneficiaries. On the other hand, the admin uses this solution to add and modify subscribers and manipulate their beneficiaries.

Key Word: EBanking, JEE, Spring MVC, Hibernate, AngularJS, REST, MySQL

Réalisé à :

Entreprise : BFI

Adresse : Rue du lac Ghar El Melh, 1053, Les Berges du Lac, Tunis

Tél. : 71962030

Email : ykacem@bfigroupe.com