|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| République Tunisienne |  |  |  | **Ingénieur en:**  Génie Informatique Industrielle  **Option :**  Réseaux et Contrôle Industriels  ***Projet de Fin d’Etudes***  **N° d’ordre:** ING-GII-53- 16 |
| Ministère de l’Enseignement Supérieur  et de la Recherche Scientifique |
|  |
| Université de Sfax |
|  |
| Ecole Nationale d’Electronique et des Télécommunications de Sfax |

# MEMOIRE

***présenté à***

L’[É](https://fr.wikipedia.org/wiki/%C3%89lectronique)cole Nationale d’[É](https://fr.wikipedia.org/wiki/%C3%89lectronique)lectronique et des Télécommunications de Sfax

***en vue de l’obtention du***

Diplôme National d’Ingénieur en :

Génie Informatique Industrielle

Option :

Réseaux [É](https://fr.wikipedia.org/wiki/%C3%89lectronique)lectrique Intelligents

***par***

Ali BEN YOUSSEF

E-BANKING

***soutenu le 14 juin 2016, devant la commission d'examen:***

|  |  |  |
| --- | --- | --- |
| **Mme.** | **Ikram AMOUS** | *Président* |
| **Mme.** | **Hana HAMDEN** | *Examinateur* |
| **M.** | **Mahmoud KARRAY** | *Encadrant* |
| **M.** | **Sofian CHARFEDDINE** | *Encadrant* |

***Dédicaces***

À ma Mère,

« Tu m’as donné la vie, la tendresse et le courage pour réussir.

Tout ce que je peux t’offrir ne pourra exprimer l’amour et la reconnaissance que je te porte.

En témoignage, je t’offre ce modeste travail pour te remercier pour tes sacrifices et pour l’affection dont tu m’as toujours entourée. »

À mon Père,

« L’épaule solide, l’œil attentif compréhensif et la personne la plus digne de mon estime et de mon respect.

Aucune dédicace ne saurait exprimer mes sentiments, que Dieu te préserve et te procure santé et longue vie ! »

À mon frère Achref et mes sœurs Imen et Arije pour leurs encouragements incessants.

À tous mes amis en souvenir des bons moments que nous avons passés ensemble, pour leur soutien continu, leur aide précieuse et leur amour.

À tous ceux qui me sont chers, avec tous mes souhaits de réussite et de bonne santé.

***Remerciements***

Je tiens, avant tout à adresser mes plus vifs remerciements à Monsieur Habib BEN HARIZ, Directeur Général de la Banque et Finance Internationale, pour m’avoir acceptée et m’avoir offert les conditions nécessaires pour l’élaboration de ce travail.

J’exprime ma gratitude à Monsieur Charfeddine SOUFIANE, mon encadrant de la société BFI, pour le temps consacré à mon projet. J’espère être à la hauteur de sa confiance.

J’exprime aussi ma gratitude et mes plus vifs remerciements à Monsieur Mahmoud KARRAY, mon encadrant de l’Ecole Nationale d'Electronique et des Télécommunications de Sfax pour son suivi, pour m’avoir consacrée de son temps précieux et m’avoir prodigué des conseils tout au long de l’élaboration de ce travail et pour son encouragement, chose qu’il me fallait pour la réalisation de ce projet.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce modeste travail.

Qu’ils puissent tous, trouver ici, l’expression de notre profonde gratitude et de notre grand respect.

**Introduction générale**

Les innovations des technologies de l'information et de la communication (TIC), adossées à la globalisation de l'économie mondiale, ont induit une accélération de mouvements de capitaux telle qu'elle requiert des systèmes transactionnels modernes et efficaces visant à sécuriser et harmoniser ces importants flux financiers. Le développement des instruments de paiement par voie électronique qui vient se substituer progressivement à l'échange physique des moyens de paiement, constitue une des réponses bancaires à cette logique visant à plus de sécurité, plus d'efficacité , plus de fluidité et de rapidité.

L'activité bancaire a effectivement connu une mutation importante en matière de distribution des services. Aux développements et perfectionnements des automates bancaires et serveurs vocaux est venue s'ajouter l'offre de services par Internet. L'agence n'est plus, comme autrefois, le canal de distribution exclusif de la banque. Ainsi les distributeurs automatiques de billets, les guichets automatiques de banque et Internet ont fait successivement voler en éclats l'unité de lieu, de temps et d'action, principe si cher aux institutions bancaires. L'objectif étant d'ajouter le plus d'éléments de satisfaction possible en vue de fidéliser une clientèle par ailleurs «  volatile » du fait d'une concurrence farouche entre établissements.

Dans ce contexte, il nous a été confié d’explorer les différentes solutions existantes en e-banking et détecter les carences en matière de sécurité, pour pouvoir ensuite concevoir et réaliser deux application qui se chargerons de satisfaire les besoins de tout client possédant un compte bancaire et souhaitant bénéficier des services bancaires à distance sans avoir à se soucier de l’interception de ses données personnelles.

Ce rapport est organisé en trois chapitres. Nous commençons par un premier chapitre qui décrit le cadre général de notre projet et les spécifications de besoins pour une meilleure compréhension du contexte. Le deuxième chapitre est consacré à analyser et présenter les différents aspects conceptuels de l'application spécifier les besoins de notre application ainsi que l’architecture de notre solution.

Le dernier chapitre porte sur la partie réalisation. Il illustre en premier temps les détails de l’environnement matériel et logiciel utilisé afin de réaliser notre application. De même, il comporte les interfaces de notre application qui permettent à l’utilisateur d’y accéder.

# Chapitre 1 : Cadre général du projet et spécification des besoin

# 1 Cadre général du projet

## 1.1 Introduction

E-banking est une abréviation de services bancaires électroniques. Il nous permet d'effectuer des transactions bancaires en ligne, au lieu de trouver une banque et d'interagir avec un banquier. La plupart des banques offrent des services e-banking, bien que l'étendue de ceux-ci puisse varier.

Dans ce chapitre, nous élaborons une introduction à notre travail en allant d’une étude de l’existant jusqu’à définir les objectifs et les besoins de notre projet.

## 1.2 Présentation de l’organisme d’accueil

BFI est spécialisée depuis 1994 dans l’édition et l’intégration de solutions 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 180 clients dans 15 pays différents notamment avec des institutions de renommée mondiale.

Forte d’une expérience de plus de 15 ans d’installations réussies en Afrique et d’une politique de proximité auprès de ses clients, BFI offre un savoir-faire reconnu dans la mise en œuvre de projets couvrant : le suivi et le pilotage, l’intégration de systèmes, la méthodologie de test et de recette, la migration des données, la formation et le transfert de compétences, le conseil et l’accompagnement et enfin le support et la maintenance.

## 1.3 Etat de l’art sur l’E-Banking

### *1.3.1 Emergence des TIC*

#### 1.3.1.1 Société de l’information

Au cours de ces vingt dernières années, les TIC ont connu une croissance spectaculaire. D'une société essentiellement industrielle, l'économie mondiale évolue progressivement vers une société dite de « l'information ». Cette évolution, généralement reprise sous l’appellation « nouvelle économie », se caractérise par un phénoménal potentiel de croissance, d'intégration et d'emploi. Les technologies numériques ont rendu l'accès, le traitement, le stockage et la transmission des informations de plus en plus aisés et de moins en moins onéreux.

#### 1.3.1.2 Position de la banque en ligne et profil des utilisateurs

Le phénomène de banque en ligne connait un bel essor, passant de 46% en 2006 à 52% en 2007. Cette hausse touche toutes les catégories de personnes. En ce qui concerne l’utilisation de la banque en ligne, les personnes ayant un niveau d’instruction moyen ou élevé l’emploient plus que celles ayant un niveau d’instruction faible.

De plus, il est intéressant de noter qu’il n’y a pas de grosses différences significatives entre les différentes tranches d’âge, à l’exception des moins de 24 ans et des plus de 64 ans qui sont tout de même un peu en retrait par rapport au reste de la population. Le fait que la tranche la plus jeune de la population soit elle aussi en retrait n’est pas surprenant, ceci peut s’expliquer aisément par la dépendance financière de ceux-ci vis-à-vis de leurs parents. Certains n’ont peut-être tout simplement pas de compte en banque.

### *1.3.2 Evolution du secteur bancaire*

#### 1.3.2.1 Relation banque-client d’aujourd’hui

S’il y a bien une certitude c’est que le client d’aujourd’hui est tout-à-fait différent de celui d’hier. Il y a 30 ans, les banques entretenaient une relation proche et personnelle avec leurs clients, lesquels se rendaient régulièrement en agences afin d’y déposer des chèques, rencontrer le personnel afin de recevoir des conseils, etc. Progressivement, ces clients ont été poussés hors des agences à cause de ou grâce à la technologie, et plus spécifiquement avec l’utilisation du téléphone et plus récemment de l’Internet. Tout cela a changé la relation banque-client. Ce dernier est maintenant à distance et en ligne, avec comme conséquence que les banquiers ne voient plus leurs clients.

Le résultat qui en découle est que la banque est devenue un fournisseur de produits de commodité : les services deviennent du « self-service » (Skinner, 2007). Cette évolution n’aurait su se faire sans la réalisation de la vision de Bill Gates « a computer on every desk and in every home», devenue réalité dans les années 2000. Cette plus grande accessibilité a pour résultat que la plupart des technologies développées aujourd’hui sont conduites non pas par les gouvernements ou les entreprises mais bien par les consommateurs. Selon (Skinner, 2007), le défi actuel des banques est de soutenir le rythme, voir anticiper, la demande des consommateurs en matière d’accès et de services.

#### 1.3.2.2 Révolution technologique

La technologie est le facteur le plus critique dans le processus bancaire car la banque est une entreprise digitale. C’est une entreprise transactionnelle basée sur des nombres. Si nous faisions une analogie, le secteur des télécommunications serait certainement le plus proche, avec comme différence principale que les banques connectent les finances tandis que les télécommunications connectent la société.

Durant les dernières décennies, un certain nombre d’innovations et de révolutions technologiques est intervenu dans le secteur bancaire. Brièvement, nous retrouvons entre autres :

* L’informatisation et l’automatisation du secteur bancaire ;
* Les cartes à puce ;
* La prolifération des outils informatiques.

#### 1.3.2.3 Première application de banque en ligne

Le premier service de banque en ligne à domicile britannique a été lancé par la Nottingham Building Society (NBS) en 1983. Le système utilisé était basé sur les terminaux Prestel (abréviation pour « Press Telephone ») des Royaumes Unis et consistait en un terminal connecté au système téléphonique. Le système permettait de visionner en ligne les états bancaires, de réaliser des transferts et le paiement de factures. Dans le but de procéder à un transfert bancaire et de régler une facture, un avis écrit avec les détails du destinataire devait être envoyé à la NBS qui encodait alors ces détails dans le système Homelink. Les destinataires typiques étaient les compagnies de gaz, d’électricité et de téléphone, ainsi que d’autres banques. Les détails de paiements étaient introduits dans le système NBS par le détenteur du compte via Prestel. Un chèque était alors envoyé par NBS au bénéficiaire et un avis reprenant les détails du paiement était envoyé au propriétaire du compte.

### *1.3.3 Fonctionnalités* *proposées de nos jours*

Le tableau suivant représente une étude comparative entre les différentes banques en Tunisie selon les services e-banking qu’elles offrent.

Tableau 1 Comparaison entre différentes banques selon les services e-banking offerts

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ouvrir un compte | Consultation soldes | Ordres de virement | Opérations de paiement | Commandes de chéquiers | Demandes de cartes | Accès aux opérations en bourse | Personnalisation du profil |
| \*BNA |  | X | X |  | X | X | X | X |
| \*UBCI | X | X | X |  | X |  |  | X |
| \*UBS |  | X | X | X | X | X | X | X |
| \*BCF | X | X | X | X | X | X | X | X |

\*BNA : Banque Nationale Agricole

\*UBCI : Union Bancaire pour le Commerce et l’Industrie

\*UBS : Union de Banques Suisses

\*BCF : Banque Cantonale de Fribourg

### *1.3.4 Atouts de l’E-banking*

#### 1.3.4.1 Du point de vue de la banque

##### Faster : un gain de temps considérable

Le fait que les clients réalisent la majorité de leurs opérations par voie électronique conduit à un gain de temps conséquent pour les employés de la banque. A cet égard, si nous nous penchons sur la manière dont les banques sont aménagées, nous remarquons que tout a été pensé et conçu de façon à ce que le client s’adresse le moins possible à une personne physique. Cette automatisation a pour résultat une accélération du temps de traitement des opérations.

##### Better : de nouvelles possibilités

A l’origine, les sites web des banques étaient réduits à de simples vitrines destinées à présenter l’entreprise. Par la suite, ils ont évolué vers de véritables outils transactionnels avec lesquels la clientèle peut obtenir des informations sur les produits bancaires proposés, suivre ses comptes, effectuer des opérations liées à la gestion de ceux-ci, voir contracter des prêts ou réaliser des opérations d’investissements.

##### Cheaper : coût de transactions réduit

Selon (DeYoung ,2003), l’Internet a fortement réduit le coût de livraison de la plupart des services financiers. La rentabilité d’une entreprise dépend en partie de ses coûts. Parmi ceux-ci, les coûts de transaction sont primordiaux. Les technologies digitales ont grandement contribué à une réduction des coûts de compilation, de traitement, et de distribution de l’information (Harper et Chan, 2003). En effet, si nous nous focalisons sur l’e-banking, nous pouvons constater que comparativement, ces coûts sont substantiellement moindres que ceux des autres canaux de distribution.

#### 1.3.4.2 Du point de vue du client

##### Anytime : disponibilité 24h sur 24

Disparition des contraintes liées aux heures et aux jours d’ouverture des agences bancaires. Le client peut réaliser ses opérations à n’importe quel moment.

##### Anywhere : une offre transfrontalière

Au-delà de la dépersonnalisation et de l’automatisation, l’Internet a pour caractéristique essentielle la distance dans les rapports client-fournisseur. (DeYoung, 2003) souligne d’ailleurs que l’Internet réduit l’importance du facteur géographique dans la production de services financiers et l’entretien de relations financières. Le client a donc la possibilité d’effectuer rapidement et aisément des opérations bancaires depuis son domicile, et de partout dans le monde pour peu qu’il dispose d’une connexion Internet.

##### Anything : de plus en plus de fonctionnalités proposées

Que ce soit du point de vue transactionnel ou non transactionnel, les fonctionnalités et les possibilités proposées aux clients sont de plus en plus nombreuses. Les applications se sont enrichies au fil des années et proposent maintenant tout un panel d’informations et de fonctions permettant une gestion de plus en plus complète.

### *1.3.5 Points sensibles de l’E-banking*

#### 1.3.5.1 Confiances et risques

Les risques opérationnels et de sécurité sont liés à des insuffisances dans la fiabilité ou l’intégrité du système. Les considérations de sécurité jouent donc un rôle prépondérant car les systèmes d’information peuvent faire l’objet d’attaques autant internes qu’externes.

#### 1.3.5.2 Complexité et évolution

La complexité concerne la difficulté de réaliser une transaction. De plus, l'un des challenges inhérents à toute solution informatique est sa capacité à s'adapter à l'évolution rapide de ce domaine.

## 1.4 Description générale du projet

Notre projet consiste à la réalisation de deux application de gestion des activités bancaires permettant aux clients abonnés aux services e-banking de bénéficier de nombreux services à distance, tels que la consultation du solde des comptes, la consultation du relevé d’opérations, la commande de chéquiers, etc.

L’application sera destinées à toute personne ayant un compte bancaire, avec probablement pour cœur de cible la ménagère de 20-65 ans étant plus alaise avec la technologie.

## 1.5 Objectifs visés

Dans nos applications, nous tenterons de résoudre les problèmes de sécurité rencontrés par les systèmes d’e-banking et mettre en place un système adapté aux nouvelles technologies.

En vue de réduire les risques et les problèmes déjà cités, il faudra :

* Assurer la disponibilité des services et des données ;
* Eviter l’interception non autorisée des données ;
* Confirmer que les données envoyées, reçues ou stockées sont complètes et non modifiées ;
* Garantir leur confidentialité par l’utilisation de mots de passe, et des techniques de cryptage ;
* Garantir une authentification fiable capable de confirmer l’identité supposée des abonnés.

# 2Spécification des besoins

Notre application doit satisfaire les exigences de la totalité des utilisateurs. Nous exposons dans ce qui suit leurs besoins fonctionnels ainsi que les besoins non fonctionnels communs à tous les acteurs.

## 2.1Besoins fonctionnels

Les besoins fonctionnels doivent expliciter ceux du client. Dans notre cas, des besoins exprimés par l’entreprise ont permis d’établir le cahier des charges suivant :

* **Consulter les comptes bancaires :** les utilisateurs ont la possibilité de consulter les soldes de leurs comptes ainsi que les détails que nous allons expliquer plus tard.
* **Consulter les mouvements d’un compte :** consiste à consulter à tout moment les mouvements des opérations bancaires (tels que virement, versement, retrait,…) d’un compte d’une période définie.
* **Effectuer des virements :** la possibilité d’effectuer des virements vers un compte interne ou externe à la banque.
* **Gérer les cartes bancaires :** les utilisateurs ont la possibilité de demander une carte bancaire et suivre son dénouement. Cette demande peut être acceptée ou refusée par la banque.
* **Gérer les chéquiers :** la banque accepte ou rejette une demande d’un chéquier provenant d’un client.
* **Gérer les crédits :** il s’agit de suivre l’état d’avancement des demandes de crédits par la banque et ses clients.
* **Communiquer:** la possibilité aux clients de communiquer avec la banque en déposant des requêtes et suivre les réponses de la banque et vice versa.
* **Gérer les utilisateurs :** la possibilité d’ajouter, modifier ou supprimer des utilisateurs du système.
* **Établir des statistiques :** cette fonctionnalité offre aux utilisateurs une vue d’ensemble des différentes opérations bancaires effectuées pendant une période.
* **Recevoir des documents :** ce module permet de recevoir un relevé de compte périodique détaillé sous format PDF qui sera conservé pendant 30 jours. Durant cette période l’utilisateur peut le consulter, l’imprimer et le télécharger.
* **E-Facture :** le client peut demander à sa banque la réception via ce module des factures (Electricité, Eau, Télécom, etc...). Ce qui lui permet, en cas d’acceptation, de les payer.
* **Effectuer les opérations en bourse:** l’utilisateur peut à travers ce module acheter et vendre des actions, consulter son portefeuille et suivre l’actualité de la bourse.
* **Suivre le taux de change :** les utilisateurs peuvent à tout moment connaitre l’appréciation des monnaies et consulter le résultat d’une conversion de devise.
* **Multimodalité :** les utilisateurs peuvent accéder aisément à cette application par n’importe quel type d’appareil (Ordinateur, Tablette, Smartphone, etc...).
* **Réutilisabilité :** cette application doit pouvoir être mise en place chez n’importe quelle institution bancaire moyennent quelques légères modifications spécifiques à chaque banque.

## 2.2Besoins non fonctionnels

Ces besoins sont les contraintes techniques exigées et les fonctionnalités nécessaires pour rendre le logiciel plus performant et qui se résument généralement en :

* **Sécurité :** les utilisateurs ne peuvent accéder aux fonctionnalités de l'application qu'après avoir réussi l'étape de l'authentification. En effet, il valide son authentification par le login et le mot de passe.

Afin d’assurer un niveau plus haut de sécurité, nous allons ajouter un clavier virtuel, les utilisateurs ainsi ne peuvent saisir leurs mots de passe qu’à travers ce clavier ; ce qui va empêcher une éventuelle attaque de type « keylogger » de ce produire. En ajoutant le déplacement aléatoire des touches de ce clavier et en cas ou un pirate arrive à craquer les mouvements et les clicks souris du client, il ne pourra en aucun cas reconstruire le vrai mot de passe parce qu’il n’obtiendra jamais une image statique de la disposition du clavier virtuel.

Pour assurer que le mot de passe stocké ne soit pas accessible en cas où le support de stockage est compromis, nous allons ajouter son hachage.

Enfin, nous avons utilisé un Time Out qui revoie l’utilisateur à la page d’authentification en cas de non activité pendant dix minutes.

* **L'ergonomie des interfaces :** l’interface de l’application doit être simple et claire : la manipulation de l'interface ne doit pas nécessiter des connaissances poussées en informatique.
* **Performance :** les interfaces doivent s’afficher à la demande des utilisateurs en un temps acceptable. Cela est assuré par l’utilisation des bonnes pratiques de développement.
* **Extensibilité :** l’application doit faciliter l’ajout de nouvelles fonctionnalités au moindre coût, pour cela nous allons utiliser le modèle MVC qui impose la séparation des couches donc la facilité d’amélioration à l’avenir.

**3 Conclusion**

Après cette brève description de notre projet et la spécification des besoins, attaquons-nous maintenant à la conception de notre application.

# 

# Chapitre 2. Conception

# Introduction

Comme étant une phase fondamentale dans le cycle de vie d’un logiciel, la phase de conception a pour objectif de fixer les choix techniques et de préparer les solutions pour trouver comment le problème est résolu.

Ceci sera illustré par des diagrammes de cas d'utilisations, diagrammes de séquences, d’activité et des diagrammes de classes.

# Identification des acteurs

L’application « E-Banking » est composée de quatre acteurs principaux. Nous citons ci-dessous les acteurs ainsi que les fonctionnalités leur permettant d’interagir et de bien mener leur travail.

**Client :**

Le client peut consulter les détails de ses comptes bancaires et ses mouvements, effectuer des virements, commander un chéquier, gérer les cartes bancaires, demander et suivre un crédit, communiquer avec sa banque, suivre les taux de change et l’évolution des différentes opérations bancaires, recevoir périodiquement les relevés de ses comptes, effectuer les opérations en bourse et recevoir et payer en ligne ces factures.

**Administrateur :**

Le rôle de l’administrateur est de gérer tout type d’utilisateur, suivre l’évolution des opérations effectuées dans sa banque et personnaliser son E-Banking.

**Conseiller clientèle** :

Le conseiller communique avec ses clients et répond à leurs demandes.

**Services communs** :

Nous appelons services communs, le système d’information déjà existant d’une banque. Cet acteur externe représente un système logiciel et permet d’offrir les « Web Services » nécessaires pour s’interfacer avec la banque.

## 2. Diagrammes des cas d'utilisation global

Nous donnons ci-dessous une vue globale concernant le comportement fonctionnel du système. Ce diagramme permet aussi de représenter les interactions entre les acteurs et les cas d’utilisation du système.

****

Figure 1: Diagramme de cas d'utilisation global

## 3 Diagrammes des cas d’utilisation détaillés

Nous allons maintenant détailler chaque cas d’utilisation vu dans le diagramme global précédent.

### *3.1 Cas d’utilisation «Gérer les opérations bancaires»*

Ce cas d’utilisation permet au client de consulter l’historique de ses mouvements bancaires entre deux dates précises et donne aussi la possibilité d’effectuer un virement de compte à compte.

Ces fonctionnalités sont représentées dans la figure suivante.



Figure 2 : Diagramme de cas d’utilisation détaillé «Gérer les opérations bancaires »

### *3.2 Cas d’utilisation «Gérer les cartes »*

À travers ce cas d’utilisation, le client peut déposer et suivre une demande d’une carte bancaire qui sera traitée par le conseiller client.

Ce diagramme est représenté à dans la figure suivante.



Figure 3 : Diagramme de cas d’utilisation détaillé «Gérer les cartes »

### *3.3 Cas d’utilisation «Gérer les chéquiers*»

À travers ce cas d’utilisation, le client peut déposer une demande de chéquier et recevra une réponse du conseiller client comme indiqué dans le diagramme suivant.



Figure 4 : Diagramme de cas d’utilisation détaillé «Gérer les chéquiers»

### *3.4 Cas d’utilisation «Établir des statistiques»*

À travers ce cas d’utilisation, le client ainsi que l’administrateur peuvent consulter les statistiques sur les opérations bancaires et les soldes comme le montre la figure suivante.



**Figure 5 :** Diagramme de cas d’utilisation détaillé «Établir des Statistiques»

### *3.5 Cas d’utilisation «Gérer les crédits»*

À travers ce cas d’utilisation, le client va pouvoir demander un crédit bancaire qui sera traité par le conseiller clientèle suivant le diagramme ci-après.



Figure 6 : Diagramme de cas d’utilisation détaillé «Gérer les crédits»

### *3.6 Cas d’utilisation «Communiquer»*

Ce cas d’utilisation donne la possibilité au client de déposer une requête qui sera traitée par le conseiller client ou l’administrateur. Il permet aussi au conseiller de passer une demande à son client. Voir le diagramme suivant.



Figure 7 : Diagramme de cas d’utilisation détaillé «Communiquer»

### *3.7 Cas d’utilisation «Tableau de bord»*

À travers ce cas d’utilisation, les clients et les conseillers vont recevoir des notifications sur les différents types de demande et l’administrateur va pouvoir suivre ses conseillers comme suit.



Figure 8 : Diagramme de cas d’utilisation détaillé «Tableau de bord»

### *3.8 Cas d’utilisation «Recevoir des documents*»

À travers ce cas d’utilisation, les clients vont recevoir chaque mois des relevés de comptes qu’ils peuvent télécharger ou imprimer à l’instar de ce qui suit.



Figure 9 : Diagramme de cas d’utilisation détaillé «Recevoir des documents»

### ***3.9 Cas d’utilisation «Suivre le taux change****»*

Les clients peuvent à tout moment consulter les taux de change et effectuer une conversion de devise telle qu’illustré dans le diagramme suivant.



Figure 10 : Diagramme de cas d’utilisation détaillé «Suivre le Taux de change»

### *3.10 Cas d’utilisation «E-Facture»*

À travers ce cas d’utilisation, le client peut demander des factures (d’électricité, d’eau, de Télécom, etc …) afin de les payer.

Nous montrons ce cas dans le diagramme suivant.



Figure 11 : Diagramme de cas d’utilisation détaillé «E-Facture»

### *3.11 Cas d’utilisation «E-Bourse»*

Ce cas d’utilisation donne la possibilité au client de consulter l’actualité de la bourse, demander l’achat ou la vente d’actions qui seront traités par son conseiller. Le client peut aussi consulter son portefeuille.

Nous présentons dans la figure suivante ce cas d’utilisation.



Figure 12 : Diagramme de cas d’utilisation détaillé «E-Bourse»

## 4 Description détaillée des diagrammes de séquences

## 4.1 Diagrammes des séquences détaillés

Un diagramme de séquence détaillé est une représentation détaillée des interactions entre les objets métiers de notre système selon un ordre chronologique. Nous exposons ainsi les différents diagrammes de séquences détaillés.

***4.1.1 Ajouter un client***

La vue « Clients » demande au conseiller : le nom, le prénom, le mail et le code client. Une fois ces informations fournies, le contrôleur « ClientsControlleur » se charge de générer un login et un mot de passe, d’enregistrer les informations associés au nouveau client et de lui envoyer un mail pour l’informer des données nécessaires lui permettant d’accéder à son compte.

Le diagramme de séquence suivant décrit ces étapes.



Figure 13 : Diagramme de séquence détaillé « Ajouter un client »

***4.1.2 Consulter les mouvements d’un compte***

En cliquant sur le menu « Mouvement », un client authentifié peut avoir la liste de ses comptes. Il peut ainsi choisir un d’entre eux afin de consulter les détails des mouvements. La vue ainsi affichée demande au client : le compte à détailler et les dates de début et de fin. Une fois ces informations fournies, le contrôleur « MouvementsController » se charge d’invoquer à travers le Web Service « executeWebService » l’opération « Mouvements » ainsi le client pourra consulter la réponse reçue par le contrôleur et formatée par la vue.

Le diagramme de séquence suivant décrit ces étapes.



Figure 14 : Diagramme de séquence détaillé « Consulter les mouvements d’un compte »

***4.1.3 Répondre à la demande de virement***

En arrivant à la vue des virements, un conseiller authentifié peut avoir la liste des demandes de virement de ses clients. Il peut ainsi choisir un d’entre eux et cliquer sur l’une des boutons accepter ou refuser ; ce qui fait appel au contrôleur qui va mettre à jour la demande au niveau de la base de données. Mais auparavant il doit invoquer à travers le Web Service « executeWebService » l’opération « Virement » qui va effectuer le virement en cas où le conseiller choisit d’accepter la demande.

La description de ces étapes figure dans le diagramme de séquence suivant.



Figure 15 : Diagramme de séquence détaillé « Répondre aux demandes de virement »

***4.1.4 Recevoir des relevés de comptes***

En cliquant sur le menu « E-Documents » par le client, le contrôleur « DocumentsController » se charge d’invoquer à travers le Web Service « executeWebService » l’opération « Comptes » afin de créer un fichier PDF par compte contenant ses mouvements. Cela est assuré par le contrôleur qui va invoquer à travers le Web Service « executeWebService » l’opération « Mouvements ».

Après la création des fichiers PDF, le contrôleur va afficher la liste des extraits de comptes. Une fois le client clique sur un extrait un fichier PDF s’affiche.

Ces opérations figurent dans le diagramme de séquence suivant.



Figure 16 : Diagramme de séquence détaillé « Recevoir des relevés de comptes»

**4.1.5 Digramme de séquence de la User story « Consulter les comptes »**

En cliquant sur le menu « Comptes », le contrôleur « ComptesController » se charge d’invoquer à travers le Web Service « executeWebService » l’opération « Comptes ». Ainsi, le client connecté pourra consulter la liste de ses comptes reçue par le contrôleur et formatée par la vue.

Le diagramme de séquence suivant décrit ces étapes.



Figure 17 : Diagramme de séquence détaillé « Consulter les comptes »

**4.1.6 Diagramme de séquence de la User story « Demander un virement »**

Lorsque le client arrive à la vue de virement, il peut choisir un numéro de compte et saisir le numéro du bénéficiaire et le montant à virer. Ces informations seront transmises au contrôleur pour être enregistrées dans la base. A la fin de l’opération d’enregistrement, le contrôleur réaffiche la vue « virement ».

Le diagramme de séquence suivant décrit ces étapes.



Figure 18 : Diagramme de séquence détaillé « Demander un virement »

**4.1.7 Diagramme de séquence de la User story « Suivre les demandes de virement »**

Un client peut suivre ses demandes de virement en cliquant sur le bouton virement. Cette action va activer le contrôleur afin de récupérer les informations déjà existantes au niveau de la base de données. Ces informations seront ensuite envoyées à la vue pour être affichées.

Le diagramme de séquence suivant décrit ces étapes.



Figure 19 : Diagramme de séquence détaillé « Suivre les demandes de virement »

# 5. Diagrammes d’activités

Le diagramme d’activité est un moyen graphique pour donner une vision d’ensemble sur des scénarios alternatifs et/ou d’exception. S’agissant d’un diagramme comportemental d’UML, il permet de modéliser un processus interactif, global ou partiel pour un système donné ([logiciel](http://fr.wikipedia.org/wiki/Logiciel), [système d’information](http://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27information)).  On peut attacher un diagramme d’activités à n’importe quel élément de modélisation afin de visualiser, spécifier, construire ou documenter le comportement de cet élément.

Dans la phase de conception, les diagrammes d’activités sont particulièrement adaptés à la description des cas d’utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d’utilisation. De plus, leur représentation sous forme d’organigrammes les rend facilement intelligibles.

Les diagrammes d’activités permettent de spécifier des traitements a priori séquentiels et offrent une vision très proche de celle des langages de programmation impératifs comme C++ ou Java. Ainsi, ils peuvent être utiles dans la phase de réalisation, car ils permettent une description si précise des opérations qu’elle autorise la génération automatique du code.

Dans ce qui suit, nous présentons les diagrammes d’activités pour les cas d’utilisation de notre système.

## 5.1 Diagramme d’activité «Authentification»



Figure 20. *Diagramme d’activité «Authentification»*

## 5.2 Diagramme d’activité « Consultation de solde »



Figure 21. *Diagramme d’activité « Consultation de solde »*

## 5.3 Diagramme d’activité « Demande carte »



Figure 22. *Diagramme d’activité « Demande carte »*

## 5.3 Diagramme d’activité « Demande chéquier »



Figure 23. *Diagramme d’activité « Demande chéquier »*

# *6.* Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d’une telle modélisation.

Contrairement au diagramme de cas d’utilisation qui montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d’utilisation. Sous forme de schéma, il est utilisé pour présenter les [classes](http://fr.wikipedia.org/wiki/Classe_(informatique)) et les [interfaces](http://fr.wikipedia.org/wiki/Interface_(informatique)) des systèmes ainsi que les différentes relations entre celles-ci. Ce [diagramme](http://fr.wikipedia.org/wiki/Diagramme) fait partie de la partie [statique](http://fr.wikipedia.org/wiki/Statique) d’[UML](http://fr.wikipedia.org/wiki/Unified_modeling_language) car il fait abstraction des aspects temporels et [dynamiques](http://fr.wikipedia.org/wiki/Dynamique).

Le diagramme de classes ci-dessous nous permet d’avoir une vue statique de l’application. Il nous montre les relations entre les différentes entités composant notre application et il nous mènera vers la solution finale. À partir de ce schéma, nous retrouvons les corps des différentes classes de notre application.

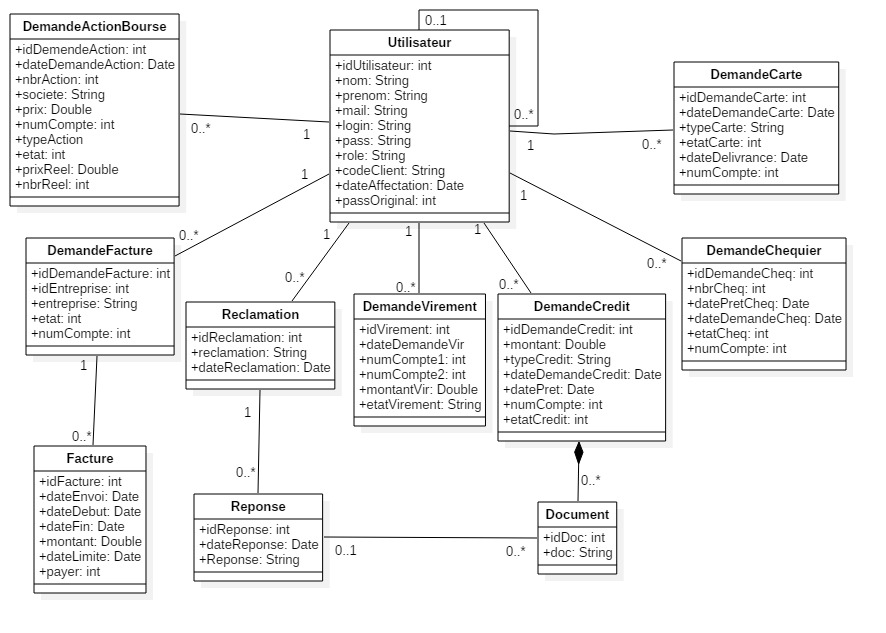


Figure 24. *Diagramme de classe*

# 7 Conclusion

Au cours de ce chapitre, nous avons décrit les besoins fonctionnels et techniques attendus de notre application en présentant les diagrammes des cas d’utilisation, des séquences et d'activités. Nous pouvons ainsi entamer la prochaine étape qui consiste à réaliser notre application.

Chapitre 3 : Réalisation

# Introduction

Dans ce chapitre, nous présentons l’architecture à partir de laquelle nous avons développé notre application, les différents outils utilisés ainsi que les composantes applicatives réalisées.

# 1. Environnements logiciels et technologiques

**Framework SpringMVC 3**

C’est un conteneur léger. Le terme « conteneur », découle du fait que Spring permet d’instancier les objets, de gérer leur cycles de vie ainsi que les dépendances entre eux, et cela grâce au principe de l’inversion de contrôle IOC (appelé également injection de dépendance). Le terme « léger » signifie que Spring est caractérisé par une infrastructure qui est similaire à un serveur d’application JEE, mais qui n’est pas liée, effectivement, à une spécification lourde comme pour le cas de JEE.

**Glassfish 4.1**

GlassFish est un serveur d’application compatible avec la plate-forme Java EE permettant de développer et de déployer des applications Java EE et des services Web Java.

**Hibernate 4.3.1**

C’est un Framework open source gérant la persistance des objets en base de données relationnelle.

Pour utiliser Hibernate dans un projet Java, nous pouvons utiliser, au choix, des fichiers XML ou des annotations Java directement intégrées dans le code. A la différence du mapping par XML, tous les attributs sont mappés par défaut dans le mapping par annotations.

Hibernate faisant le lien entre une application Java et une base de données, l’architecture d’un projet Hibernate est la suivante :

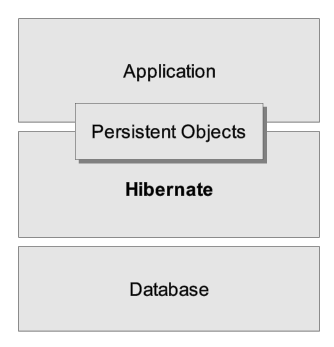


Figure 25 : Architecture d’Hibernate

**JSP**

Le JavaServer Pages ou JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web.

**JavaScript**

Le JavaScript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du [serveur](http://www.commentcamarche.net/contents/222-environnement-client-serveur) web.

**JQuery**

JQuery est un framework Javascript qui permet de faciliter des fonctionnalités communes de Javascript.

L’utilisateur de cette bibliothèque permet de gagner du temps de développement lors de l’interaction sur le code HTML d’une page web, l’AJAX ou la gestion des événements.

**Ajax**

AJAX (Asynchronous JavaScript and XML) n'est pas une technologie en soi, mais un terme désignant une nouvelle approche utilisant un ensemble de technologies existantes, dont : HTML ou XHTML, les feuilles de styles CSS, JavaScript, le modèle objet de document (DOM), XML, XSLT, et l'objet XMLHttpRequest. Lorsque ces technologies sont combinées dans le modèle AJAX, les applications Web sont capables de réaliser des mises à jour rapides de l'interface utilisateur sans devoir recharger la page entière du navigateur. Les applications fonctionnent plus rapidement et sont plus réactives aux actions de l'utilisateur.

**Bootstrap**

Bootstrap est un framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation… Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées.

**Service web**

Un service Web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.

**Architecture d’un Web Service**

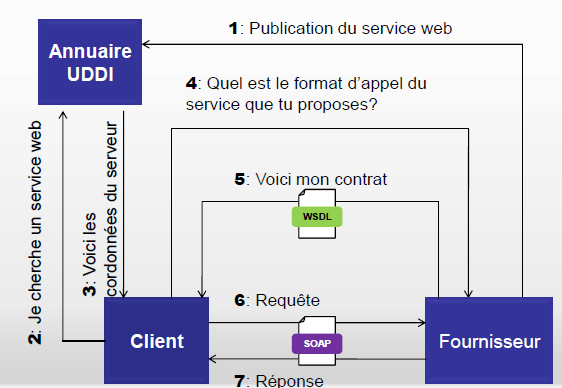


Figure 26 : Architecture d’un Web Service

Comme indiqué dans la figure ci-dessus, afin de mettre en œuvre un service web trois composants sont nécessaires :

* Un langage pour décrire le service web: **WSDL**.
* Un protocole de communication pour écrire les messages échangés entre le consommateur et le fournisseur : **SOAP**.

Dans notre projet, le consommateur est notre application E-Banking et le fournisseur est le système d’information de la banque.

* Un protocole de transport afin de faire circuler les informations sur Internet, dans notre cas **HTTP**.

**Responsive webdesign**

C’est une approche de conception web qui vise à l’élaboration de sites offrant une expérience de lecture et de navigation optimales pour l’utilisateur quelle que soit sa gamme d’appareil (téléphones mobiles, tablettes, liseuses, moniteurs d’ordinateur de bureau).

## Système de gestion de base de données (MYSQL)

MySQL est un Système de Gestion de Bases de Données Relationnelles (SGBDR) Open Source qui permet de gérer des bases de données. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels. Développé dans un souci de performances élevées en lecture, il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est [multi-thread](http://fr.wikipedia.org/wiki/Processus_l%C3%A9ger) et multi-utilisateur et utilise pour cela le langage SQL.

**Enterprise Architect 11**

C’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 depuis la gestion des exigences, en passant par les phases de conception, la construction, les tests et la maintenance.

**NetBeans IDE 8.1**

Une plate-forme de développement d’application s’appuyant sur le langage Java, et dont les spécifications sont gérées par la société SUN.

# 2. Architecture

Avant de se lancer dans la conception et le développement, nous allons préparer notre architecture. Dans cette partie, nous nous intéressons à l’architecture opérationnelle et logicielle de notre application.

## 2.1Architecture opérationnelle

La solution proposée pour la mise en place de la plateforme E-Banking se base sur l’utilisation d’une architecture distribuée en trois tiers. Cette solution consiste, donc, à faire communiquer un poste client simple avec le serveur qui, à son tour, va accéder à la base de données. Dans ce but, l’architecture trois tiers applique les principes suivants :

* Gestion des données de façon centralisée par un système de gestion de bases de données.
* Couche de présentation prise en charge par le poste client.
* Logique applicative prise en charge par un serveur intermédiaire.

La figure ci-dessous, met en évidence les interactions entres les couches dans le cadre d’une architecture trois tiers.

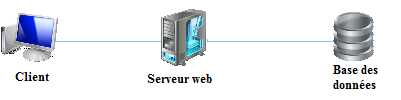


Figure 27 : Architecture trois tiers

## 3. Spé**cification logicielle du système**

Notre architecture trois tiers est mise en œuvre suivant le style architectural MVC (Modèle Vue Contrôleur).

Ce modèle d'architecture impose la séparation entre les données, les traitements et la présentation, ce qui donne trois parties fondamentales dans l'application :

Le modèle, la vue et le contrôleur

**Le Modèle** :

Représente le comportement de l'application : traitements des données, interactions avec la base de données, etc... Il décrit les données manipulées par l'application et définit les méthodes d'accès.

**La Vue** :

La vue correspond à l’interface avec laquelle l’utilisateur interagit, elle contient les composants graphiques. Sa première tâche est de présenter les résultats renvoyés par le modèle qui lui sont passés par le contrôleur. Sa seconde tâche est de recevoir toutes les actions de l’utilisateur (clic de souris, sélection d’une entrée, boutons, soumission de formulaire…). Ces différents événements sont envoyés au contrôleur. La vue n’effectue aucun traitement, elle se contente d’afficher les résultats des traitements effectués par le modèle.

**Le Contrôleur** :

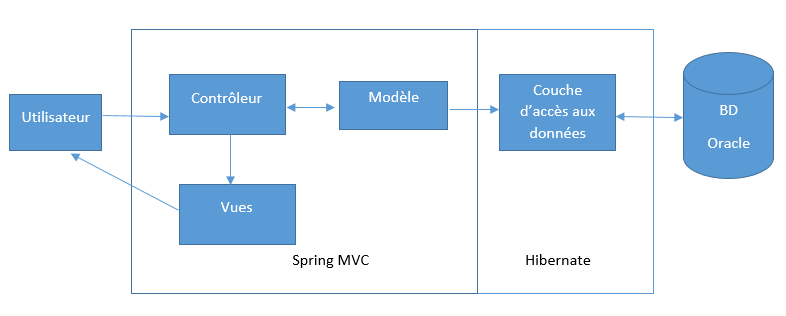
Il prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l’utilisateur et enclenche les actions à effectuer. 

Figure 28 : Architecture MVC

Dans le cadre de ce projet, notre application doit s’interfacer avec l’application de la banque, pour cela nous allons invoquer des Web Services, ceci a une implication directe sur le schéma de la figure 18. En effet, nous avons jugé utile de faire apparaître les Web Services invoqués à partir des services communs comme le montre la figure suivante.

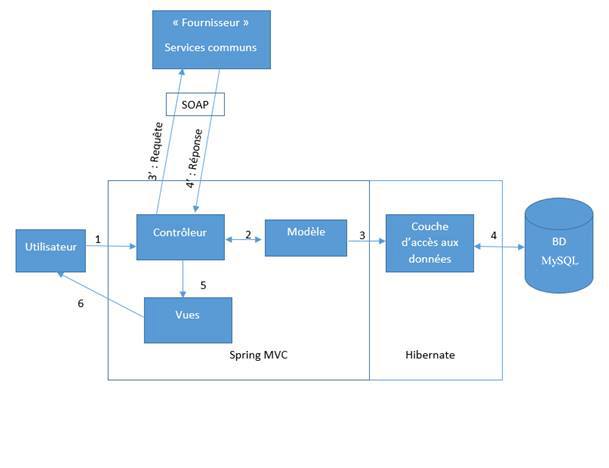


Figure 29 : Architecture du système

## Interface Graphique

Dans cette partie, nous allons présenter les différentes interfaces réalisées.

Ci-dessous, des imprimes écrans relatifs aux interfaces du premier sprint.

* + 1. **Ajouter un client**

La figure suivante représente l’interface d’ajout d’un client E-Banking. Seul l’administrateur à l’accès à cette interface.

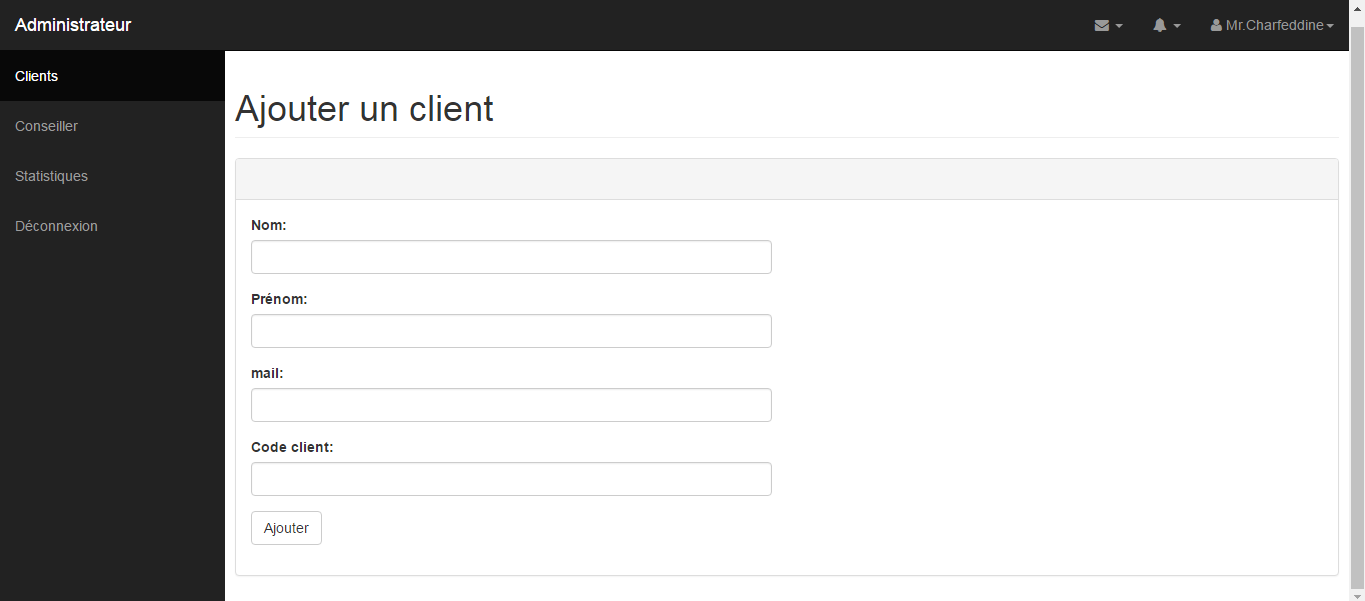


Figure  : Interface graphique « Ajouter un client »

* + 1. **Consulter les comptes**

Comme le montre la figure ci-dessous, nous trouvons dans l’interface de consultation une liste détaillée des comptes du client connecté.

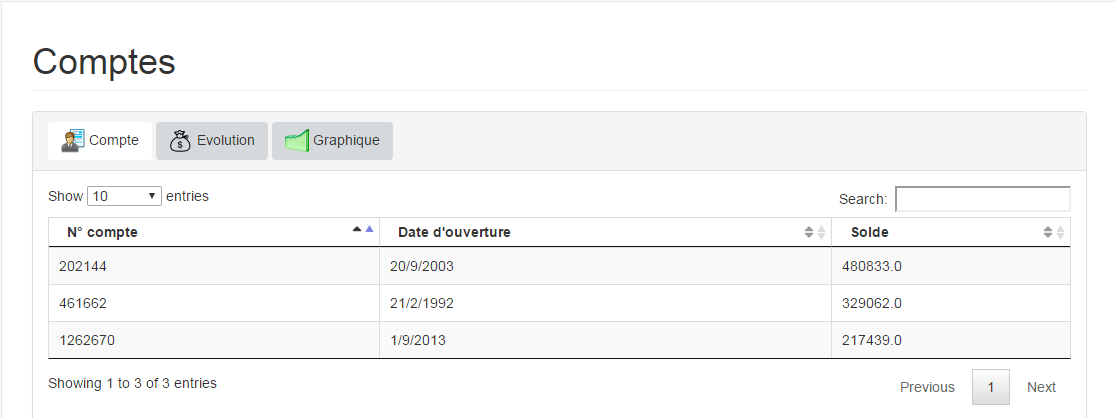


Figure  : Interface graphique « Consulter les comptes »

* + 1. **Consulter les mouvements d’un compte**

L’interface illustrée dans la figure suivante permet aux clients de la banque de consulter les informations de leurs opérations bancaires effectuées pendant une période sur un compte.