MINISTERE DE L’ENSEIGNEMENT SUPERIEUR ET DE LA

** RECHERCHE SCIENTIFIQUE**

****

**UNIVERSITE TUNIS EL MANAR**

**INSTITUT SUPERIEUR D’INFORMATIQUE**

**RAPPORT DE FIN D’ÉTUDES**

Présenté en vue de l’obtention du

Diplôme National d’Ingénieur en Sciences Appliquées et Technologies

**Spécialité : Génie Logiciel et Systèmes d’Information**

Par

**Tasnime OMRANI**

**Développement d’un E-Banking en JEE**

**Encadrant professionnel Monsieur Charfeddine SOUFIANE**

**Encadrant académique Monsieur Jamil DIMASSI**

**Réalisé au sein de la Banque et Finance Internationale**



Année Universitaire 2014/2015

**Encadrant Entreprise**

**Signature et cachet :**

**Encadrant ISI**

**Signature :**

**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 Saber et mes sœurs Asma et Nessrine pour leurs encouragements incessants.

À tous mes amis et spécialement Manel 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.

Je remercie également l’ensemble du personnel de la BFI en particulier Monsieur Ridha

RAZOUANE pour son chaleureux accueil.

J’exprime aussi ma gratitude et mes plus vifs remerciements à Monsieur Jamil DIMASSI, mon encadrant de l’Institut Supérieur d’Informatique 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.

Avec beaucoup d'égard, je ne manquerai pas d'exprimer ma grande reconnaissance à l’encadrante linguistique Madame Sonia ZAOUALI qui a su m’orienter dans un apprentissage continu, progressif et cohérent.

Table des matières

[Introduction générale 11](#_Toc421467256)

[**Chapitre 1. Cadre général du projet 12**](#_Toc421467257)

[Introduction 12](#_Toc421467258)

[1. Présentation de l’organisme d’accueil 12](#_Toc421467259)

[2. Présentation du sujet 12](#_Toc421467260)

[2.1. Présentation du projet 12](#_Toc421467261)

[2.2. Problématique 12](#_Toc421467262)

[2.3. Ébauche de la solution 13](#_Toc421467263)

[3. État de l’art 13](#_Toc421467264)

[3.1. Les solutions E-Banking existantes sur le marché 13](#_Toc421467265)

[3.2. Comparaison 15](#_Toc421467266)

[3.3. Critiques 16](#_Toc421467267)

[3.4. Solution BFI-E-Banking 17](#_Toc421467268)

[4. Méthodologie de gestion de projet 18](#_Toc421467269)

[4.1. Choix de la méthodologie 18](#_Toc421467270)

[4.2. Présentation de SCRUM 18](#_Toc421467271)

[4.2.1. Cycle de vie de la méthode SCRUM 19](#_Toc421467272)

[4.2.2. Rôles 20](#_Toc421467273)

[Conclusion 20](#_Toc421467274)

[**Chapitre 2. Analyse et spécification des besoins 21**](#_Toc421467275)

[Introduction 21](#_Toc421467276)

[1. Spécification des besoins 21](#_Toc421467277)

[1.1. Besoins fonctionnels 21](#_Toc421467278)

[1.2. Besoins non fonctionnels 22](#_Toc421467279)

[2. Identification des acteurs 23](#_Toc421467280)

[3. Diagramme des cas d’utilisation global 23](#_Toc421467281)

[4. Diagrammes des cas d’utilisation détaillés 25](#_Toc421467282)

[5. Pilotage de projet avec SCRUM 31](#_Toc421467283)

[5.1. Equipe et rôles 31](#_Toc421467284)

[5.2. Backlog du produit 32](#_Toc421467285)

[5.3. Planification des sprints 36](#_Toc421467286)

[6. Environnements logiciels et technologiques 36](#_Toc421467287)

[7. Architecture 39](#_Toc421467288)

[7.1. Architecture opérationnelle 39](#_Toc421467289)

[7.2. Spécification logicielle du système 40](#_Toc421467290)

[Conclusion 42](#_Toc421467291)

[**Chapitre 3. Sprint 1 43**](#_Toc421467292)

[Introduction 43](#_Toc421467293)

[1. Backlog du sprint 43](#_Toc421467294)

[2. Analyse 43](#_Toc421467295)

[2.1. Descriptions 43](#_Toc421467296)

[2.2. Maquette 44](#_Toc421467297)

[3. Conception 45](#_Toc421467298)

[3.1. Diagrammes des séquences détaillés 45](#_Toc421467299)

[3.2. Diagramme de classes 49](#_Toc421467300)

[4. Réalisation 50](#_Toc421467301)

[4.1. Codage 50](#_Toc421467302)

[4.2. Interface Graphique 51](#_Toc421467303)

[Conclusion 54](#_Toc421467304)

[**Chapitre 4. Sprint 2 55**](#_Toc421467305)

[Introduction 55](#_Toc421467306)

[1. Backlog du sprint 55](#_Toc421467307)

[2. Analyse 55](#_Toc421467308)

[2.1. Descriptions 55](#_Toc421467309)

[2.2. Maquette 57](#_Toc421467310)

[3. Conception 57](#_Toc421467311)

[3.1. Diagrammes des séquences détaillés 57](#_Toc421467312)

[3.2. Diagramme de classes 61](#_Toc421467313)

[4. Réalisation 61](#_Toc421467314)

[4.1. Codage 61](#_Toc421467315)

[4.2. Interface Graphique 63](#_Toc421467316)

[Conclusion 68](#_Toc421467317)

[**Chapitre 5. Sprint 3 69**](#_Toc421467318)

[Introduction 69](#_Toc421467319)

[1. Backlog du sprint 69](#_Toc421467320)

[2. Analyse 69](#_Toc421467321)

[2.1. Descriptions 69](#_Toc421467322)

[2.2. Maquette 70](#_Toc421467323)

[3. Conception 70](#_Toc421467324)

[3.1. Diagrammes de séquences détaillés 70](#_Toc421467325)

[3.2. Diagramme de classes 72](#_Toc421467326)

[4. Réalisation 72](#_Toc421467327)

[Conclusion 76](#_Toc421467328)

[**Chapitre 6. Sprint 4 77**](#_Toc421467329)

[Introduction 77](#_Toc421467330)

[1. Backlog du sprint 77](#_Toc421467331)

[2. Analyse 77](#_Toc421467332)

[2.1. Descriptions 77](#_Toc421467333)

[2.2. Maquettes 78](#_Toc421467334)

[3. Conception 79](#_Toc421467335)

[3.1. Diagrammes de séquences détaillés 79](#_Toc421467336)

[3.2. Diagramme de classes 82](#_Toc421467337)

[4. Réalisation 83](#_Toc421467338)

[5. Diagramme de Gantt 88](#_Toc421467339)

[Conclusion 89](#_Toc421467340)

[Conclusion générale 90](#_Toc421467341)

[Bibliographique 91](#_Toc421467342)

[Annexe 93](#_Toc421467343)

Table des figures

[Figure 1: Ébauche de la solution 13](#_Toc421392680)

[Figure 2: Fonctionnalités principales d'E-Banking 18](#_Toc421392681)

[Figure 3: Cycle de vie SCRUM 19](#_Toc421392682)

[Figure 4 : Diagramme de cas d’utilisation global 24](#_Toc421392683)

[Figure 5 : Diagramme de cas d’utilisation détaillé «Gérer les opérations bancaires » 25](#_Toc421392684)

[Figure 6 : Diagramme de cas d’utilisation détaillé «Gérer les cartes » 26](#_Toc421392685)

[Figure 7 : Diagramme de cas d’utilisation détaillé «Gérer les crédits» 27](#_Toc421392686)

[Figure 8 : Diagramme de cas d’utilisation détaillé «Communiquer» 28](#_Toc421392687)

[Figure 9 : Diagramme de cas d’utilisation détaillé «Tableau de bord» 28](#_Toc421392688)

[Figure 10 : Diagramme de cas d’utilisation détaillé «Recevoir des documents» 29](#_Toc421392689)

[Figure 11 : Diagramme de cas d’utilisation détaillé «Suivre le Taux de change» 29](#_Toc421392690)

[Figure 12 : Diagramme de cas d’utilisation détaillé «E-Facture» 30](#_Toc421392691)

[Figure 13 : Diagramme de cas d’utilisation détaillé «E-Bourse» 31](#_Toc421392692)

[Figure 14 : Rôles sous SCRUM 32](#_Toc421392693)

[Figure 15 : Architecture d’Hibernate 37](#_Toc421392694)

[Figure 16 : Architecture d’un Web Service 38](#_Toc421392695)

[Figure 17 : Architecture trois tiers 40](#_Toc421392696)

[Figure 18 : Architecture MVC 41](#_Toc421392697)

[Figure 19 : Architecture du système 41](#_Toc421392698)

[Figure 20 : Maquette de la User Story « Consulter les mouvements d’un compte » 45](#_Toc421392699)

[Figure 21 : Diagramme de séquence détaillé « Ajouter un client » 46](#_Toc421392700)

[Figure 22 : Diagramme de séquence détaillé « Consulter les mouvements d’un compte » 47](#_Toc421392701)

[Figure 23 : Diagramme de séquence détaillé « Répondre aux demandes de virement » 48](#_Toc421392702)

[Figure 24 : Diagramme de séquence détaillé « Recevoir des relevés de comptes» 49](#_Toc421392703)

[Figure 25 : Diagramme de classes du Sprint 1 50](#_Toc421392704)

[Figure 26 : Interface graphique « Ajouter un client » 51](#_Toc421392705)

[Figure 27 : Interface graphique « Consulter les comptes » 52](#_Toc421392706)

[Figure 28 : Interface graphique « Consulter les mouvements d’un compte » 52](#_Toc421392707)

[Figure 29 : Interface graphique « Evolution de solde » 53](#_Toc421392708)

[Figure 30 : Interface graphique « Demander un virement et suivre les demandes » 53](#_Toc421392709)

[Figure 31 : Interface graphique « Liste des relevés de comptes » 54](#_Toc421392710)

[Figure 32 : Interface graphique « Extrait de compte » 54](#_Toc421392711)

[Figure 33 : Maquette de la User Story « Répondre aux demandes de cartes bancaires » 57](#_Toc421392712)

[Figure 34 : Diagramme de séquence détaillé « Statistiques des opérations bancaires » 58](#_Toc421392713)

[Figure 35 : Diagramme de séquence détaillé « Répondre aux demandes de cartes bancaires » 59](#_Toc421392714)

[Figure 36 : Diagramme de séquence détaillé « Demander un crédit» 60](#_Toc421392715)

[Figure 37 : Diagramme de classes du Sprint 2 61](#_Toc421392716)

[Figure 38 : Interface graphique « S’authentifier » 63](#_Toc421392717)

[Figure 39 : Interface graphique « Statistiques des opérations bancaires » 64](#_Toc421392718)

[Figure 40 : Interface graphique « Changer un conseiller » 64](#_Toc421392719)

[Figure 41 : Interface graphique « Affecter un client » 65](#_Toc421392720)

[Figure 42 : Interface graphique « Demander et suivre les cartes bancaires » 66](#_Toc421392721)

[Figure 43 : Interface graphique « Répondre à une demande d’une carte bancaire » 66](#_Toc421392722)

[Figure 44 : Interface graphique « Simuler un crédit » 67](#_Toc421392723)

[Figure 45 : Interface graphique « Demander un crédit » 68](#_Toc421392724)

[Figure 46 : Maquette de la User Story « Consulter les réponses aux réclamations» 70](#_Toc421392725)

[Figure 47 : Diagramme de séquence détaillé « Répondre à une réclamation » 71](#_Toc421392726)

[Figure 48 : Diagramme de classes du Sprint 3 72](#_Toc421392727)

[Figure 49 : Interface graphique « Déposer une réclamation » 73](#_Toc421392728)

[Figure 50 : Interface graphique « Liste des réclamations » 73](#_Toc421392729)

[Figure 51 : Interface graphique « Consulter les réponses à la réclamation » 74](#_Toc421392730)

[Figure 52 : Interface graphique « Recevoir les notifications du conseiller » 74](#_Toc421392731)

[Figure 53 : Interface graphique « Suivre les conseillers » 75](#_Toc421392732)

[Figure 54 : Maquette de la User Story « Recevoir des relevés de comptes » 79](#_Toc421392733)

[Figure 55 : Modèle d’une facture 79](#_Toc421392734)

[Figure 56 : Diagramme de séquence détaillé « Payer une facture » 80](#_Toc421392735)

[Figure 57 : Diagramme de séquence détaillé « Demander l’achat d’actions » 81](#_Toc421392736)

[Figure 58 : Diagramme de séquence détaillé « Consulter le portefeuille » 82](#_Toc421392737)

[Figure 59 : Diagramme de classes du Sprint 4 83](#_Toc421392738)

[Figure 60 : Interface graphique « Liste des factures » 84](#_Toc421392739)

[Figure 61 : Interface graphique « Demander une facture » 84](#_Toc421392740)

[Figure 62 : Interface graphique « Payer une facture » 85](#_Toc421392741)

[Figure 63 : Interface graphique « Consulter l’actualité de la bourse » 85](#_Toc421392742)

[Figure 64 : Interface graphique « Demander l’achat d’actions » 86](#_Toc421392743)

[Figure 65 : Interface graphique « Liste des demandes des actions de bourse » 86](#_Toc421392744)

[Figure 66 : Interface graphique « Répondre à la demande d’achat d’actions » 87](#_Toc421392745)

[Figure 67 : Interface graphique « Consulter le portefeuille » 87](#_Toc421392746)

[Figure 68 : Interface graphique « Consulter taux de change et convertir une devise » 88](#_Toc421392747)

[Figure 69 : Diagramme de Gantt 88](#_Toc421392748)

[Figure 70 : Diagramme de cas d’utilisation détaillé «Gérer les chéquiers» 93](#_Toc421392749)

[Figure 71 : Diagramme de cas d’utilisation détaillé «Établir des Statistiques» 94](#_Toc421392750)

[Figure 72 : Diagramme de cas d’utilisation détaillé «Personnaliser l’interface» 94](#_Toc421392751)

[Figure 73 : Diagramme de cas d’utilisation détaillé «Gérer les nouveautés» 95](#_Toc421392752)

[Figure 74 : Diagramme de séquence détaillé « Consulter les comptes » 95](#_Toc421392753)

[Figure 75 : Diagramme de séquence détaillé « Demander un virement » 96](#_Toc421392754)

[Figure 76 : Diagramme de séquence détaillé « Suivre les demandes de virement » 97](#_Toc421392755)

[Figure 77 : Interface graphique « Statistiques globales » 97](#_Toc421392756)

[Figure 78 : Interface graphique « Répondre à la demande de virement » 98](#_Toc421392757)

[Figure 79: Recevoir les notifications des clients 98](#_Toc421392758)

Liste des tableaux

[Tableau 1 : Comparaison des tops 5 des E-Banking 16](#_Toc421392759)

[Tableau 2 : Rôles SCRUM 20](#_Toc421392760)

[Tableau 3 : Backlog du produit 33](#_Toc421392761)

[Tableau 4 : Planning des sprints 36](#_Toc421392762)

[Tableau 5 : Backlog du Sprint 1 43](#_Toc421392763)

[Tableau 6 : Description de la User Story « Consulter les mouvements d’un compte » 43](#_Toc421392764)

[Tableau 7 : Description de la User Story « Répondre aux demandes de virement » 44](#_Toc421392765)

[Tableau 8 : Description de la User Story «Recevoir des relevés de comptes» 44](#_Toc421392766)

[Tableau 9 : Backlog du Sprint 2 55](#_Toc421392767)

[Tableau 10 : Description de la User Story « Statistiques des opérations bancaires » 56](#_Toc421392768)

[Tableau 11 : Description de la User Story «Changer un conseiller » 56](#_Toc421392769)

[Tableau 12 : Description de la User Story «Demander une carte » 56](#_Toc421392770)

[Tableau 13 : Description de la User Story «Répondre aux demandes de cartes bancaires » 56](#_Toc421392771)

[Tableau 14 : Backlog du Sprint 3 69](#_Toc421392772)

[Tableau 15 : Description de la User Story «Déposer une réclamation » 69](#_Toc421392773)

[Tableau 16 : Description de la User Story «Consulter les réponses aux réclamations » 70](#_Toc421392774)

[Tableau 17 : Backlog du Sprint 4 77](#_Toc421392775)

[Tableau 18 : Description de la User Story « Demander une facture » 77](#_Toc421392776)

[Tableau 19 : Description de la User Story «Payer une facture» 78](#_Toc421392777)

[Tableau 20 : Description de la User Story « Acheter des actions » 78](#_Toc421392778)

# Introduction générale

Depuis des années, la société est en perpétuelle évolution sur tous les plans. Ainsi que les besoins de l’être humain. Et, en raison de sa mobilité permanente et le principe de gain de temps, des moyens d’épanouissement sont alors mis en œuvre afin de garantir une meilleure gestion des ressources et maintenir la relation avec le client comme but majeur de l’entreprise.

Pour ces raisons, les entreprises d’aujourd’hui mettent en valeur l’effort informatique en s’invertissant dans le domaine des recherches technologiques d’une manière de plus en plus remarquable ; ce qui nous mène à l’utilisation de solutions informatisées au sein de la majorité des entreprises et notamment le E-Banking pour les banques.

En effet, l’Electronic-Banking ou encore Internet Banking est apparu depuis le milieu des années 90 en France et désigne l’accès à des services offerts par une banque donnée via une interface interactive. De nos jours, il est devenu un service très demandé par, à la fois, l’entreprise et le client.

Dans ce cadre, nous étions amenés à réaliser un projet qui nous permettra de modéliser le concept d’E-Banking pour la société Banque et Finance Internationale (BFI) notre organisme d’accueil.

La BFI est une société d’édition et d’intégration de solutions logicielles destinées aux banques et institutions financières.

Nous présenterons notre projet au fil des six chapitres suivants.

Le premier chapitre, Cadre général, sera consacré à la description de la BFI et du sujet mais aussi à une comparaison de quelques méthodologies de conception.

Le deuxième, Analyse et spécification des besoins, comportera la spécification des besoins fonctionnels et opérationnels de notre future application, le planning de travail, l’environnement de développement et l’architecture de notre projet.

Les quatre chapitres suivants, seront concentrés sur les sprints de notre projet via l’analyse, la conception et la réalisation de chacun.

Enfin, nous conclurons et proposerons nos perspectives.

# Chapitre 1. Cadre général du projet

# ****Introduction****

**Dans ce chapitre, nous commençons par présenter l’organisme d’accueil de notre projet. Nous analysons ensuite le sujet en terme de fonctionnalités, devant être offertes et de contraintes à respecter. Puis nous présentons un état de l’art pour donner un cadre de référence à notre projet. À la fin de ce chapitre, nous exposons la méthodologie qui a conduit à la réalisation de notre projet.**

# Présentation de l’organisme d’accueil

La Banque et Finance International (**BFI)**, créée en 1994, est une société Tunisienne d’édition et d’intégration de solutions logicielles destinées aux banques et aux institutions financières.

La 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 tels que la gestion des clients, la gestion des risques, le contrôle des engagements, les crédits, la trésorerie, le marché des capitaux, les moyens de paiement, la comptabilité… [1]

# Présentation du sujet

Cette section a pour objectif de présenter notre projet, dégager la problématique afin d’exposer une ébauche de notre solution.

## Présentation du projet

Dans le cadre de l’expansion de ses services, la BFI se propose de développer un service de banque en ligne (E-Banking) qui permet aux clients des banques d’effectuer les opérations bancaires en ligne, les rendant ainsi plus rapides, plus disponibles et plus pratiques et ce afin d’éviter le déplacement des clients aux agences.

Pour les banques, cette solution permettra à terme la réduction du nombre ainsi que la taille de ses agences. En outre, ce type de service permettra une réduction des charges d’exploitation (papiers, imprimantes, etc.).

## Problématique

De nos jours, avec l’avancée technologique accrue dans un environnement très concurrentiel, rendre les services de la banque accessibles 24h/24 et 7j/7 n’est plus un luxe. Afin de répondre à cet objectif, plusieurs clients de la BFI sont demandeurs d’une solution d’E-Banking qui respecte les contraintes suivantes :

- Ce service doit être le plus générique possible pour permettre sa réutilisation par plusieurs banques.

- Ce service doit être fortement découplé des systèmes bancaires existants (BackOffice et Front office).

- Ce service doit être multimodal afin de permettre un accès aisé de n’importe quel type d’appareil (Ordinateur, Tablette, Smartphone, etc.)

N’ayant pas une solution d’E-Banking, la BFI a décidé de développer ce service répondant aux besoins des banques clientes ainsi qu’aux usagers.

Dans ce cadre, l’objectif de notre projet de PFE est de répondre aux besoins de la BFI, en développant une solution d’E-Banking.

## Ébauche de la solution

Vu les besoins préliminaires, nous nous proposons dans ce projet de développer une application permettant la couverture de l’ensemble des exigences de la BFI. A priori, et en étudiant ces besoins, l’application doit être basée sur une technologie web pour satisfaire la contrainte de multi-modalité. En plus cette solution doit être fortement basée sur les architectures SOA pour gagner en découplage. Notre solution se résume dans la figure suivante.

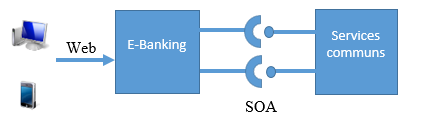


Figure : Ébauche de la solution

# État de l’art

Nous essayons dans ce qui suit d’étudier les différentes solutions existantes afin de tirer profit de chacune d’elle dans notre développement.

## Les solutions E-Banking existantes sur le marché

Plusieurs solutions d’E-Banking existent sur le marché international. D’après nos différentes lectures, les solutions d’E-Banking sont utilisées en général par trois types d’utilisateurs à savoir : les clients finaux, les opérateurs front-office des banques ainsi que les opérateurs back-office. Chacun de ces types bénéficie d’un pack de fonctionnalités lui permettant d’interagir avec les services de la banque.

Dans ce qui suit, nous allons présenter, les tops cinq de ces solutions. Pour chacune d’elle, nous nous intéressons essentiellement aux services offerts pour chaque type d’utilisateur ainsi que les technologies utilisées [2]

* + 1. **NexorOne** [3]

Fondée en 2005, cette solution est déployée sur plus de 320 banques dans le monde reparties à travers 40 pays.

Grâce à ce logiciel, l’utilisateur peut :

* consulter ses comptes,
* consulter ses cartes bancaires,
* recharger une carte bancaire,
* effectuer des virements interne et externe,
* communiquer avec le conseiller,
* suivre les mouvements bancaires,
* gérer son profil.

Ce logiciel offre à l’administrateur les fonctionnalités de :

* gestion des différents utilisateurs,
* consultation et création des comptes bancaires,
* gestion des transactions.

Les technologies utilisées afin de développer cette application sont :

* PHP basé sur le framework Zend comme language de développement,
* MySQL comme système de gestion des données.

**3.1.2. BankWare.net** [4]

C’est une solution fondée en 1985. Elle offre aux différents clients de la banque la possibilité d’accéder à leurs informations bancaires 24 heures par jour et 7 jours par semaine.

BankWare.net fournit les fonctionnalités suivantes dans le but de satisfaire les clients :

* consulter des comptes et des soldes,
* consulter des mouvements de compte,
* transférer de compte à compte,
* communiquer avec la banque à travers un système d’e-mail,
* demander un crédit,
* consulter carte bancaire.

Ce système possède un service SMS Banking qui envoie des messages aux clients pour les informer des soldes ou des mouvements de leurs comptes.

Les technologies utilisées afin de développer ce système sont :

* architecture : architecture 3 tiers client/serveur,
* langage de programmation : Microsoft C++/C#,
* base de~~s~~ données : SQL.
  + 1. **Canopus E-pay suite** [5]

C’est une solution de banque en ligne fondée en 1992.

Le logiciel donne la possibilité au client de :

* consulter ses comptes,
* communiquer avec la banque par le système de mail,
* effectuer des virements,
* gérer son profil,
* demander et recharger une carte bancaire,
* demander un crédit.
  + 1. **ARMnet** [6]

C’est une solution de banque en ligne Fondée en 2007.

A travers ce système, l’utilisateur peut :

* consulter ses comptes,
* communiquer avec la banque,
* effectuer des virements,
* gérer son profil,
* demander et recharger une carte bancaire,
* demander un crédit.

Afin de mieux satisfaire les clients, une version mobile est disponible.

* + 1. **CoBis** [7]

Cette solution est fondée en 2008, basée sur les technologies C#, Visual basic.net et JavaScript. Elle fournit aux clients les services suivants :

* consulter les comptes,
* consulter les mouvements de compte,
* virement,
* communiquer avec la banque,
* consulter une carte bancaire.

## Comparaison

Nous présentons dans le tableau suivant un comparatif entre les tops des solutions d’E-Banking citées précédemment.

Tableau  : Comparaison des tops 5 des E-Banking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E-Banking  Service | NexorOne | BankWore.net | Canopus E-pay suite | ARMnet | CoBis |
| Consulter les comptes |  |  |  |  |  |
| Consulter les mouvements |  |  |  |  |  |
| Virement |  |  |  |  |  |
| Communiquer avec la banque |  |  |  |  |  |
| Demander un crédit |  |  |  |  |  |
| Gérer les cartes |  |  |  |  |  |
| Technologies | PHP | C++/C# | Non mentionné | Non mentionné | PHP |
| SMS Banking |  |  |  |  |  |
| Version mobile |  |  |  |  |  |

## Critiques

Notre évaluation des logiciels se base sur, principalement, trois critères à savoir la fonctionnalité, la technologie et l’intégration. Nous entendons par intégration le fait de pouvoir mettre en relation différents logiciels.

**Côté technologie et intégration**

Les deux langages utilisés pour développer les solutions présentées sont PHP et C#.

Bien que PHP présente une certaine facilité de développement et de déploiement puisqu’il ne nécessite pas une infrastructure serveur spécifique à l’inverse de C# qui s’intègre dans le Framework .Net, ce langage (PHP) présente de très nombreux inconvénients à savoir :

* Une mauvaise structuration du code ; vu la multitude de composants utilisés.
* Une défaillance au niveau de la gestion de la sécurité, puisque celle-ci reste à la charge du développeur.

D’un autre coté C# (.Net) présente l’inconvénient majeur de la « dépendance » à l’environnement « Windows » qui n’est pas fréquemment utilisé dans le domaine bancaire.

Par ailleurs, vu la nature et le nombre d’applications déjà développées par BFI, le framework JEE s’impose naturellement pour notre projet. Ce framework présente les avantages suivants par rapport à ses concurrents :

* JEE a la capacité de maintenir ses fonctionnalités et ses performances en cas de montée en charge, grâce au serveur d’application.
* Il permet de garantir la sécurité.
* Grâce à son architecture, un projet avec JEE est facile à intégrer avec d’autres plateformes.

**Côté fonctionnalité**

Le tableau comparatif sus-indiqué montre la présence des fonctionnalités suivantes dans toutes les solutions existantes : la consultation des comptes, la consultation des mouvements, le virement et la gestion des cartes.

Par contre, nous pouvons constater l’absence de quelques fonctionnalités telles que « Demander un crédit » pour les solutions : Nexorne et CoBis. Et « Communiquer avec un conseiller » pour les solutions : Nexorne , BankWore et CoBis.

En ce qui concerne la version mobile, seules Canpus et ARMnet l’ont implémenté.

## Solution BFI-E-Banking

Lors des différentes réunions avec l’équipe du projet de la BFI, au cours desquelles nous avons présenté notre vision et avancement du projet, il a été décidé de développer une solution d’E-Banking générique, qui couvre toutes les fonctionnalités mentionnées dans le tableau comparatif en utilisant la technologie JEE ; la BFI entend par généricité le concept de minimiser le changement des Systèmes d’Information déjà existants dans les banques clientes, le E-Banking sera facilement adaptable pour toute nouvelle installation.

Par ailleurs, la BFI a exprimé le besoin d’ajouter des services concernant la gestion des portefeuilles de placement boursier, de paiement des factures tiers ainsi que l’approfondissement des fonctionnalités de collaboration entre les clients et la banque.

La figure suivante montre les fonctionnalités principales demandées par la BFI avant l’ajout des services susmentionnés.

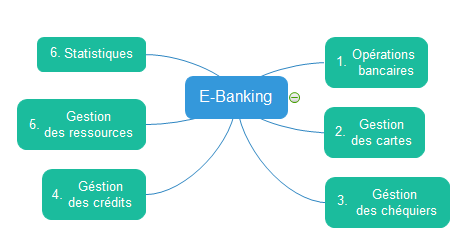


Figure : Fonctionnalités principales d'E-Banking

# Méthodologie de gestion de projet

Nous présentons dans cette partie, la méthodologie adoptée afin de développer notre application.

## Choix de la méthodologie

Plusieurs méthodes existent. Choisir entre elles dépend de la nature du projet.

Si le domaine du projet est maîtrisé, un cycle de vie en cascade suffit. En cas où, nous ne pouvons pas tout prévoir dès le début ou si les besoins sont incomplets comme dans notre cas, il faut utiliser les méthodes itératives et incrémentales telles que les méthodes agiles.

Une méthode Agile garantit une meilleure qualité de communication avec l’utilisateur, une meilleure visibilité du client sur l’avancement des travaux, un meilleur contrôle de qualité par le fait que les tests sont effectués en continu, ce qui permet de détecter rapidement les problèmes. Elle intègre aussi la notion de travail en équipe.

Parmi les méthodes Agiles, nous pouvons citer « SCRUM » qui sera utilisé dans la réalisation de notre projet.

## Présentation de SCRUM

Le principe de la méthode agile SCRUM est de concentrer l’équipe de développement sur un ensemble de fonctionnalités à réaliser de façon itératif, dans des itérations d’une durée de deux à quatre semaines, appelées des Sprints. Chaque Sprint doit aboutir à la livraison d’un produit partiel. [8]

### Cycle de vie de la méthode SCRUM

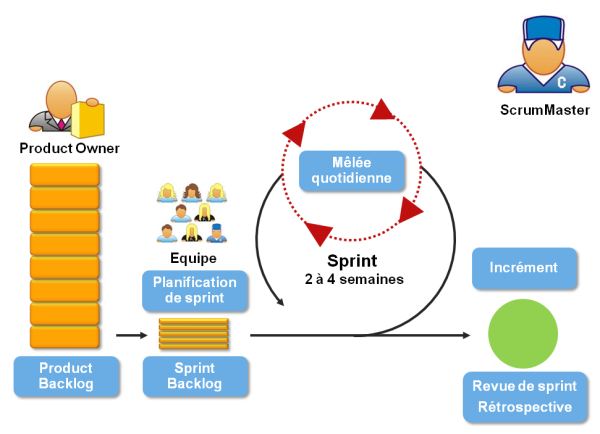


Figure : Cycle de vie SCRUM

Comme indiqué dans la figure ci-dessus, pour mettre en place la méthode SCRUM, il faut dégager dans un premier lieu le maximum de fonctionnalités à réaliser pour former le Backlog du produit. En second lieu, définir les priorités des fonctionnalités et choisir lesquelles seront réalisées dans chaque itération.

Par la suite, focaliser l’équipe de façon itérative sur l’ensemble des fonctionnalités à réaliser dans des itérations appelées Sprints.

Un sprint aboutit toujours à la livraison d’un produit partiel fonctionnel appelé incrément. Ainsi, vers la fin de chaque Sprint, une réunion aura lieu pour effectuer la revue de l’itération. L’objectif de cette réunion consiste à valider l’incrément qui été produit pendant l’itération.

### Rôles

SCRUM définit quatre rôles principaux. Ces rôles seront expliqués dans le tableau suivant.

Tableau  : Rôles SCRUM

|  |  |
| --- | --- |
| Product Owner  (Directeur du produit) | Il représente les utilisateurs (les clients), clarifie leurs besoins, détermine leurs priorités et prend les décisions importantes concernant l’orientation du projet. |
| Scrum Master (Meneur de l’équipe, facilitateur) | C’est un facilitateur et un coach plus qu’un superviseur. Il n’est pas responsable de la distribution des tâches. Il travaille de façon rapprochée avec le client et l’équipe. Il protège l’équipe de tous les éléments perturbateurs, ainsi que des problèmes non techniques. |
| Développement Team (L’équipe) | Il y n’a ni hiérarchie ni rôle prédéfini dans l’équipe. Cette équipe s’autogère. Les membres prennent les décisions ensemble et personne ne donne d’ordre à l’équipe sur sa façon de procéder. L’équipe s’adresse directement au client. |
| Stakeholders (Les intervenants) | Représentent les personnes qui souhaitent avoir une vue sur le projet sans réellement s’y investir, tels que les experts techniques ou les agents de direction qui souhaitent avoir une vue très éloignée de l’avancement du projet. |

# ****Conclusion****

**Dans ce chapitre, nous avons présenté l’organisme d’accueil et une brève description du projet à réaliser, en déterminant la problématique et en proposant la solution envisagée pour faire face à la situation actuelle. Nous avons aussi exposé la méthodologie adoptée pour le suivi de notre projet.**

**Le chapitre suivant s’intéressera à la modélisation des cas d’utilisation comme moyen de spécification du Backlog de produit et préparation du planning de travail.**

# Chapitre 2. Analyse et spécification des besoins

# Introduction

Dans ce chapitre, nous commençons par la capture des différents besoins pour les regrouper dans un diagramme des cas d’utilisation qui va être par la suite détaillé afin de produire le Backlog initial. Ensuite, nous identifions les rôles des utilisateurs et nous préparons notre première planification des sprints.

À la fin de ce chapitre, nous présentons l’environnement de développement de notre solution et l’architecture adoptée.

# Spécification 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.

## Besoins 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.

## Besoins 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.

# 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.

# Diagramme 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  : Diagramme de cas d’utilisation global

# Diagrammes des cas d’utilisation détaillés

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

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

* 1. **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  : Diagramme de cas d’utilisation détaillé «Gérer les cartes »

* 1. **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  : Diagramme de cas d’utilisation détaillé «Gérer les crédits»

* 1. **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  : Diagramme de cas d’utilisation détaillé «Communiquer»

* 1. **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  : Diagramme de cas d’utilisation détaillé «Tableau de bord»

* 1. **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  : Diagramme de cas d’utilisation détaillé «Recevoir des documents»

* 1. **Cas d’utilisation «Suivre le taux change»**

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



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

* 1. **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  : Diagramme de cas d’utilisation détaillé «E-Facture»

* 1. **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  : Diagramme de cas d’utilisation détaillé «E-Bourse»

La liste des autres diagrammes de cas d’utilisation détaillés est placée en Annexe.

# Pilotage de projet avec SCRUM

Cette partie est constituée de trois éléments principaux du SCRUM, qui sont l'équipe avec des rôles bien définis, le Backlog du produit et la planification.

## Equipe et rôles

SCRUM définit quatre rôles principaux :

* Le directeur de produit (Product Owner)
* Le facilitateur/animateur (Scrum Master)
* L’équipe
* Les intervenants

Dans le cadre de ce projet l’équipe se limite en un seul membre qui est moi-même. Par ailleurs, la liste des rôles affectés à ce projet est présentée dans la figure suivante :

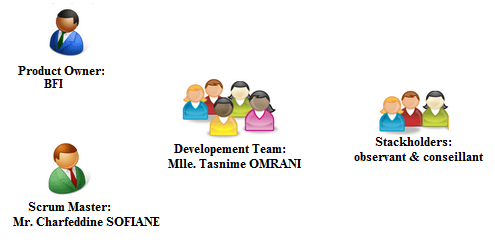


Figure  : Rôles sous SCRUM

## Backlog du produit

Un Backlog est une liste de fonctionnalités ou de tâches, jugées nécessaires et suffisantes pour la réalisation satisfaisante du projet. [9]

Les éléments du Backlog sont appelés User Stories (histoires). Chacune de ces Stories doit avoir un but métier et, par la suite, représente un cas d’utilisation détaillé.

Dans ce but, nous dressons le Backlog de notre application illustré dans le tableau ci-dessous en nous basant sur les diagrammes des cas d'utilisation détaillés représentés précédemment.

Dans ce tableau chaque histoire utilisateur est caractérisée par une importance attribuée par le Product Owner. Pour le traitement de ces histoires, nous choisissons de commencer avec les cas d’utilisation les plus importants ; ceux ayant la valeur la plus élevée.

En plus de l’importance, chaque histoire utilisateur possède une estimation qui est l’évaluation initiale de l’équipe sur la quantité de travail nécessaire pour implémenter cette exigence. Cette estimation est calculée en points d’histoire qui correspond aux jours hommes idéaux.

Ce Backlog servira à piloter l’équipe de développement et pourra évoluer tout au long de notre projet.

Tableau  : Backlog du produit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **User story** | **Description** | **Importance** | **Estimation** |
| **Sprint 1** | | | | |
| 1 | Ajouter un client | En tant qu’administrateur, je peux ajouter un client. | 80 | 19 |
| Consulter les comptes | En tant que client, je peux consulter mes comptes et ses détails. |
| Consulter les mouvements d’un compte | En tant que client, je peux choisir une période afin de consulter les différents mouvements de mon compte. |
| Evolution de solde | En tant que client, je peux consulter l’évolution des soldes de mes comptes. |
| Ajouter un conseiller | En tant qu’administrateur, je peux ajouter un conseiller clientèle. |
| Demander un virement | En tant que client, je peux ordonner un virement intra ou inter bancaire. |
| Suivre les demandes de virement | En tant que client, je peux consulter l’état de mes ordres de virement. |
| Répondre aux demandes de virement | En tant que conseiller, je peux accepter un ordre de virement. |
| Recevoir des relevés de comptes | En tant que client, je peux, mensuellement, consulter, télécharger et imprimer un relevé de compte. |
| **Sprint 2** | | | | |
| 2 | S’authentifier | En tant qu’utilisateur, je peux m’authentifier afin d’accéder à mon espace personnel. | 60 | 21 |
| Statistiques des opérations bancaires | En tant que client, je peux consulter les statistiques de mes opérations bancaires. |
| Statistiques globales | En tant qu’administrateur, je peux suivre les statistiques des mouvements bancaires effectués pendant les 3 dernières années. |
| Changer un conseiller | En tant qu’administrateur, je peux changer le conseiller d’un client. |
| Affecter un client | En tant qu’administrateur, je peux affecter des clients à un conseiller. |
| Demander une carte bancaire | En tant que client, je peux demander une carte bancaire. |
| Suivre les demandes de cartes | En tant que, client je peux consulter l’état de ma demande de carte bancaire. |
| Répondre aux demandes de cartes | En tant que conseiller, je peux répondre aux demandes de cartes bancaires provenant des clients. |
| Demander un chéquier | En tant que client, je peux demander un chéquier. |
| Suivre les demandes de chéquiers | En tant que client, je peux suivre l’état de ma demande d’un chéquier. |
| Répondre aux demandes de chéquiers | En tant que conseiller, client je peux répondre aux demandes de chéquiers. |
| Demander un crédit | En tant que client, je peux demander un crédit. |
| Ajouter des documents | En tant que client, je peux ajouter des documents à une demande de crédit. |
| Suivre la demande de crédit | En tant que client, je peux suivre l’état de ma demande de crédit. |
| Répondre aux demandes de crédits | En tant que conseiller, je peux refuser, accepter une demande de crédit ou demander des documents supplémentaires. |
| **Sprint 3** | | | | |
| 3 | Déposer une réclamation | En tant que client, je peux passer une réclamation. | 40 | 15 |
| Consulter les réponses à une réclamation | En tant que client, je peux consulter les réponses de mes réclamations. |
| Répondre à une réclamation | En tant que conseiller, je peux répondre aux réclamations de mes clients. |
| Demander un document | En tant que conseiller, je peux demander un document à un client. |
| Envoyer un document | En tant que client, je peux envoyer un document à mon conseiller. |
| Recevoir les notifications d’un conseiller | En tant que client, je peux recevoir les notifications des réponses de mon conseiller. |
| Suivre les conseillers | En tant qu’administrateur, je peux contrôler le déroulement des tâches des conseillers. |
| Recevoir les notifications des clients | En tant que conseiller, je peux recevoir les notifications des demandes de mes clients. |
| **Sprint 4** | | | | |
| 4 | Demander une facture | En tant que client, je peux demander la réception de mes factures. |  |  |
| Payer une facture | En tant que client, je peux payer mes factures. |
| Répondre aux demandes de factures | En tant que conseiller, je peux accepter ou refuser les demandes des factures de mes clients. |
| Consulter l’actualité de la bourse | En tant que client, je peux consulter l’actualité de la bourse. | 20 | 17 |
| Demander l’achat des actions | En tant que client, je peux demander l’achat des actions en bourse. |
| Demander la vente des actions | En tant que client, je peux demander la vente de mes actions en bourse. |
| Consulter le portefeuille | En tant que client, je peux consulter mon portefeuille. |
| Répondre aux demandes d’achat d’actions | En tant que conseiller, je peux répondre aux demandes d’achat d’actions de mes clients. |
| Répondre aux demandes de vente d’actions | En tant que conseiller, je peux répondre aux demandes de vente d’actions de mes clients. |
| Consulter les taux de change | En tant que client, je peux consulter l’appréciation des monnaies. |
| Convertir une devise | En tant que client, je peux consulter le résultat de la conversion d’une devise. |
| Changer le logo | En tant qu’administrateur, je peux changer le logo de l’E-Banking |
| Ajouter une nouveauté | En tant qu’administrateur, je peux ajouter une nouveauté. |
| Consulter les nouveautés | En tant que client, je peux consulter les nouveautés de ma banque. |

## Planification des sprints

La réunion de la planification des sprints est l’événement le plus important dans SCRUM. Le but de cette réunion est de préparer le planning de travail et d’identifier le Backlog des sprints.

La figure suivante résume notre planning de travail.

Tableau  : Planning des sprints

|  |  |  |
| --- | --- | --- |
|  | Date début | Date fin |
| Sprint 1 | 26/02 | 19/03 |
| Sprint 2 | 20/03 | 13/04 |
| Sprint 3 | 14/04 | 30/04 |
| Sprint 4 | 01/05 | 20/05 |

# Environnements logiciels et technologiques

**Framework SpringMVC 3** [10]

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** [10]

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 :

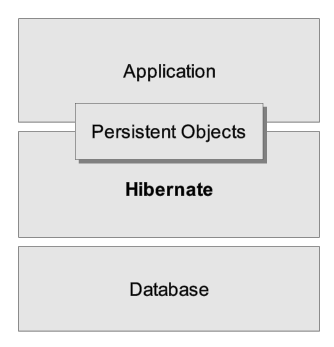


Figure  : Architecture d’Hibernate

**JSP** [11]

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** [12]

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** [13]

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** [14]

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** [15]

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** [16]

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**

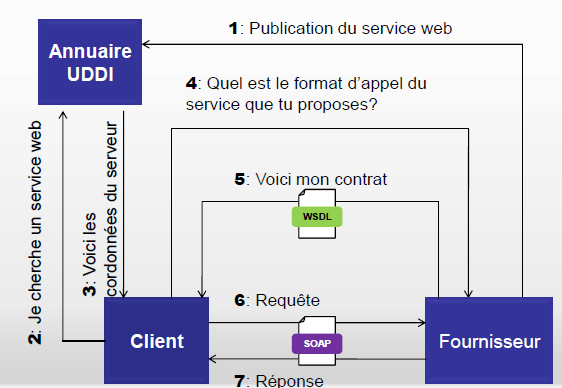


Figure 16 : Architecture d’un Web Service

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** [17]

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).

**Oracle 10g** [18]

Oracle Database est un système de gestion de base de données relationnel (SGBDR) qui depuis l’introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO).

Fourni par Oracle corporation, il a été développé par Larry Ellison, accompagné d’autres personnes telles que Bob Miner et Ed Oates.

**Enterprise Architect 11** [19]

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 8.0.2**

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

# 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.

## Architecture 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.

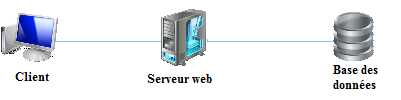


Figure  : Architecture trois tiers

## Spécification 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 [20]

**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** :

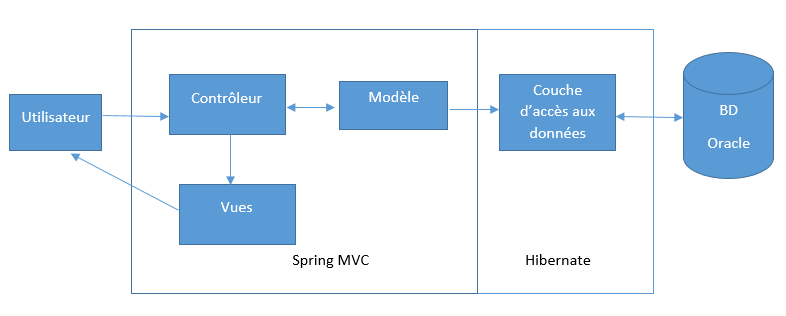
Il prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l’utilisateur et enclenche les actions à effectuer. 

Figure  : 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.

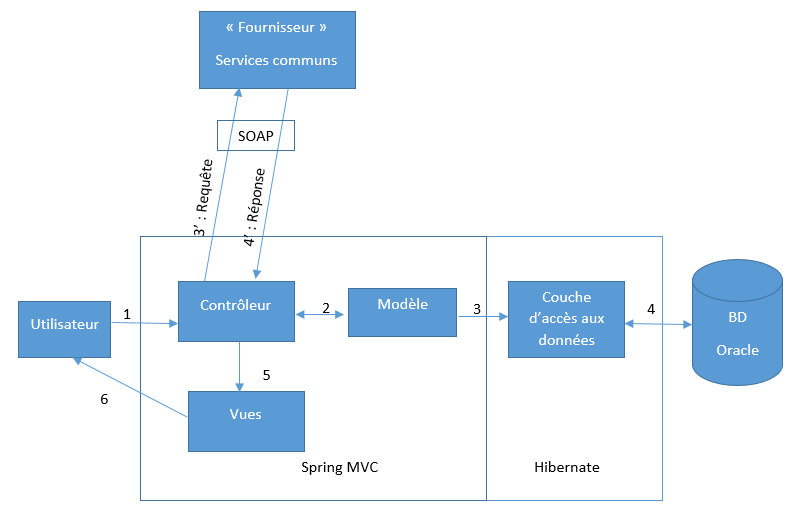


Figure  : Architecture du système

# Conclusion

Dans ce chapitre, nous avons capturé les besoins de notre application, préparer notre planning de travail et les rôles des utilisateurs et par la suite nous avons exposé les environnements logiciels et l’architecture de notre application.

Nous présenterons dans les chapitres qui suivent les sprints dans lesquels nous jouons le rôle de développeur et de concepteur.

# Chapitre 3. Sprint 1

# Introduction

Dans un sprint nous pouvons dégager trois activités principales qui sont l’analyse, la conception et la réalisation. Tout au long de ce sprint, nous allons respecter ces activités pour construire le plan de notre travail.

# Backlog du sprint

Le tableau ci-dessous définit les histoires de notre Backlog du produit qui seront incluses dans le Backlog du sprint.

Tableau  : Backlog du Sprint 1

|  |  |  |
| --- | --- | --- |
| **ID** | **Histoire** | **Estimation** |
| 001 | Ajouter un client | 1 |
| 002 | Ajouter conseiller | 1 |
| 003 | Consulter les comptes | 3 |
| 004 | Consulter les mouvements d’un compte | 4 |
| 005 | Evolution de solde | 2 |
| 006 | Demander un virement | 1 |
| 007 | Suivre les demandes de virement | 1 |
| 008 | Répondre aux demandes de virement | 2 |
| 009 | Recevoir des relevés de comptes | 4 |

# Analyse

La phase d’analyse permet de répondre à la question « Que fait le système ? ». Cette réponse sera représentée sous deux formes. La 1ère sera réservée aux descriptions des User Stories et la 2ème pour les maquettes qui seront détaillées plus tard.

## Descriptions

Une User Story est caractérisé par une description simple et compréhensible de sa fonctionnalité. Nous présentons ci-dessous les différentes descriptions des histoires utilisateurs du Sprint 1.

Tableau  : Description de la User Story « Consulter les mouvements d’un compte »

|  |  |
| --- | --- |
| #004 | Consulter les mouvements d’un compte |
| **En tant que** client  **Je veux** voir les mouvements de mon compte | |
| **Vu que** je suis arrivé à la page des mouvements  **Vu que** je choisis un compte et je saisis une date de début et une date de fin  **Quand** je clique sur le bouton « Consulter »  **Alors** la liste des mouvements de mon compte s’affiche | |

Tableau : Description de la User Story « Répondre aux demandes de virement »

|  |  |
| --- | --- |
| #008 | Répondre aux demandes de virement |
| **En tant que** conseiller  **Je veux** répondre à une demande de virement | |
| **Vu que** je suis sur la page des virements  **Quand** je clique sur un bouton « accepter »  **Alors** l’action de virement s’effectue et la page se réaffiche  **Quand** je clique sur un bouton « refuser »  **Alors** l’action s’applique et la page des virements se réaffiche | |

Tableau : Description de la User Story «Recevoir des relevés de comptes»

|  |  |
| --- | --- |
| #009 | Recevoir des relevés de comptes |
| **En tant que**  client  **Je veux** recevoir un relevé par compte et à la fin de chaque mois | |
| **Vu que** je suis sur le site  **Quand** je clique sur le menu « E-Document »  **Alors** la liste de mes relevés s’affiche  **Quand** je clique sur un relevé  **Alors** un fichier PDF, contient mes mouvements bancaires, s’affiche | |

## Maquette

Une présentation des besoins sous forme de spécification fonctionnelle et d’enchaînement de processus est difficile à comprendre par nos clients. C’est pour cette raison que nous avons réalisé des maquettes reflétant les besoins des clients qui seront avalisées et validées par ces derniers avant de commencer le développement.

Nous présentons ci-dessous une maquette du Sprint 1.

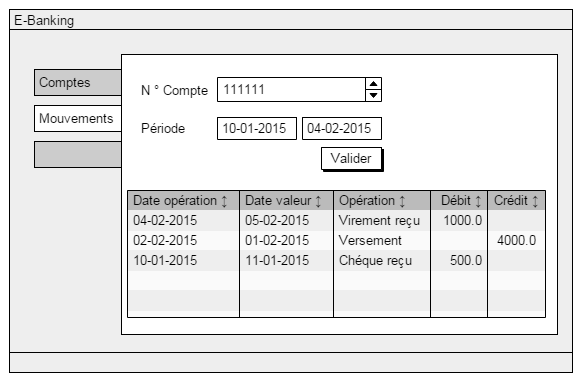
****

Figure  : Maquette de la User Story « Consulter les mouvements d’un compte »

# Conception

Nous arrivons maintenant à la partie conception où nous allons présenter les différents diagrammes de séquences détaillés ainsi que le diagramme de classes pour ce Sprint.

## 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 des histoires utilisateurs du Sprint 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  : Diagramme de séquence détaillé « Ajouter un client »

* + 1. **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  : Diagramme de séquence détaillé « Consulter les mouvements d’un compte »

* + 1. **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  : Diagramme de séquence détaillé « Répondre aux demandes de virement »

* + 1. **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  : Diagramme de séquence détaillé « Recevoir des relevés de comptes»

La liste des autres diagrammes de séquences détaillés est placée en Annexe.

## Diagramme de classes

Alors que le diagramme de cas d'utilisation 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 classes ainsi que les différentes relations structurelles entre ces classes.

Ci-dessous le diagramme de classes utilisé pour le développement du 1er Sprint.



Figure  : Diagramme de classes du Sprint 1

# Réalisation

Dans cette partie, nous allons présenter un exemple de code responsable de l’invocation des Web Services ainsi que les interfaces de notre application.

## Codage

Nous présentons ci-dessous le code permettant d’invoquer un Web Service. Dans notre cas, le Web Service « executeWebService » permet de retourner les informations des comptes relatives à un client.

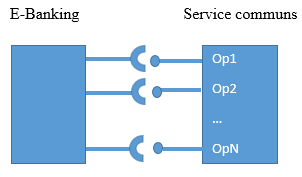
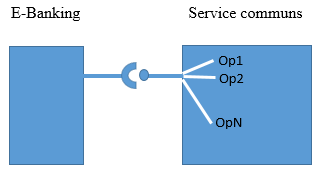


Le Web Service n’est qu’une simulation du Système d’information gérant les services communs de la banque.

Comme le montre ce code, nous invoquons un service web générique nommé « executeWebService ». Ce service prend en paramètre le nom de l’opération spécifique à exécuter.

Avec ce mécanisme nous garantissons la généricité de notre solution.

Il est à signaler qu’une autre alternative aurait été de développer un Web Service pour chaque opération ; ceci aurait pour effet de multiplier les Web Services et de créer un couplage fort au niveau code entre notre système de E-Banking et les services communs.

Sans généricité Notre solution

## 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.

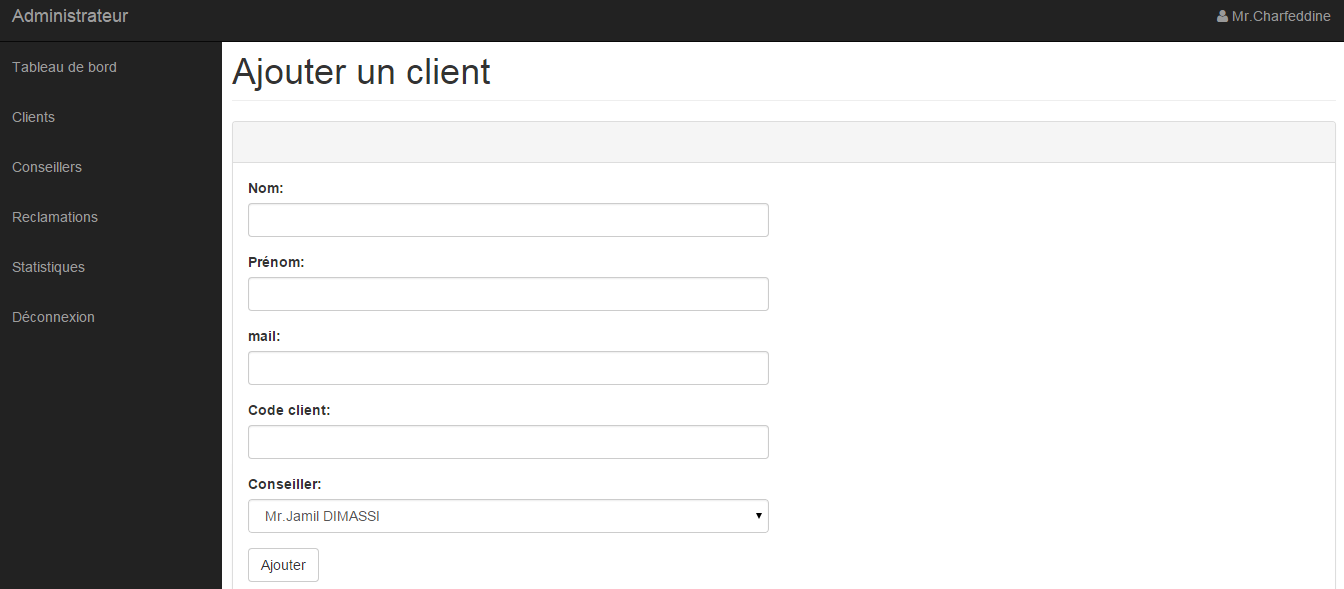


Figure  : Interface graphique « Ajouter un client »

* + 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.

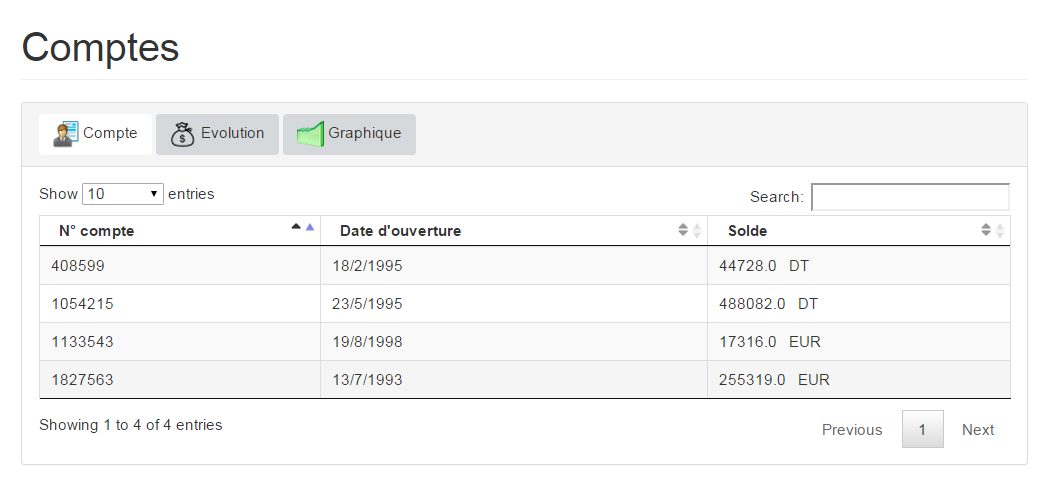


Figure  : Interface graphique « Consulter les comptes »

* + 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.

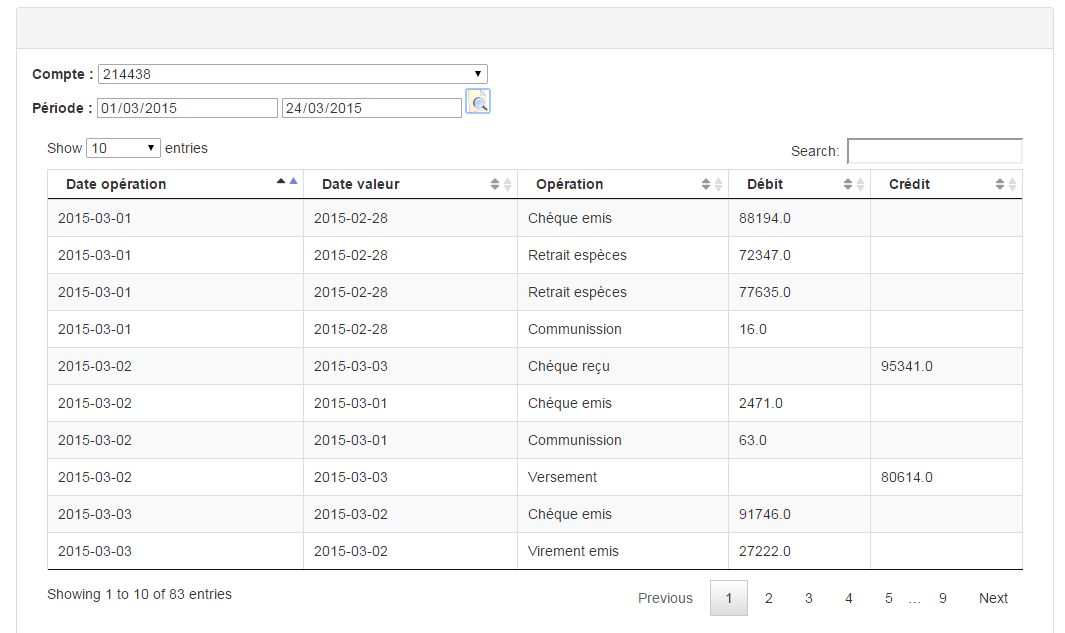
****

Figure  : Interface graphique « Consulter les mouvements d’un compte »

* + 1. **Evolution de solde**

La figure suivante représente l’interface permettant au client de consulter l’évolution du solde d’un compte dès sa création à ce jour.

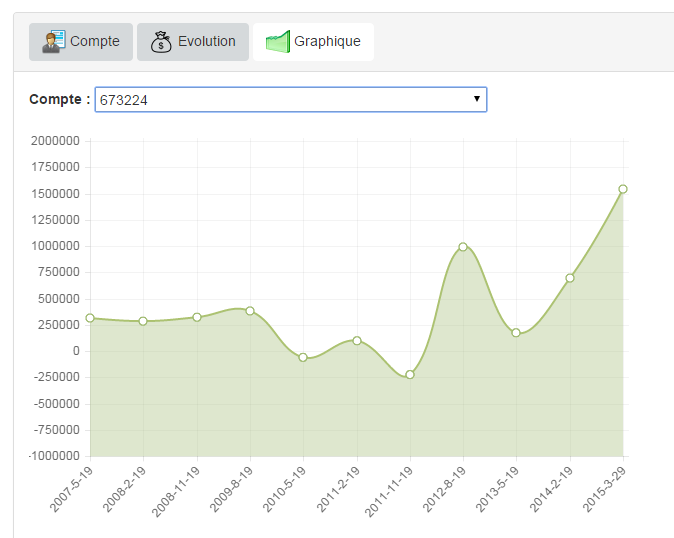
****

Figure  : Interface graphique « Evolution de solde »

* + 1. **Demander un virement & Suivre les demandes de virement**

Pour demander un virement, le client doit passer par l’interface de la figure ci-dessous qui intègre la liste des états des demandes déjà effectuées.

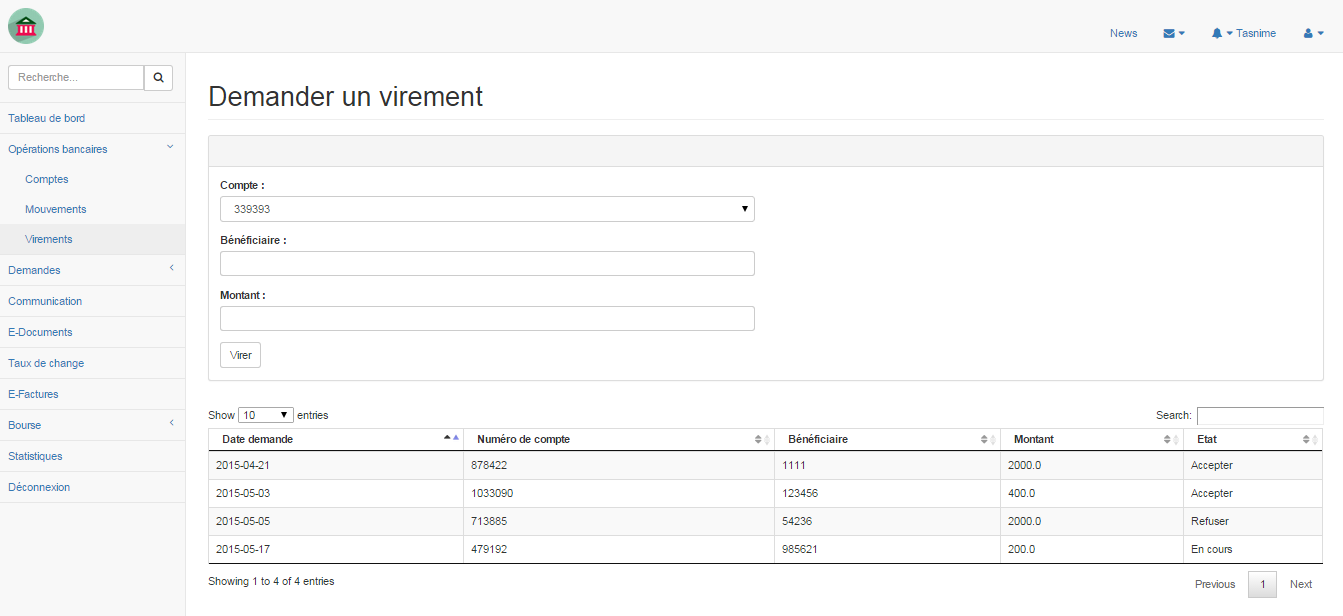


Figure  : Interface graphique « Demander un virement et suivre les demandes »

* + 1. **Recevoir des relevés de comptes**

Notre application permet aux clients de recevoir mensuellement un relevé de chaque compte.

L'interface graphique suivante met en lumière les différents relevés du client :

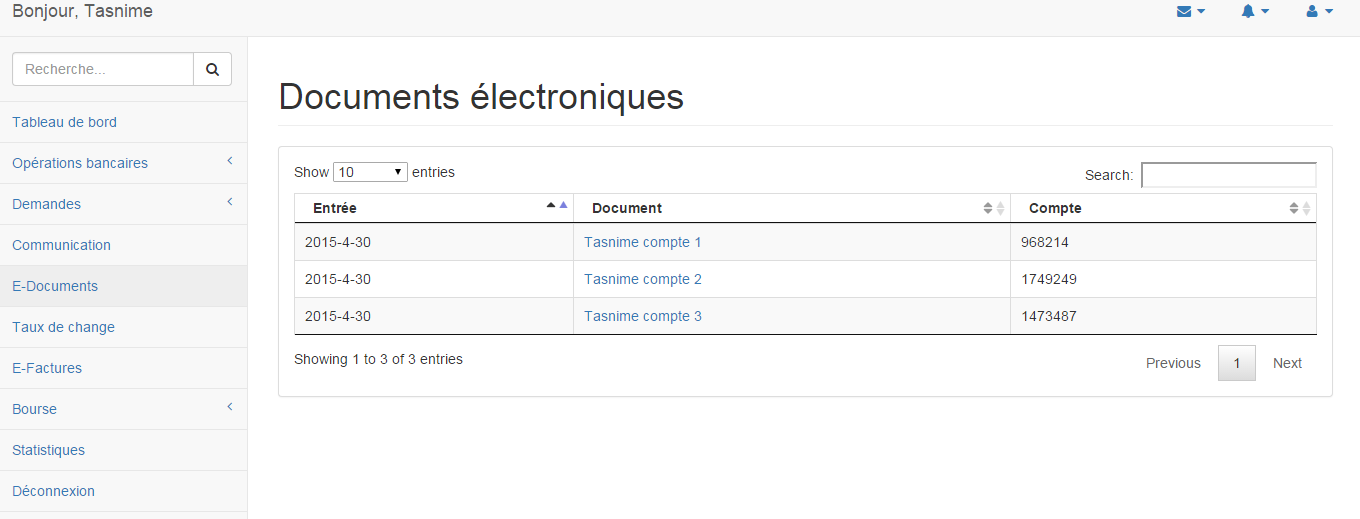


Figure  : Interface graphique « Liste des relevés de comptes »

Lorsque le client clique sur un compte, le relevé s’affiche sous forme PDF comme le montre la figure suivante et il pourra par la suite le télécharger ou l’imprimer.

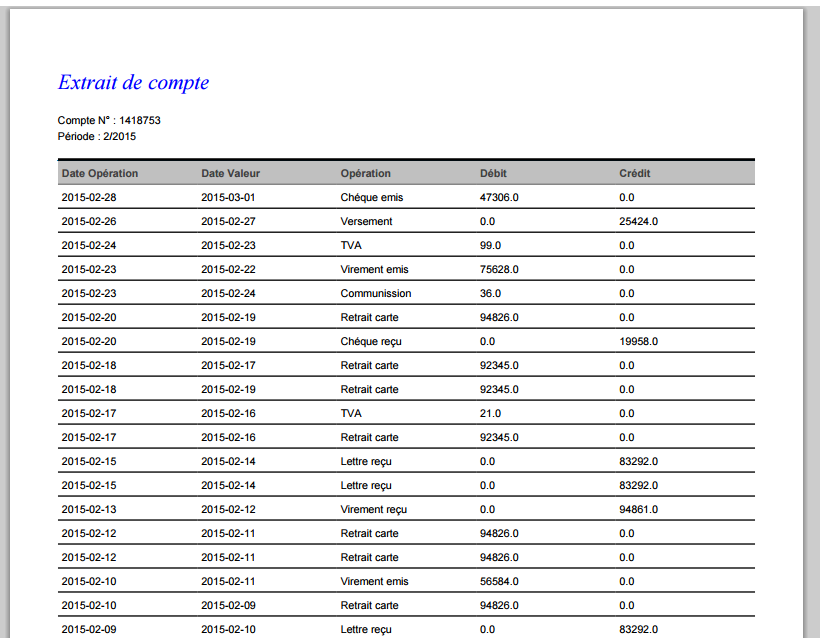


Figure  : Interface graphique « Extrait de compte »

La liste des autres interfaces est placée en Annexe.

# Conclusion

Tout le long de ce chapitre, nous avons présenté le module du Sprint 1, en passant par son analyse, sa conception et sa réalisation.

Dans le chapitre suivant nous ferons la même chose pour le deuxième sprint.

# Chapitre 4. Sprint 2

# Introduction

Au cours de ce chapitre, nous allons traiter le deuxième Sprint. Pour cela, nous commençons par le Backlog du sprint, pour passer ensuite à l’analyse et la conception. Nous terminons par la réalisation.

# Backlog du sprint

Le tableau ci-dessous définit les histoires du 2ème sprint.

Tableau : Backlog du Sprint 2

|  |  |  |
| --- | --- | --- |
| **ID** | **Histoire** | **Estimation** |
| 010 | S’authentifier | 3 |
| 011 | Statistiques des opérations bancaires | 4 |
| 012 | Statistiques globales | 2 |
| 013 | Changer un conseiller | 1 |
| 014 | Affecter un client |
| 015 | Demander une carte bancaire | 1 |
| 016 | Suivre les demandes de cartes | 1 |
| 017 | Répondre aux demandes de cartes | 1 |
| 018 | Demander un chéquier | 1 |
| 019 | Suivre les demandes de chéquiers | 1 |
| 020 | Répondre aux demandes de chéquiers | 1 |
| 021 | Demander un crédit | 2 |
| 022 | Ajouter un document | 1 |
| 023 | Suivre les demandes de crédits | 1 |
| 024 | Répondre aux demandes de crédits | 1 |

# Analyse

La phase d’analyse permet de répondre à la question « Que fait le système ? ». Cette réponse sera représentée sous deux formes. La 1ère sera réservée aux descriptions des User Stories et la 2ème pour les maquettes qui seront détaillées plus tard.

## Descriptions

Nous présentons ci-dessous les différentes descriptions de quelques histoires utilisateurs du Sprint 2.

Tableau : Description de la User Story « Statistiques des opérations bancaires »

|  |  |
| --- | --- |
| #011 | Statistiques des opérations bancaires |
| **En tant que** client  **Je veux** voir des statistiques sur mes opérations bancaires | |
| **Vu que** je suis arrivé à la page des statistiques  **Vu que** je choisis un compte et je saisis une date de début et une date de fin  **Quand** je clique sur le bouton « Consulter »  **Alors** les statistiques s’affichent sous format graphique | |

Tableau : Description de la User Story «Changer un conseiller »

|  |  |
| --- | --- |
| #013 | Changer un conseiller |
| **En tant qu’**administrateur  **Je veux** changer le conseiller d’un client | |
| **Vu que** je suis arrivé à la page des clients  **Quand** je clique sur « Modifier conseiller »  **Alors** un formulaire s’affiche  **Vu que** je choisi un conseiller  **Quand** je clique sur « Valider »  **Alors** la modification s’effectue et un message de succès s’affiche | |

Tableau : Description de la User Story «Demander une carte »

|  |  |
| --- | --- |
| #015 | Demander une carte |
| **En tant que** client  **Je veux** passer une demande d’une carte bancaire | |
| **Vu que** je suis arrivé à la page des cartes  **Vu que** je choisis un compte parmi mes comptes et un type de carte  **Quand** je clique sur le bouton « demander »  **Alors** la demande sera envoyée au conseiller et la page des cartes se réaffiche | |

Tableau : Description de la User Story «Répondre aux demandes de cartes bancaires »

|  |  |
| --- | --- |
| #017 | Répondre aux demandes de cartes bancaires |
| **En tant que** conseiller  **Je veux** répondre à une demande d’une carte | |
| **Vu que** je suis sur la page des cartes  **Quand** je clique sur un bouton « accepter »  **Alors** un formulaire de saisie d’une date de délivrance s’affiche  **Vu que** je saisis une date de délivrance  **Quand** je clique sur le bouton « Répondre »  **Alors** la réponse sera envoyée par mail au client et la page des cartes se réaffiche  **Quand** je clique sur un bouton « refuser »  **Alors** la réponse sera enregistrée et la page des cartes se réaffiche | |

## Maquette

Nous présentons ci-dessous une maquette du Sprint 2.

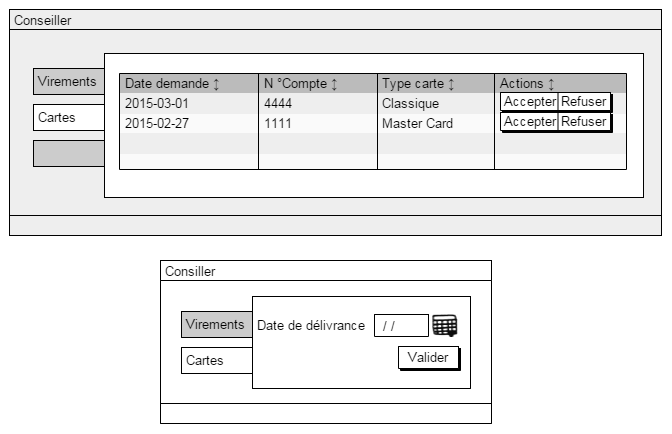


Figure  : Maquette de la User Story « Répondre aux demandes de cartes bancaires »

# Conception

Dans cette partie, nous allons présenter les différents diagrammes de séquences détaillés ainsi que le diagramme de classes pour ce Sprint.

## Diagrammes des séquences détaillés

Nous exposons ci-dessous des diagrammes de séquences détaillés des histoires utilisateurs du Sprint 2.

* + 1. **Statistiques des opérations bancaires**

En arrivant à la vue Statistiques, un client authentifié peut choisir un compte et les dates de début et de fin. Une fois ces informations fournies, le contrôleur « StatistiquesControlleur » se charge d’invoquer à travers le Web Service « executeWebService » l’opération « Mouvements » ainsi le client pourra consulter, sous forme graphique, les statistiques de ses opérations bancaires reçues par le contrôleur et formatées par la vue.

Le diagramme de séquence suivant décrit ces étapes.



Figure  : Diagramme de séquence détaillé « Statistiques des opérations bancaires »

* + 1. **Répondre aux demandes de cartes bancaires**

En arrivant à la vue « Cartes », un conseiller authentifié peut avoir la liste des demandes de cartes de ses clients. Il peut ainsi choisir un d’entre eux afin de répondre à la demande.

En cas ou le choix du conseiller est positif, le contrôleur « CartesControlleur » se charge d’activer la vue « ReponseCarte ». La vue ainsi affichée demande au conseiller : la date de délivrance de la carte. Une fois cette information fournie, le contrôleur « ReponseCarteController » se charge de mettre à jour la base et d’envoyer un mail au client.

En cas ou le choix du conseiller est négatif, le contrôleur « ReponseCarteControlleur » enregistre directement la réponse dans la base.

Le diagramme de séquence suivant décrit ces étapes.



Figure  : Diagramme de séquence détaillé « Répondre aux demandes de cartes bancaires »

* + 1. **Demander un crédit**

La vue « Credit » demande au client connecté : le montant du crédit, la durée de remboursement et son revenu mensuel et ce, pour la simulation du crédit.

En cas de succès de simulation, la vue « DemandeCredit » est affichée et demande au client : le compte, le type de crédit et son dossier. Une fois ces informations fournies, le contrôleur « DemandeCreditControlleur » se charge d’enregistrer la demande dans la base.

En cas d’erreur de simulation, un message d’erreur s’affiche.

Le diagramme de séquence suivant décrit ces étapes.



Figure  : Diagramme de séquence détaillé « Demander un crédit»

## Diagramme de classes

Ci-dessous le diagramme de classes utilisé pour le développement du 2ème Sprint.

Ce diagramme est un incrément du diagramme de classes du sprint 1.



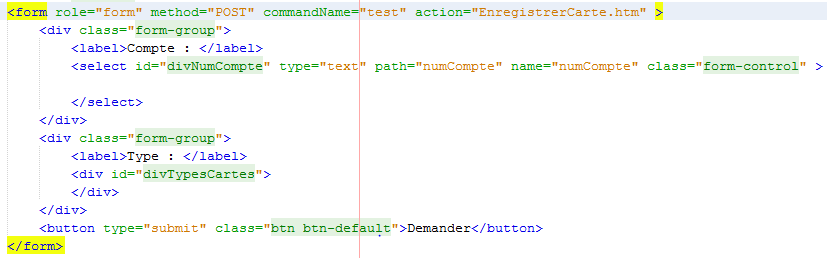
Figure  : Diagramme de classes du Sprint 2

# Réalisation

Dans cette partie, nous allons présenter les interfaces de notre application.

## Codage

Nous exposons ci-dessous un exemple de code de la User Story « Demander carte ».



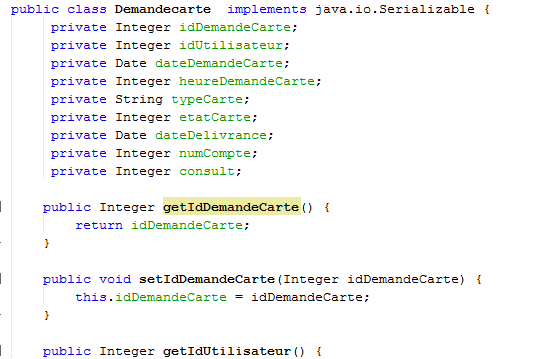
Après avoir choisi le numéro de compte et le type de carte, le clic sur le bouton « Demander » va activer le contrôleur responsable de l’enregistrement de la demande. Nous présentons ci-dessous le code responsable à l’enregistrement.



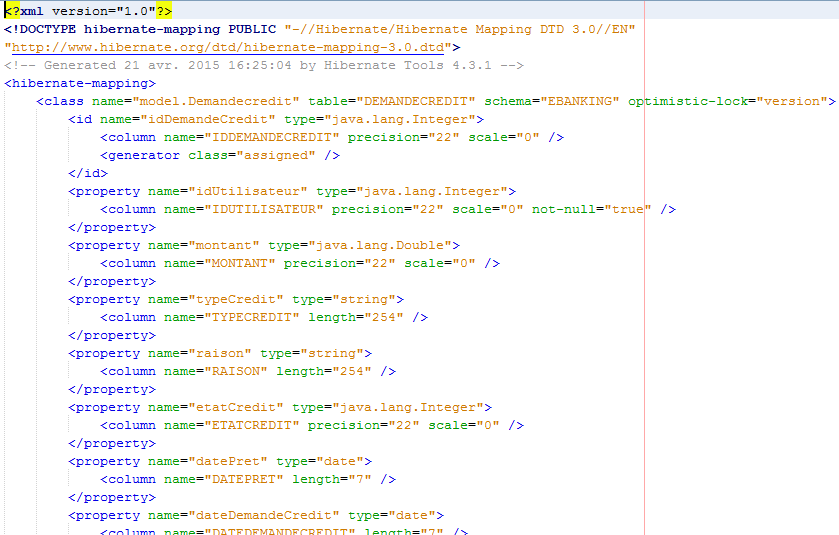
Pour enregistrer des nouvelles données dans la base, il suffit de créer une nouvelle instance de la classe encapsulant les données (« carte » dans notre exemple) et d'appeler la méthode save() de la session en lui passant en paramètre l'objet encapsulant les données.

La méthode save() n'a aucune action directe sur la base de données. Pour enregistrer les données dans la base, il faut réaliser un commit sur la transaction.

Ce contrôleur communique avec le modèle du code suivant :



Et enfin, nous devons assurer le mapping de la base de données à travers la portion de code suivante :



## Interface Graphique

Nous présentons ci-dessous, des imprimes écrans relatifs à quelques interfaces du deuxième sprint.

* + 1. **S’authentifier**

La figure suivante représente la première interface vue au démarrage de l’application. L’utilisateur introduit un login et un mot de passe pour pouvoir accéder à l’application et profiter de nos services.

****

Figure  : Interface graphique « S’authentifier »

* + 1. **Statistiques des opérations bancaires**

L’E-Banking offre la possibilité au client de consulter des statistiques de ses opérations bancaires effectuées pendant une période comme le montre la figure suivante.

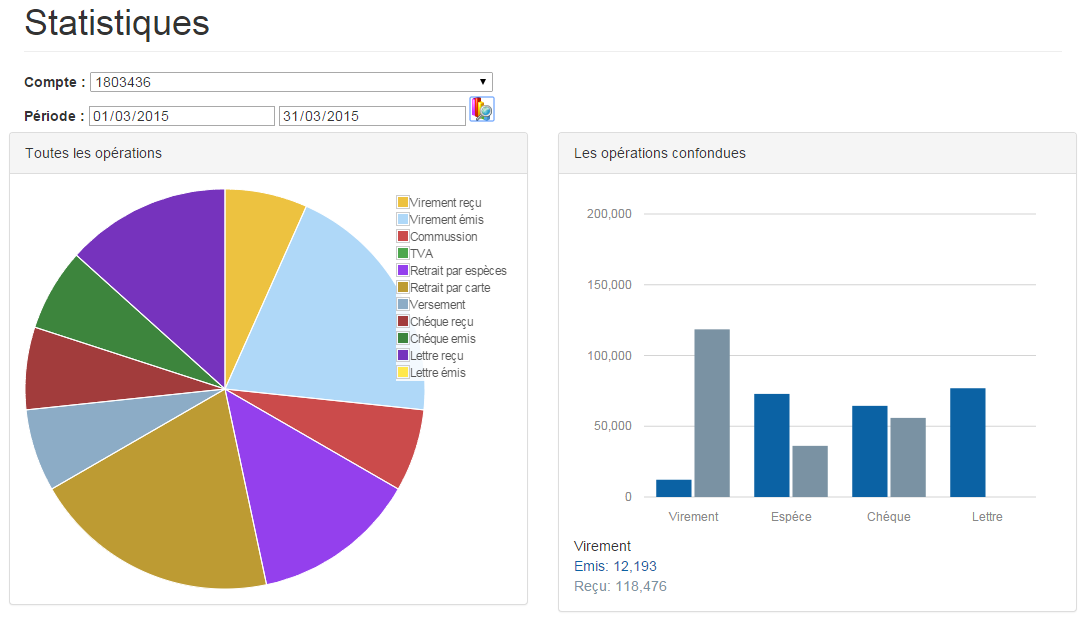


Figure  : Interface graphique « Statistiques des opérations bancaires »

* + 1. **Changer un conseiller**

Afin de changer le conseiller d’un client, l’administrateur passe par l’interface suivante :

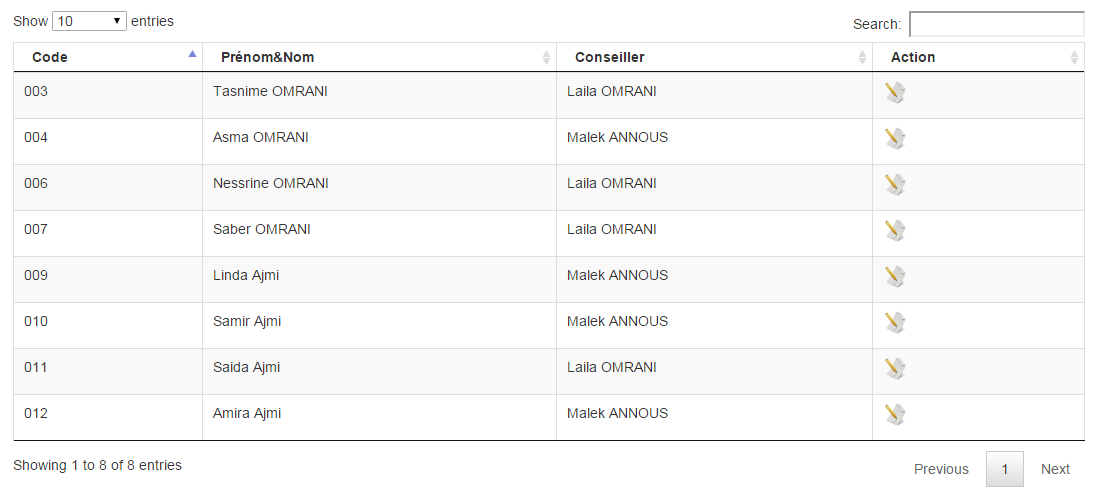
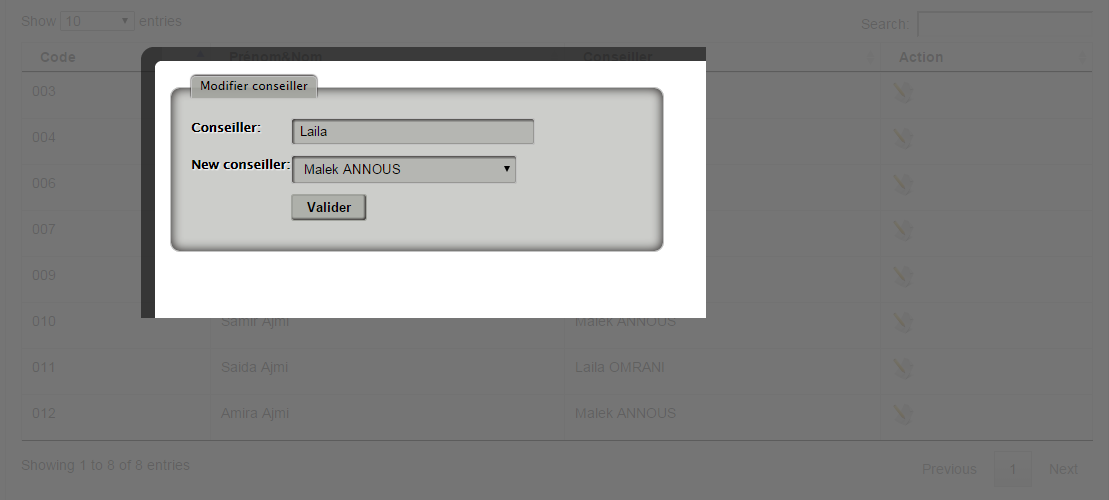


Figure  : Interface graphique « Changer un conseiller »

Après avoir demandé la liste des clients, l’administrateur clique sur le bouton « modifier » relatif à un client, un pop-up s’affiche comme le montre la figure suivante.



* + 1. **Affecter un client**

Dans cette interface, nous trouvons une liste de tous les conseillers. Lorsque l’administrateur clique sur un conseiller, la liste des clients de ce dernier s’affiche permettant à l’administrateur de supprimer l’un d’eux et de l’affecter à un autre conseiller.

Cette interface est représentée par la figure suivante :

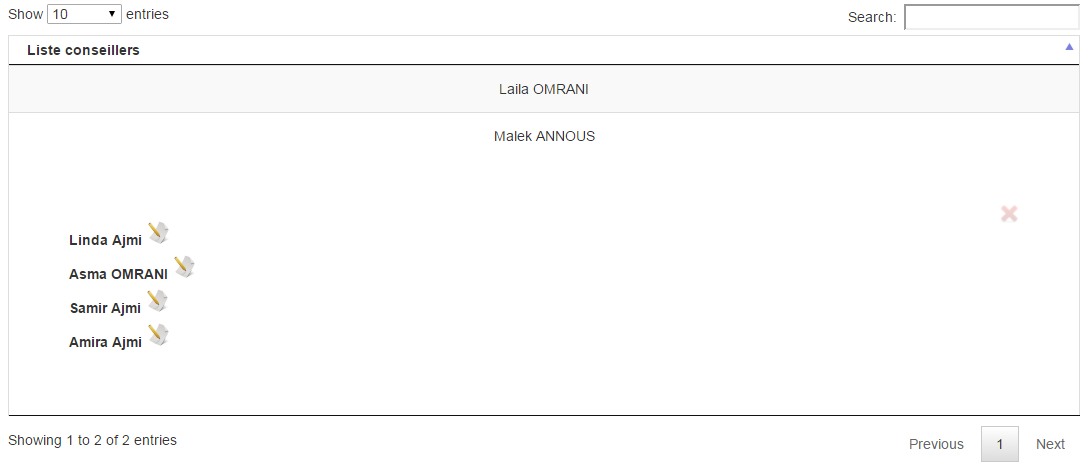


Figure  : Interface graphique « Affecter un client »

* + 1. **Demander une carte & Suivre les demandes des cartes bancaires**

Pour passer une demande d’une carte bancaire, il suffit de remplir le formulaire suivant.

Cette interface contient aussi la liste des demandes déjà effectuées.

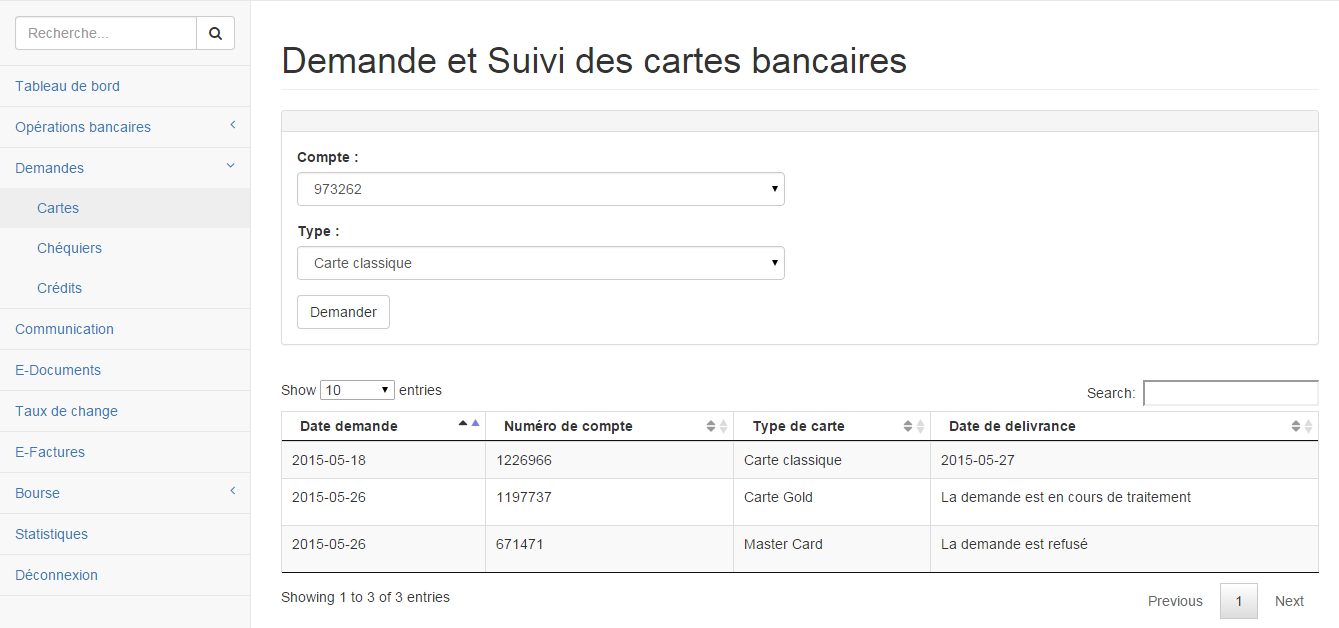


Figure  : Interface graphique « Demander et suivre les cartes bancaires »

* + 1. **Répondre à une demande d’une carte bancaire**

Pour répondre à une demande d’une carte bancaire, le conseiller doit suivre les étapes suivantes :

1. Cliquez sur le l’action (accepter ou refuser) de la demande.



Figure  : Interface graphique « Répondre à une demande d’une carte bancaire »

1. Remplir le formulaire suivant :



* + 1. **Demander un crédit**

Pour demander un crédit il faut :

1. Passer par l’étape de simulation de crédit.
2. Remplir le formulaire y afférent.

Les deux interfaces graphiques suivantes montrent comment nous pouvons simuler un crédit :

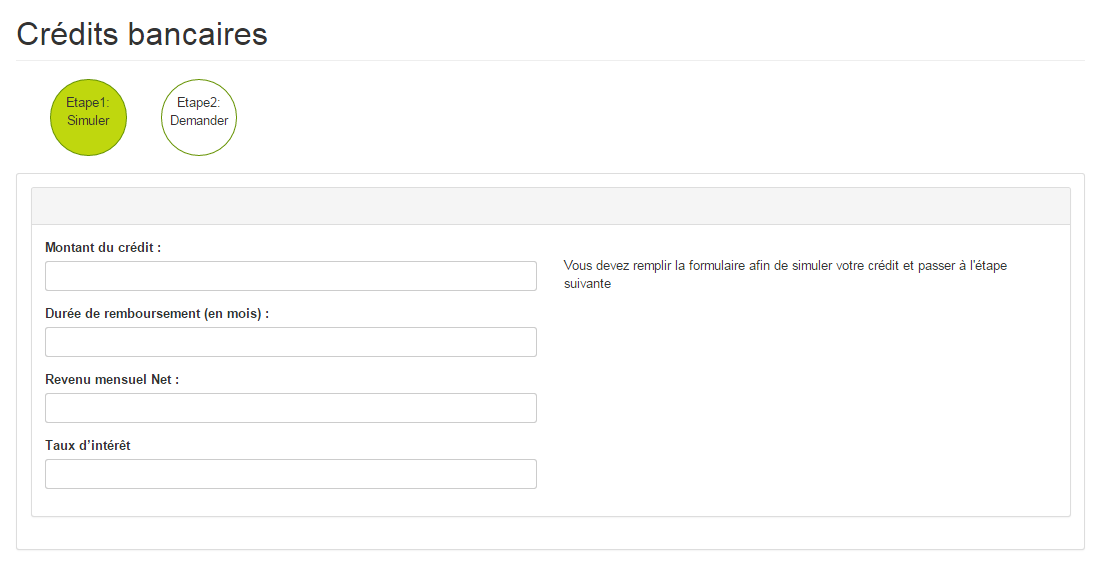
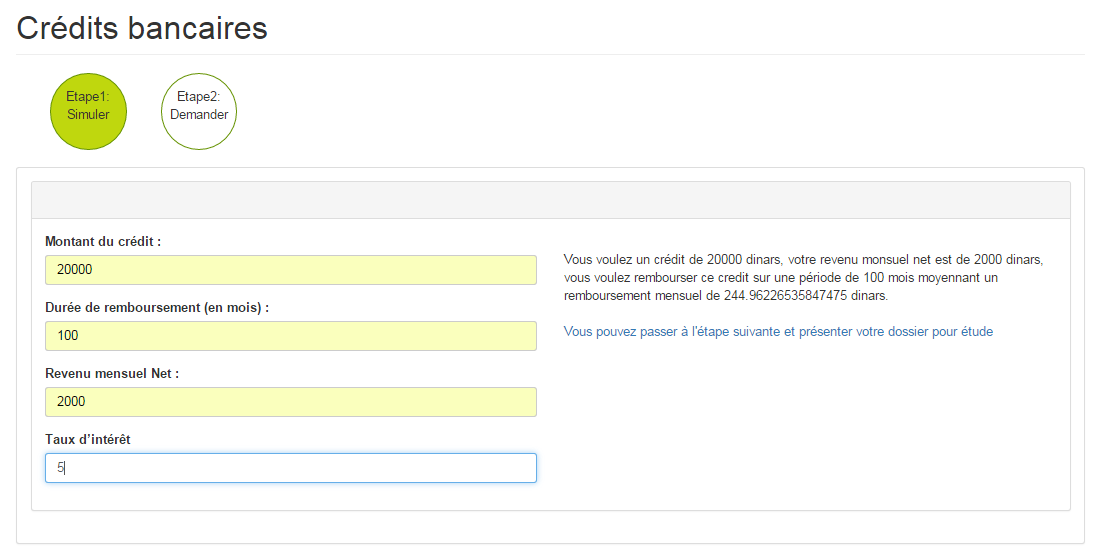


Figure  : Interface graphique « Simuler un crédit »

Après avoir rempli correctement le formulaire, l’accès à la demande de crédit sera possible, comme le montre la figure suivante :



L'interface graphique ci-dessous offre au client la possibilité de poser son dossier de crédit.

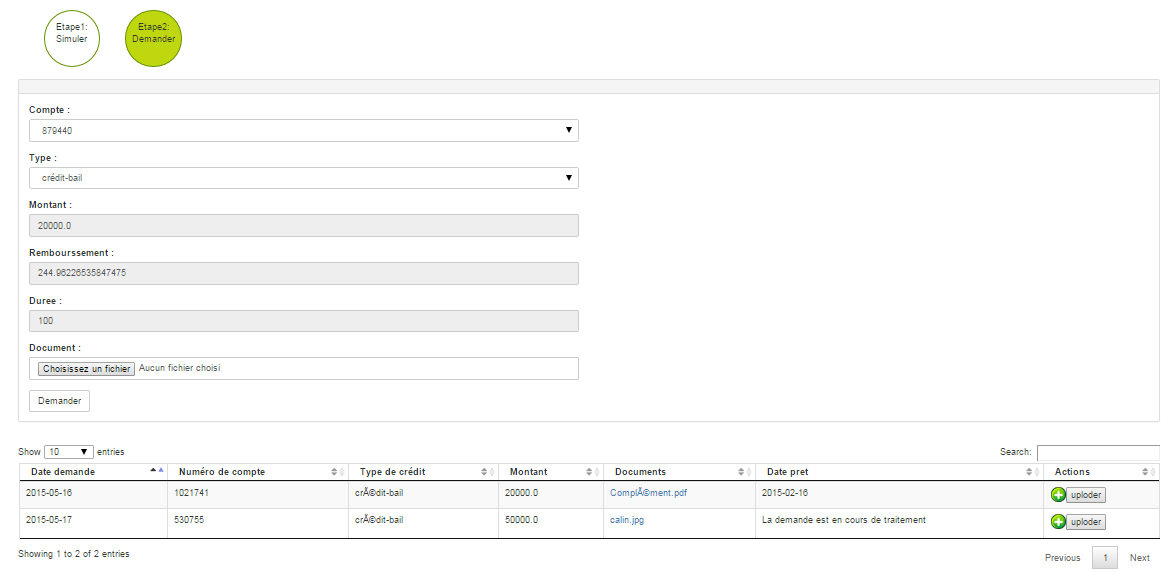


Figure  : Interface graphique « Demander un crédit »

# Conclusion

Dans ce chapitre, nous avons présenté le deuxième sprint, à travers les étapes d’analyse, de conception et de réalisation.

Dans le chapitre suivant, nous détaillerons le troisième sprint.

# Chapitre 5. Sprint 3

# Introduction

Dans ce chapitre, nous allons présenter le troisième Sprint. Nous commençons d’abord par le Backlog du sprint. Ensuite, nous passons à l’analyse, la conception pour l’achever avec la partie réalisation.

# Backlog du sprint

Le tableau ci-dessous définit les histoires du 3ème sprint.

Tableau : Backlog du Sprint 3

|  |  |  |
| --- | --- | --- |
| **ID** | **Histoire** | **Estimation** |
| 025 | Déposer une réclamation | 1 |
| 026 | Consulter les réponses aux réclamations | 3 |
| 027 | Répondre à une réclamation | 1 |
| 028 | Demander un document | 1 |
| 029 | Envoyer un document | 1 |
| 030 | Recevoir des notifications de conseiller | 3 |
| 031 | Suivre les conseillers | 3 |
| 032 | Recevoir les notifications des clients | 2 |

# Analyse

La phase d’analyse permet de répondre à la question « Que fait le système ? ». Cette réponse sera représentée sous deux formes. La 1ère sera réservée aux descriptions des User Stories et la 2ème pour les maquettes qui seront détaillées plus tard.

## Descriptions

Nous présentons ci-dessous les différentes descriptions de quelques histoires utilisateurs du Sprint 3.

Tableau : Description de la User Story «Déposer une réclamation »

|  |  |
| --- | --- |
| #025 | Déposer une réclamation |
| **En tant que** client  **Je veux** communiquer avec ma banque afin de passer une réclamation | |
| **Vu que** je suis arrivé à la page des réclamations  **Vu que** je saisis une réclamation  **Quand** je clique sur le bouton « envoyer »  **Alors** la réclamation sera envoyée au conseiller et la page des réclamations se réaffiche | |

Tableau : Description de la User Story «Consulter les réponses aux réclamations »

|  |  |
| --- | --- |
| #026 | Consulter les réponses aux réclamations |
| **En tant que** client  **Je veux** consulter les réponses à mes réclamations | |
| **Vu que** je suis sur la page des réclamations  **Quand** je clique sur une réclamation  **Alors** les réponses s’affichent | |

## Maquette

Nous présentons ci-dessous une maquette du Sprint 3.

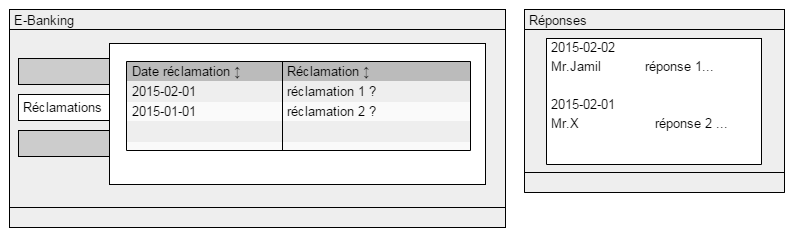


Figure  : Maquette de la User Story « Consulter les réponses aux réclamations»

# Conception

Dans cette partie, nous allons présenter les différents diagrammes de séquences détaillés ainsi que le diagramme de classes pour ce Sprint.

## Diagrammes de séquences détaillés

Nous exposons ci-dessous quelques diagrammes de séquences détaillés des histoires utilisateurs du Sprint 3.

**Répondre à une réclamation**

En arrivant à la vue « Communication », un conseiller authentifié peut avoir la liste des réclamations de ses clients. Il peut ainsi choisir un d’entre eux et pourra par la suite consulter les réponses déjà existantes reçues par le contrôleur « ReponseControlleur » et formatées par la vue « Reponses ». Ensuite, une fonction de vérification d’état de la réclamation se déclenche et renvoie le statut de la réclamation.

Si la réclamation est ouverte, le conseiller peut : soit saisir une réponse ; ce qui va activer le contrôleur « EnregisterReponse » afin de sauvegarder la réponse, soit changer le statut par « fermé » ou « résolu ». Dans ce cas le contrôleur va mettre à jour la table « Communication ».



Figure  : Diagramme de séquence détaillé « Répondre à une réclamation »

## Diagramme de classes

Ci-dessous le diagramme de classes utilisé afin de développer le 3ème Sprint.

Ce diagramme est un incrément du diagramme de classes du sprint 2.



Figure  : Diagramme de classes du Sprint 3

# Réalisation

Nous présentons ci-dessous, des imprimes écrans relatifs à quelques interfaces du troisième sprint.

* 1. **Déposer une réclamation**

À travers l’interface suivant, le client peut déposer une requête à son conseiller ou à l’administrateur d’E-Banking.

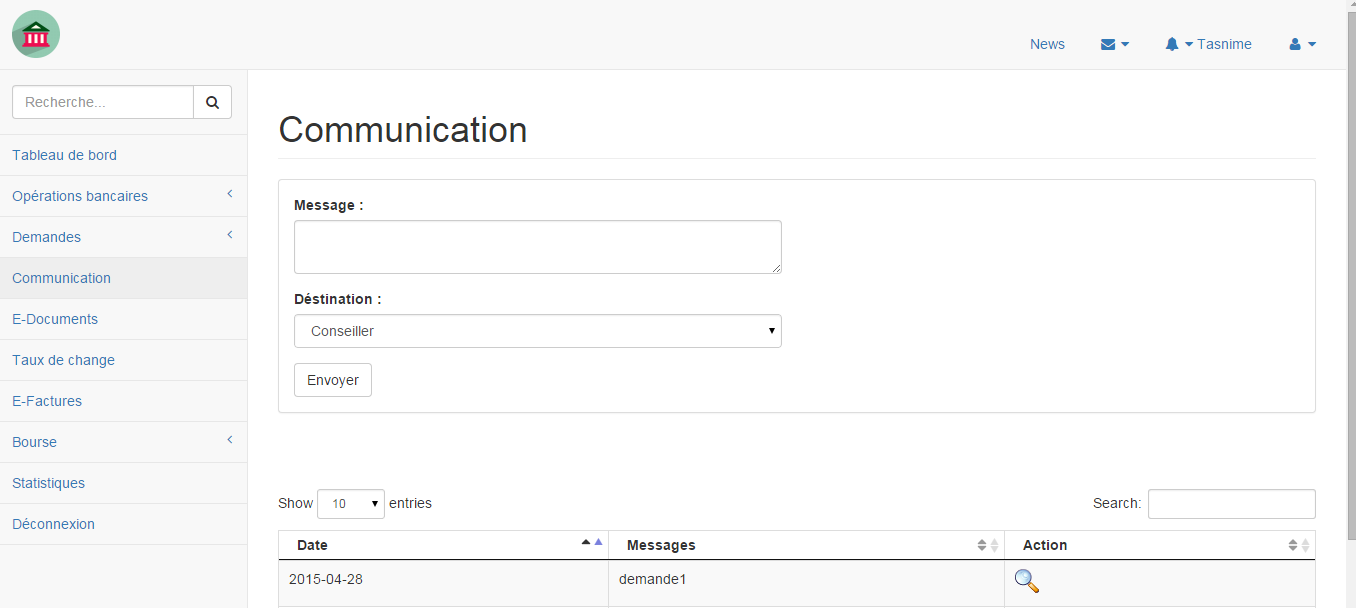


Figure  : Interface graphique « Déposer une réclamation »

* 1. **Consulter les réponses aux réclamations**

L'interface graphique suivante permet aux clients de parcourir leurs réclamations ainsi les messages envoyés par leurs conseillers. Quand un client veut accéder aux réponses, il doit cliquer le bouton à droite.

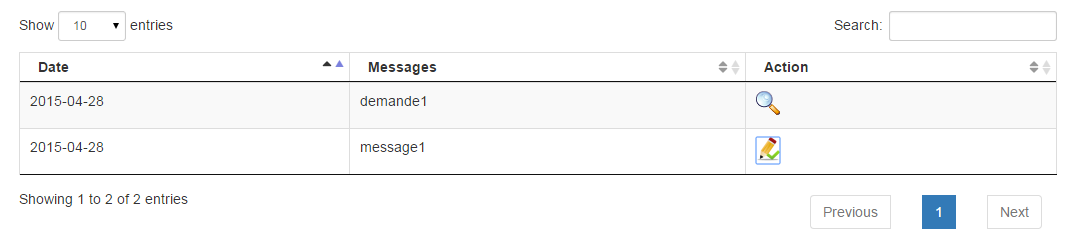


Figure  : Interface graphique « Liste des réclamations »

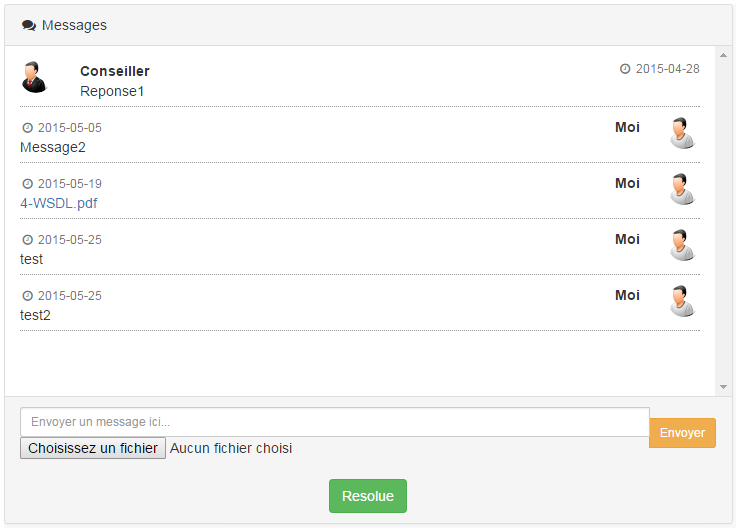


Figure  : Interface graphique « Consulter les réponses à la réclamation »

* 1. **Recevoir les notifications du conseiller**

À travers l’interface présentée dans la figure suivante, le client consulte les notifications des réponses envoyées par son conseiller et peut aussi consulter la valeur de son portefeuille de la bourse.

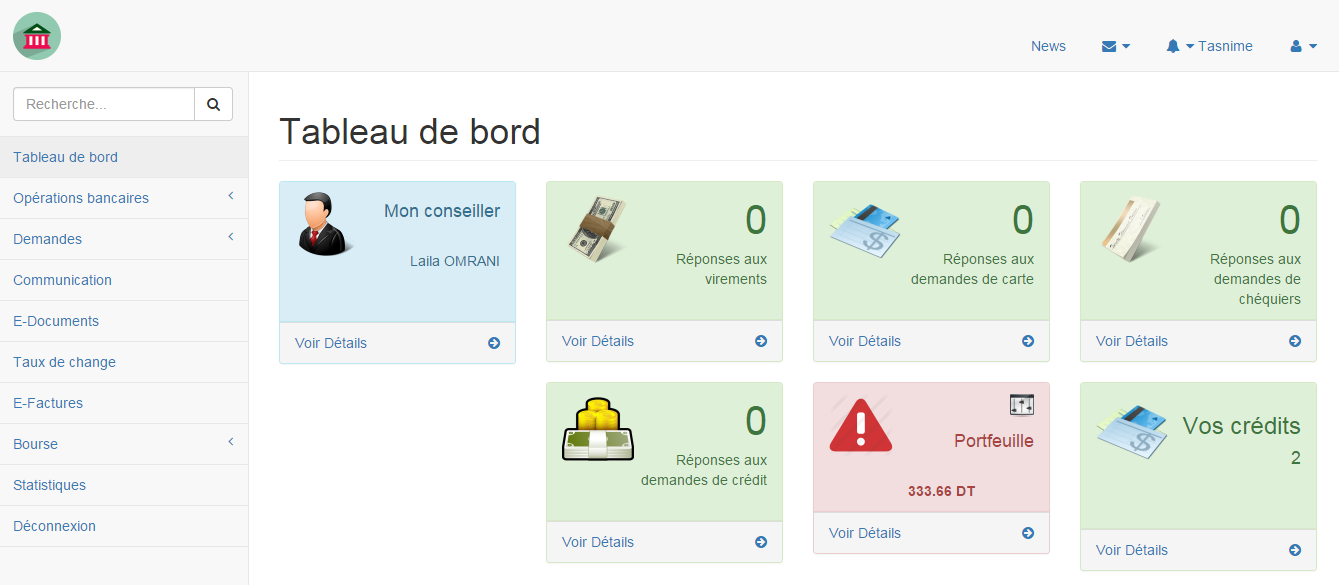


Figure  : Interface graphique « Recevoir les notifications du conseiller »

Afin de suivre ses crédits, le client clique sur le bouton « Détails de vos crédits » vu dans la figure précédente, ce qui fait apparaitre l’interface suivante :



* 1. **Suivre les conseillers**

L’Administrateur a une vue globale sur toutes les tâches des conseillers de ce jour. Les tâchent non encore traitées s’affichent en haut. Il peut demander ainsi l’affichage détaillée en cliquant sur le bouton « consulter ».

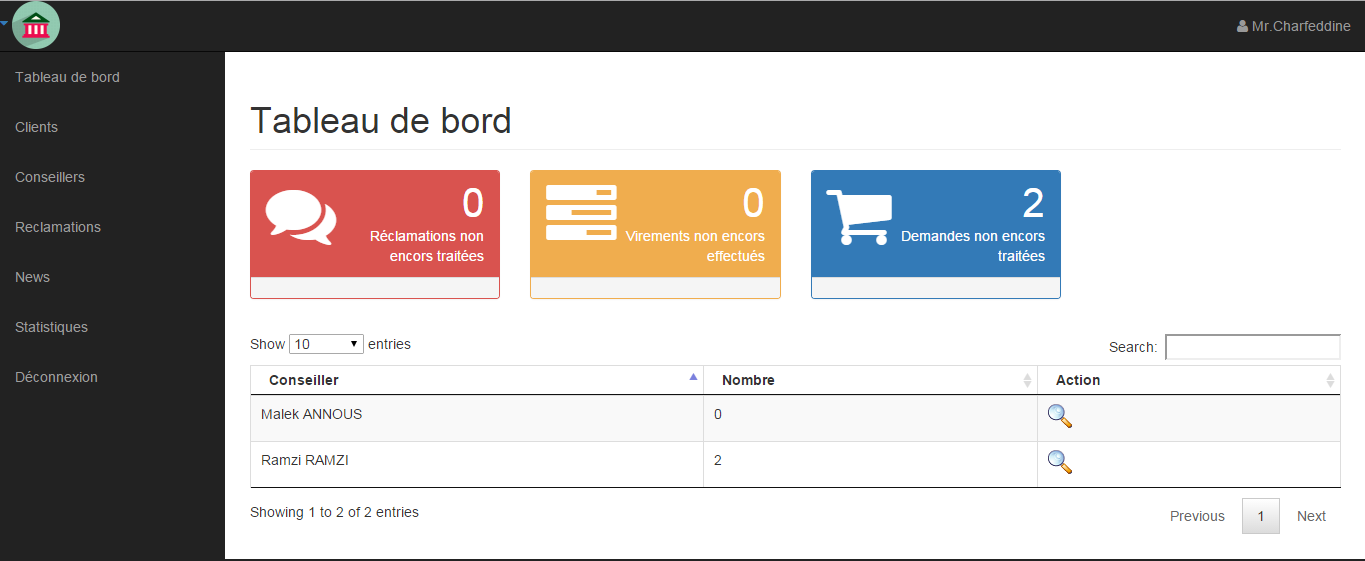


Figure  : Interface graphique « Suivre les conseillers »

En cliquant sur le bouton « consulter », le pop-up suivant s’affiche.



# Conclusion

Dans ce chapitre, nous avons présenté le troisième sprint à travers l’étape d’analyse, de conception et de réalisation.

Dans le chapitre suivant nous détaillerons le quatrième et dernier sprint.

# Chapitre 6. Sprint 4

# Introduction

Dans ce chapitre, nous allons présenter le dernier Sprint. Pour cela, nous commençons par le Backlog du sprint, pour passer ensuite à l’analyse, la conception et la réalisation. Nous allons, à la fin, montrer le diagramme de Gantt.

# Backlog du sprint

Le tableau ci-dessous définit les histoires du 4ème sprint.

Tableau : Backlog du Sprint 4

|  |  |  |
| --- | --- | --- |
| **ID** | **Histoire** | **Estimation** |
| 033 | Demander une facture | 2 |
| 034 | Payer une facture | 2 |
| 035 | Répondre aux demandes de factures | 2 |
| 036 | Consulter l’actualité de la bourse | 2 |
| 037 | Demander l’achat d’actions | 2 |
| 038 | Demander la vente d’actions | 1 |
| 039 | Consulter le portefeuille | 1 |
| 040 | Répondre aux demandes d’achat d’actions | 1 |
| 041 | Répondre aux demandes de vente d’actions |
| 042 | Consulter les taux de change | 2 |
| 043 | Convertir une devise | 1 |
| 044 | Changer le logo |
| 045 | Ajouter une nouveauté | 1 |
| 046 | Consulter les nouveautés |

# Analyse

La phase d’analyse permet de répondre à la question « Que fait le système ? ». Cette réponse sera représentée sous deux formes. La 1ère sera réservée aux descriptions des User Stories et la 2ème pour les maquettes qui seront détaillées plus tard.

## Descriptions

Nous présentons ci-dessous les différentes descriptions d’histoires utilisateurs du Sprint 4.

Tableau  : Description de la User Story « Demander une facture »

|  |  |
| --- | --- |
| #033 | Demander une facture |
| **En tant que**  client  **Je veux** recevoir mes factures dans l’E-Banking | |
| **Vu que** je suis à la page des factures  **Quand** je clique sur un type de facture  **Alors** un formulaire de demande s’affiche  **Vu que** je choisis un compte parmi mes comptes et je saisis mon identifiant dans l’entreprise  **Quand** je clique sur demander  **Alors** la demande sera envoyée à mon conseiller et la page des factures se réaffiche. | |

Tableau : Description de la User Story «Payer une facture»

|  |  |
| --- | --- |
| #034 | Payer une facture |
| **En tant que**  client  **Je veux** payer mes factures dans l’E-Banking | |
| **Vu que** je suis à la page des factures  **Quand** je clique sur un type de facture  **Alors** ma facture s’affiche  **Quand** je clique sur payer  **Alors** la facture sera payée et la page des factures se réaffiche. | |

Tableau : Description de la User Story « Acheter des actions »

|  |  |
| --- | --- |
| #037 | Demander l’achat d’actions |
| **En tant que**  client  **Je veux** acheter des actions en bourse | |
| **Vu que** je suis à la page d’achat d’actions  **Vu que** je saisis le nombre d’actions à acheter, le prix d’achat maximal et je choisis un numéro de compte et le nom de la société.  **Quand** je clique sur le bouton « Acheter»  **Alors** la demande sera envoyée à mon conseiller afin de la traiter. | |

## Maquettes

Nous présentons ci-dessous quelques maquettes du quatrième Sprint.

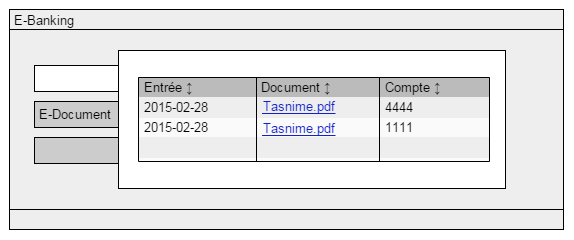


Figure  : Maquette de la User Story « Recevoir des relevés de comptes »

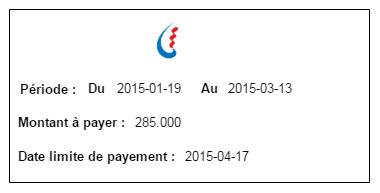


Figure  : Modèle d’une facture

# Conception

Dans cette partie, nous allons présenter les différents diagrammes de séquences détaillés, ainsi que le diagramme de classes pour ce Sprint.

## Diagrammes de séquences détaillés

Nous exposons ci-dessous quelques diagrammes de séquences détaillés des histoires utilisateurs du Sprint 4.

* + 1. **Payer une facture**

Pour que le client paye une facture, il faut tout d’abord passer par le cas d’utilisation « Recevoir une facture ». À ce stade, le client consulte sa facture et peut cliquer sur le bouton « payer ».

Le contrôleur « PayerFactureController » se charge d’invoquer à travers le Web Service « executeWebService » l’opération « Solde » afin de récupérer le solde du compte et vérifier ensuite la disponibilité du montant nécessaire pour le payement de la facture. Si le solde est suffisant, le même contrôleur va invoquer l’opération « Virement » puis mettre à jour l’état de la facture. Si le solde est insuffisant, un message d’insuffisance est affiché.



Figure  : Diagramme de séquence détaillé « Payer une facture »

* + 1. **Demander l’achat d’actions**

En cliquant sur le menu « Acheter », un client authentifié peut avoir les numéros de ses comptes et l’actualité de la bourse. La vue ainsi affichée demande au client : le numéro de compte, le nombre d’actions qu’il souhaite acheter et le prix maximum d’achat. Une fois ces informations fournies, le contrôleur « EnregistrerActionsAcheterControlleur » se charge d’enregistrer la demande dans la base des données.

Le diagramme de séquences suivant décrit ces étapes.



Figure  : Diagramme de séquence détaillé « Demander l’achat d’actions »

* + 1. **Consulter le portefeuille**

Un client authentifié peut cliquer sur le menu portefeuille ; ce qui va activer le contrôleur pour sélectionner les actions achetées et vendues en bourse et invoquer à travers le Web Service « executeWebService » l’opération responsable de l’actualité de la bourse afin de calculer la valeur du portefeuille et envoyer à la vue « Portefeuille » la liste des actions achetées et vendues ainsi que la valeur du portefeuille.



Figure  : Diagramme de séquence détaillé « Consulter le portefeuille »

## Diagramme de classes

Ci-dessous le diagramme de classes utilisé afin de développer le 4ème Sprint.

Ce diagramme est un incrément du diagramme de classes du sprint 3 et représente le diagramme de classe complet.



Figure  : Diagramme de classes du Sprint 4

# Réalisation

Nous présentons ci-dessous, des imprimes écrans relatifs aux interfaces du quatrième sprint.

* 1. **Demander une facture**

Les figures 60 et 61 représentent les interfaces de demande d’une facture. Seuls les clients ont accès à ces interfaces.

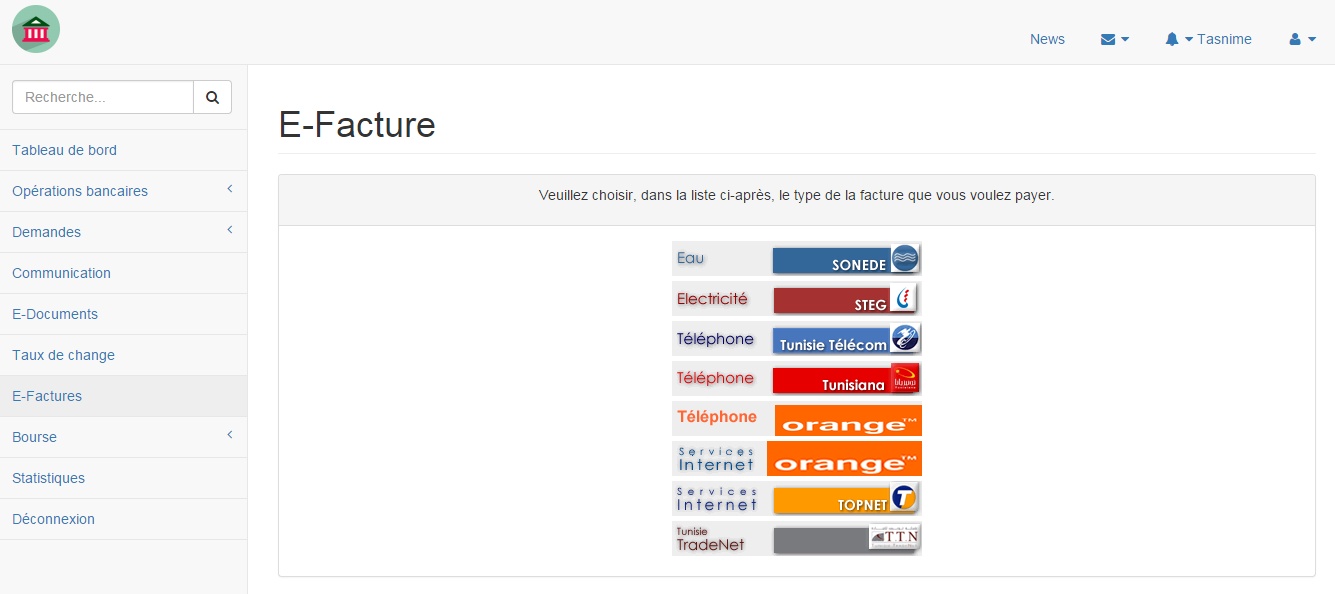


Figure  : Interface graphique « Liste des factures »

Après avoir choisi une facture à travers un clic sur l’une des images vues dans la figure précédente, le client doit remplir le champ relatif à son identifiant dans l’entreprise en question et sélectionner un numéro de compte comme le montre la figure 61.

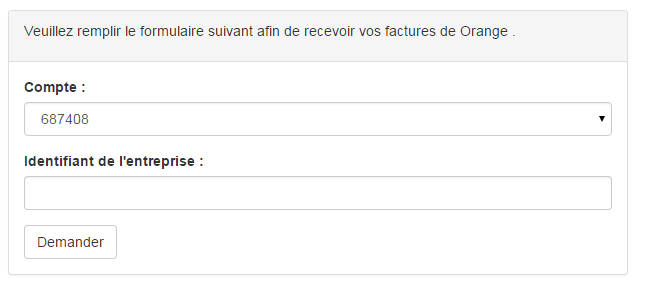


Figure  : Interface graphique « Demander une facture »

* 1. **Payer une facture**

Dans la fenêtre présentée dans la figure ci-dessous, le client peut payer sa facture en cliquant sur le bouton « Payer »

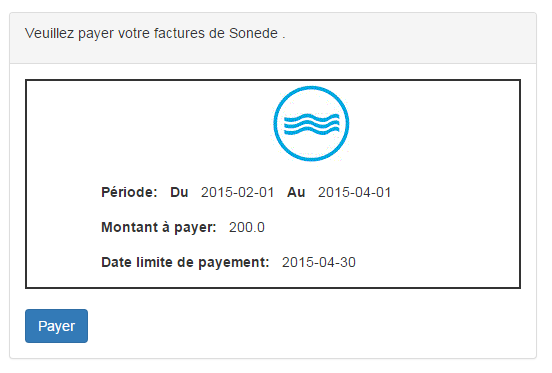


Figure  : Interface graphique « Payer une facture »

* 1. **Consulter l’actualité de la bourse**

L’interface graphique suivante permet aux clients de consulter en temps réel l’actualité de la bourse.



Figure  : Interface graphique « Consulter l’actualité de la bourse »

* 1. **Demander l’achat d’actions**

Le client peut accéder à l’interface présentée dans la figure ci-dessous afin de passer une demande d’achat d’actions en bourse et suivre l’état d’avancement de ces demandes.



Figure  : Interface graphique « Demander l’achat d’actions »

* 1. **Répondre aux demandes d’achats d’actions**

Pour répondre à une demande d’achat d’actions en bourse, le conseiller client doit passer par les deux interfaces illustrées dans les deux figures suivantes.

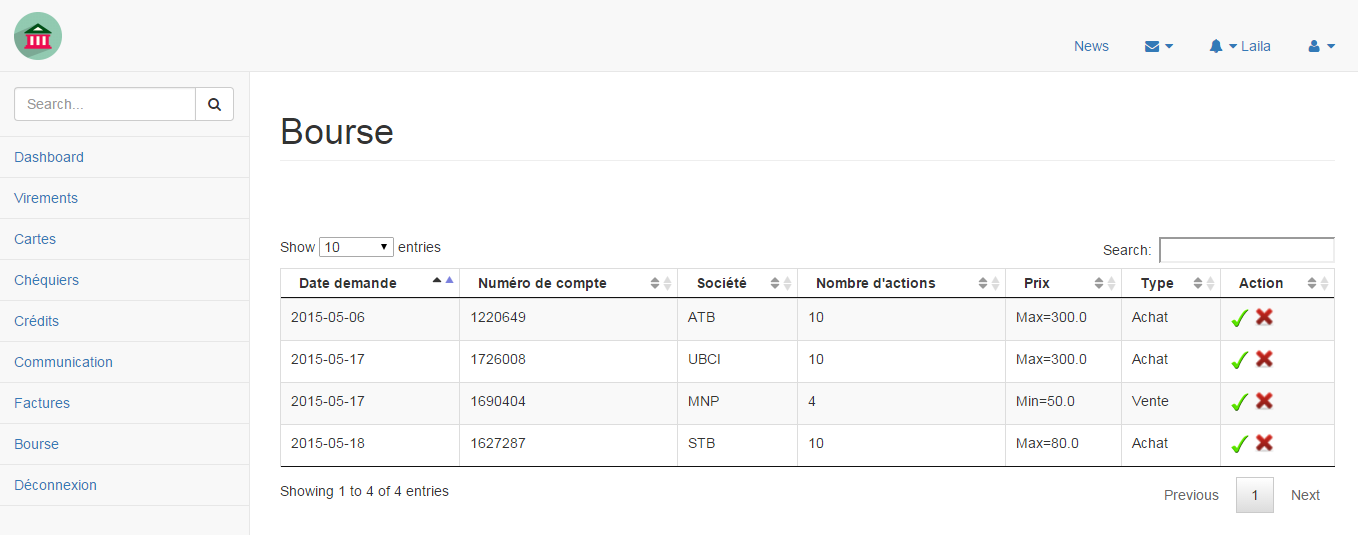


Figure  : Interface graphique « Liste des demandes des actions de bourse »

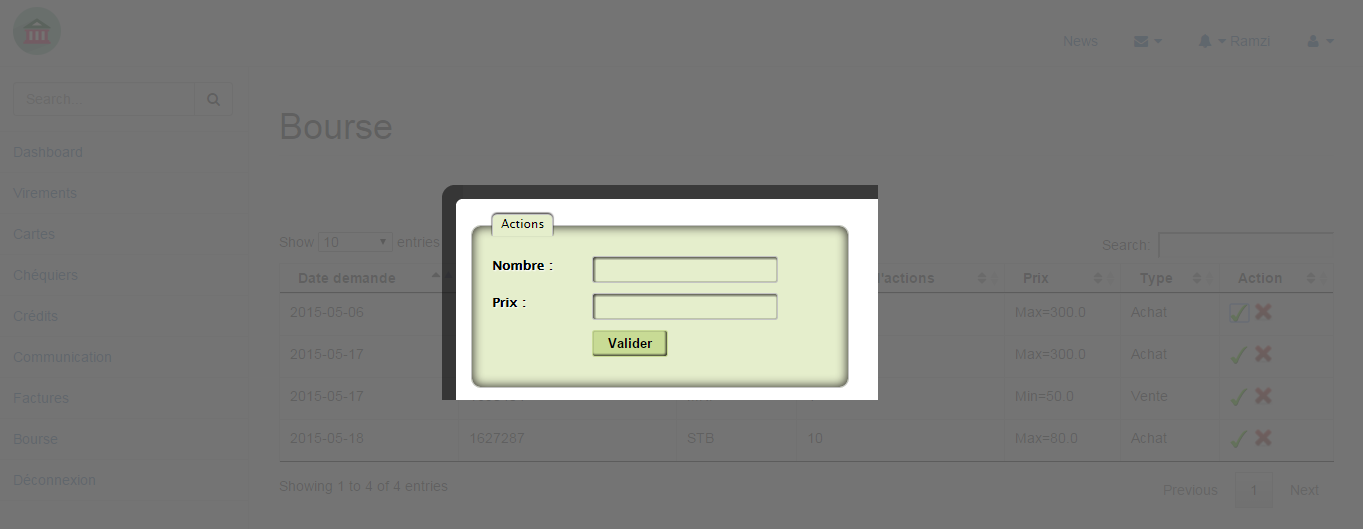


Figure  : Interface graphique « Répondre à la demande d’achat d’actions »

* 1. **Consulter le portefeuille**



Figure  : Interface graphique « Consulter le portefeuille »

* 1. **Consulter le taux de change et convertir une devise**

À travers l’interface représentée dans la figure ci-dessous, le client peut à tout moment connaitre l’appréciation des monnaies et effectuer une opération de conversion de devise.



Figure  : Interface graphique « Consulter taux de change et convertir une devise »

# Diagramme de Gantt

Dans cette section, nous allons présenter notre démarche de réalisation du projet. En fait, d'après cette figure, nous précisions les étapes qui ont été suivies afin d'atteindre les objectifs de notre projet.

Figure  : Diagramme de Gantt

# Conclusion

Au cours de ce chapitre, nous avons exposé le quatrième sprint à travers les étapes d’analyse, de conception et de réalisation.

Ce quatrième et dernier chapitre achevé, nous passons à la conclusion générale.

# Conclusion générale

Au cours de ce projet nous avons eu comme objectif de développer une solution de E-Banking générique pour le compte de la société BFI.

À cet effet, nous avons essayé d’appliquer la méthodologie SCRUM ainsi que la notation UML pour la gestion, la spécification et la conception de ce projet. Il est clair que notre première expérience dans l’utilisation de SCRUM reste à perfectionner vu que nous n’avions pas assez de connaissances sur le métier bancaire ; ce qui a rendu les tâches d’estimation et de planification un peu délicates.

Par ailleurs, sur le plan architectural nous avons utilisé le style MVC ; ce qui nous a permis d’avoir une séparation complète entre les composants du projet.

L’utilisation des frameworks JEE (Spring/ hibernate) ainsi que les technologies web (Bootstrap, Ajax, JQuery, Javascript et Responsive webdesign) a permis de consolider notre expertise en développement.

Notre principal apport réside dans la généricité de notre application. En effet, en entend par généricité le concept de minimiser le changement des systèmes d’information déjà existants dans les banques clientes afin d’assurer que notre E-Banking est adaptable pour toute nouvelle installation.

Néanmoins, notre solution se limite à un prototype qu’il serait judicieux de tester sur un système bancaire réel; ce qui offre de vraies conditions d’exécution. Une telle mise en œuvre nous permettra d’évaluer le fonctionnement de notre solution selon plusieurs critères, tels que la capacité de montée en charge, la performance d’exécution de nombreuses transactions simultanées et la sécurité du traitement des informations qui sont particulièrement sensibles et critiques dans le domaine des systèmes bancaires.

# Bibliographique

[1] BFI. Solution informatique pour banque et finance. Disponible sur : <http://www.bfigroupe.com/Fr/presentation-de-bfi_11_5>. Consulté le 02-02-2015.

[2] Capterra, Top Banking Systems Software Products. Disponible sur: <http://www.capterra.com/banking-systems-software/>. Consulté le 03-02-2015.

[3] NexorONE. The Oneline Banking Software Solution. Disponible sur: <https://www.nexorone.com> . Consulté le 05-02-2015

[4]Banking Software. International Financial Systems. Disponible sur : <http://www.banking-software.com/>. Consulté le 06-02-2015

[5] Canopus Innovative Technologies. E-pay Suite. Disponible sur : <http://canopuslab.com/products/canopus-epaysuite/> . Consulté le 07-02-2015

[6] Axcess Canada. ARMnet. Diponible sur : <http://www.axcesscanada.com/WhyARMnet.html> . Consulté le 08-02-2015

[7] Cobis smf solutions. COBIS – Banking Software. Disponible sur : <http://www.cobismfisolutions.com/> . Consulté le 09-02-2015

[8] L’agiliste. Guide de démarrage Scrum. Juin 2013. Disponible sur : <http://www.agiliste.fr/fiches/guide-demarrage-scrum/> . Consulté le 19-02-2015

[9] Pierre PEZZIARD, « Agile de A à Z ». Edition eBook, 2011.

[10] Gary Mak, « Spring par l’exemple ». Pearson,2008.

[11] Amélie Lambert. Java et Objet. Disponible sur : <http://cedric.cnam.fr/~lamberta/CNAM/Java/cours/jsp.pdf> . Consulté le 20-02-2015

[12] Comment ça marche. .Javascript - Introduction au langage Javascript. Disponible sur : <http://www.commentcamarche.net/contents/577-javascript-introduction-au-langage-javascript>. Consulté le 25-05-2015

[13] !nfoWebMaster. JQuery. Disponible sur: <http://glossaire.infowebmaster.fr/jquery/> . Consulté le 25-05-2015

[14] Amoks. Ajax. Disponible sur : <http://www.amoks.com/rep-lexique/ido-192/ajax.html>. Consulté le 05-03-2015

[15] OPENCLASSROOMS. Prenez en main Bootstrap. Disponible sur : <http://openclassrooms.com/courses/prenez-en-main-bootstrap/mise-en-route-8> . Consulté le 26-02-2015

[16] OPENCLASSROOMS. Les services web. Disponible sur : <http://openclassrooms.com/courses/les-services-web> . Consulté le 25-02-2015

[17] asacréations. C’est quoi Responsive Web Design. Disponible sur: <http://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html> . Consulté le Consulté le 26-02-2015

[18] Oracle. A propos d’Oracle. Disponible sur: <http://www.oracle.com/fr/corporate/index.html> . Consulté le 26-02-2015

[19] Sparx systems. Exceptional High-End Modeling Power. Disponible sur : <http://www.sparxsystems.com/products/ea/index.html> . Consulté le 14-02-2015

[20] OPENCLASSROOMS. Mieux structurer son code :Le Parttern MVC0 Disponible sur : <http://openclassrooms.com/courses/apprenez-a-programmer-en-java/mieux-structurer-son-code-le-pattern-mvc> . Consulté le 25-02-2015

# Annexe

1. **Digrammes des cas d’utilisation**
   1. **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  : Diagramme de cas d’utilisation détaillé «Gérer les chéquiers»

* 1. **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  : Diagramme de cas d’utilisation détaillé «Établir des Statistiques»

* 1. **Cas d’utilisation «Personnaliser l’interface»**

Ce cas d’utilisation donne la possibilité à l’Administrateur de changer le logo de son E-Banking.



Figure  : Diagramme de cas d’utilisation détaillé «Personnaliser l’interface»

* 1. **Cas d’utilisation «Gérer les nouveautés»**

La figure ci-dessous représente le diagramme des cas d’utilisation relatif à la gestion des nouveautés. À travers ce module l’administrateur peut ajouter les nouveautés de sa banque qui seront consultées par les utilisateurs non connectés ou ceux connectés.



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

1. **Diagrammes des séquences détaillés**
   1. **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  : Diagramme de séquence détaillé « Consulter les comptes »

* 1. **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  : Diagramme de séquence détaillé « Demander un virement »

* 1. **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  : Diagramme de séquence détaillé « Suivre les demandes de virement »

1. **Réalisation**
   1. **Interface graphique de la User story « Statistiques globales »**

L’E-Banking offre la possibilité à l’administrateur de consulter les statistiques des différentes opérations bancaires effectuées pendant les trois dernières années comme le montre la figure suivante.

****

Figure  : Interface graphique « Statistiques globales »

* 1. **Interface graphique de la User story  « Répondre à la demande de virement »**

Si un client passe une demande de virement, cette dernière sera affichée au niveau de l’administrateur tel qu’indiqué dans la figure suivante. L’administrateur peut accepter ou refuser cette demande en cliquant sur le bouton approprié.

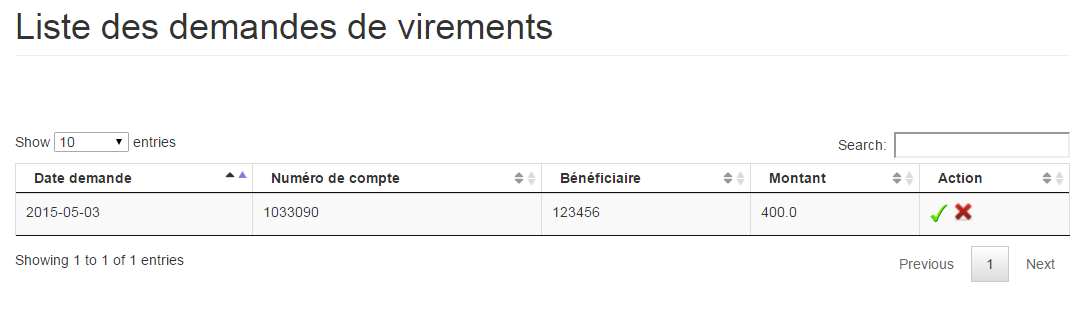


Figure  : Interface graphique « Répondre à la demande de virement »

* 1. **Recevoir les notifications des clients**

À travers l’interface présentée dans la figure suivante, le conseiller consulte toutes les demandes de ses clients.

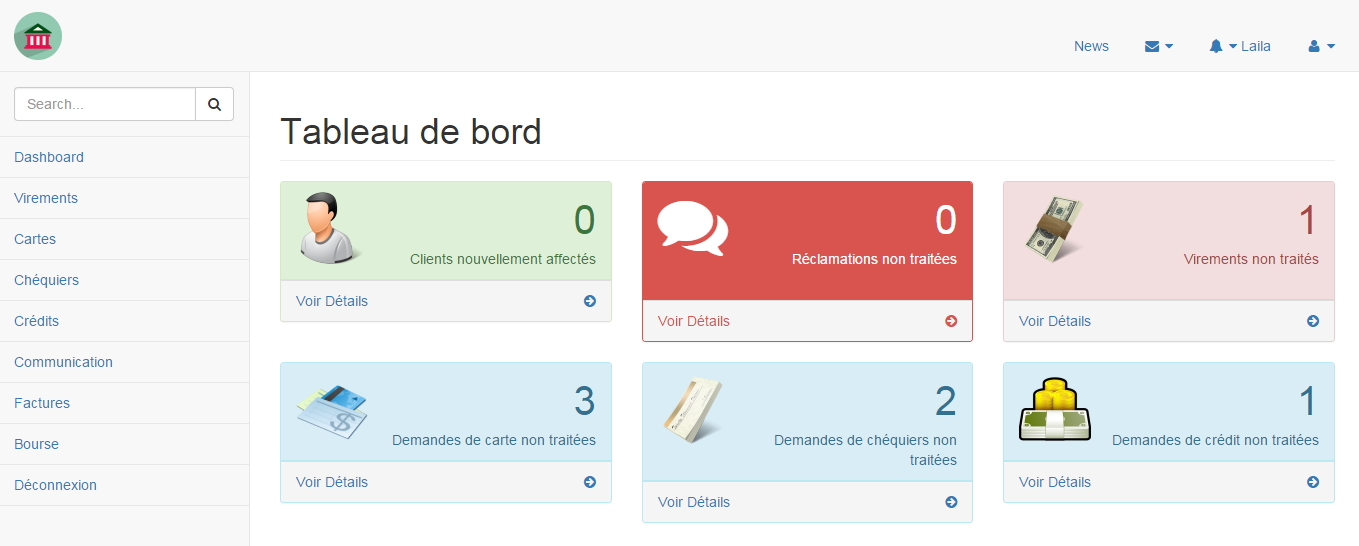


Figure : Recevoir les notifications des clients