



Réf : Ing-GI-2016-31

Rapport de Projet de Fin d'Études

Pour obtenir le

Diplôme d'Ingénieur en Génie Informatique

Option : Systèmes, Réseaux et Sécurité

Présenté et soutenu publiquement le 27 septembre 2016

Par

Asma BENABDERRAHMEN

Conception et réalisation d'une application web et mobile de gestion des opérations bancaires «Attijari Mobile Banking»

Composition du jury

Madame BEN AZZOUZ Lamia Présidente

Monsieur GARALI Wajdi Rapporteur

Monsieur AMLOUK Mohamed Hedi Encadrant Entreprise

Monsieur GHORBEL Khaled Encadrant ENSIT

Dédicaces

Je dédie ce travail à mes très chers parents ; à mon père qui était toujours à mes côtés à tout moment, à ma mère qui ne cesse jamais de m'encourager.

Que dieu les protège et que ce modeste travail, soit l'exaucement de vos veux tant formulés et de vos prières quotidiennes.

A mon frère et mes sœurs, Mohamed Aziz, Marwa et Oumaima pour leur amour et leur soutien inconditionnel,

A toute ma famille,

Et à tous mes amis.

Remerciements

J'exprime mes sentiments de gratitude à de nombreuses personnes, ceux qui nous ont permis d'avoir cette opportunité de travailler sur ce projet.

Tout d'abord, je remercie mes encadrants M. Khaled GHORBEL et M. Mohamed Hedi AMLOUK de leur aide et leur participation dans chaque étape de ce processus. Je suis reconnaissant aux membres du jury qui ont été assez amical d'évaluer mon travail.

Je veux aussi exprimer mon appréciation à mon école, ENSIT, pour m'avoir fourni un paisible environnement dans lequel j'ai pu travailler dans les meilleures conditions.

Et finalement, je remercie ma famille et mes amis pour leur soutien moral. Sans vos encouragements, je n'aurais pas pu être là.

Grand Merci à Tous...

Sommaire

Introduction générale	1
Chapitre 1 : Etude préalable.....	2
Introduction.....	2
1. Présentation de l'organisme d'accueil	2
1.1. Présentation générale d'Attijari bank	2
1.2. Organigramme de la direction de Système Informatique.....	3
2. Étude et critique de l'existant	4
2.1. Analyse de l'existant.....	4
2.2. Critique de l'existant :	4
3. Solution proposée	5
3.1. Fonctionnalités à mettre en place.....	5
4. Méthodologie de conception	6
4.1. Langages de modélisation	6
4.2. Scrum	6
4.2.1. Equipe et rôle.....	8
4.2.2. Terminologie de Scrum.....	8
Conclusion	9
Chapitre 2 : Sprint 0	10
Introduction.....	10
1. Capture des besoins	10
1.1. Identification des acteurs	10
1.2. Les besoins fonctionnels.....	11
1.3. Les besoins non fonctionnels	12
1.4. Le diagramme de cas d'utilisation global	12
2. Pilotage du projet avec Scrum.....	13

2.1. Équipe et rôle	13
2.2. Le Backlog de produit	14
2.3. Planification des Releases	17
3. Conception générale	18
3.1. Diagramme de classe.....	18
3.2. Architecture général de l'application Mobile front office	19
3.3. Pattern et style architectural de l'application Back Office.....	20
3.3.1. Style architectural.....	20
3.3.2. Pattern architectural.....	21
4. Environnement technique du travail.....	22
4.1. Outil matériel.....	22
4.2. Outil de conception	22
4.3. Outils de développement	22
5. Langage de programmation	24
6. Plateforme et Framework utilisé.....	25
Conclusion	26
Chapitre 3 : Release 1	27
Introduction.....	27
1. Sprint 1 : Authentification	27
1.1. Le Backlog de sprint 1.....	27
1.2. Spécification fonctionnelle	28
1.2.1. Diagramme de Cas d'utilisation de sprint 1.....	28
1.2.2. Description textuelle du cas d'utilisation	28
1.3. Conception	29
1.3.1. Diagramme de classe participante	29
1.3.2. Diagramme de séquence	30
1.4. Test	32

2. Sprint 2: Accès aux services client	33
2.1. Le Backlog de Sprint 2	33
2.2. Spécification fonctionnelle	34
2.2.1. Diagramme de Cas d'utilisation de sprint 2	34
2.2.2. Description détaillée de cas d'utilisation	35
2.3. Conception	39
2.3.1. Diagramme de classe participante	39
2.3.2. Diagramme de séquence	41
2.4. Test	42
3. Sprint 3: Accès aux services publics	45
3.1. Le Backlog du sprint	45
3.2. Spécification fonctionnelle	46
3.2.1. Diagramme de cas d'utilisation de sprint 3	46
3.2.2. Description textuelle détaillée	47
3.3. Conception	49
3.3.1. Diagramme de classe participante	50
3.3.2. Diagramme de séquence	50
3.4. Test	52
Conclusion	55
Chapitre 4 : Release 2	56
Introduction	56
1. Sprint 1 : Gérer les agences, les comptes et les abonnés.	56
1.1. Le Backlog du sprint 1	56
1.2. Spécification fonctionnelle	57
1.2.1. Diagramme de Cas d'utilisation	57
1.2.2. Description textuelle détaillée	57
1.3. Conception	58

1.3.1. Diagramme de classe participante	58
1.3.2. Diagramme de séquence	59
1.4. Test	61
2. Sprint 2 : Gérer les virements, Gérer les cartes, Gérer les chéquiers, Gérer les contacts	62
2.1. Le Backlog du sprint 2.....	62
2.2. Analyse et Spécification.....	64
2.2.1. Diagramme de cas d'utilisation global de sprint 2	64
2.2.2. Description textuelle détaillée.....	65
2.3. Conception	65
2.3.1. Diagramme de classe participante	66
2.3.2. Diagramme de séquence	66
2.4. Test	68
Conclusion	70
Conclusion général.....	71
Webographie	72

Table des figures

Figure 1. Logo Attijari Bank	2
Figure 2. Organigramme de la direction SSI.....	3
Figure 3. Attijari Real time.....	4
Figure 4. Le processus Scrum	6
Figure 5. Comparaison des méthodologies Agiles	7
Figure 6. Diagramme de cas d'utilisation globale	13
Figure 7. Planification des Releases	18
Figure 8. Diagramme de classe.....	19
Figure 9. Architecture 3- tiers de l'application mobile.....	20
Figure 10. Niveaux de l'architecture 3 tiers	21
Figure 11. Couche MVC	22
Figure 12. Diagramme de cas d'utilisation de sprint 1.....	28
Figure 13. Description détaillée du cas d'utilisation «s'authentifier ».....	29
Figure 14. Diagramme de classe de cas d'utilisation «S'authentifier ».....	30
Figure 15. Diagramme de séquence de cas d'utilisation « S'authentifier »	30
Figure 16 : diagramme de séquence «Modifier mot de passe».....	31
Figure 17. Interface d'authentification	32
Figure 18. Interface de modification du mot de passe	33
Figure 19. Diagramme de Cas d'utilisation de Sprint 2	35
Figure 20. Diagramme de classe participante de Cas d'utilisation	40
Figure 21. Diagramme de classe participante de cas d'utilisation	40
Figure 22. Diagramme de séquences du cas d'utilisation «Consulter les comptes»	41
Figure 23. Diagramme de séquences du cas d'utilisation «Demander une carte».	42
Figure 24. Interface espace client	42
Figure 25.Interface de consultation des comptes.....	43
Figure 26. Interface de mouvement.....	43
Figure 27. Interface demande une carte.....	44
Figure 28. Interface de virement.....	44
Figure 29. Diagramme de cas d'utilisation globale du Sprint 3	46
Figure 30. Diagramme de classe participante du cas d'utilisation «Suivre les cours de devise» ..	50

Figure 31. Diagramme de classe du cas d'utilisation «Prendre un rendez-vous»	50
Figure 32. Diagramme de séquences du cas d'utilisation «Suivre les cours de devise».....	51
Figure 33. Diagramme de séquences du cas d'utilisation «Prendre un rendez-vous»	52
Figure 34. Espace public	53
Figure 35. Interface de cours de devise.	53
Figure 36. Interface de simulation de crédit.	54
Figure 37. Interface de consultation des agences	54
Figure 38. Interface de consultation des agences sur carte.....	55
Figure 39. Interface de demande de rendez-vous	55
Figure 40. Diagramme de cas d'utilisation global du Sprint 1.....	57
Figure 41. Diagramme de classe participante du cas d'utilisation «Gérer les comptes».	59
Figure 42. Diagramme de séquence du cas d'utilisation «Gérer les comptes»	59
Figure 43. Diagramme de séquence du cas d'utilisation «Supprimer un compte»	60
Figure 44. Diagramme de séquence du cas d'utilisation «Modifier un compte»	60
Figure 45. Interface de gestion des comptes bancaires.....	61
Figure 46. Interface d'ajout d'un compte bancaire.	61
Figure 47. Interface de modification d'un compte bancaire.....	62
Figure 48. Diagramme de cas d'utilisation globale de sprint 2	64
Figure 49. Diagramme de classe participante du cas d'utilisation «Gérer les demandes d'obtention d'une carte ».....	66
Figure 50. Diagramme de classe participante du cas d'utilisation «Gérer les rendez-vous »	66
Figure 51. Diagramme de séquence du cas d'utilisation généré «Gérer les demandes d'obtention d'une cartes ».....	67
Figure 52. Le diagramme de séquence du cas d'utilisation généré «Gérer les rendz-vous ».....	68
Figure 53. Interface de gestion d'obtention d'une carte	69
Figure 54. Interface demande de carte	69
Figure 55. Interface gestion des rendez-vous	70
Figure 56. Interface d'un rendez-vous sélectionné.....	70
Figure 57. Validation de rendez-vous.....	70

Table des tableaux

Tableau 1. Identification d'équipe et rôle Scrum	14
Tableau 2. Le Backlog de produit	17
Tableau 3. Backlog du sprint 1	27
Tableau 4. Backlog de Sprint 2	34
Tableau 5. Description détaillée de cas d'utilisation « Consulter les comptes »	36
Tableau 6. Description détaillée de cas d'utilisation « Demander un virement »	36
Tableau 7. Description détaillée de cas d'utilisation « Demander une carte »	37
Tableau 8. Description détaillée de cas d'utilisation « Demander un chéquier »	37
Tableau 9. Description détaillée de cas d'utilisation « Opposer un chéquier»	38
Tableau 10. Description détaillée de cas d'utilisation «Opposer une carte»	39
Tableau 11. Description détaillée de cas d'utilisation « Suivre les demandes »	39
Tableau 12. Backlog du sprint 3	45
Tableau 13. Description détaillée de cas d'utilisation «Suivre les cours de devise »	47
Tableau 14. Description détaillée de cas d'utilisation «Simuler un crédit».....	47
Tableau 15. Description détaillée de cas d'utilisation «Contacter la banque».....	48
Tableau 16. Description détaillée de cas d'utilisation «Prendre un rendez-vous».....	48
Tableau 17. Description détaillée de cas d'utilisation «Consulter le système de géolocalisation»	49
Tableau 18. Description détaillée de cas d'utilisation «Consulter la liste de point de contact » ..	49
Tableau 19. Backlog du sprint 1	57
Tableau 20. Description détaillée du cas d'utilisation «Gérer les comptes».....	58
Tableau 22. Backlog du sprint 2	63
Tableau 23. Description détaillée du cas d'utilisation «Gérer les demandes d'obtention d'une carte ».....	65
Tableau 24. Description détaillée du cas d'utilisation «Gérer les rendez-vous»	65

Introduction générale

Vu le marché compétitif, les banques aspirent toujours à améliorer et consolider ses relations avec leurs clients en offrant un ensemble des services et répondre rapidement à leurs besoins. En effet, Les banques proposent un nombre croissant des services délivrés en ligne ce qu'on appelle La banque électronique (e-Banking en anglais) .Mais dans un environnement concurrentiel, un avantage compétitif est de bénéficier de l'information et du service voulu, au moment et à l'endroit voulus. La réponse : Le **Mobile Banking** ou **m-Banking** qui constitue une solution idéale à cette situation.

Le mobile Banking est l'utilisation du téléphone portable pour fournir des services financiers qui peuvent être des transactions d'argent ou des échanges d'informations entre le client et les institutions financières. Il est basé sur l'idée d'utiliser en micro finance un outil de communication, téléphone portable, qui s'est très fortement répandu ces dernières années, et ceci afin de diversifier et d'améliorer l'offre de services auprès de la clientèle actuelle.

C'est dans ce cadre s'inscrit notre projet de fin d'études au sein d'Attijari bank. En effet, le but de ce projet est de réaliser une application web et mobile de gestion des opérations bancaires qui permet les échanges des données avec le système d'information de la Banque afin de réaliser les opérations bancaires envisagées par ce service.

Afin de bien présenter notre rapport, nous avons choisi de le découper en quatre chapitres. Le premier chapitre présente une étude préalable décrivant le cadre général du projet qui consiste à la présentation de l'organisme d'accueil, l'étude de l'existant et les choix méthodologiques et techniques. Le deuxième chapitre « planning et architecture » qui consiste la première phase dans le cycle Scrum ainsi que le Backlog produit mais aussi l'analyse des besoins et l'architecture de l'application. Le troisième chapitre comporte notre premier release. Nous enchaînerons ensuite avec le deuxième release lors du quatrième chapitre.

Nous terminons ce rapport par une conclusion général en proposant des nouvelles perspectives d'amélioration et d'optimisation de notre application.

Chapitre 1 : Etude préalable

Introduction

L'étude préalable est une étape primordiale pour la conception de tout système d'information. Dans ce chapitre, nous présenterons le contexte général du projet qui sera décliné en deux parties : la première partie présentera la société d'accueil Attijari Banque au sein duquel nous avons effectué notre projet, et la seconde partie décrira le contexte, la problématique, l'objectif attendu du projet ainsi que les fonctionnalités à mettre en place et le choix méthodologique et technique.

1. Présentation de l'organisme d'accueil

1.1. Présentation générale d'Attijari bank



Figure 1. Logo Attijari Bank

Attijari bank est une banque tunisienne fondée par Sadok Ben Jemâa sous l'appellation de Banque du Sud .Elle possède le premier réseau bancaire en Tunisie, 201 agences et centres d'affaires. En 2015, la banque a été élue, pour la deuxième année consécutive, « La banque de l'année en Tunisie» « Bank of The Year - Tunisia » par le prestigieux magazine The Banker Elle est fortement présentée sur l'ensemble des segments de marchés particuliers, professionnels, entreprises et opère, à travers des filiales spécialisées, dans tous les métiers de la Finance : Leasing, Gestion d'actifs, Intermédiation boursière, Capital Risque, Conseil, Assurance, ...

A travers de la mise en place de son projet de développement, Attijari bank contribue d'une façon active à la performance du système bancaire tunisien et au développement du pays.

La banque tunisienne Attijari bank est filiale du Groupe Attijariwafa bank, 1ier groupe bancaire et financier du Maghreb présent dans 24 pays : en Afrique, en Europe et en Asie et compte 7,9 millions de clients et 3534 agences. Grâce à l'appui du Groupe, Attijari bank Tunisie accompagne ses clients dans leur développement à l'international et notamment en Afrique [1].

1.2. Organigramme de la direction de Système Informatique

A travers son projet de développement, la banque tunisienne Attijari bank a mis en place une organisation axée sur une spécialisation en Business Units afin de permettre aux différentes lignes métiers de la banque de se focaliser sur ses objectifs et cela notamment à travers :

- ❖ La séparation et la filialisation des activités.
- ❖ La spécialisation front-back : séparation entre les services qui initient les opérations et ceux qui sont en charge de leur comptabilisation.
- ❖ L'obtention du statut de Banque d'Affaires pour l'activité de conseil (accès aux appels d'offres publics).
- ❖ La refonte du système d'information avec la mise en place sur 3 ans d'un système de global banking répondant aux besoins de la banque et aux exigences des nouveaux standards de la profession.
- ❖ La mise en place des principes de bonne gouvernance. La banque a mis en place très rapidement les structures recommandées par les standards internationaux en la matière.

Cette figure illustre la structure du service informatique :

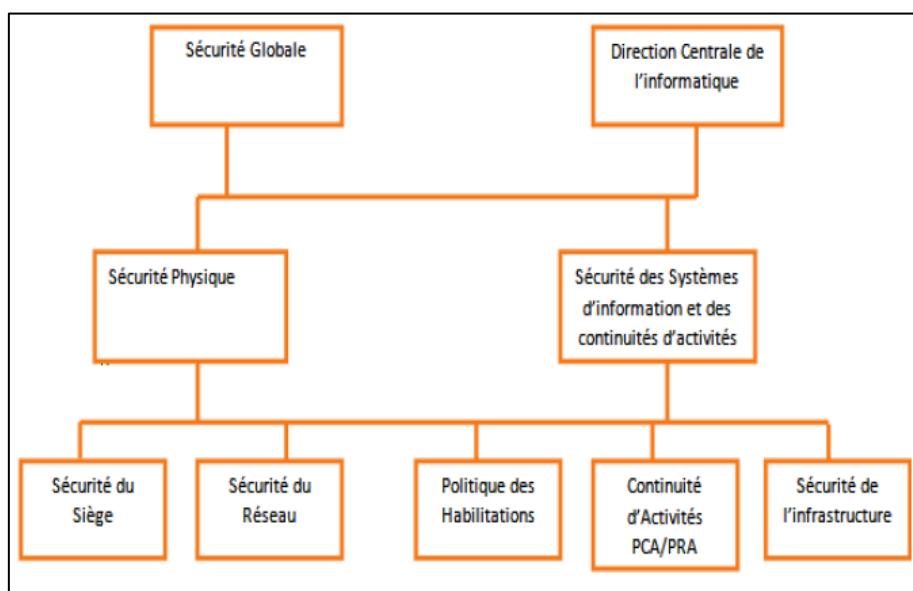


Figure 2. Organigramme de la direction SSI

2. Étude et critique de l'existant

2.1. Analyse de l'existant

L'analyse de l'existant consiste à présenter les canaux diversifiés d'Attijari Bank qui offre à ses clients plusieurs services tels que les opérations de virement compte à compte, la consultation de solde, etc. En effet, le client d'Attijari Bank peut profiter de ses services à travers deux modes à concevoir : le mode classique et le mode à distance.

- ❖ **Le mode classique :** Attijari Bank met à la disposition de ses clients des agences pour servir leurs besoins. Pour se bénéficier de ses services, le client se trouve souvent obligé de se déplacer vers l'agence.
- ❖ **Le mode à distance :** Attijari Bank met à la disposition de ses clients un service électronique E-Banking (Attijari Real time) pour suivre des comptes en temps réel, initier des virements et consulter des avis d'opérations et autres documents 7j/7 et 24h/24. Pour se bénéficier de cette offre, le client doit passer par le site web d'Attijari Bank: <https://www.attijarirealtime.com.tn>, se connecter sur le site et choisir le service désiré, la figure 3 suivante représente l'interface du site web Attijari Bank:

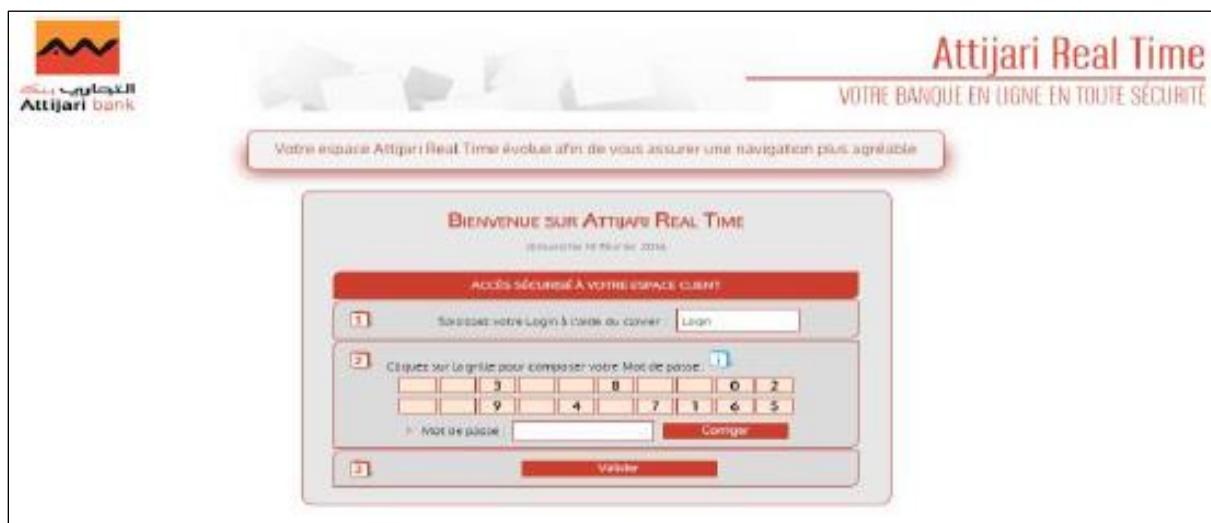


Figure 3. Attijari Real time

2.2. Critique de l'existant :

Après L'analyse et l'étude de deux types de services offerts par la banque, nous passons au critique et à la description des inconvénients présentés par ces deux modes:

- ❖ **Services classiques:** Malgré l'existence de 200 agences qui sont misent à la disposition des clients par Attijari Bank afin de se servir confortablement, mais vu

l'augmentation de l'inscription au sein de la banque, le client est appelé à faire la queue et attendre son tour.

- ❖ **Services à distance:** Sans doute, cet offre peut simplifier la vie et facilite les relations avec notre banque. Mais il exige toujours l'existence d'un ordinateur et d'une connexion internet.

3. Solution proposée

Après avoir examiné la situation actuelle, on va essayer de proposer une application mobile sous Android qui traite les différents services bancaires et qui propose une solution facilement accessible et qui sera soutenu par une application Web assurant et facilitant son administration afin d'être la plus pertinente possible.

3.1. Fonctionnalités à mettre en place

Les fonctionnalités de notre application Mobile Banking sont divisées en deux modules : un premier dédié au Front Office et un second pour le Back Office.

❖ Module1 : Front Office

Une application Android qui sera installée sur un terminal mobile développé en langage de programmation JAVA permet de:

- ✓ Consulter les comptes et leurs mouvements.
- ✓ Demander et/ou opposer un chéquier.
- ✓ Demander et/ou opposer une carte bancaire.
- ✓ Effectuer des virements.
- ✓ Consulter la boîte de réception et envoyer des messages.
- ✓ Suivre les demandes
- ✓ Trouver et lister les agences et les DAB.
- ✓ Consulter les cours des devises.
- ✓ Simuler un crédit.
- ✓ Contacter la banque.
- ✓ demander un rendez-vous.

❖ Module2 : Back Office

C'est une plateforme web J2EE pour l'administration de l'application mobile. En effet, elle permet la mise à jour de la base de données et de l'application mobile. La gestion des demandes clients sont les principales fonctionnalités de cette application web.

4. Méthodologie de conception

4.1. Langages de modélisation

Nous utiliserons, pour la spécification et la conception de ce travail, le langage UML (Unified Modeling Language), c'est un langage de modélisation qui permet de représenter les différentes vues du système.

Ce langage est utilisé dans les projets logiciels et offre une multitude de schéma, appelés des diagrammes permettant de présenter le logiciel à développer

4.2. Scrum

Pour la gestion de notre projet, nous avons recours aux méthodes agiles et plus précisément la méthodologie Scrum. Le terme Scrum (qui signifie «mêlée» en rugby) se rapproche plus d'une gestion de ressources humaines plutôt que d'une réelle méthode de développement. Il s'agit ici de ne pas oublier le côté humain du développement.

Scrum s'appuie sur le découpage d'un projet en boîtes de temps, nommées « sprints ». Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé. Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective. Cette technique analyse le déroulement du sprint achevé, afin d'améliorer ses pratiques. L'adaptation et la réactivité de l'équipe de développement est facilitée par son auto-organisation [2].

La figure suivante illustre la méthode Scrum :

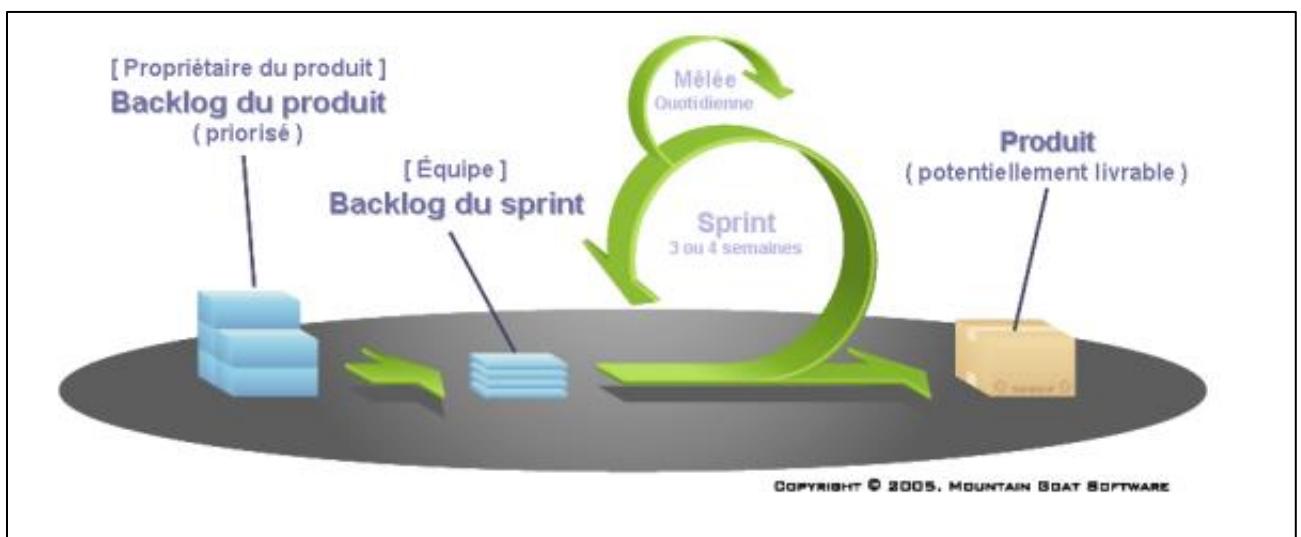


Figure 4. Le processus Scrum

Les principales caractéristiques de Scrum sont :

- ❖ Identifier les changements très tôt.
- ❖ Donner toute confiance aux développeurs et les laisser faire leur travail.
- ❖ Faire des itérations variantes (généralement de 30 jours), appelés aussi «sprints» pour laisser le temps de coder. Chaque itération a un objectif bien précis ou «backlog» et fournit une nouvelle fonctionnalité testée (une démonstration est faite à la fin de chaque sprint).
- ❖ Faire des réunions tous les jours (daily meeting) et chaque semaine (weekly meeting) pour encadrer les équipes et recaler les objectifs.

Pourquoi SCRUM ?

Le choix de Scrum comme une méthodologie de pilotage pour notre projet s'est basé sur les atouts de ce dernier. Il se résume comme suit:

- ❖ Plus de souplesse et de réactivité,
- ❖ La grande capacité d'adaptation au changement grâce à des itérations courtes.
- ❖ Et la chose plus importante, c'est que Scrum rassemble les deux cotés théorique et pratique et se rapproche beaucoup de la réalité.

En effet, ci-dessous une figure de comparaison aux différentes méthodologies agiles.

Condition	XP	Lean	FDD	Scrum
Petite équipe	✓	✓	✗	✓
Exigences très variables	✓	✓	✓	✓
Equipes distribuées	✗	✓	✓	✓
Systèmes hautement critiques	✗	-	-	-
Clients/Parties prenantes multiples	✗	✓	✓	✓

Figure 5. Comparaison des méthodologies Agiles

4.2.1. Équipe et rôle

SCRUM définit trois rôles, qui sont comme suit :

- ❖ **Le responsable (Scrum master)** : Il doit s'assurer que la méthode SCRUM doit être mise en œuvre correctement, que la démarche de l'équipe de développement s'inscrit dans les pratiques de SCRUM, son rôle est de :
 - ✓ Diriger et aider les membres de l'équipe.
 - ✓ Gérer d'une manière efficace le carnet du produit, afin qu'il soit concis et clair pour l'équipe.
 - ✓ Aider l'équipe à éliminer les obstacles à son progrès
 - ✓ S'assurer que l'équipe est productive.
- ❖ **Le propriétaire du produit (Product owner)** : c'est le responsable du produit, il représente les utilisateurs finaux, son rôle est de :
 - ✓ Définir la liste des fonctionnalités du produit.
 - ✓ Prioriser les fonctionnalités en fonction de leur importance et leur valeur ajoutée pour l'entreprise qu'il présente.
 - ✓ Choisir la date de livraison des versions ainsi que leurs contenus.
 - ✓ Valider les lots livrés avec l'équipe de développement.
 - ✓ Clarifier les besoins à l'équipe de développement si nécessaire.
- ❖ **Équipe de développement (Team)** : L'équipe de développement est constituée de 3 à 9 personnes et a pour responsabilité de livrer à chaque fin d'itération une nouvelle version de l'application enrichie de nouvelles fonctionnalités et respectant le niveau de qualité nécessaire pour être livré.

4.2.2. Terminologie de Scrum

Les artefacts de la méthodologie Scrum sont :

- ❖ **Backlog Product** : Il sert à piloter l'équipe de développement, et comporte une liste ordonnée présentant les exigences, les besoins et les fonctionnalités requises par le client.
- ❖ **Sprint Backlog** : Il contient la liste des histoires utilisateur présentes au sein du Product Backlog qui ont été sélectionnées par l'équipe pour être traitées pendant un sprint donné.
- ❖ **User Story** : Est une exigence du système à développer, formulée en une ou deux phrases dans le langage de l'utilisateur.

- ❖ **Burndown Chart** : est un graphique qui est généré dès le début de chaque sprint, permettant de visualiser la progression de l'équipe.

Conclusion

Dans ce premier chapitre, nous avons commencé par présenter notre organisme d'accueil Attijari Bank. Ensuite, nous avons décrit le contexte du projet et la solution proposée qui constituera le sujet de notre projet. Enfin, après avoir choisi Scrum comme méthodologie de gestion de projet, nous nous dirigerons naturellement vers le Sprint 0.

Chapitre 2 : Sprint 0

Introduction

Dans le chapitre précédent, nous avons choisi d'adopter la méthodologie Scrum pour la conception de notre application. En fait, Scrum est organisé suivant trois phases dont la première est la phase de planification et architecture «appelé aussi sprint 0» qui ne se termine pas nécessairement par une livraison. Cette phase est la plus importante dans le cycle de développement Scrum puisqu'elle influence directement sur la réussite des sprints et en particulier le premier. Au cours de cette phase, il faut préparer et mettre en place tout ce qui est nécessaire au lancement des activités. C'est donc dans ce chapitre que nous allons définir les besoins, identifier les différents acteurs, élaborer le Backlog du produit initial et planifier les sprints.

1. Capture des besoins

1.1. Identification des acteurs

Un acteur est une entité physique (personne) ou abstraite (logiciel) capable d'utiliser le système afin de répondre à un besoin bien défini. Les acteurs de notre application sont :

- ❖ **Client d'Attijari Bank** : Tout client de la banque possède le pack « Mobile Banking » afin de pouvoir utiliser l'application Attijari Mobile Banking. Il s'authentifie afin qu'il puisse accéder à son espace client où il bénéficiera d'un ensemble des fonctionnalités (consultation compte, demande carte ou chéquier, virements ...). Il a également l'accès à un espace public qui regroupe un ensemble des services (trouver une agence ou distributeur, convertisseur de devise...).
- ❖ **Client potentiel d'Attijari Bank** : Toute personne utilise l'application Mobile Banking d'Attijari Bank, qui n'est pas encore client, et qui pourra le devenir. Il a l'accès à un espace public de l'application où il pourra bénéficier d'un ensemble de fonctionnalités tel que la recherche d'une agence, simulation de crédit et contact la banque...
- ❖ **Administrateur Mobile Banking d'Attijari Bank** : Il s'agit de la personne qui possède toutes les permissions sur le système. Il est responsable de la gestion des abonnés Mobile Banking, la gestion des agences, des comptes, des contacts et des demandes clients (demandes d'opposition ou d'obtention de carte ou de chéquier, demandes des virements...)

1.2. Les besoins fonctionnels

Dans cette partie, nous allons dégager les besoins fonctionnels qui doivent correspondre aux attentes des utilisateurs. Nous présentons les besoins fonctionnels de notre application par acteur.

Le client d'Attijari Bank accède à l'espace client de l'application pour :

- ❖ Consulter les comptes : En tant qu'un client Attijari Bank, cette fonction offre le droit de consulter le solde de chaque compte.
- ❖ Consulter les extraits d'un compte : En tant qu'un client Attijari Bank, cette fonction offre le droit de visualiser les opérations réalisées sur un compte.
- ❖ Demander ou opposer une carte bancaire : En tant qu'un client d'Attijari Bank, cette fonction permet de faire toute les opérations liées à la carte bancaire, c'est-à-dire le client peut envoyer une demande, passer une réclamation à une carte portant son identité.
- ❖ Demande ou opposer un chéquier : En tant qu'un client d'Attijari Bank, cette fonction permet d'envoyer une demande ou passer une réclamation à un chéquier.
- ❖ Suivre les demandes : En tant qu'un client d'Attijari Bank, cette fonction permet de suivre les demandes (accepté, refusé).
- ❖ Consulter la boîte de réception et envoyer un message : En tant qu'un client d'Attijari Bank, cette fonction permet d'envoyer un message à la banque pour une réclamation ou une information et de consulter les messages reçus.

Le client potentiel d'Attijari Bank accède à l'espace public de l'application pour :

- ❖ Consulter les devises : En tant qu'utilisateur Mobile Banking, cette opération permet de suivre les cours de devise afin d'être au courant de changement de devise.
- ❖ Simuler un crédit : En tant qu'utilisateur Mobile Banking, le client peut simuler un crédit.
- ❖ Contacter la banque : En tant qu'utilisateur Mobile Banking, le client peut contacter la banque
- ❖ Prendre un rendez-vous : En tant qu'utilisateur Mobile Banking, le client peut demander un rendez-vous
- ❖ Consulter les agences : En tant qu'utilisateur Mobile Banking, le client a la possibilité de trouver les contacts de la banque pour soit annoncer des réclamations, soit poser des questions à propos les différentes services de la banques.

L'administrateur Mobile Banking d'Attiajri Bank accède à l'application back office pour :

- ❖ La gestion des comptes : L'application doit donner la main à l'administrateur pour qu'il puisse gérer les comptes des utilisateurs en ajoutant, supprimant ou modifiant des comptes.
- ❖ La gestion des abonnés : L'application doit donner la main à l'administrateur pour qu'il puisse gérer les abonnés en ajoutant ou modifiant des abonnés.
- ❖ La gestion des chéquiers : L'application doit donner la main à l'administrateur pour qu'il puisse gérer les demandes des chéquiers.
- ❖ La gestion des cartes : L'application doit donner la main à l'administrateur pour qu'il puisse gérer les demandes des cartes.
- ❖ La gestion des contacts : L'application doit donner la main à l'administrateur pour qu'il puisse gérer les demandes de rendez-vous et répond à un message.

1.3. Les besoins non fonctionnels

L'application développée doit satisfaire les besoins non fonctionnels suivants :

- ❖ La sécurité : L'application doit assurer que chaque utilisateur ne peut accéder qu'à son espace et qu'après une authentification pour garantir la confidentialité de l'accès.
- ❖ La performance : L'application doit répondre aux exigences des usagers d'une manière optimale.
- ❖ L'ergonomie : Les interfaces doivent être bien organisées, simples à manipuler puisque l'application sera utilisée par des utilisateurs de différents niveaux de maîtrise de l'outil informatique.
- ❖ L'évolutivité : Le système doit avoir la capacité de s'adapter aux changements et aux futurs besoins de l'entreprise.

1.4. Le diagramme de cas d'utilisation global

Nous avons dégagé précédemment les différents besoins fonctionnels de notre application, nous les présentons maintenant d'une manière informelle en utilisant le diagramme de cas d'utilisation d'UML. La figure 6 présente le diagramme de cas d'utilisation global.

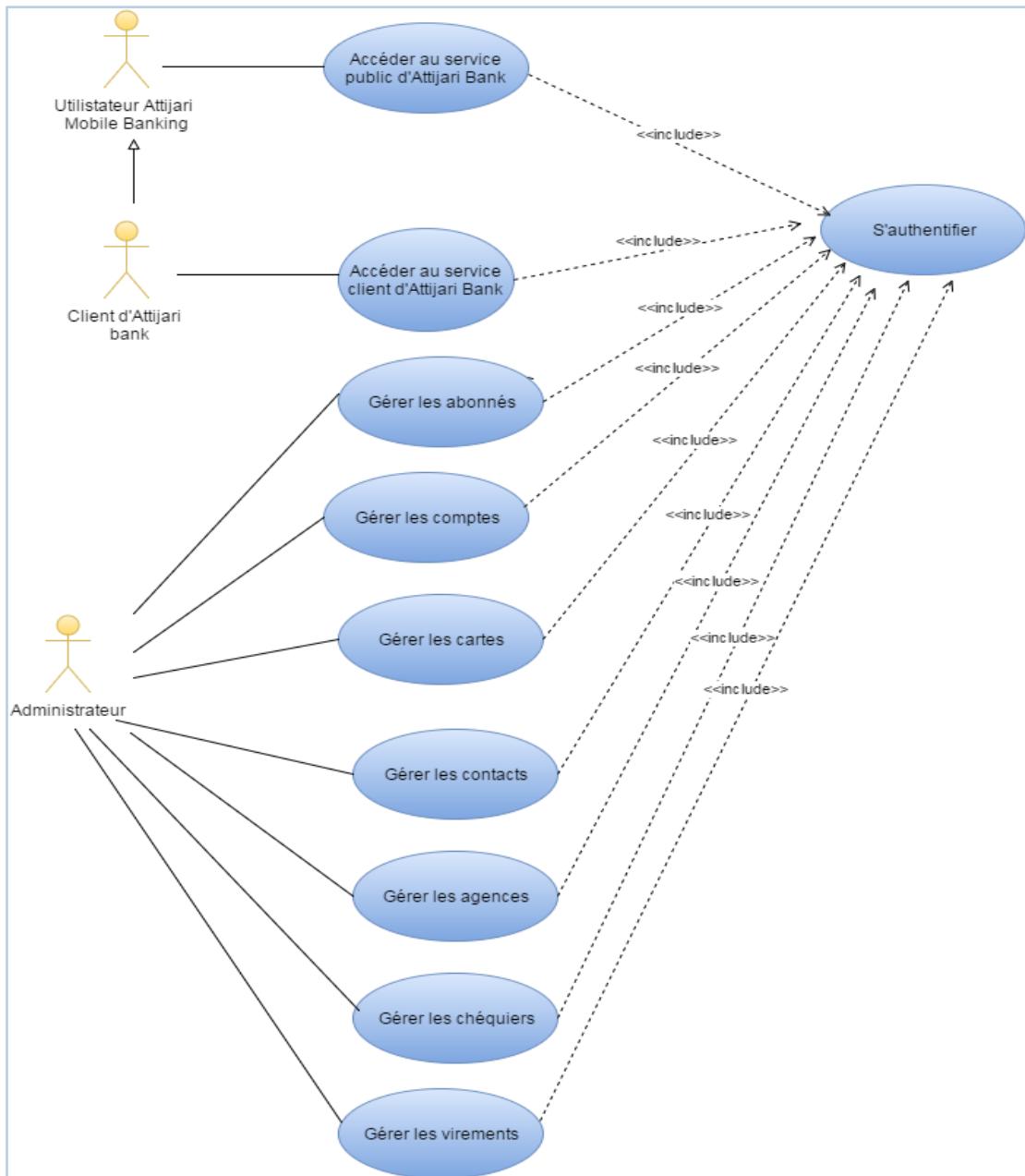


Figure 6. Diagramme de cas d'utilisation globale

2. Pilotage du projet avec Scrum

2.1. Équipe et rôle

Scrum définit trois rôles principaux qui nous avez défini dans le chapitre précédent.

Dans le contexte de notre projet, Monsieur Mohamed Hedi AAMLOUK sera le propriétaire et Monsieur Khaled GHORBEL le directeur de produit et on forme moi-même le seul membre de l'équipe Scrum.

Dans le tableau 1 suivant, nous avons défini les membres de l'équipe Scrum ainsi que le rôle de chacun.

Rôle Scrum	Personne Affectées
Product Owner	M. Mohamed Hedi AMLOUK
Scrum Master	M. Khaled GHORBEL
Team	Asma BENABDERRAHMEN

Tableau 1. Identification d'équipe et rôle Scrum

2.2. Le Backlog de produit

Le Backlog de produit est un artefact très important dans Scrum, il contient l'ensemble des fonctionnalités attendues d'un produit. Chaque fonctionnalité est représentée sous forme d'histoires utilisateur (user story). Chacune de ces Stories doit avoir un but métier, son ajout au Backlog doit généralement justifier une valeur ajoutée.

Dans ce but, nous dressons le Backlog de notre application illustré par le tableau 2 ci-dessous.

Feature	ID	User Story	Priorité
Accéder au service public d'Attijari Bank	1	En tant qu'utilisateur Mobile Banking, je veux suivre les cours de devise afin d'être au courant des changements de devise.	20
	2	En tant qu'utilisateur Mobile Banking, je veux changer un montant afin de connaître le montant converti.	20
	3	En tant qu'utilisateur Mobile Banking je veux prendre un rendez-vous afin de planifier une rencontre avec le chargé clientèle de mon agence	20
	4	En tant qu'utilisateur Mobile Banking, je veux consulter le système de géo-localisation afin de visualiser sur la carte les agences d'Attijari Bank.	20
	5	En tant qu'utilisateur Mobile Banking, je veux consulter la liste des agences afin d'avoir leurs adresses.	20
	6	En tant qu'utilisateur Mobile Banking je veux simuler un crédit afin d'avoir la mensualité de mon crédit	20

	7	En tant qu'utilisateur Mobile Banking je veux appeler le service clientèle d'Attijari bank afin d'être en contact avec le service clientèle de banque.	20
Accéder au service client d'Attijari Bank	8	En tant que client d'Attijari Bank, je veux demander des virements compte à compte ou vers bénéficiaire afin de transférer d'argent.	20
	9	En tant que client d'Attijari Bank, je veux demander une carte afin de faciliter mes échanges quotidiens.	20
	10	En tant que client d'Attijari Bank, je veux un nouveau chéquier afin de faciliter mes échanges quotidiens.	20
	11	En tant que client d'Attijari Bank, je veux consulter mes comptes avec ces informations (solde,...) afin de visualiser les coordonnées de mes comptes.	20
	12	En tant que client, je veux consulter mon extrait de compte afin de visualiser les mouvements du compte	20
	13	En tant que client d'Attijari Bank, je veux opposer ma carte afin de bloquer son utilisation.	20
	14	En tant que client d'Attijari Bank, je veux opposer mon chéquier afin de bloquer son utilisation.	20
	15	En tant que client d'Attijari Bank, je veux envoyer un message à ma banque afin d'envoyer une réclamation ou demande d'une information.	20
	16	En tant qu'administrateur, je veux ajouter un ou plusieurs abonnés afin que mon application soit mise à jour.	10
	17	En tant qu'administrateur, je veux modifier un ou plusieurs abonnés afin que mon application soit MAJ	10
Gérer les comptes	18	En tant qu'administrateur, je veux ajouter un ou plusieurs comptes afin que mon application soit mise à jour.	10
	19	En tant qu'administrateur, je veux modifier un ou plusieurs comptes afin que mon application soit mise à jour.	10

	20	En tant qu'administrateur, je veux supprimer un ou plusieurs comptes afin que mon application soit mise à jour.	10
Gérer les cartes	21	En tant qu'administrateur, je veux accepter la demande d'une carte afin de répondre à la demande de client.	10
	22	En tant qu'administrateur, je veux refuser la demande d'une carte afin de répondre à la demande de client	10
	23	En tant qu'administrateur, je veux supprimer la demande d'une carte afin de mettre à jour les demandes.	10
	24	En tant qu'administrateur, je veux accepter la demande d'opposition d'une carte afin de répondre à la demande de client.	10
	25	En tant qu'administrateur, je veux refuser la demande d'opposition d'une carte afin de répondre à la demande de client.	10
	26	En tant qu'administrateur, je veux supprimer la demande d'opposition d'une carte afin de mettre à jour les demandes.	10
Gérer les chéquiers	27	En tant qu'administrateur, je veux accepter la demande d'un chéquier afin de répondre à la demande de client.	10
	28	En tant qu'administrateur, je veux refuser la demande d'un chéquier afin de répondre à la demande de client.	10
	29	En tant qu'administrateur, je veux supprimer la demande d'un chéquier afin de mettre à jour les demandes.	10
	30	En tant qu'administrateur, je veux accepter la demande d'opposition d'un chéquier afin de répondre à la demande de client.	10
	31	En tant qu'administrateur, je veux refuser la demande d'opposition d'un chéquier afin de répondre à la demande de client.	10
	32	En tant qu'administrateur, je veux supprimer la demande de chéquier afin de mettre à jour les demandes.	10

Gérer les contacts	33	En tant qu'administrateur, je veux valider un rendez-vous afin de planifier un rendez-vous.	10
	34	En tant qu'administrateur, je veux consulter la liste de rendez-vous planifiés afin de vérifier les données.	10
	35	En tant qu'administrateur, je veux consulter la liste des messages afin de répondre aux réclamations des clients.	10
Gérer les virements	36	En tant qu'administrateur, je veux accepter la demande d'un virement afin de répondre à la demande de client.	10
	37	En tant qu'administrateur, je veux refuser la demande d'un virement afin de répondre à la demande de client.	10
Gérer les agences	38	En tant qu'administrateur, je veux ajouter une ou plusieurs agences afin que mon application soit mise à jour.	10
	39	En tant qu'administrateur, je veux modifier une ou plusieurs agences afin que mon application soit mise à jour.	10
S'authentifier	40	En tant que client d'Attijari Bank je veux m'authentifier afin d'accéder à mon espace et bénéficier des fonctionnalités de l'application.	20
	41	En tant que client d'Attijari Bank je veux changer mon mot de passe afin de mieux personnaliser mes paramètres de connexion.	20
	42	En tant qu'administrateur d'Attijari Mobile Banking, je veux m'authentifier afin d'accéder à mon espace.	10

Tableau 2. Le Backlog de produit

2.3. Planification des Releases

Après avoir mis au point l'ensemble des fonctionnalités attendues par notre système, nous devons diviser le travail selon des Releases. Chaque release présente une série de sprints successifs. Dans notre cas, nous avons décidé de découper notre projet en deux releases.

Release 1 : présente la partie back office de notre application, elle est divisée en trois sprints. En effet nous allons commencer par l'authentification dans le premier sprint. Ensuite nous

allons traiter les services clients de notre application mobile dans le deuxième sprint. Et dans le dernier sprint, nous allons traiter les services publics.

Release 2 : présente la partie front office de notre application, il est divisé en deux sprint. En effet nous allons commencer par l'authentification, gérer les abonnés puis gérer les comptes dans le premier sprint. Ensuite nous passons au gérer les demandes et gérer les contacts.

La figure 7 schématisse le découpage en Releases.

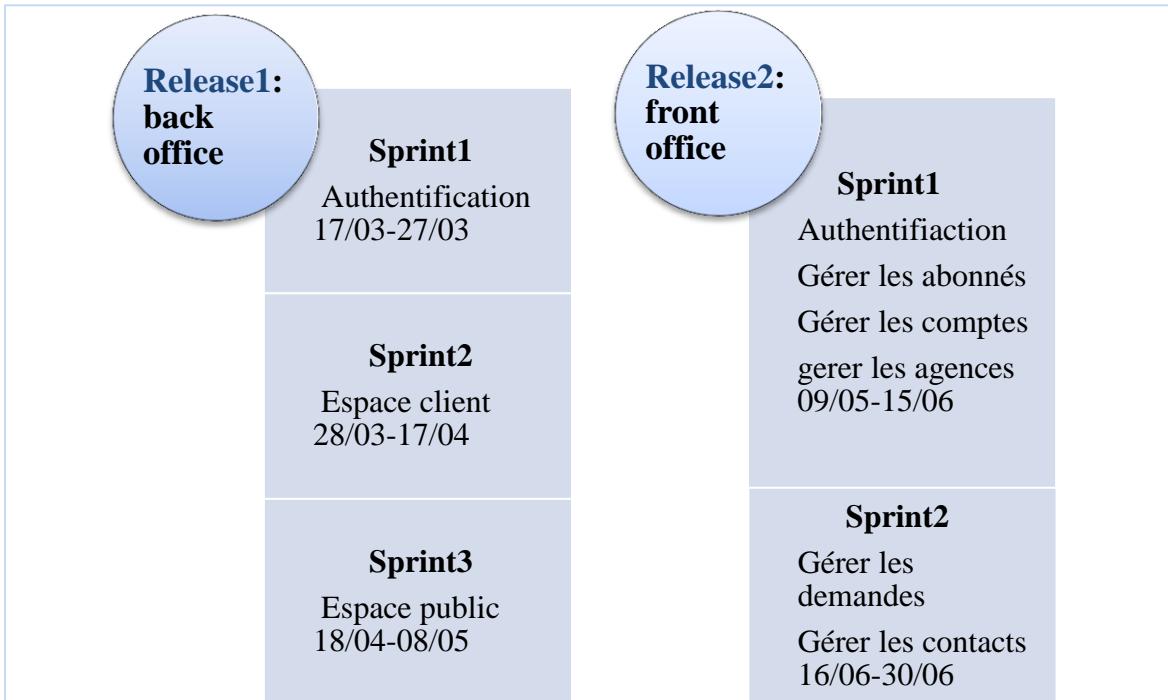


Figure 7. Planification des Releases

3. Conception générale

Cette partie traite des aspects techniques liés à l'implémentation et la mise en œuvre de notre application

3.1. Diagramme de classe

La modélisation statique d'un système consiste à décrire les composantes de ce dernier sans tenir compte de leur évolution dans le temps.

Dans ce qui suit, le diagramme de classe représente les classes d'un système mais également les associations (les relations) entre celles-ci.

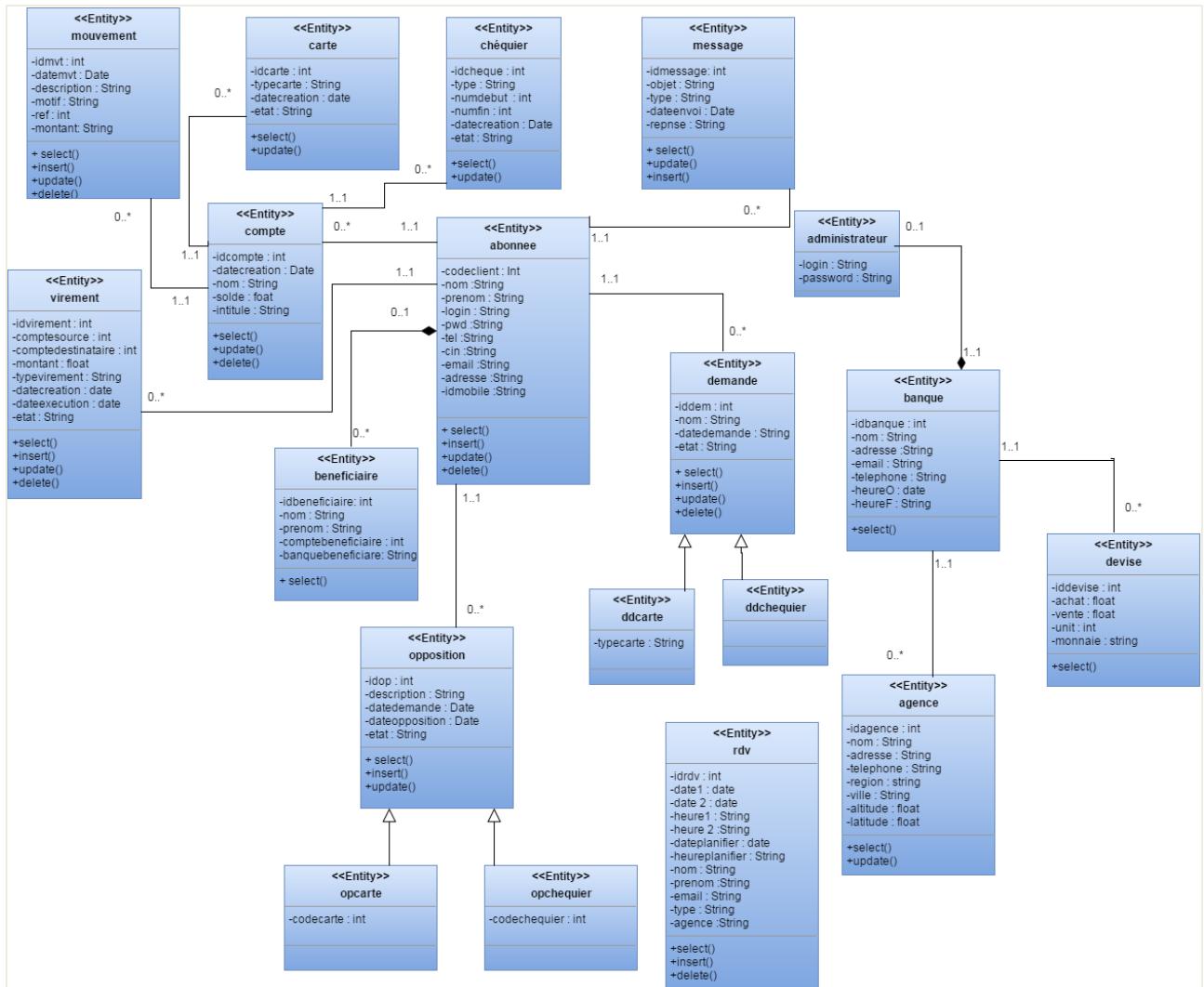


Figure 8. Diagramme de classe

3.2. Architecture général de l'application Mobile front office

Au niveau de cette partie, on va décrire le fonctionnement de notre système développé.

L'application «Attijari Mobile Banking» suit l'architecture trois-tiers : Le client Android, le serveur web et le serveur de base de donnée. Lorsque notre application Android s'exécute, elle se connectera au script PHP à l'aide de protocole http, Le script PHP va récupérer les données depuis la base de données MySQL. Ensuite les données seront encodées au format JSON et envoyées au système Android, l'application Android va obtenir ces données codées. Il les analysera et les affichera sur l'appareil Android.

La figure suivante illustre bien la façon d'échanger les données entre le client Android et la partie des serveurs.

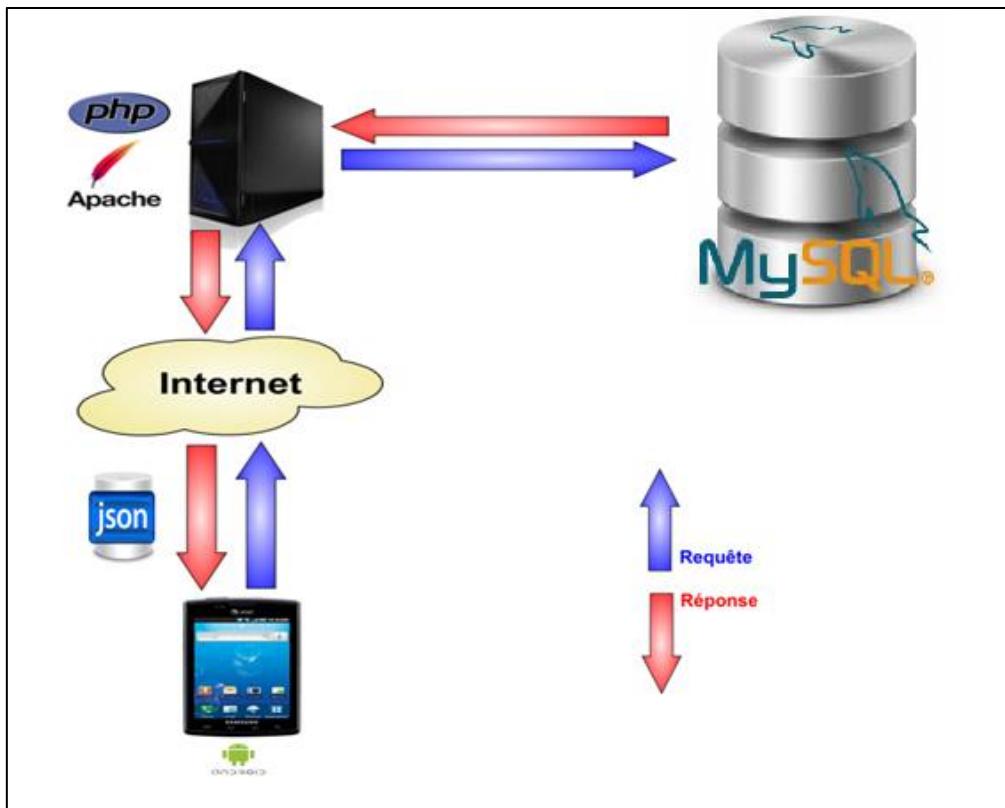


Figure 9. Architecture 3- tiers de l'application mobile

3.3. Pattern et style architectural de l'application Back Office

3.3.1. Style architectural

Nous avons développé l'application en nous basant sur l'architecture trois-tiers. En adoptant cette architecture, le système sera divisé en trois couches (ou niveaux) différents.

- ❖ Présentation des données.
- ❖ Logique métier.
- ❖ Accès aux données.

En effet, cette architecture est un modèle logique d'architecture applicative qui vise à séparer, très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et à présenter cette application comme un empilement de trois couches, étages, ou niveaux qui sont les suivantes :

Couche Présentation (premier niveau) : Elle a pour rôle de représenter l'interface de l'application qui est visible et interactive avec les utilisateurs. Présenter les résultats renvoyés par le modèle On parle d'interface homme-machine. La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier, et lui présente en retour les informations renvoyées par les traitements de cette couche.

Couche Métier / Business (second niveau) : Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la «logique», et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs effectuées au travers de la couche présentation.

Couche Accès aux données (troisième niveau) : C'est la partie qui gère l'accès aux données du système. Ces données peuvent être propres au système, ou gérées par un autre système.

La figure 10 schématise l'architecture générale de notre application web.

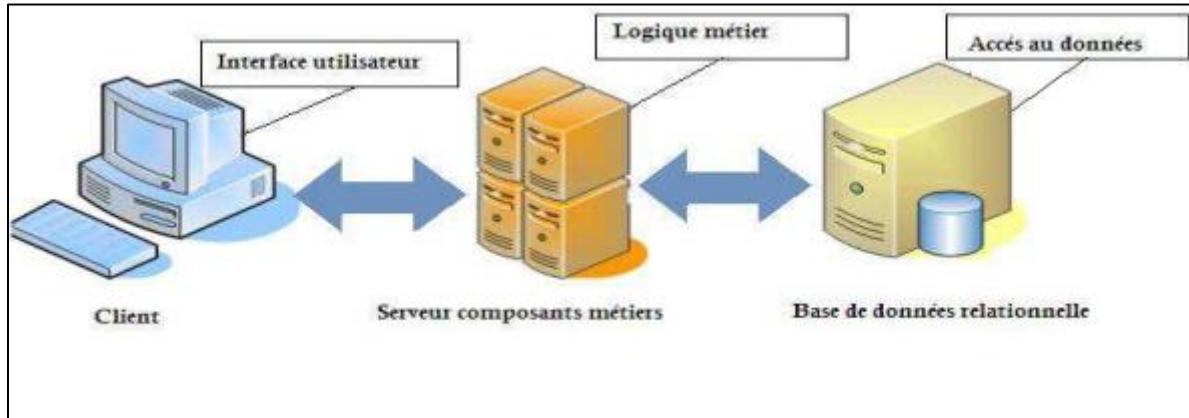


Figure 10. Niveaux de l'architecture 3 tiers

3.3.2. Pattern architectural

En ce qui concerne le patron de conception, nous avons choisi le modèle MVC (modèle-Vue-Contrôleur), c'est un concept très puissant intervenant dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur).

Le rôle d'une telle structure est de diviser l'application en trois parties distinctes.

- ❖ **Modèle** : représente les données de l'application (les bases de données en fait partie) et définissent la logique de manipulation de ces données.
- ❖ **Vue** : représentent la vue de l'application, l'interface avec laquelle l'utilisateur communique.
- ❖ **Contrôleur** : contient les fonctionnalités nécessaires pour gérer et contrôler la couche intermédiaire entre la vue et le modèle [3].

La figure qui suit présente le schéma d'interaction entre les composants de ce modèle

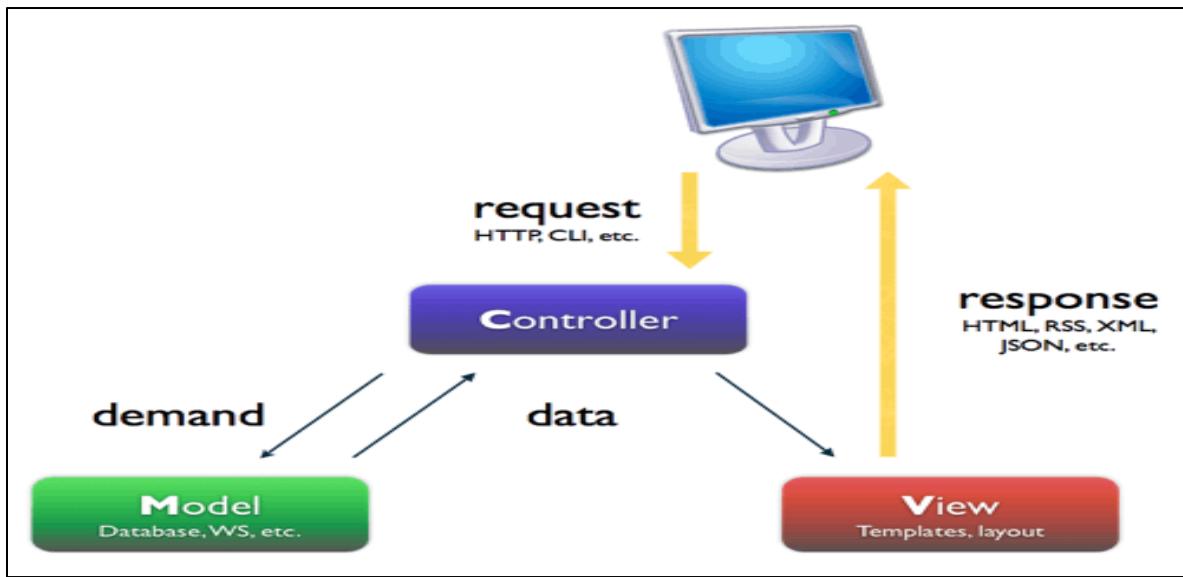


Figure 11. Couche MVC

4. Environnement technique du travail

4.1. Outil matériel

Pour la réalisation de notre application, nous avons disposé d'un ordinateur Dell caractérisé par :

- ❖ Système d'exploitation : Windows 7/64bits.
- ❖ Processeur : Core i3.
- ❖ Mémoire vive : 4 Go.

4.2. Outil de conception

Draw.io : C'est une application Cloud de conception UML, utilisé pour tracer les diagrammes de conceptions (Use cases, diagrammes de classes...).

4.3. Outils de développement

Dans cette partie nous allons décrire les outils de développement utilisés pour réaliser ce projet.

- ❖ **NetBeans**

C'est un environnement de développement libre et open source intégré pour le développement d'applications sur les systèmes d'exploitation Windows, Mac, Linux et Solaris. L'IDE simplifie le développement de web, entreprise, bureau, et les applications mobiles [4].



❖ **AndroidStudio**

C'est un environnement de développement pour développer des applications Android. Il permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android et propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément [5].



❖ **Genymotion**

C'est un émulateur Android relativement rapide qui permet de simuler un terminal Android sur un PC. Cet émulateur a surtout été conçu pour les développeurs pour tester leurs applications sur Smartphones que des tablettes Android virtuels sur le PC [6].



❖ **ApacheTomcat**

C'est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process2, il est paramétrable par des fichiers XML et des propriétés, Il comporte également un serveur http [7].



❖ **Mysql**

C'est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde [8].



5. Langage de programmation

❖ Java

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (JSP) [9].

❖ PHP (HyperText Preprocessor)

C'est un langage de programmation libre principalement utilisé pour produire des pages web dynamiques via un serveur http [10].

❖ JSON (JavaScript Object Notation)

JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendant du C, comme par exemple : C lui-même, C++, C#, Java, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal [11].

Pourquoi Json ?

JSON est plus simple pour retirer des données sur le serveur, et les utiliser, c'est comme une mémoire permanente pour un programme. Il vous faut connaître la structure des données pour l'utiliser, de préférence vous devez être le propriétaire du fichier. Il est plus léger que XML et économise les ressources.



❖ Web services REST

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 des fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et de manière synchrone. REST ou « Representation State Transfer » est un style d'architecture de services web qui utilise les standards Web déjà très utilisé, plus spécifiquement http. Dans notre application, nous allons choisir PHP pour le codage des web services et JSON pour la sérialisation vu son apport de point de vue volume des données communiquées par rapport à XML et YAML [12].

6. Plateforme et Framework utilisé

Nous détaillons dans ce qui suit les différents moyens utilisés pour le développement. De nombreux langages de programmation sont utilisables pour développer des sites web dynamiques. Nous avons choisi parmi ces langages :

- ❖ **JEE** : Java Enterprise Edition, ou Java EE (anciennement J2EE), est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise [13].



- ❖ **Hibernate** : Hibernate est un Framework open source gérant la persistance des objets en base de données relationnelle. Il est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement Java EE complet. Hibernate apporte une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau [14].



- ❖ **JPA** : La Java Persistence API (abrégée en JPA), est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java [15].



- ❖ **JSF** : Java Server Faces (abrégé en JSF) est un Framework Java pour le développement d'applications Web. À l'inverse des autres Framework MVC traditionnels à base d'actions, JSF est basé sur la notion de composants, comparable à celle de Swing ou SWT, où l'état d'un composant est enregistré lors du rendu de la page, pour être ensuite restauré au retour de la requête. JSF est agnostique à la technologie de présentation [16].



- ❖ **Prime faces** : Prime faces est une bibliothèque open source de composantes JSF. Il est basé côté serveur sur l'API standard de ce dernier [17].



- ❖ **Spring Security** est un cadre qui met l'accent sur la fourniture de l'authentification et l'autorisation d'applications Java.



Conclusion

Nous avons présenté le Sprint 0. C'est une phase de préparation pendant laquelle on a construit notre Backlog du produit en se basant sur les besoins fonctionnels et non fonctionnels. En plus, nous avons découpé notre projet en des releases qui formeront les chapitres suivants. Et finalement, nous avons défini les technologies et les environnements de développement ainsi que l'architecture globale de l'application.

Chapitre 3 : Release 1

Introduction

Le terme release désigne une version distribuée de notre application .on parle de release pour considérer la période de temps qui va du début du travail sur cette version jusqu'à sa livraison et qui passe par une série des sprints successifs qui se terminent quand les incrément de ces derniers construisent un produit présentant suffisamment de valeur à l'utilisateur.

Notre premier release sera composé de trois sprints, Tous au long de ce chapitre, nous allons définir le but de chacune et traiter les histoires utilisateurs (User Story) de nos sprints.

1. Sprint 1 : Authentification

Le sprint est le cœur de Scrum. Il s'agit d'un bloc de temps durant lequel un incrément du produit sera réalisé. Avant de se lancer dans un sprint, on définit un objectif et on lui associe une liste de fonctionnalités qui constitueront le Backlog de sprint.

Ci-dessous la définition de l'objectif et des développeurs de ce sprint.

- ❖ Objectif : Les clients de l'application Attijari Mobile Banking sont capables de s'authentifier avec succès avec la possibilité de changer le mot de passe en cas de besoin.
- ❖ Développeurs : Team.

1.1. Le Backlog de sprint 1

Après la définition de but de notre sprint, il est temps de décider quelles histoires inclure dans ce dernier. Plus précisément, quelles histoires utilisateurs de notre Backlog du produit seront incluses dans le Backlog du sprint.

Le tableau suivant résume donc le Backlog de notre premier sprint :

Id	User Story	Priorité
30	En tant que un client d'Attijari Bank je veux m'authentifier afin de bénéficier des fonctionnalités de l'application.	20
31	En tant que un client d'Attijari Bank je veux changer mon mot de passe afin de mieux personnaliser mes paramètres de connexion.	20

Tableau 3. Backlog du sprint 1

Par la suite on va passer à la spécification fonctionnelle ou on va représenter les deux parties conception et test.

1.2. Spécification fonctionnelle

Dans cette partie, nous présentons le diagramme de cas d'utilisation du sprint puis une description textuelle de chaque cas.

1.2.1. Diagramme de Cas d'utilisation de sprint 1

Les différents cas d'utilisation de ce sprint sont :

- ❖ S'authentifier : permet au client de s'authentifier.
- ❖ Changer mot de passe : permet au client de changer son mot de passe.

Les différentes fonctionnalités de ce sprint sont modélisées par le diagramme de cas d'utilisation dans la figure

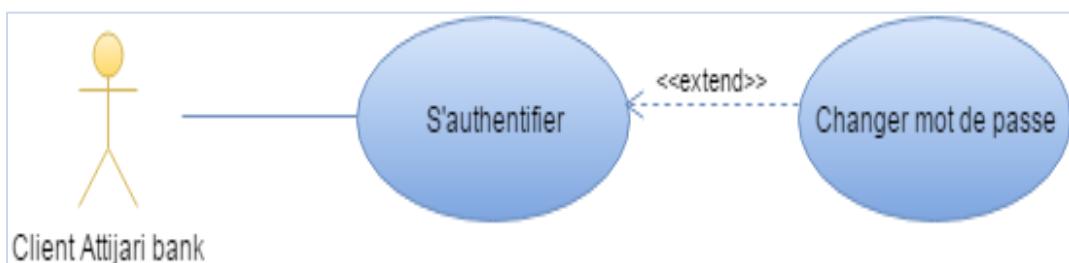


Figure 12. Diagramme de cas d'utilisation de sprint 1

1.2.2. Description textuelle du cas d'utilisation

La description détaillée de cas d'utilisation «s'authentifier» est donnée par le tableau suivant.

Cas d'utilisation	S'authentifier
Acteurs	Client d'Attijari bank
Pré condition	Paramètres d'accès valides.
Post-condition	Utilisateur authentifié.
Scénario principale	<ul style="list-style-type: none">-L'utilisateur saisit son code et son mot de passe.-Il clique sur le bouton « Connecter ».-Le système vérifie la combinaison login et mot de passe. S'ils sont corrects, le système affiche l'interface du menu client.
Extension	<p>L'utilisateur peut choisir de changer son mot de passe :</p> <ul style="list-style-type: none">-L'utilisateur saisit son code et son mot de passe et son nouveau mot de passe.-Il clique sur le bouton « Modifier ».

	-Le système modifie le mot de passe et il affiche un message indiquant que la modification a été effectuée.
Exceptions	Si le login et/ou le mot de passe sont erronés, le système réaffiche la page d'authentification avec un message d'erreur expliquant le cas et demandant à l'utilisateur de s'authentifier avec des paramètres d'accès valide, sinon l'utilisateur peut quitter en cliquant sur le bouton «quitter » ou accéder au menu principal en cliquant sur l'icône « home ».

Figure 13. Description détaillée du cas d'utilisation «s'authentifier »

1.3. Conception

La conception est la deuxième activité dans un sprint. Elle se traduit par le diagramme de séquence et le diagramme de classe participante.

1.3.1. Diagramme de classe participante

Le diagramme de classe participante permet de représenter une vue statique du système d'information.

- ❖ Classe interface ou « Boundary » : sert à modéliser les interactions entre le système et ses acteurs. Elle présente souvent des abstractions des fenêtres, des formulaires, d'interfaces de communication.
- ❖ Classe contrôle ou « Control » : utiliser pour représenter la coordination, le séquencement, les transactions et le contrôle des autres objets. Elle est en général reliée à un cas d'utilisation particulier.
- ❖ Classe entité ou « Entity » : sert à modéliser les informations persistantes et durables.

❖ Cas d'utilisation «S'authentifier»

Nous présentons dans cette partie le diagramme de classe participante de cas d'utilisation «S'authentifier »

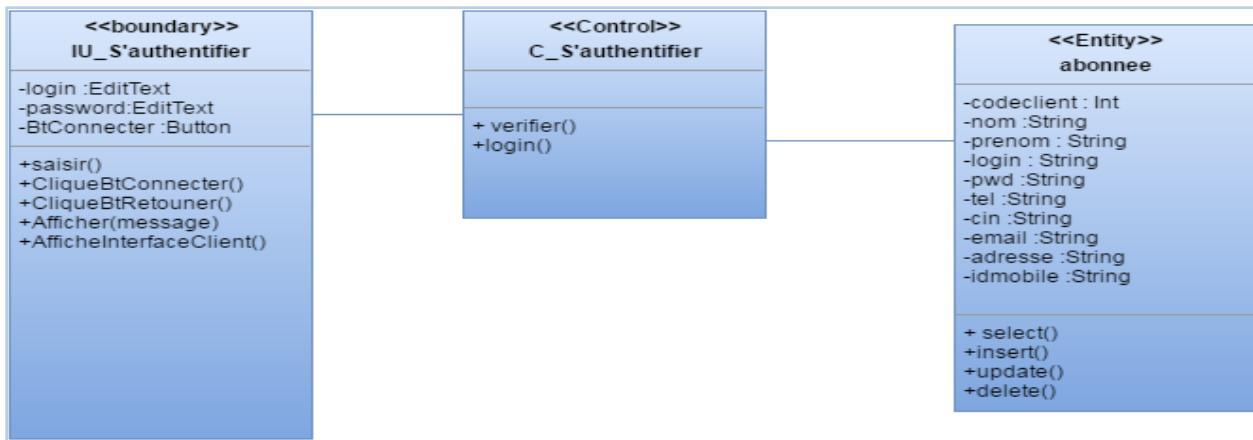


Figure 14. Diagramme de classe de cas d'utilisation «S'authentifier»

1.3.2. Diagramme de séquence

Le diagramme de séquence permet de représenter l'interaction acteur système d'une manière dynamique. Il permet de décrire les scénarios de chaque cas d'utilisation.

❖ Diagramme de séquence du cas d'utilisation «S'authentifier»

La figure 15 illustre le diagramme de séquence du cas d'utilisation «S'authentifier». En effet, le client d'Attijari bank doit s'authentifier pour accéder à l'espace client de notre application. Il remplit le formulaire d'authentification et clique sur le bouton «Connexion». Le système vérifie la combinaison login et mot de passe à travers le web service «Login». S'ils sont corrects, le système affiche l'interface du menu client.

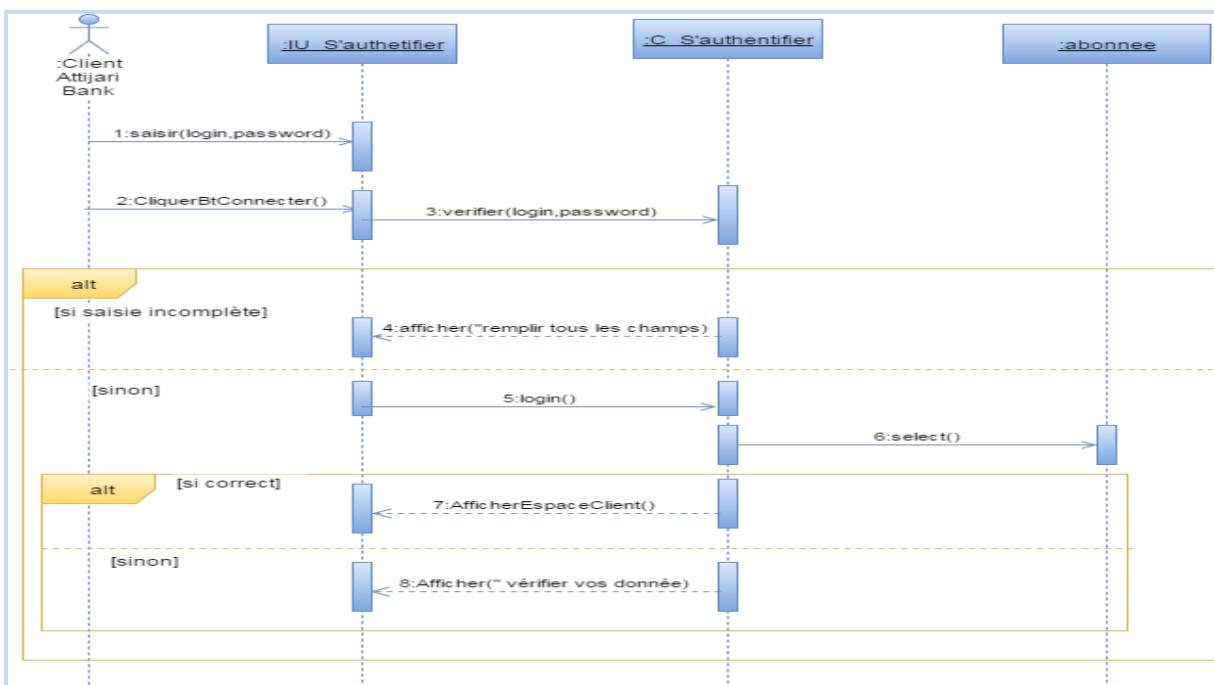


Figure 15. Diagramme de séquence de cas d'utilisation « S'authentifier »

❖ Cas d'utilisation «Modifier mot de passe»

Cette figure illustre le diagramme de séquences du cas d'utilisation «Modifier mot de passe». Le client d'Attijari bank remplit le formulaire de modification de mot de passe puis il clique sur le bouton «Modifier». Le système vérifie la combinaison login et mot de passe à travers le web service «Login». S'ils sont corrects, le système enregistre le nouveau mot de passe et affiche l'interface d'authentification sinon un message d'erreur s'affiche pour lui informer que les coordonnées sont erronées.

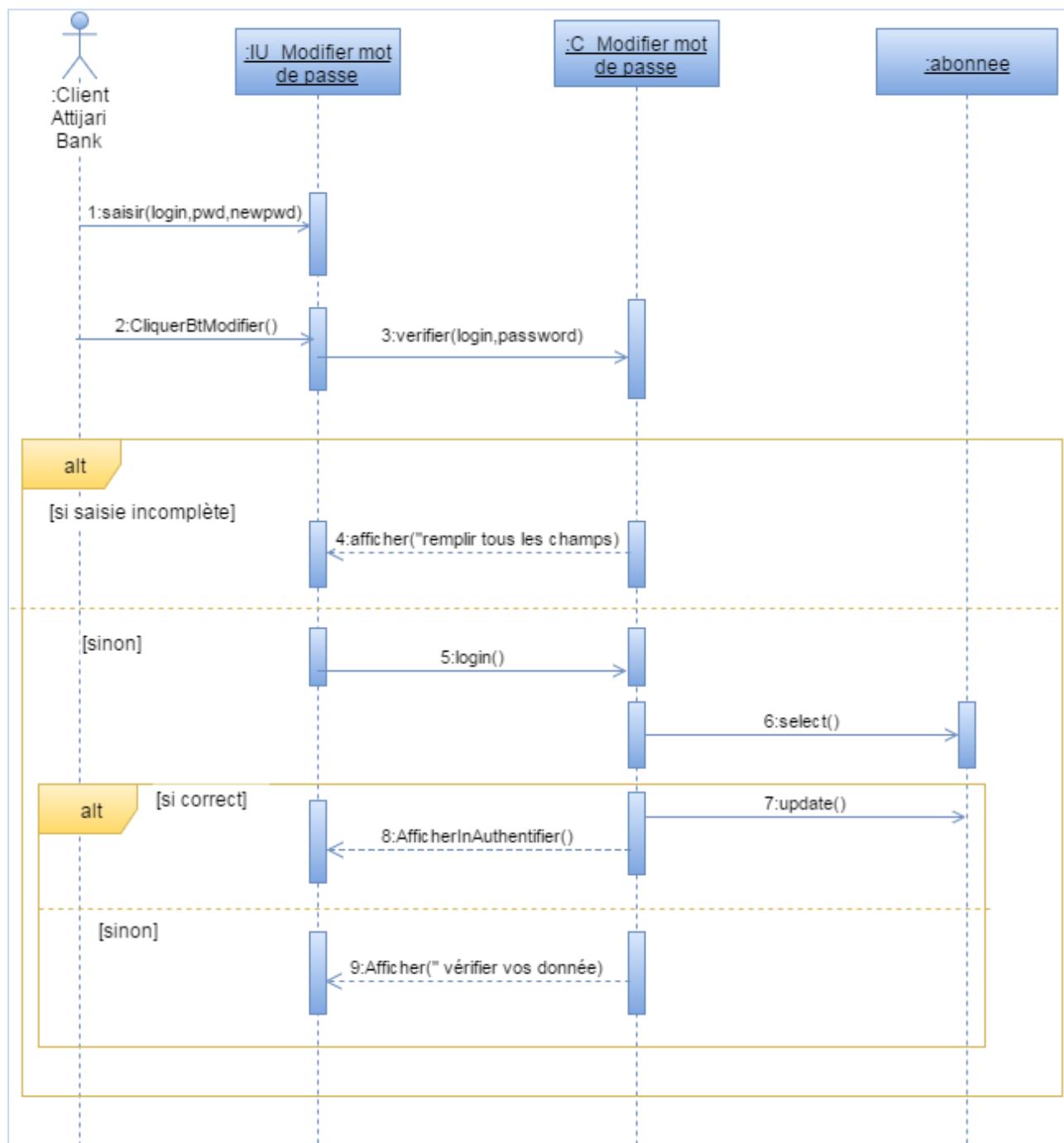


Figure 16 : diagramme de séquence «Modifier mot de passe»

1.4. Test

Dans cette partie, nous nous intéressons à la présentation de quelques interfaces de notre système pour ce premier sprint.

Après avoir démarré l'application, le client doit s'authentifier en saisissant son login et son mot de passe pour pouvoir accéder aux services offerts par notre application. Au cas d'erreur un message sera affiché.

La figure 17 qui suit présente l'interface d'authentification :

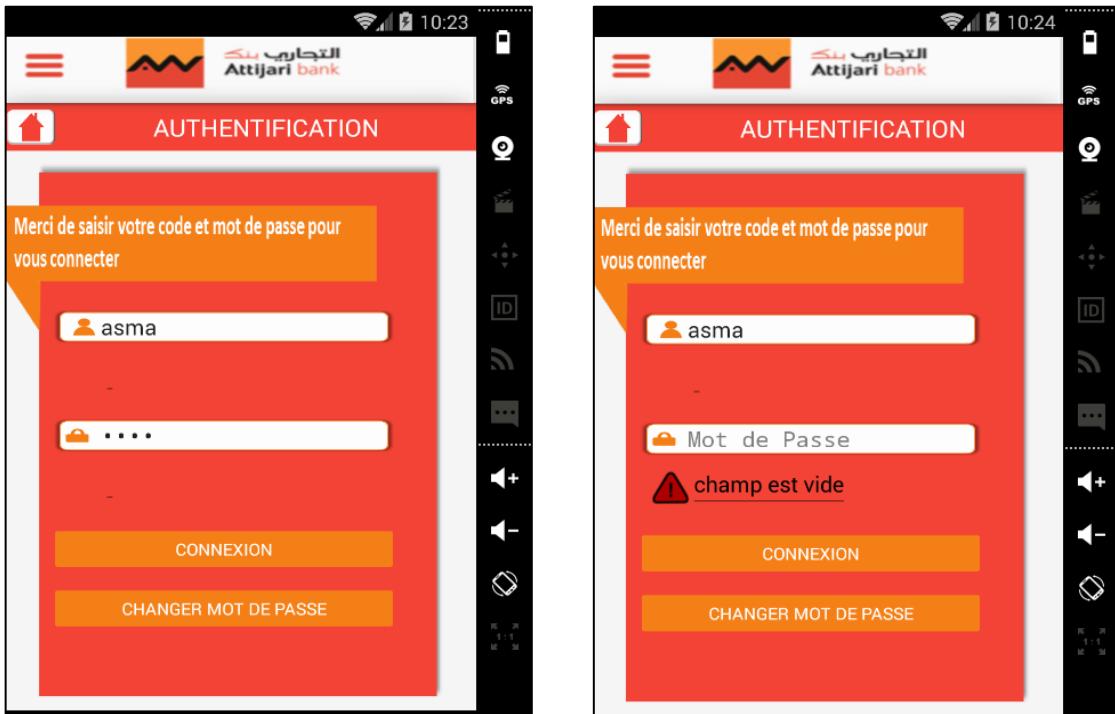


Figure 17. Interface d'authentification

Le client peut aussi changer son mot de passe en tapant ses anciennes informations avec son nouveau mot de passe.

La figure 18 illustre l'interface de modification du mot de passe

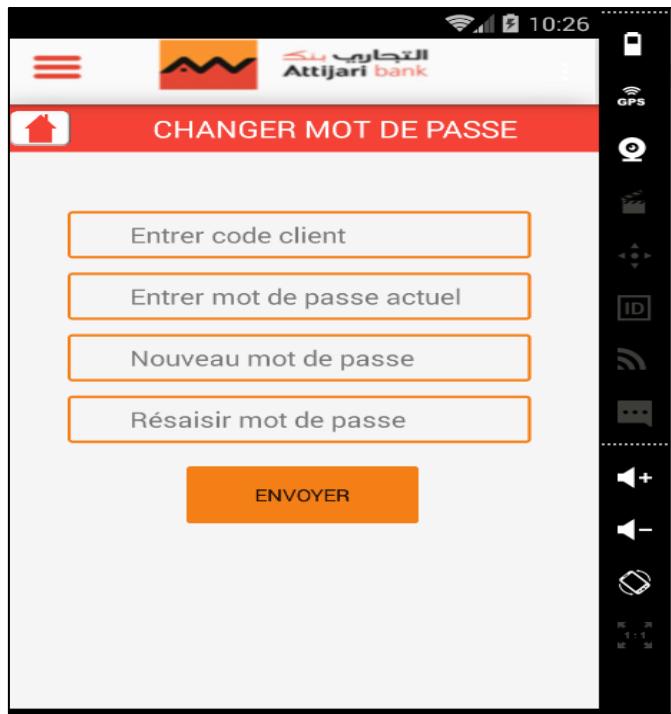


Figure 18. Interface de modification du mot de passe

2. Sprint 2: Accès aux services client

Comme chaque début sprint, on doit préciser notre objectif et on lui associe ses fonctionnalités qui constitueront le Backlog de sprint. Ci-dessous la définition de l'objectif et les développeurs de ce sprint.

- ❖ Objectif attendu : Les clients de l'application Attijari Mobile Banking sont capables d'accéder au service client offert par la banque par exemple : opposition d'une carte et/ou un chéquier et des demandes de virement, consulter les compte etc.
- ❖ Développeurs : Team.

2.1. Le Backlog de Sprint 2

Dans celui-ci, nous allons traiter les user stories de notre deuxième sprint présenté dans le tableau suivant pour produire un incrément potentiellement livrable.

ID	User story	Priorité
8	En tant que client d'Attijari Bank, je veux demander des virements compte à compte ou vers bénéficiaire.	20
9	En tant que client d'Attijari Bank, je veux demander une carte afin de faciliter mes échanges quotidiens.	20

10	En tant que client d'Attijari Bank, je veux un nouveau chéquier afin de faciliter mes échanges.	20
11	En tant que client d'Attijari Bank, je veux consulter mes comptes avec ces informations (solde,..) afin de visualiser leurs coordonnées.	20
12	En tant que client d'Attijari Bank, je veux consulter mon extrait de compte afin de visualiser les mouvements.	20
13	En tant que client d'Attijari Bank, je veux opposer ma carte afin de bloquer son utilisation.	20
14	En tant que client d'Attijari Bank, je veux opposer mon chéquier afin de bloquer son utilisation.	20
15	En tant que client d'Attijari Bank, je veux envoyer un message à ma banque pour une réclamation ou demande d'une information.	20

Tableau 4. Backlog de Sprint 2

2.2. Spécification fonctionnelle

Lors de la première étape de chaque itération, la spécification fonctionnelle se traduit par un diagramme de cas d'utilisation.

2.2.1. Diagramme de Cas d'utilisation de sprint 2

Les différents cas d'utilisation de ce sprint sont :

- ❖ Consulter les comptes : permet de consulter les comptes avec leurs soldes et leurs mouvements.
- ❖ Demander une carte : permet de demander une nouvelle carte ou/et opposer une carte.
- ❖ Demander un virement : permet de demander un virement compte à compte ou vers bénéficiaire.
- ❖ Demander un chéquier : permet de demander un nouveau chéquier ou/et opposer un chéquier.
- ❖ Suivre les demandes : permet de consulter l'état de demande (accepté ou pas encore).
- ❖ Consulter la boîte de réception : permet de consulter les messages et envoyer un nouveau message.

La figure suivante illustre le diagramme de cas d'utilisation globale du sprint 2.

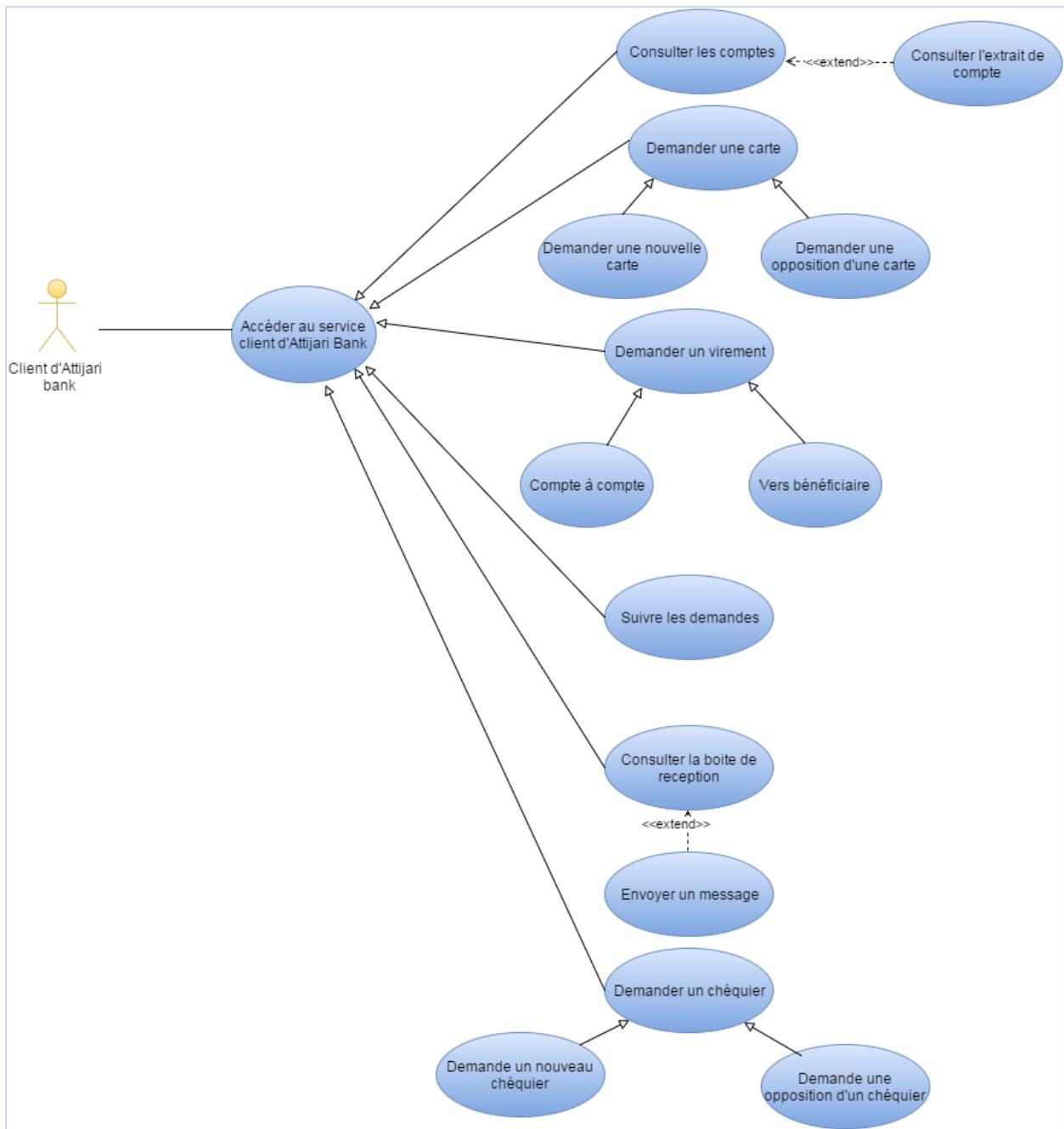


Figure 19. Diagramme de Cas d'utilisation de Sprint 2

2.2.2. Description détaillée de cas d'utilisation

- ❖ **Description détaillée de cas d'utilisation « Consulter les comptes »**

La description détaillée de cas d'utilisation «Consulter les comptes » est donnée par le tableau suivant :

Cas d'utilisation	Consulter les comptes
Acteurs	Client d'Attijari Bank
Pré condition	Il faut que les champs du formulaire soit tous remplis et les paramètres d'accès sont valides
Post-condition	Le client consulte son extrait du compte avec succès.
Scénario principale	<ul style="list-style-type: none"> -Le client de la banque clique sur l'icône « compte ». -Le système affiche les listes des comptes avec ses informations. -Le client choisit un des comptes. -Le système affiche l'interface des mouvements du compte.

Tableau 5. Description détaillée de cas d'utilisation « Consulter les comptes »

❖ **Description détaillée de cas d'utilisation « Demander un virement »**

La description détaillée de cas d'utilisation « Demander un virement » est donnée par le tableau suivant :

Cas d'utilisation	Demander un virement
Acteurs	Client d'Attijari Bank
Pré condition	Il faut que les champs du formulaire soient tous remplis et les paramètres d'accès sont valides.
Post-condition	Virement effectué.
Scénario principale	<ul style="list-style-type: none"> - le client de la banque demande un virement en cliquant sur l'icône « virement ». - le client saisit son code client et choisir le type de virement. -Le client de la banque doit remplir le formulaire affiché. - les informations seront envoyés au serveur afin qu'il vérifie les champs. -Le serveur envoie une réponse de validation d'envoie au client.
Exceptions	Si l'un des champs est mal rempli une réponse d'échec sera envoyée au client.

Tableau 6. Description détaillée de cas d'utilisation « Demander un virement »

❖ **Description détaillée de cas d'utilisation « Demander une carte »**

La description détaillée de cas d'utilisation « Demander une carte » est donnée par le tableau suivant :

Cas d'utilisation	Demander une carte
Acteurs	Client d'Attijari Bank
Pré condition	Il faut que les champs du formulaire soit tous remplis et les paramètres d'accès sont valides
Post-condition	Demande de client envoyé à la boite de réception de la banque.
Scénario principale	<p>-Le client de la banque clique sur l'icône « carte » puis choisit «demander une carte».</p> <p>-Un formulaire de demande est affiché.</p> <p>-il remplit le formulaire qui contient ses informations personnels et le type de la carte souhaité y avoir, après il clique sur le bouton « envoyer » pour confirmer sa demande.</p>
Exceptions	Si le client saisit une fausse information ou laisse une case vide, alors un message d'erreur va être affiché, le client donc est invité à ressaisir les données convenablement.

Tableau 7. Description détaillée de cas d'utilisation « Demander une carte »

❖ Description détaillée de cas d'utilisation « Demander un chéquier »

La description détaillée de cas d'utilisation «Demander un chéquier » est donnée par le tableau suivant :

Cas d'utilisation	Demander un chéquier
Acteurs	Client d'Attijari Bank
Pré condition	Il faut que les champs du formulaire soient tous remplis et les paramètres d'accès sont valides
Post-condition	Demande de client envoyé à la boite de réception de la banque.
Scénario principale	<p>-Le client clique sur l'icône « chéquier» puis choisit «demander un chéquier».</p> <p>-Un formulaire de demande est affiché.</p> <p>-il remplit le formulaire, après il clique sur le bouton « envoyer » pour confirmer sa demande.</p>
Exceptions	Si le client saisit une fausse information ou laisse une case vide, alors un message d'erreur va être affiché, le client donc est invité à ressaisir les données convenablement.

Tableau 8. Description détaillée de cas d'utilisation «Demander un chéquier »

❖ **Description détaillée de cas d'utilisation « Opposer un chéquier »**

La description détaillée de cas d'utilisation «Opposer un chéquier» est donnée par le tableau suivant :

Cas d'utilisation	Opposer un chéquier
Acteurs	Client d'Attijari Bank
Pré condition	Il faut que les champs du formulaire soient tous remplis et les paramètres d'accès soit valides
Post-condition	Demande de client envoyé à la boite de réception de la banque.
Scénario principale	<p>Si notre client a un /des problèmes concernant un chèque, il a le droit d'envoyer une réclamation à la banque.</p> <p>-Le client de la banque clique sur l'icône « chéquier» puis choisit «opposer un chéquier».</p> <p>-Il remplit un formulaire qui contient ses informations personnels et explique sa réclamation concernant un chèque spécifique, après il clique sur le bouton « envoyer » pour que la réclamation soit à la boite de réception de la banque.</p>
Exceptions	Si le client saisit une fausse information ou laisse une des cases vide, alors un message d'erreur va être affiché, le client donc est invité à ressaisir les données convenablement.

Tableau 9. Description détaillée de cas d'utilisation «Opposer un chéquier»

❖ **Description détaillée de cas d'utilisation « Opposer une carte »**

La description détaillée de cas d'utilisation «Opposer une carte» est donnée par le tableau suivant :

Cas d'utilisation	Opposer une carte
Acteurs	Client d'Attijari Bank
Pré condition	Il faut que les champs du formulaire soit tous remplis et Paramètres d'accès valides
Post-condition	Demande de client envoyé à la boite de réception de la banque.
Scénario principale	Si notre client a un /des problèmes concernant une carte, il a le droit d'envoyer une réclamation à la banque.

	<p>-Le client de la banque clique sur l'icône « carte» puis choisit «opposer une carte».</p> <p>-Il remplit un formulaire qui contient ses informations personnels et explique sa réclamation concernant une carte, après il clique sur le bouton « envoyer » pour que la réclamation soit à la boite de réception de l'agent de la banque ou bien il quitte pour revenir au menu.</p>
Exceptions	<p>Si le client saisit une fausse information ou laisse une case vide, alors un message d'erreur va être affiché, le client donc est invité à ressaisir les données convenablement.</p>

Tableau 10. Description détaillée de cas d'utilisation «Opposer une carte»

❖ **Description détaillée de cas d'utilisation « Suivre les demandes »**

La description détaillée de cas d'utilisation «Suivre les demandes» est donnée par le tableau suivant :

Cas d'utilisation	Suivre les demandes
Acteurs	Client d'Attijari Bank
Pré condition	Client authentifié
Post-condition	Les états de demande sont affichés
Scénario principale	<ul style="list-style-type: none"> -Le client clique sur « suivis mes demandes » -Le système affiche la liste de demandes avec les états (accepté ou non)

Tableau 11. Description détaillée de cas d'utilisation « Suivre les demandes »

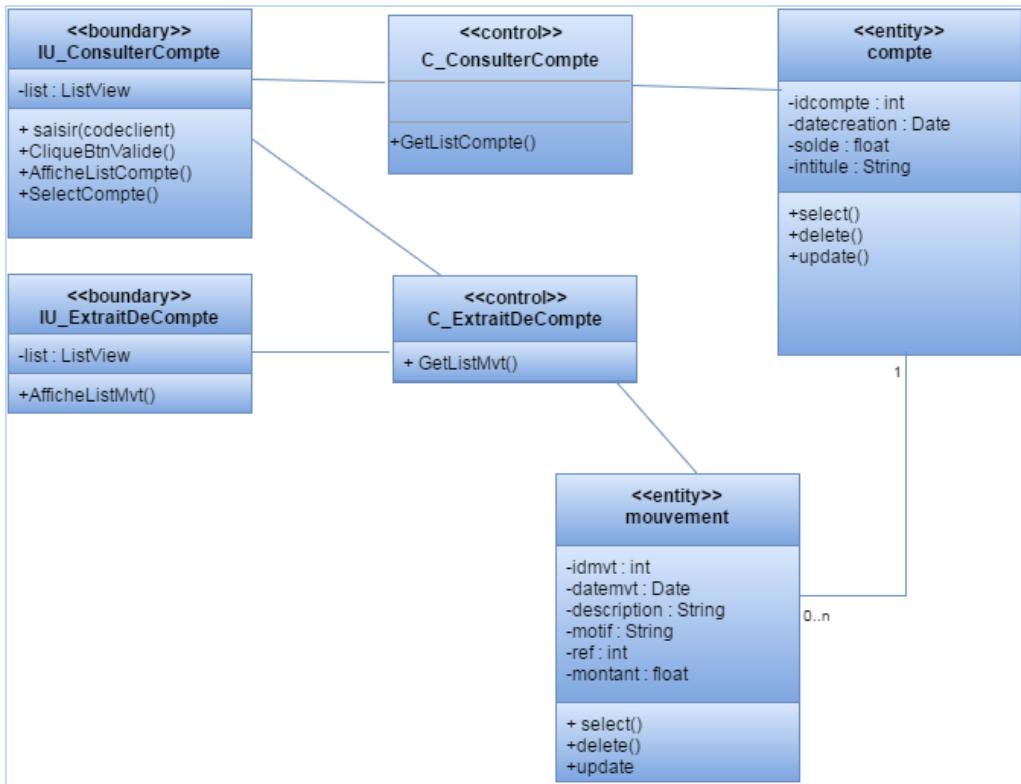
2.3. Conception

En nous référant aux descriptions textuelles dans la section précédente, nous présentons les diagrammes de séquences et les diagrammes des classes participantes de ce sprint. Sur la base de ces descriptions, nous pouvons constater que certains cas d'utilisations sont similaires. C'est pour cette raison qu'on a choisi de sélectionner quelques exemples pour les traiter.

2.3.1. Diagramme de classe participante

❖ **Cas d'utilisation « Consulter les comptes »**

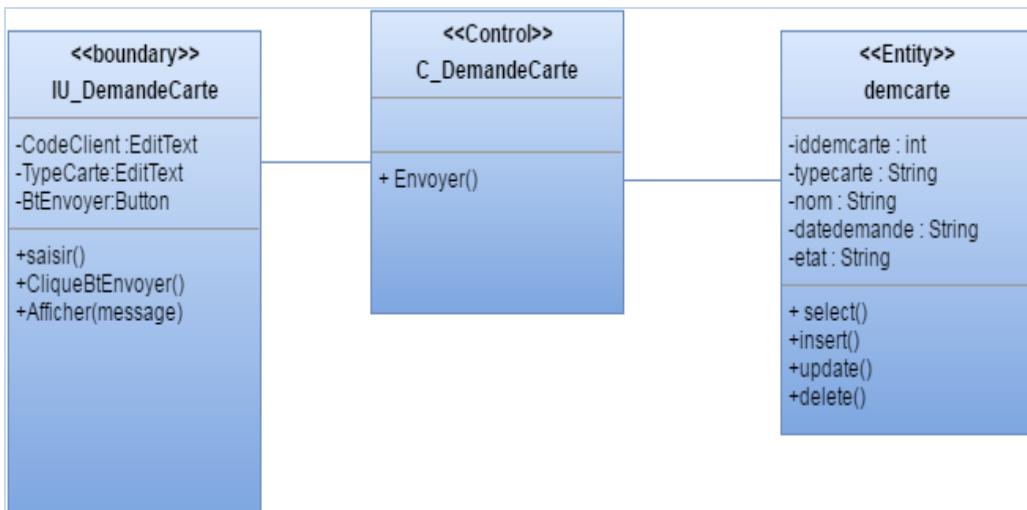
Nous présentons le diagramme de classe participante de cas d'utilisation« Consulter les comptes» dans la figure 20 :



**Figure 20. Diagramme de classe participante de Cas d'utilisation
« Consulter les comptes »**

❖ Cas d'utilisation «Demander une carte bancaire»

Nous présentons le diagramme de classe participante de cas d'utilisation «Demander une carte» dans la figure suivante :



**Figure 21. Diagramme de classe participante de cas d'utilisation
« Demander une carte »**

2.3.2. Diagramme de séquence

❖ Cas d'utilisation «Consulter les comptes»

La figure 22 illustre le diagramme de séquence du cas d'utilisation «Consulter les comptes». Le client d'Attijari bank clique sur le bouton «Consulter les comptes». Pour cela, deux web services sont nécessaires «GetListCompte» pour afficher la liste des comptes de client et «GetListMvt» pour la consultation de mouvement d'un compte sélectionné.

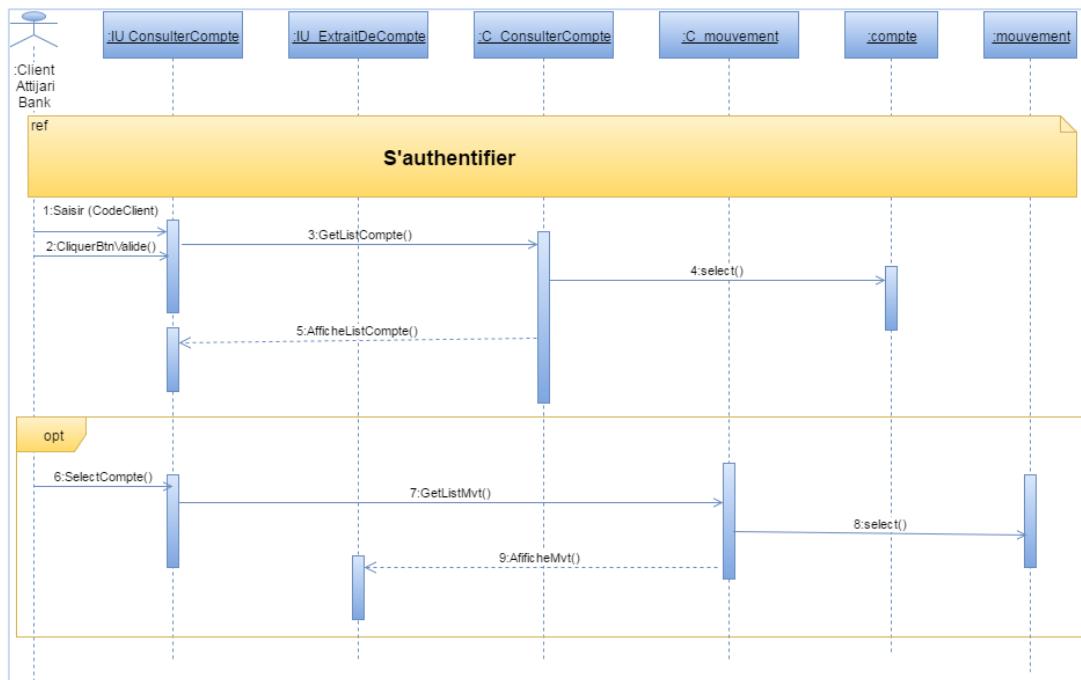


Figure 22. Diagramme de séquences du cas d'utilisation «Consulter les comptes»

❖ Cas d'utilisation «Demander une carte bancaire»

La figure 23 illustre le diagramme de séquences du cas d'utilisation «Demander une carte». Le client d'Attijari bank clique sur le bouton « demander une carte» puis remplit le formulaire, les données seront enregistrées dans la base avec le web service «envoyer».

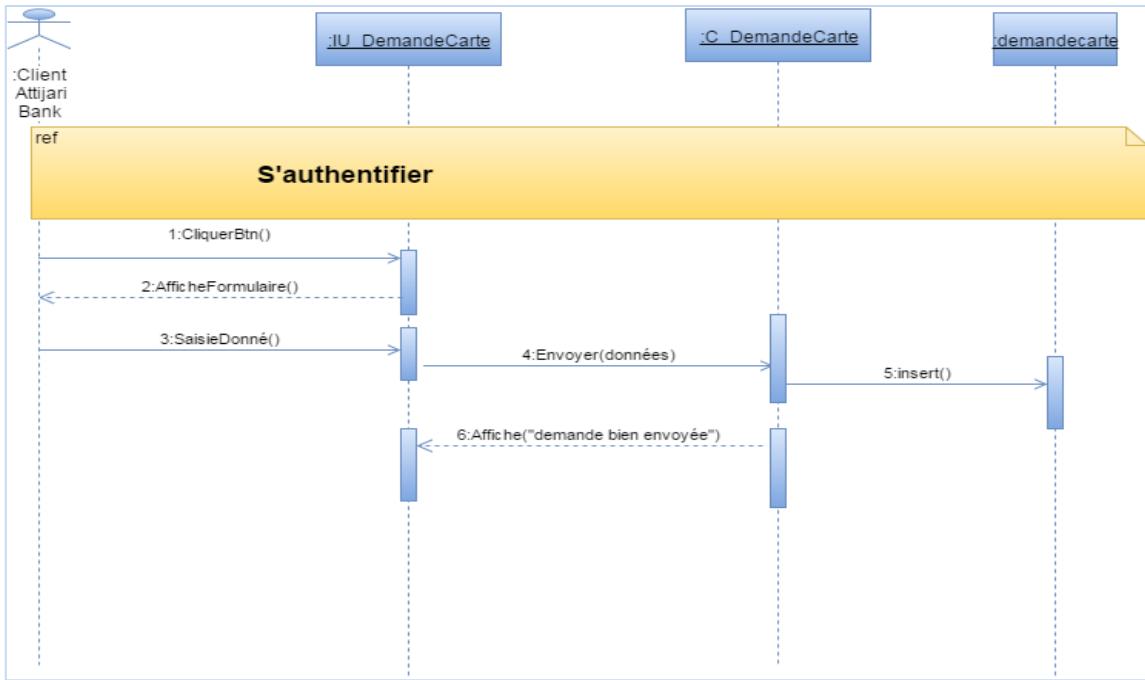


Figure 23. Diagramme de séquences du cas d'utilisation «Demander une carte».

2.4. Test

À ce stade, quelques interfaces de l'application à travers différents imprimés écrans réalisés seront exposées.

Cette interface est la première fenêtre qu'aperçoit le client authentifié et il peut accéder aux fonctionnalités de l'application.

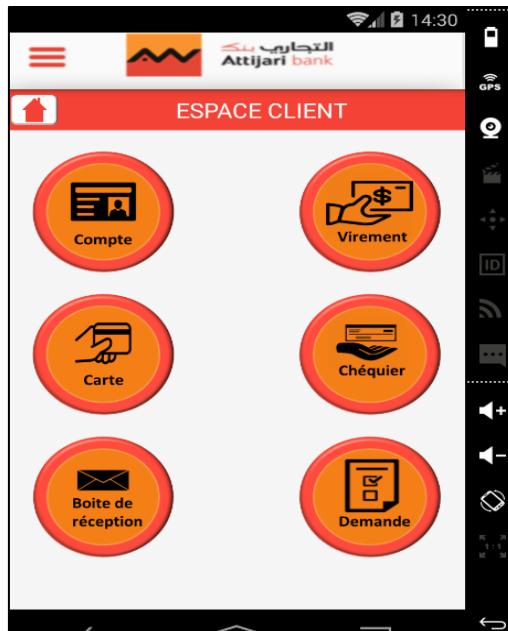


Figure 24. Interface espace client

Le client peut consulter le solde de son ou ses compte(s), ainsi que les mouvements effectués pour chaque. Cette fonctionnalité permet à l'abonné de consulter plusieurs informations qui concernent la liste des mouvements effectués dans chacun de ses comptes.

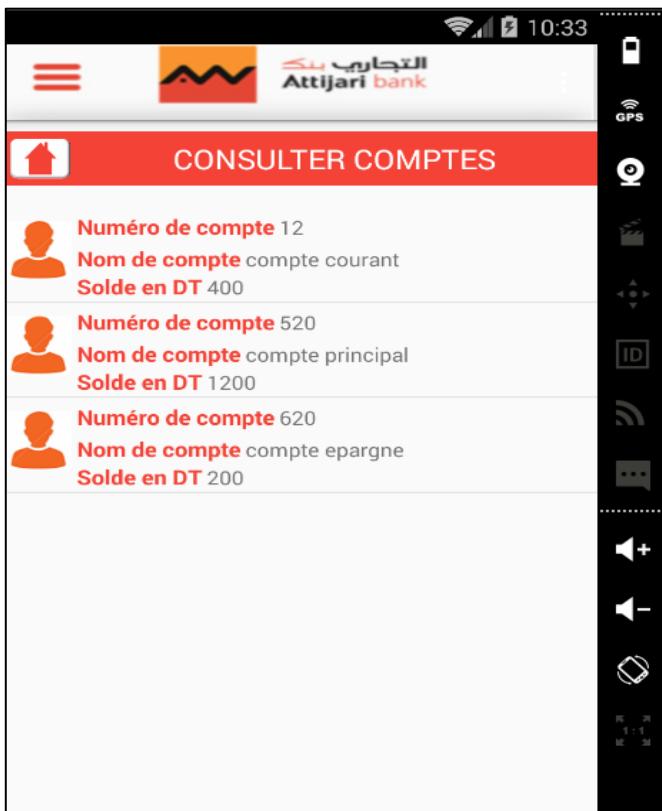


Figure 25. Interface de consultation des comptes

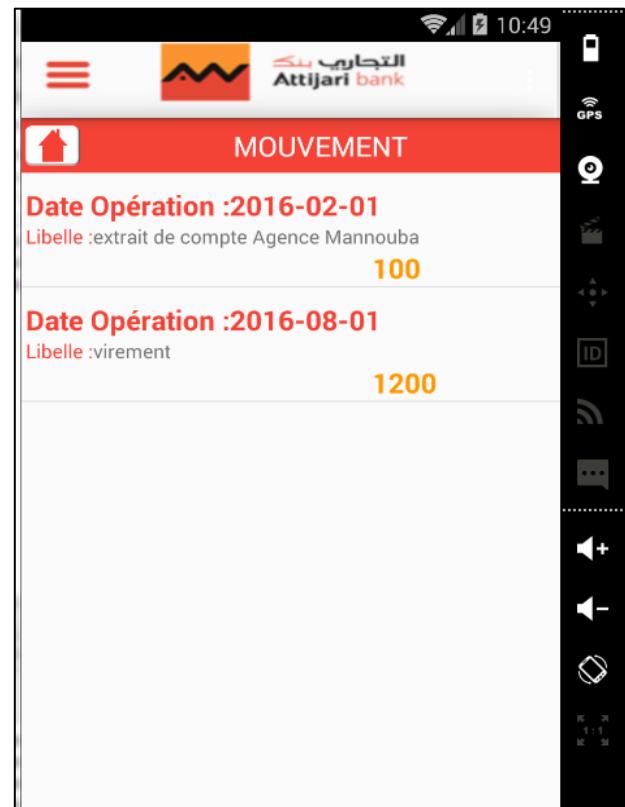


Figure 26. Interface de mouvement

Le client peut aussi envoyer une demande par exemple demande d'une carte, il remplit le formulaire avec le type de carte puis clique sur le bouton envoyer et sa demande sera enregistré dans la base.

La figure 27 illustre l'interface de demande d'une carte.

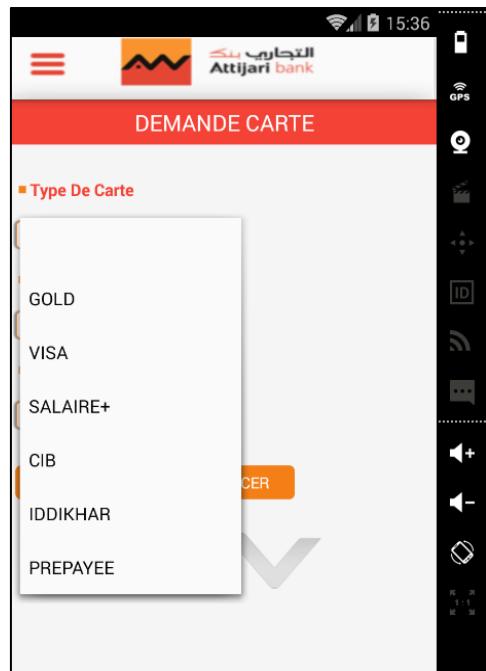


Figure 27. Interface demande une carte

Le client a éventuellement la possibilité d'effectuer des virements, soit de l'un de ses comptes à un autre, soit vers le compte d'un bénéficiaire inscrit dans le contrat.

La figure 28 illustre l'interface de demande d'un virement.

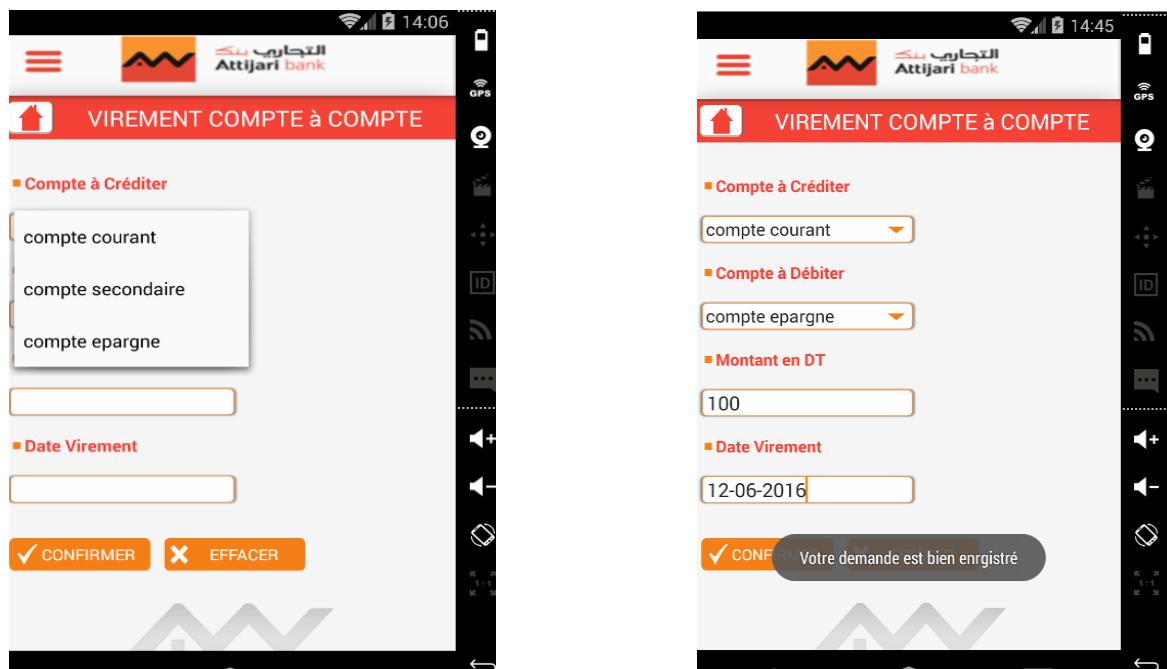


Figure 28. Interface de virement

3. Sprint 3: Accès aux services publics

Comme chaque début de sprint, on doit préciser notre objectif et on lui associe ses fonctionnalités qui constitueront le Backlog de sprint. Et dans les normes de Scrum, l'objectif et la composition de l'équipe ne peuvent être modifiés. Ci-dessous la définition de l'objectif et des développeurs de ce sprint.

- ❖ Objectif attendu : Les utilisateurs peuvent installer et utiliser l'application «Attijari Mobile Banking» sont avoir authentifié. Ils sont capables d'accéder à l'espace public et bénéficieront d'un ensemble de fonctionnalités (consulter le système de géo localisation, consulter les cours de devise...).
- ❖ Développeurs : Team.

3.1. Le Backlog du sprint

Une fois nous avons défini l'objectif de notre sprint, on va définir le Backlog du sprint.

Le tableau 12 résume donc le Backlog du troisième sprint.

Id	User Story	Priorité
1	En tant qu'utilisateur Mobile Banking, je veux suivre les cours de devise afin d'être au courant des changements de devise.	20
2	En tant qu'utilisateur Mobile Banking, je veux changer un montant afin de connaître le montant converti.	20
3	En tant qu'utilisateur Mobile Banking je veux prendre un rendez-vous afin de planifier une rencontre avec le chargé clientèle de mon agence	20
4	En tant qu'utilisateur Mobile Banking, je veux consulter le système de géo localisation afin de visualiser sur la carte les agences d'Attijari Bank.	20
5	En tant qu'utilisateur Mobile Banking, je veux consulter la liste des agences afin d'avoir leurs adresses.	20
6	En tant qu'utilisateur Mobile Banking je veux simuler un crédit afin d'avoir la mensualité de mon crédit	20
7	En tant qu'utilisateur Mobile Banking je veux appeler le service clientèle d'Attijari bank afin d'être en contact avec le service clientèle de banque	20

Tableau 12. Backlog du sprint 3

Par la suite on va passer aux activités et le cycle de développement. Dans un sprint nous pouvons présenter trois étapes parties : spécification fonctionnelle, conception, et le test.

3.2. Spécification fonctionnelle

Dans cette partie, on va présenter un diagramme de cas d'utilisation du sprint puis une description textuelle.

3.2.1. Diagramme de cas d'utilisation de sprint 3

Il s'agit de la première étape de ce sprint. Un utilisateur public peut accéder aux différentes fonctionnalités spécifiques offertes par l'application Mobile Banking d'Attijari Bank sans avoir authentifié.

Cette figure illustre le diagramme de cas d'utilisation globale du troisième Sprint.

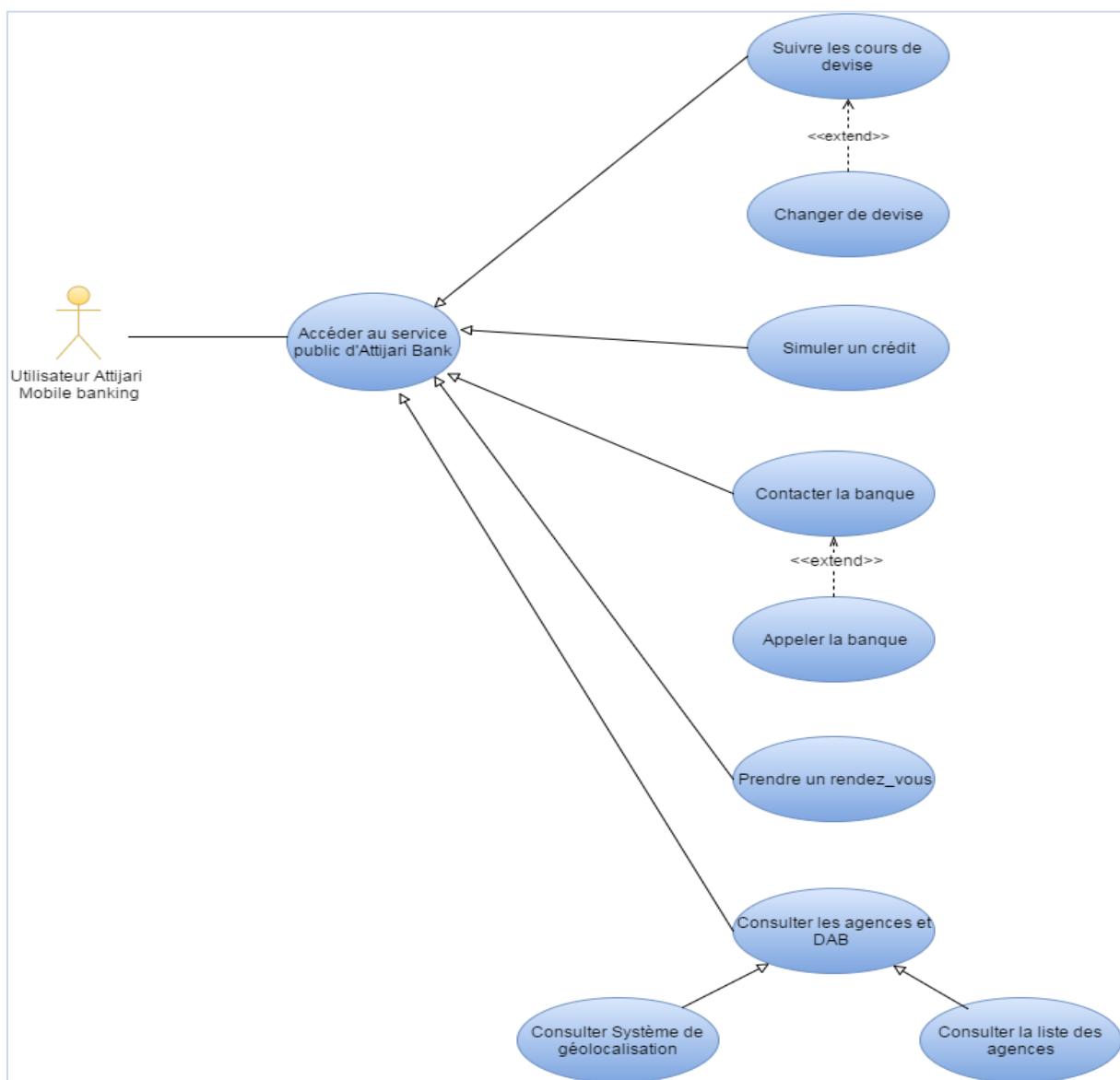


Figure 29. Diagramme de cas d'utilisation globale du Sprint 3

3.2.2. Description textuelle détaillée

Ci-dessous la description textuelle des différents cas d'utilisation de troisième sprint

❖ Cas d'utilisation « suivre les cours de devise »

La description détaillée de cas d'utilisation «Suivre les cours de devise » est donnée par le tableau suivant :

Cas d'utilisation	Suivre les cours de devise
Acteurs	L'utilisateur Attijari Mobile Banking
Pré condition	Le client possède l'application Mobile Banking sur son terminal mobile.
Post-condition	La liste de devise est affichée
Scénario principale	-L'utilisateur sélectionne dans l'interface publique «cours de change» - l'utilisateur est informé sur les prix de vente et d'achat par pays.
Extension	L'utilisateur peut choisir de convertir un montant .Il entre le montant souhaité puis clique sur le bouton «convertir » et le système affiche le résultat.

Tableau 13. Description détaillée de cas d'utilisation «Suivre les cours de devise »

❖ Cas d'utilisation « Simuler un crédit »

La description détaillée de cas d'utilisation «Simuler un crédit» est donnée par le tableau suivant :

Cas d'utilisation	Simuler un crédit
Acteurs	L'utilisateur Attijari Mobile Banking
Pré condition	L'utilisateur possède l'application Mobile Banking sur son terminal mobile.
Post-condition	le résultat de simulation est affiché.
Scénario principale	-L'utilisateur sélectionne «Simuler crédit ». -Le système affiche l'interface appropriée. -L'utilisateur remplit les champs de simulation et clique sur le bouton «Simuler». -Le système affiche le résultat calculé.

Tableau 14. Description détaillée de cas d'utilisation «Simuler un crédit»

❖ Cas d'utilisation « contacter la banque »

La description détaillée de cas d'utilisation «Contacter la banque» est donnée par le tableau suivant :

Cas d'utilisation	Contacter la banque
Acteurs	L'utilisateur Attijari Mobile Banking
Pré condition	Le client possède l'application Mobile Banking sur son terminal mobile.
Post-condition	le contact de banque est affiché.
Scénario principale	-L'utilisateur sélectionne dans l'interface public « Contacter la banque». -Le système affiche l'interface appropriée qui contient les coordonnées de la banque
Extension	L'utilisateur peut choisir d'appeler la banque en cliquant sur « appeler la banque »

Tableau 15. Description détaillée de cas d'utilisation «Contacter la banque»

❖ Cas d'utilisation « prendre un rendez-vous »

La description détaillée de cas d'utilisation «Prendre un rendez-vous» est donnée par le tableau suivant :

Cas d'utilisation	Prendre un rendez-vous
Acteurs	L'utilisateur Attijari Mobile Banking
Pré condition	L'utilisateur possède l'application Mobile Banking sur son terminal mobile.
Post-condition	Demande de rendez-vous envoyée
Scénario principale	-L'utilisateur sélectionne «prendre un rendez-vous». -Le système affiche l'interface appropriée. -L'utilisateur remplit les champs et clique sur le bouton «envoyer». -l'utilisateur sera répondu par un Email.

Tableau 16. Description détaillée de cas d'utilisation «Prendre un rendez-vous»

❖ Cas d'utilisation « Consulter le système de géolocalisation»

La description détaillée de cas d'utilisation «Consulter le système de géolocalisation» est donnée par le tableau suivant :

Cas d'utilisation	Consulter le système de géolocalisation
Acteurs	L'utilisateur Attijari Mobile Banking
Pré condition	L'utilisateur possède l'application Mobile Banking sur son terminal mobile.
Post-condition	Carte géographique affichée
Scénario principale	<ul style="list-style-type: none"> -Dans l'interface publique le client clique sur « DAB et Agence» puis choisit «Localiser sur carte » -Le système affiche l'interface appropriée qui contient une carte où sont affichées les agences.

Tableau 17. Description détaillée de cas d'utilisation «Consulter le système de géolocalisation»

❖ Cas d'utilisation « Consulter la liste de point de contact»

La description détaillée de cas d'utilisation «Consulter la liste de point de contact » est donnée par le tableau suivant :

Cas d'utilisation	Consulter la liste de point de contact
Acteurs	L'utilisateur Attijari Mobile Banking
Pré condition	L'utilisateur possède l'application Mobile Banking sur son terminal mobile.
Post-condition	La liste des agences affichée.
Scénario principale	<ul style="list-style-type: none"> -Dans l'interface publique le client clique sur « DAB et Agence» puis choisit «Localiser sur carte » -Le système affiche l'interface qui contient les régions. -Le client choisit une région. -Le système affiche la liste appropriée.

Tableau 18. Description détaillée de cas d'utilisation «Consulter la liste de point de contact »

3.3. Conception

Maintenant on va passer à la partie conception qui va être présenté par un diagramme de classe participante et un diagramme de séquences, à titre d'exemple nous prenons les deux cas « prendre un rendez-vous» et «suivre les cours de devise»

3.3.1. Diagramme de classe participante

❖ Cas d'utilisation «Suivre les cours de devise»

La figure suivante illustre le diagramme de classe participante du cas d'utilisation «Suivre les cours de devise».

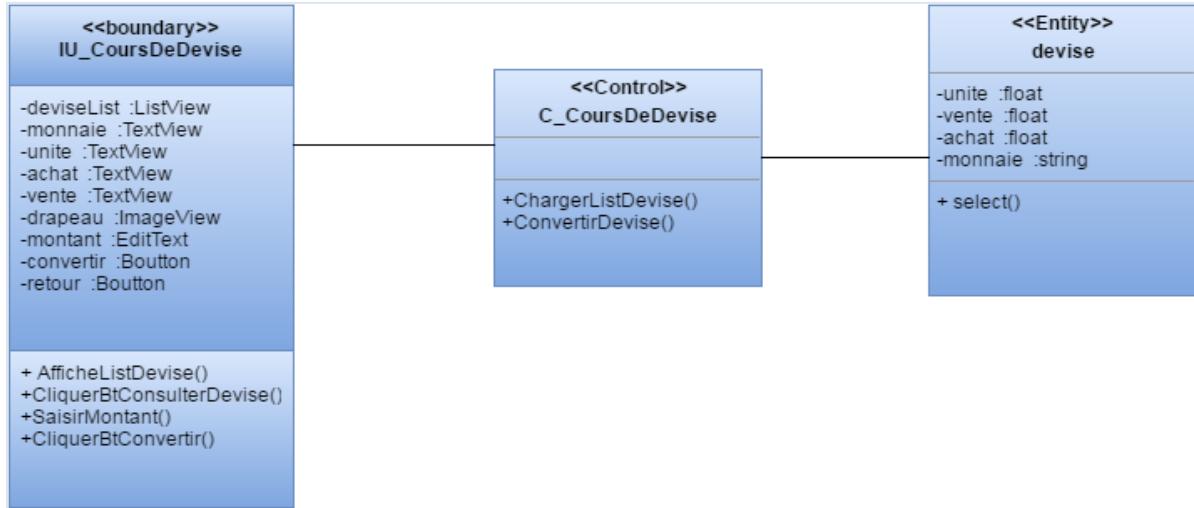


Figure 30. Diagramme de classe participante du cas d'utilisation «Suivre les cours de devise»

❖ Cas d'utilisation «Prendre un rendez-vous»

La figure suivante illustre le diagramme de classe du cas d'utilisation «Prendre un rendez-vous».

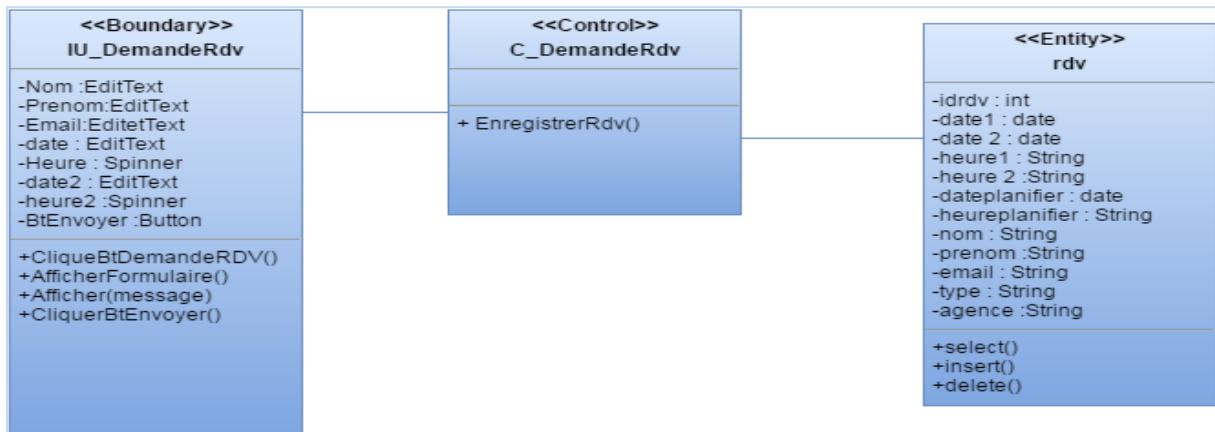


Figure 31. Diagramme de classe du cas d'utilisation «Prendre un rendez-vous».

3.3.2. Diagramme de séquence

❖ Cas d'utilisation «Suivre les cours de devise»

La figure 32 illustre le diagramme de séquences du cas d'utilisation «Suivre les cours de devise».

L'utilisateur de l'application choisit le menu «cours de change», le système affiche la liste de devise à travers le web service «ChargerListDevise».

Pour convertir les devises, l'utilisateur tape le montant à convertir le système affiche le montant après la conversion à travers le web service «ConvertirDevise».

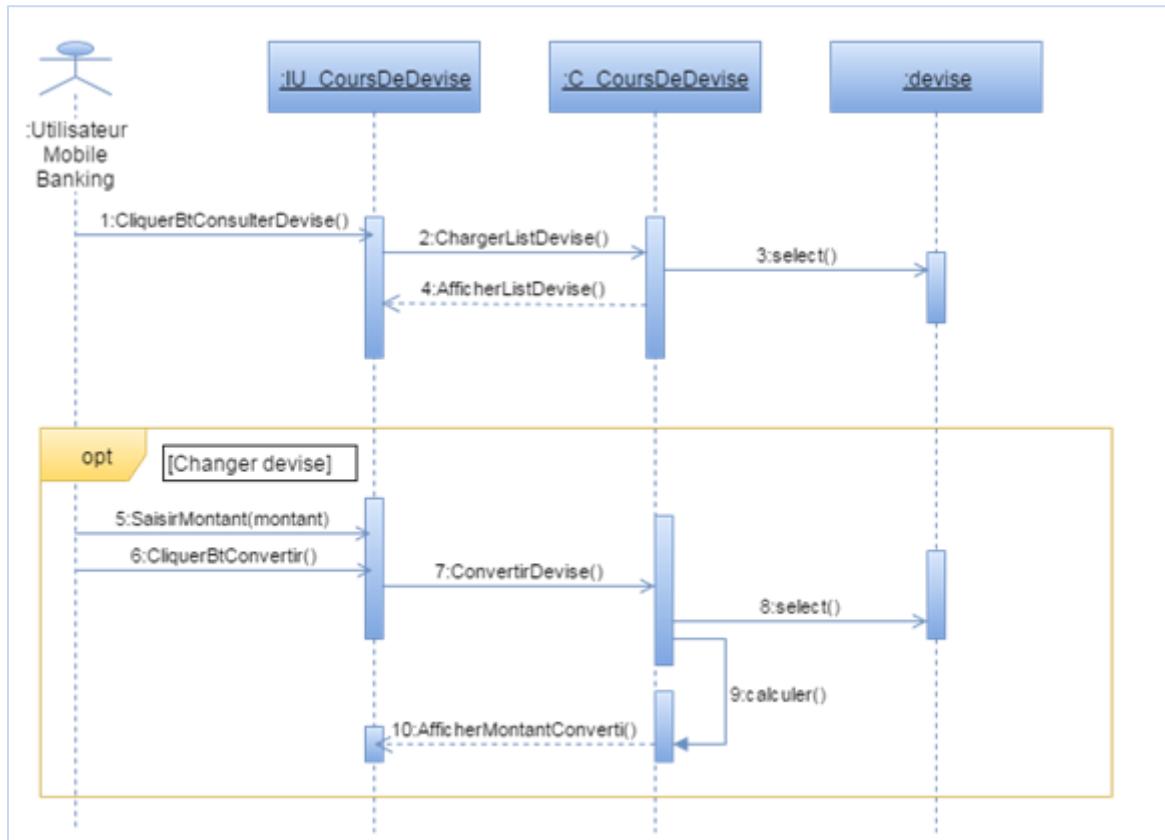


Figure 32. Diagramme de séquences du cas d'utilisation «Suivre les cours de devise»

❖ Cas d'utilisation «Prendre un rendez-vous»

La figure 33 illustre le diagramme de séquences du cas d'utilisation «Prendre un rendez-vous».

L'utilisateur de l'application choisit le menu «rendez-vous», le système affiche le formulaire.

L'utilisateur remplit le formulaire et clique sur «envoyer» le demande sera ajouter avec la fonction «EnregistrerRdv».

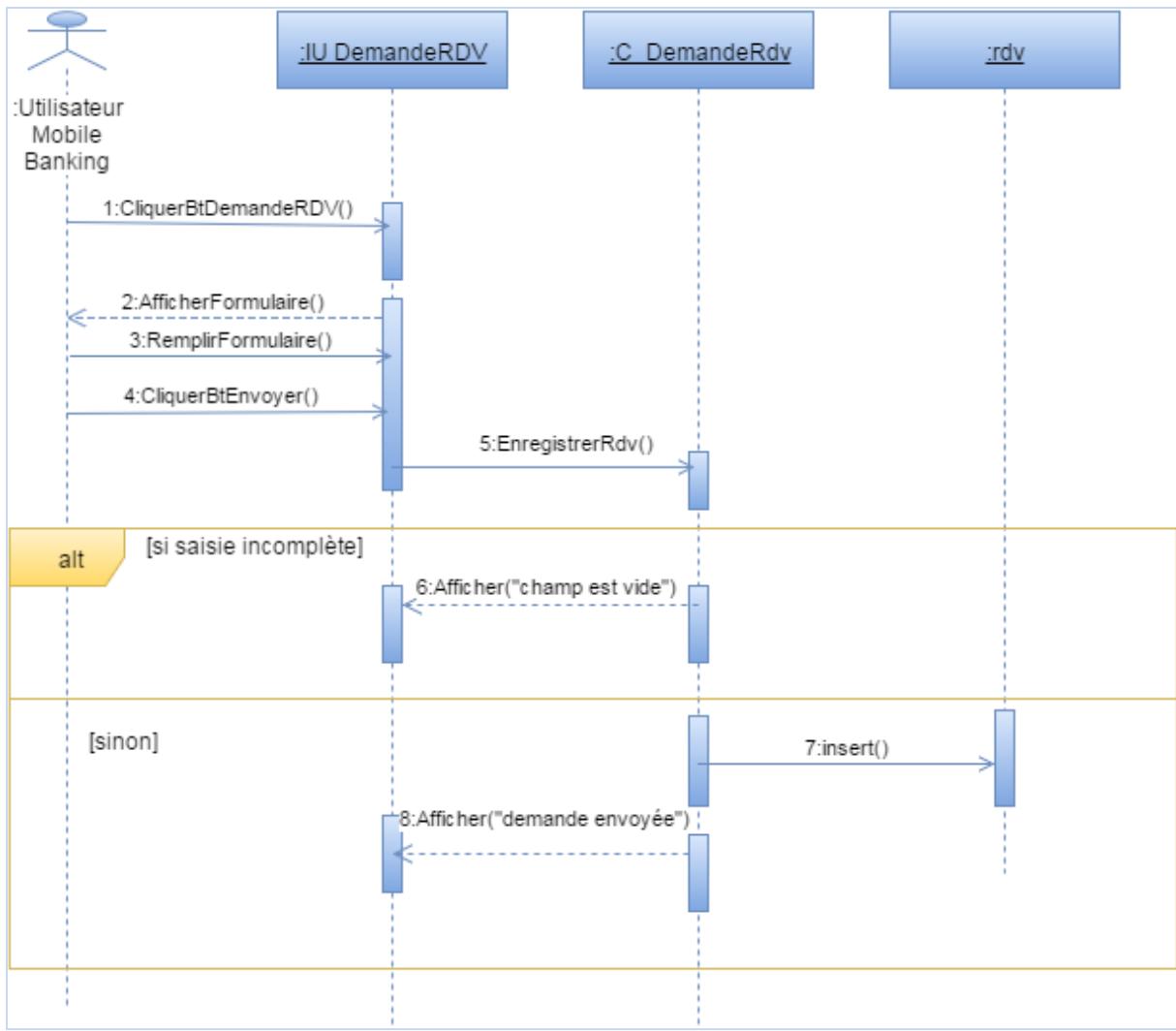


Figure 33. Diagramme de séquences du cas d'utilisation «Prendre un rendez-vous»

3.4. Test

Maintenant on va passer à la phase de test, dans laquelle on va présenter la partie réalisation intégré dans chaque sprint selon la méthode Scrum. Nous illustrons à ce niveau les captures d'écran réalisées.

Lorsque l'utilisateur lance notre application, l'interface principale est apparue.

La figure 34 illustre l'interface du menu public.



Figure 34. Espace public

L'utilisateur Appui sur l'icône « Cours de change » pour consulter le cours de change et changer un montant saisi.

La figure 35 présente l'interface de cours de devise.

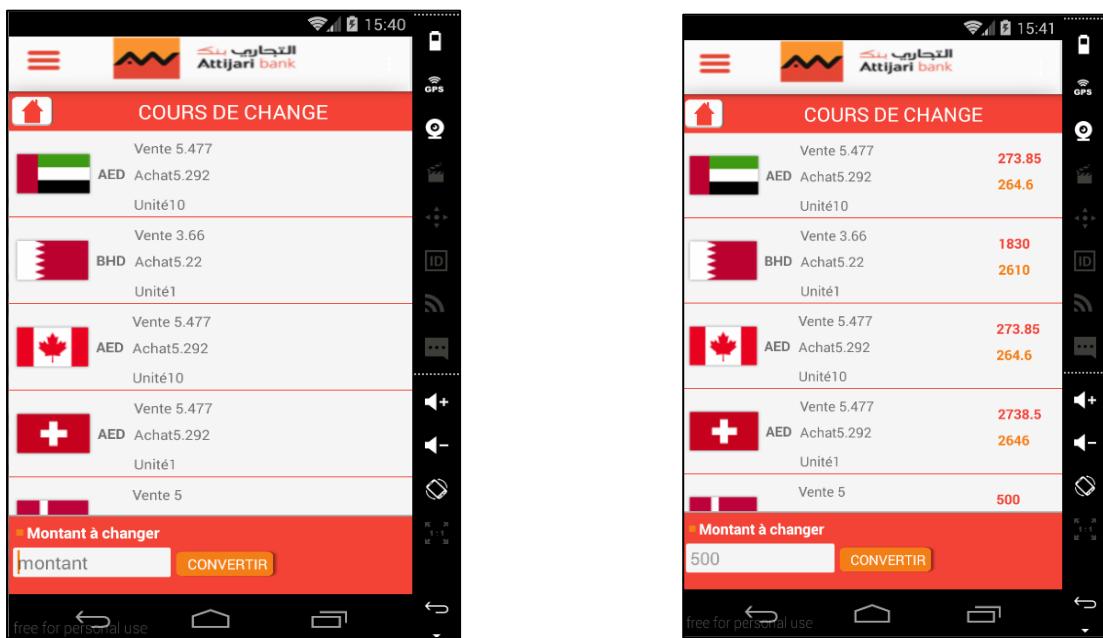


Figure 35. Interface de cours de devise.

L'utilisateur a accès au simulateur de crédit pour avoir une idée sur l'échéance ou le montant mensuel à payer en cas d'une demande.

La figure 36 illustre l'interface de simulation de crédit.

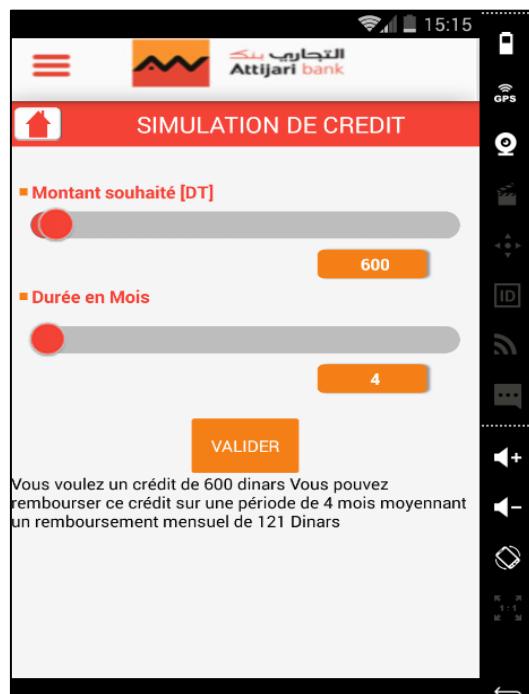


Figure 36. Interface de simulation de crédit.

L'utilisateur peut aussi consulter les agences et leurs coordonnées en cliquant sur l'icône «agence» puis choisit une région.

La figure suivante présente l'interface de consultation des agences par région.

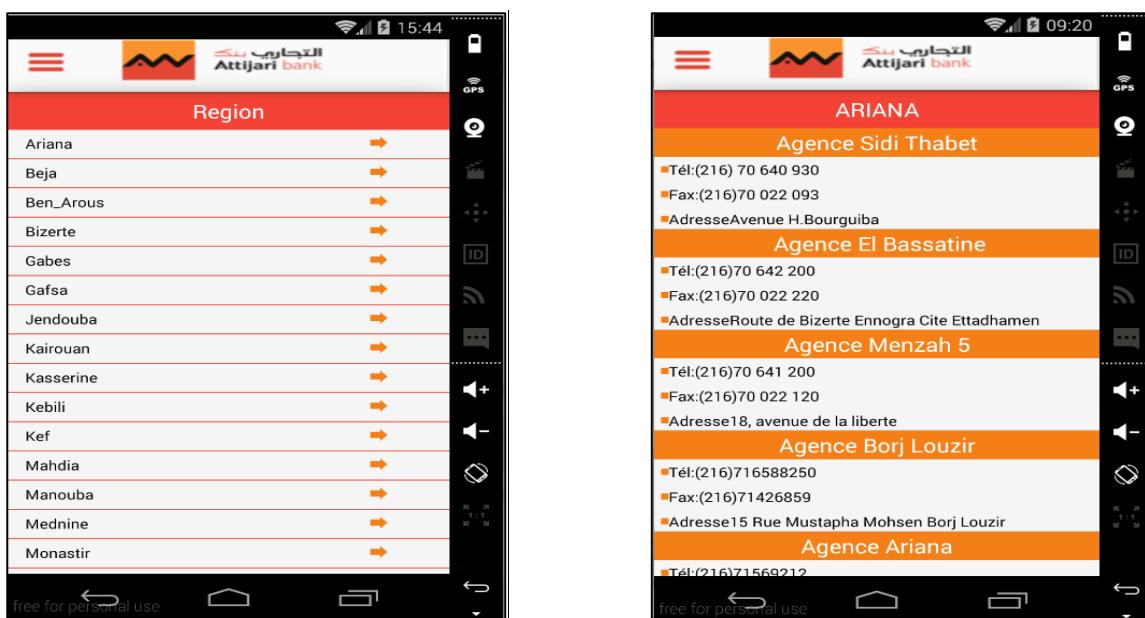


Figure 37. Interface de consultation des agences

L'utilisateur peut aussi visualiser les agences sur la carte géographique et demander un rendez-vous.

La figure 38 présente l'interface de consultation des agences sur carte géographique.

La figure 39 présente l'interface de demande de rendez-vous .

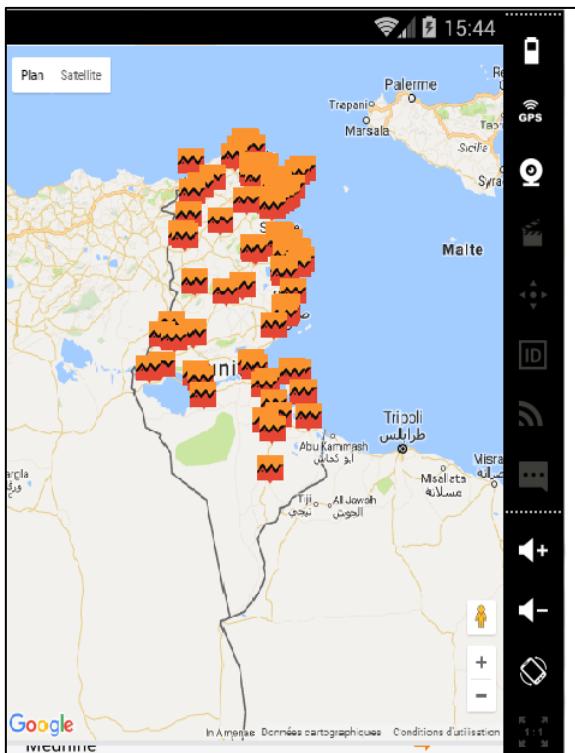


Figure 38. Interface de consultation des agences sur carte

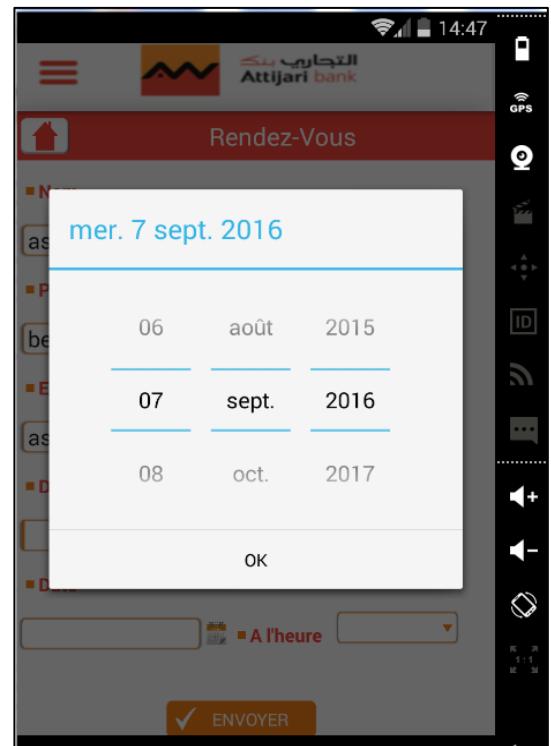


Figure 39. Interface de demande de rendez-vous

Conclusion

A la fin de ce chapitre, nous avons réussi à produire un incrément ayant une valeur pour notre client. Le chapitre suivant sera consacré pour produire un nouveau release couvrant les fonctionnalités de gestion de l'application mobile

Chapitre 4 : Release 2

Introduction

Maintenant on va représenter le quatrième chapitre qui présente le deuxième release qui présente la partie web de notre application et qui contient deux sprints. Nous allons traiter les différentes fonctionnalités d'administrateur pour produire un ensemble d'incrémentations potentiellement livrable. Alors on va définir planification, conception et test de chaque sprint.

1. Sprint 1 : Gérer les agences, les comptes et les abonnés.

Dans chaque sprint, on définit un objectif et les différentes fonctionnalités qui constitueront le backlog puis le diagramme de cas d'utilisation par la suite conception et les différentes interfaces de notre réalisation et en finit par le test.

- ❖ Objectif : l'administrateur peut gérer les agences, les comptes et les abonnés après authentification.
- ❖ Développeurs : Team.

1.1. Le Backlog du sprint 1

Après la définition de la but, dans le tableau ci-dessous on va définir l'histoire de notre Backlog du produit inclure dans le Backlog du sprint.

ID	User Story	Priorité
42	En tant qu'administrateur, je veux m'authentifier	10
16	En tant qu'administrateur, je veux ajouter un ou plusieurs abonnés afin que mon application soit mise à jour.	10
17	En tant qu'administrateur, je veux modifier un ou plusieurs abonnés afin que mon application soit mise à jour.	10
18	En tant qu'administrateur, je veux ajouter un ou plusieurs comptes afin que mon application soit mise à jour.	10
19	En tant qu'administrateur, je veux modifier un ou plusieurs comptes afin que mon application soit mise à jour.	10
20	En tant qu'administrateur, je veux supprimer un ou plusieurs abonnés afin que mon application soit mise à jour.	10
38	En tant qu'administrateur, je veux ajouter un ou plusieurs agences afin que mon application soit mise à jour.	10

39	En tant qu'administrateur, je veux modifier un ou plusieurs agences afin que mon application soit mise à jour.	10
----	----------------------------------------------------------------------------------------------------------------	----

Tableau 19. Backlog du sprint 1

Par la suite on va passer à la spécification fonctionnelle et les deux parties conception et test.

1.2. Spécification fonctionnelle

1.2.1. Diagramme de Cas d'utilisation

Les différents cas d'utilisation de ce sprint sont :

- ❖ S'authentifier : permet à l'administrateur de s'authentifier.
- ❖ Gérer les abonnés : permet à l'administrateur d'ajouter, de modifier ou supprimer un abonné.
- ❖ Gérer les agences : permet à l'administrateur d'ajouter, ou modifier une agence.
- ❖ Gérer les comptes : permet à l'administrateur d'ajouter, de modifier ou supprimer un compte.

La figure ci-dessous illustre le diagramme de cas d'utilisation global du Sprint 1

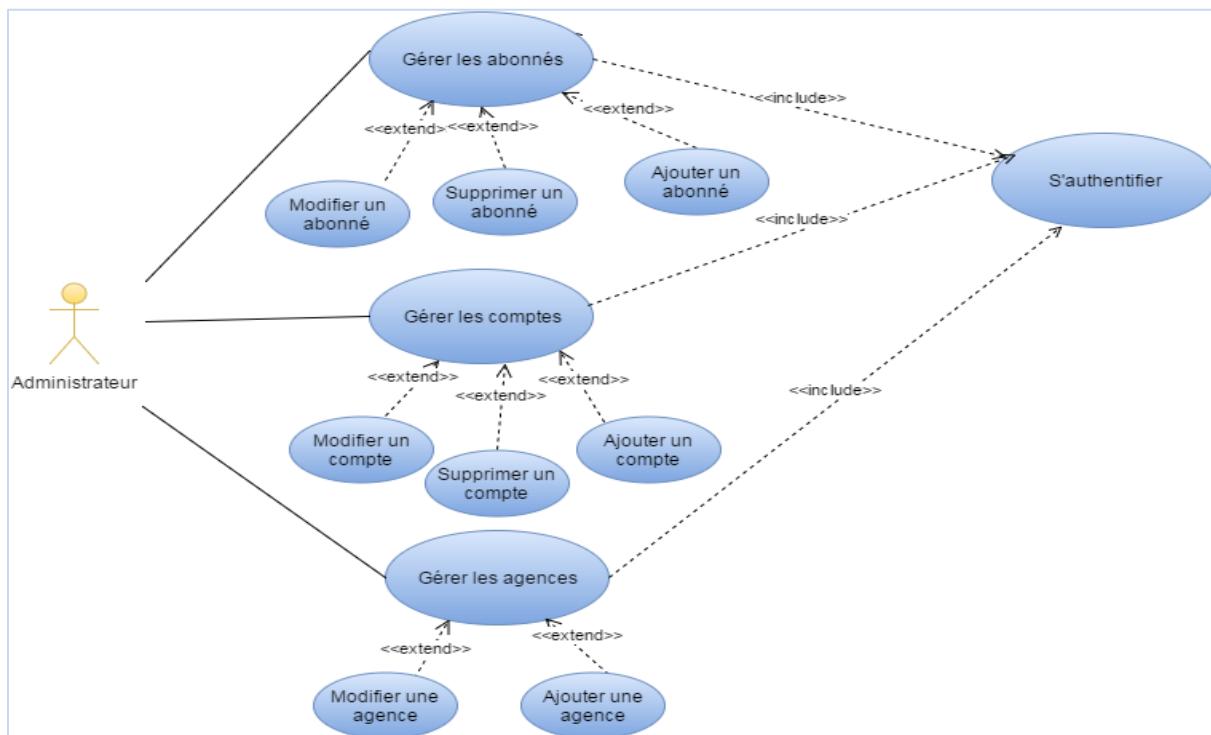


Figure 40. Diagramme de cas d'utilisation global du Sprint 1

1.2.2. Description textuelle détaillée

A titre d'exemple, nous détaillons le cas d'utilisation «Gérer les Comptes» et les autres cas d'utilisation «Gérer les agences» et «Gérer les abonnés» suivent le même principe.

❖ Description détaillée du cas d'utilisation « Gérer les comptes »

Le tableau suivant illustre la description détaillée du cas d'utilisation «Gérer les comptes».

Cas d'utilisation	Gérer les comptes
Acteurs	Administrateur
Pré condition	L'administrateur du système authentifié
Post-condition	Une nouvelle Compte est ajouté, modifié ou supprimé avec succès.
Scénario principal	<p>Le système affiche l'interface principale.</p> <p>-L'administrateur clique sur l'onglet Gérer Comptes.</p> <p>-Le système affiche l'interface appropriée.</p>
Point d'extension : Ajouter un compte	<p>-Le système affiche l'interface relative à «Ajouter un compte »</p> <p>-L'administrateur remplit les champs demandés et clique sur Enregistrer.</p> <p>-Le système enregistre les données</p>
Point d'extension : Modifier un compte	<p>-Le système affiche l'interface relative à «Modifier les coordonnées d'une comptes»</p> <p>-L'administrateur remplit les champs demandés et clique sur Modifier.</p> <p>-Le système enregistre les données</p>
Point d'extension : Supprimer un compte	<p>-L'administrateur clique sur le bouton «supprimer»</p> <p>-Le système supprime les données.</p>

Tableau 20. Description détaillée du cas d'utilisation «Gérer les comptes».

1.3. Conception

Maintenant on va passer à la partie conception qui va être présenté par un diagramme de classe participante et un diagramme de séquences.

1.3.1. Diagramme de classe participante

Le diagramme de classes du cas d'utilisation «Gérer les Comptes» se traduit par les différentes classes participantes à la réalisation de ce dernier.

Cette figure illustre le diagramme de classe participante du cas d'utilisation «Gérer les comptes».

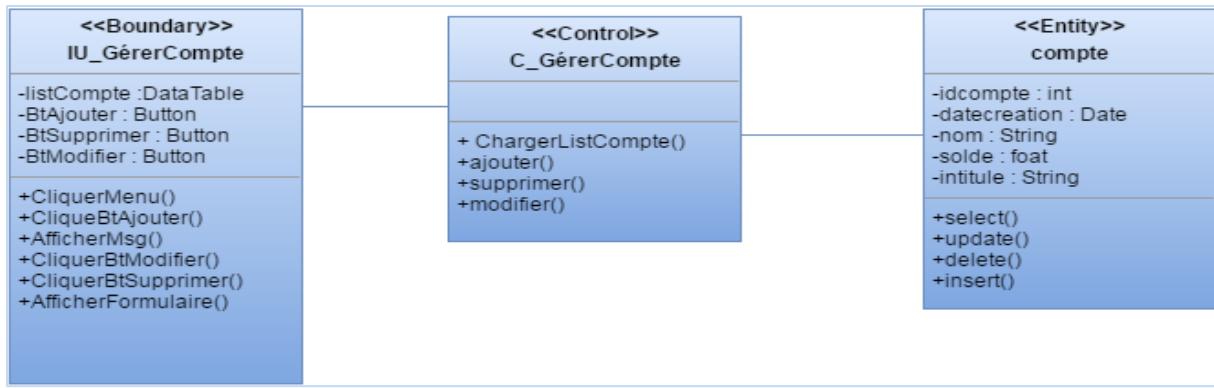


Figure 41. Diagramme de classe participante du cas d'utilisation «Gérer les comptes».

1.3.2. Diagramme de séquence

A partir de ces diagrammes, nous allons décrire le scénario du cas d'utilisation «Gérer les comptes». Ce scénario est déclenché par l'administrateur en cliquant sur «gérer les comptes», le système affiche l'interface approprié avec la liste des comptes et trois cas sont possibles :

- ❖ Ajouter un compte

Si l'administrateur clique sur le bouton «ajouter», le système lui affiche un formulaire à remplir. Il remplit ce formulaire et lance la fonction «Ajouter», le système les insère dans la table et lui affiche un message pour approuver que l'ajout est effectué avec succès, sinon il lui affiche un message d'erreur comme il montre la figure 41 :

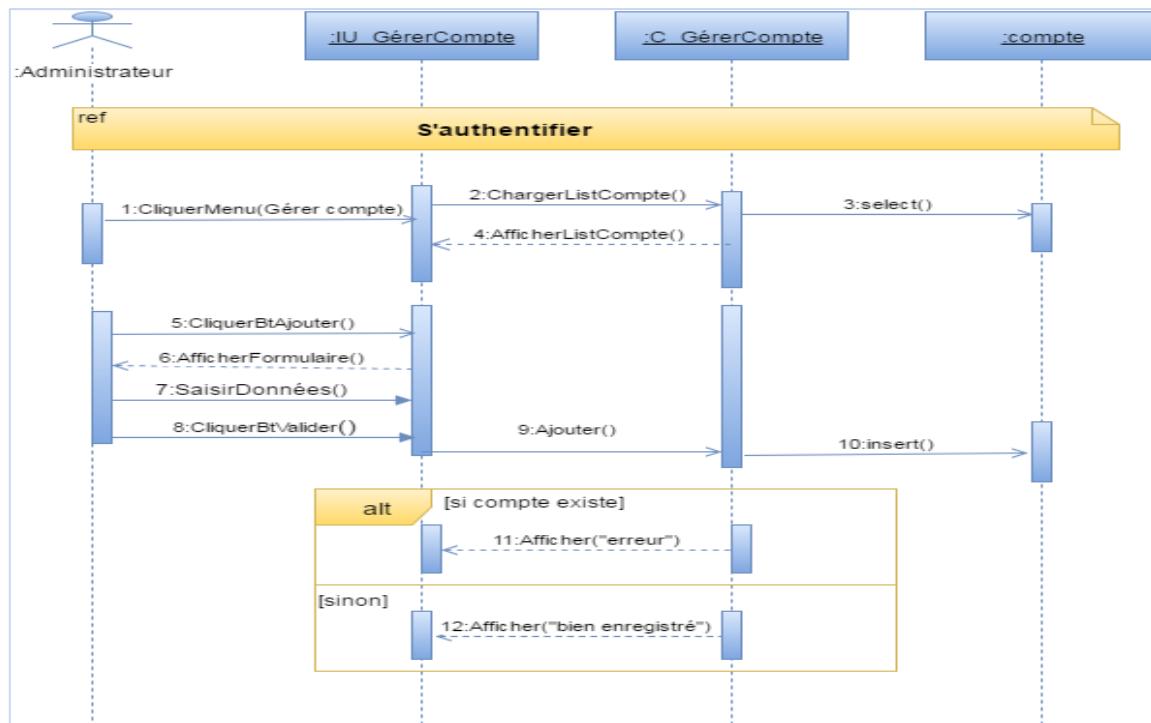


Figure 42. Diagramme de séquence du cas d'utilisation «Gérer les comptes»

❖ Supprimer un compte

Pour que l'administrateur puisse supprimer un compte, il doit lancer la méthode ‘supprimer’.

Dans ce cas notre système supprime le compte sélectionné.

Le scénario de suppression est illustré dans la figure 42 :

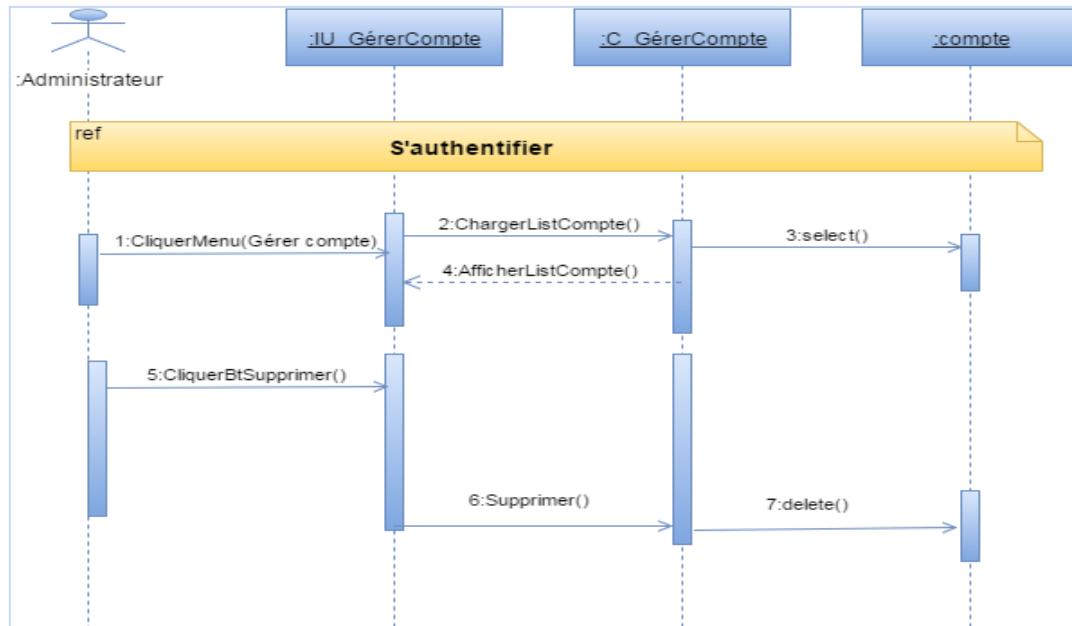


Figure 43. Diagramme de séquence du cas d'utilisation «Supprimer un compte»

❖ Modifier un compte

Le diagramme exposé dans la Figure suivante décrit le scénario possible lors d'une opération de modification. En effet, l'administrateur choisit le compte à modifier. Le système à son tour l'affiche dans un formulaire. L'administrateur saisit les nouvelles informations et valide.

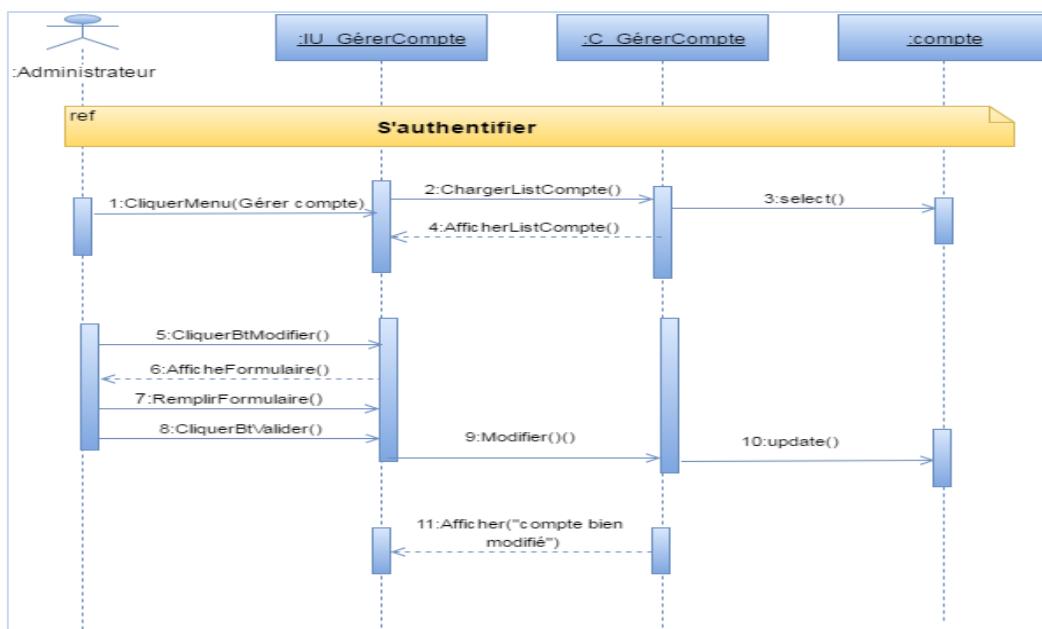


Figure 44. Diagramme de séquence du cas d'utilisation «Modifier un compte»

1.4. Test

Nous présentons dans cette partie notre implémentation de ce sprint à travers des interfaces.

A titre d'exemple nous présentons le cas de gestion de compte bancaire avec suppression, modification et ajout d'un compte.

Une fois l'administrateur clique sur gestion des comptes, la liste des comptes est apparue.

La figure 44 présente l'interface de gestion des comptes bancaires.

The screenshot shows a web-based application titled "Attijari mobile administrateur". At the top, there is a navigation bar with links: "Gestion Client", "Gestion Agence", "Demande Transactionnel", "Gestion Contact", and "Déconnection". Below the title, there is a toolbar with icons for search, sort, and filters, followed by a page number "1" and a total count "5". A "Ajouter" button is located in the top right corner. The main area displays a table with four columns: "ID", "Date de creation", "Intitûde", and "Solde". There are three rows of data:

ID	Date de creation	Intitûde	Solde	
12	2016-08-09 00:00:00.0	compte courant	300.0	<button>supprimer</button> <button>modifier</button>
13	2016-09-09 00:00:00.0	compte principal	300.0	<button>supprimer</button> <button>modifier</button>
20	2016-09-13 00:00:00.0	compte secondaire	200.0	<button>supprimer</button> <button>modifier</button>

Figure 45. Interface de gestion des comptes bancaires

La figure suivante présente l'interface d'ajout d'un compte bancaire.

The screenshot shows a modal dialog box titled "Attijari mobile administrateur" for adding a new bank account. The form contains the following fields:

- "idcompte": Input field containing "17".
- "date de creation": Input field (disabled).
- "intitûde": Input field (disabled).
- "solde": Input field (disabled).
- "Abonne": Input field (disabled).
- "Ajouter": A yellow "Ajouter" button.

On the right side of the dialog, there is a calendar for September 2016, showing the days of the week from Sunday to Saturday and the dates from 1 to 30.

Figure 46. Interface d'ajout d'un compte bancaire.

La figure suivante présente l'interface de modification d'un compte bancaire.

The screenshot shows a window titled 'Modification d'un compte bancaire'. It contains five input fields with labels: 'idcompte:' (containing '12'), 'date de creation:' (containing '2016-08-09'), 'intitule:' (containing 'compte courant'), 'solde:' (containing '300.0'), and 'Abonne' (containing '1241992'). Below these fields is a dropdown menu. At the bottom left is a yellow 'Ajouter' button.

Figure 47. Interface de modification d'un compte bancaire.

2. Sprint 2 : Gérer les virements, Gérer les cartes, Gérer les chéquiers, Gérer les contacts

Dans chaque sprint, on définit un objectif et les différentes fonctionnalités qui constitueront le Backlog puis le diagramme de cas d'utilisation par la suite conception et les différentes interfaces de notre réalisation et en finit par le test.

- ❖ Objectif : l'administrateur peut gérer les demandes de virement, carte et chéquier, rendez-vous, et les messages
- ❖ Développeurs : Team.

2.1. Le Backlog du sprint 2

Après la définition de la but, dans le tableau ci-dessous on va définir l'histoire de notre backlog du produit inclure dans le backlog du sprint.

ID	User Story	Priorité
21	En tant qu'administrateur, je veux accepter la demande d'une carte afin de répondre à la demande de client.	10
22	En tant qu'administrateur, je veux refuser la demande d'une carte afin de répondre à la demande de client	10
23	En tant qu'administrateur, je veux supprimer la demande d'une carte afin de mettre à jour les demandes.	10
24	En tant qu'administrateur, je veux accepter la demande d'opposition d'une carte afin de répondre à la demande de client.	10

25	En tant qu'administrateur, je veux refuser la demande d'opposition d'une carte afin de répondre à la demande de client.	10
26	En tant qu'administrateur, je veux supprimer la demande d'opposition d'une carte afin de mettre à jour les demandes.	10
27	En tant qu'administrateur, je veux accepter la demande d'un chéquier afin de répondre à la demande de client.	10
28	En tant qu'administrateur, je veux refuser la demande d'un chéquier afin de répondre à la demande de client.	10
29	En tant qu'administrateur, je veux supprimer la demande d'un chéquier afin de mettre à jour les demandes.	10
30	En tant qu'administrateur, je veux accepter la demande d'opposition d'un chéquier afin de répondre à la demande de client.	10
31	En tant qu'administrateur, je veux refuser la demande d'opposition d'un chéquier afin de répondre à la demande de client.	10
32	En tant qu'administrateur, je veux supprimer la demande d'opposition d'un chéquier afin de mettre à jour les demandes.	10
33	En tant qu'administrateur, je veux valider un rendez-vous afin planifier un rendez-vous.	10
34	En tant qu'administrateur, je veux consulter la liste de rendez-vous planifié afin de vérifier les données.	10
35	En tant qu'administrateur, je veux consulter la liste des messages afin de répondre aux réclamations des clients.	10
36	En tant qu'administrateur, je veux accepter la demande d'un virement afin de répondre à la demande de client.	10
37	En tant qu'administrateur, je veux refuser la demande d'un virement afin de répondre à la demande de client.	10
42	En tant qu'administrateur d'Attijari Mobile Banking, je veux m'authentifier afin d'accéder à mon espace.	10

Tableau 21. Backlog du sprint 2

2.2. Analyse et Spécification

2.2.1. Diagramme de cas d'utilisation global de sprint 2

Les différents cas d'utilisation de ce sprint sont :

- ❖ Gérer les cartes : permet à l'administrateur d'accepter, de refuser ou de supprimer une demande d'obtention ou d'opposition d'une carte
- ❖ Gérer les chéquiers : permet à l'administrateur d'accepter, de refuser ou de supprimer une demande d'obtention ou d'opposition d'un chéquier
- ❖ Gérer les virements : permet à l'administrateur accepter ou refuser une demande d'un virement.
- ❖ Gérer les contacts : permet à l'administrateur de valider un rendez-vous ou de consulter les anciens rendez-vous et la liste des messages.

La figure suivante présente le diagramme de cas d'utilisation globale de sprint 2.

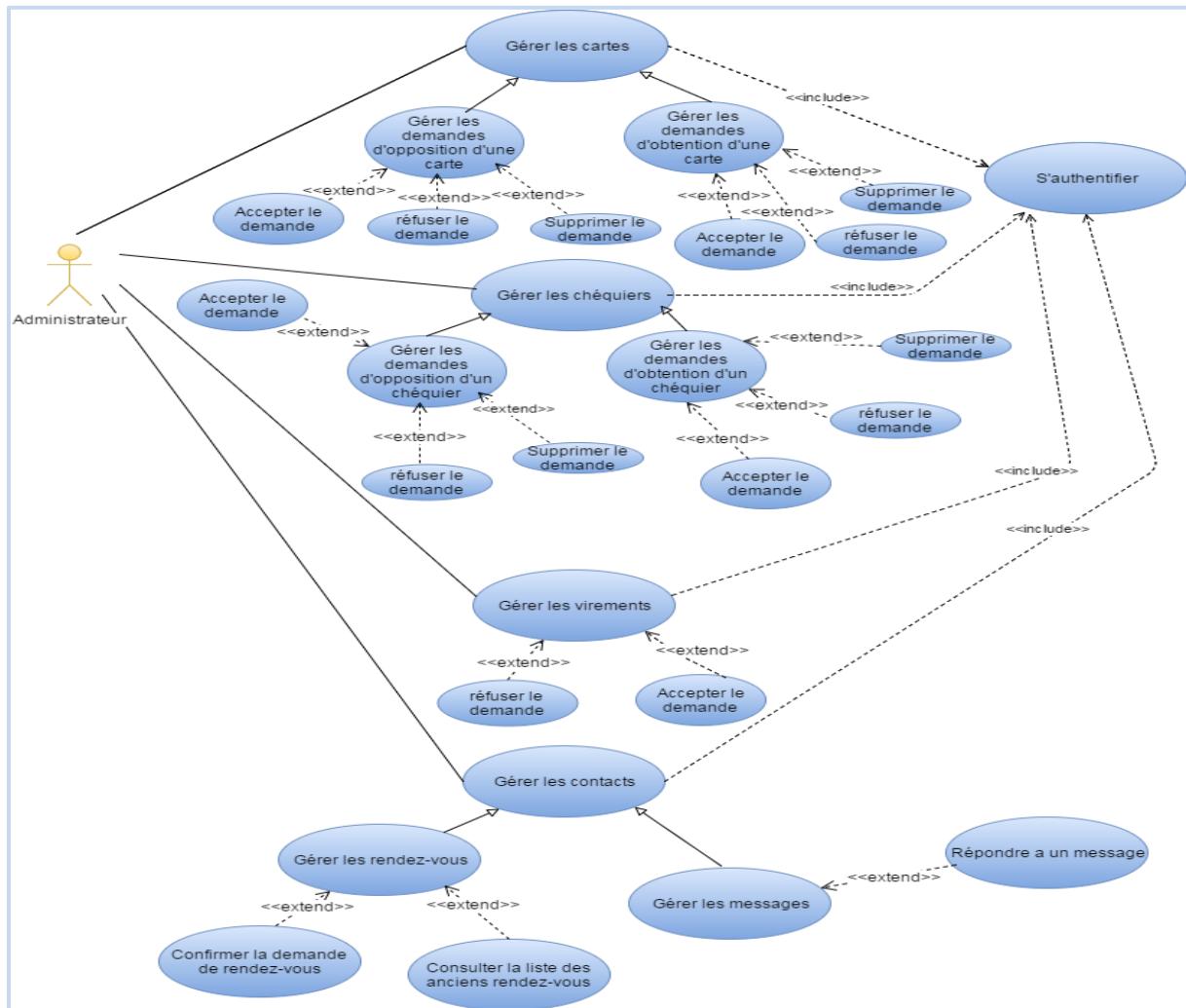


Figure 48. Diagramme de cas d'utilisation globale de sprint 2

2.2.2. Description textuelle détaillée

A titre d'exemple, nous détaillons le cas d'utilisation «Gérer les demandes d'obtention d'une carte » et «Gérer les rendez-vous» et les autres cas d'utilisation «Gérer les chéquiers», «Gérer les messages» et «Gérer les cartes» suivent le même principe.

❖ Description détaillée du cas d'utilisation «Gérer les demandes d'obtention d'une carte»

La description détaillée du cas d'utilisation «Gérer les demandes d'obtention d'une carte » est donnée par le tableau suivant :

Cas d'utilisation	Gérer les demande d'obtention d'une carte
Acteurs	Administrateur
Pré condition	L'administrateur du système authentifié
Post-condition	Une demande est acceptée ou refusé.
Scénario principal	<p>Le système affiche l'interface principale.</p> <p>-L'administrateur clique sur l'onglet Gérer Carte.</p> <p>-Le système affiche l'interface appropriée avec la liste des demandes d'obtention d'une carte</p> <p>-l'administrateur accepte ou refuse demande.</p>

Tableau 22. Description détaillée du cas d'utilisation «Gérer les demandes d'obtention d'une carte »

❖ Description détaillée du cas d'utilisation «Gérer les rendez-vous»

La description détaillée du cas d'utilisation «Gérer les rendez-vous» est donnée par le tableau suivant :

Cas d'utilisation	Gérer les rendez-vous
Acteurs	Administrateur
Pré condition	L'administrateur du système authentifié
Post-condition	Un rendez-vous est validé.
Scénario principal	<p>-L'administrateur clique sur l'onglet Gérer rendez-vous.</p> <p>-Le système affiche l'interface appropriée avec la liste des rendez-vous.</p>

Tableau 23. Description détaillée du cas d'utilisation «Gérer les rendez-vous»

2.3. Conception

Maintenant on va passer à la partie conception qui va être présenté par un diagramme de classe participante et un diagramme de séquences.

2.3.1. Diagramme de classe participante

❖ cas d'utilisation «Gérer les demandes d'obtention d'une carte »

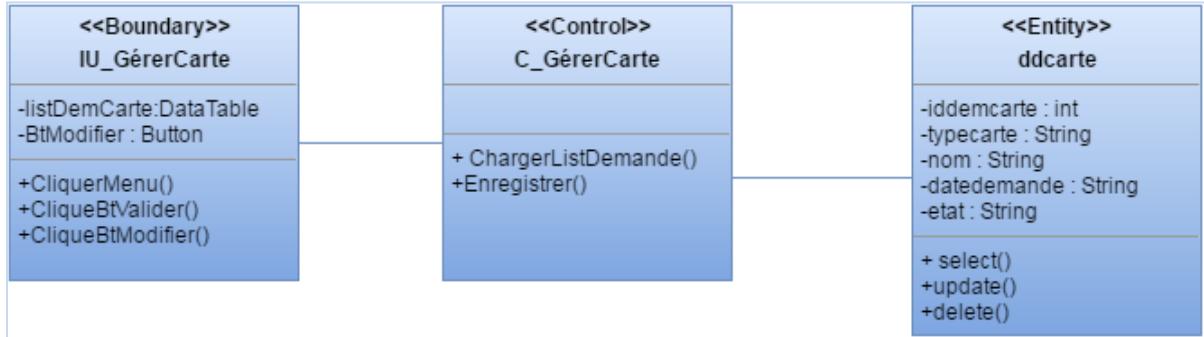


Figure 49. Diagramme de classe participante du cas d'utilisation «Gérer les demandes d'obtention d'une carte »

❖ cas d'utilisation «Gérer les rendez-vous»

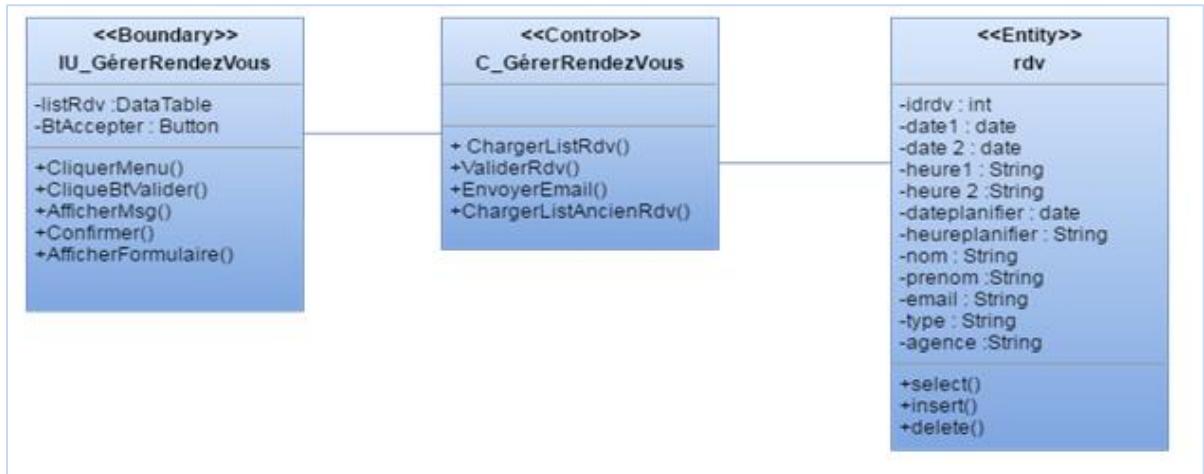


Figure 50. Diagramme de classe participante du cas d'utilisation «Gérer les rendez-vous »

2.3.2. Diagramme de séquence

❖ cas d'utilisation «Gérer les demandes d'obtention d'une carte »

L'administrateur gère les demandes d'obtention et d'opposition des cartes ; Dans les deux cas il peut soit accepter une demande ou la refuser. Egalement, l'administrateur peut consulter les demandes qui sont déjà traité auparavant. Et encore il peut effectuer une recherche avancée par code client, nom et d'autres critères.

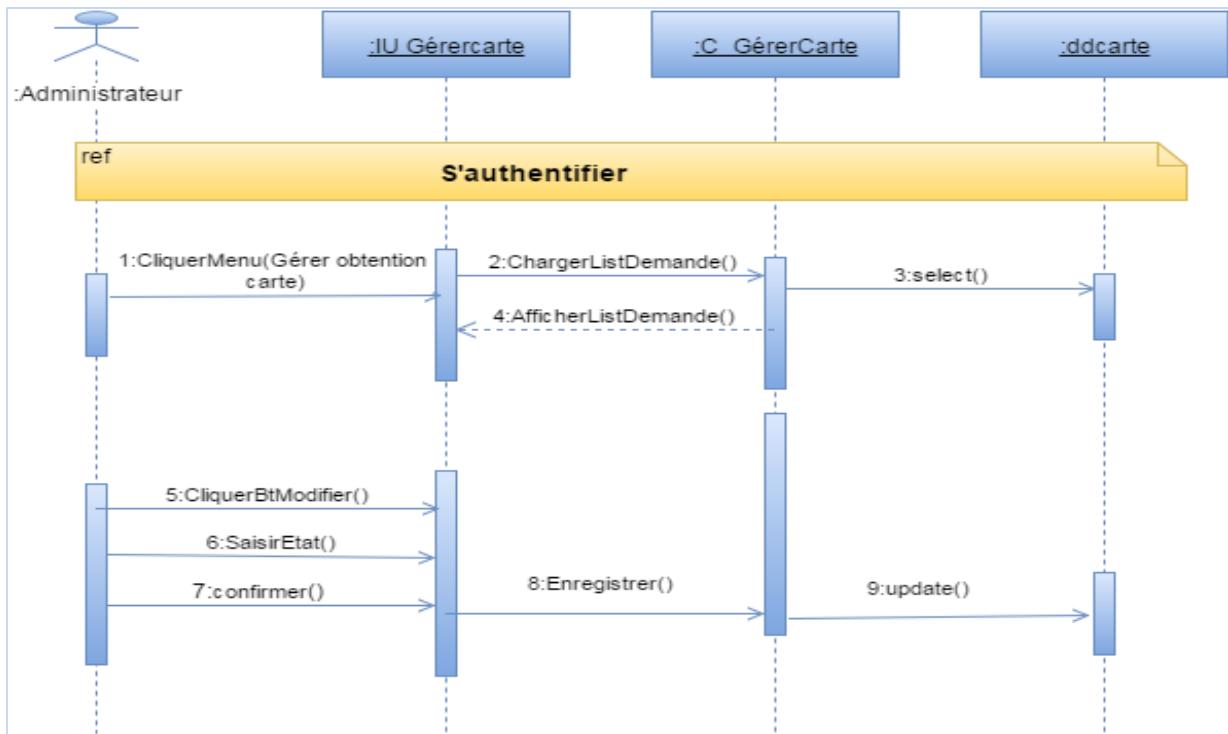


Figure 51. Diagramme de séquence du cas d'utilisation généré «Gérer les demandes d'obtention d'une cartes ».

❖ cas d'utilisation «Gérer les rendez-vous»

Après authentification, l'administrateur aura accès à l'interface de gestion des rendez-vous où il pourra soit confirmer la demande de rendez-vous dans l'une des dates proposés par le client ou proposer une autre date selon la disponibilité du chargé clientèle de son agence.

D'autre part, l'administrateur peut consulter la liste des anciens rendez-vous. Dans les deux cas, il peut faire une recherche avancée pour faciliter sa tâche.

La figure 51 présente le scénario de gestion des rendez-vous :

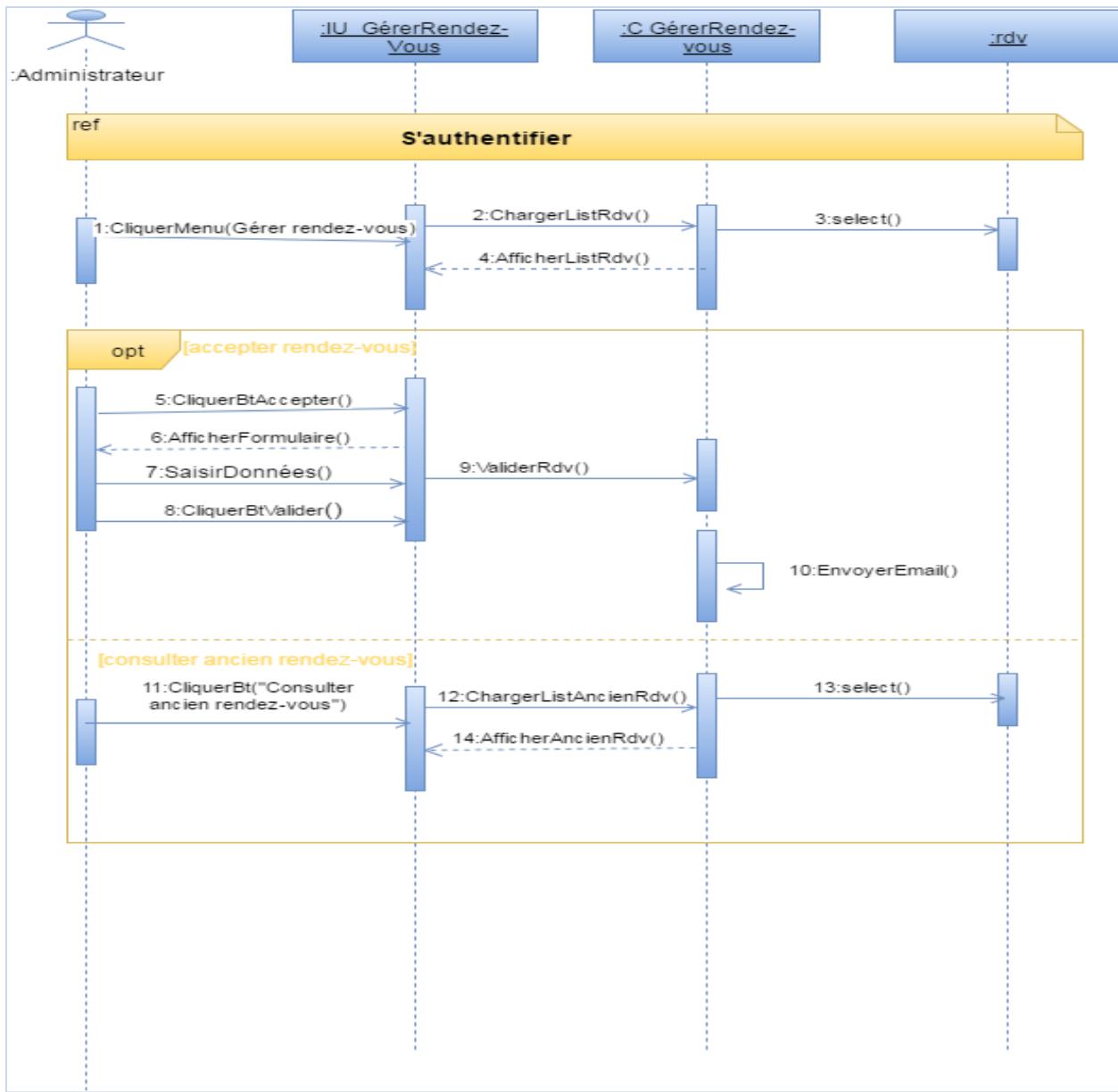


Figure 52. Le diagramme de séquence du cas d'utilisation généré «Gérer les rendez-vous».

2.4. Test

Nous présentons dans cette partie notre implémentation de ce dernier sprint à travers des interfaces, nous présentons les deux cas «Gérer les demandes d'obtention d'une carte» et «Gérer les rendez-vous»

Une fois l'administrateur choisit la gestion de demande d'une carte le système affiche la liste des demandes et l'administrateur soit accepter demande ou supprimer.

La figure 52 illustre l'interface de gestion d'obtention d'une carte



Figure 53. Interface de gestion d'obtention d'une carte

Une fois l'administrateur clique sur bouton «éditer» un formulaire est affiché avec les informations d'une demande sélectionnée et choisit d'accepter ou refuser la demande comme il montre la figure 53 suivante :

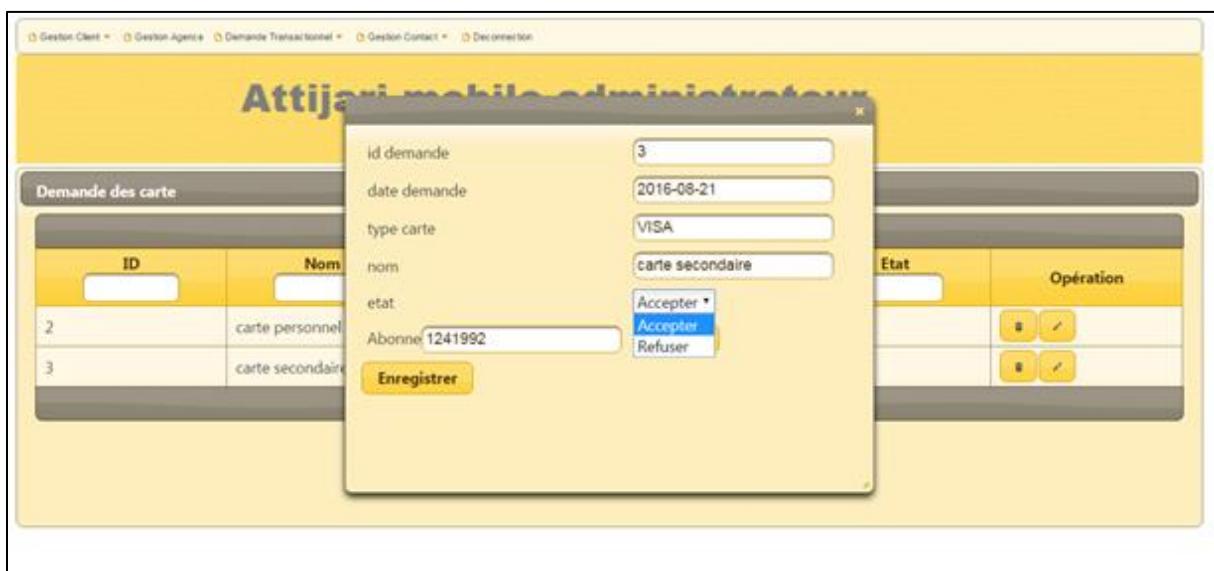


Figure 54. Interface demande de carte

Une fois l'administrateur choisit la gestion des rendez-vous, une liste des rendez-vous est affichée.

S'il sélectionne une ligne, un formulaire est affiché avec les informations d'une demande sélectionnée, il saisit date et heure planifié puis valide la demande, un email sera envoyé au destinataire comme il montre les figures 54 et 55.



Figure 55. Interface gestion des rendez-vous

Email	asouma1292@gmail.com
Nom	asma
Prenom	benab
Heure 1	10
Heure 2	10
Heure planifié	10 h
Agence	ariana
date planifié	12/08/2016
Type	client

Enregistrer

Figure 56. Interface d'un rendez-vous sélectionné

Une fois l'administrateur accepte la demande un email sera envoyée.

Pour le test nous utilisons notre propre email comme le montre la figure suivante :

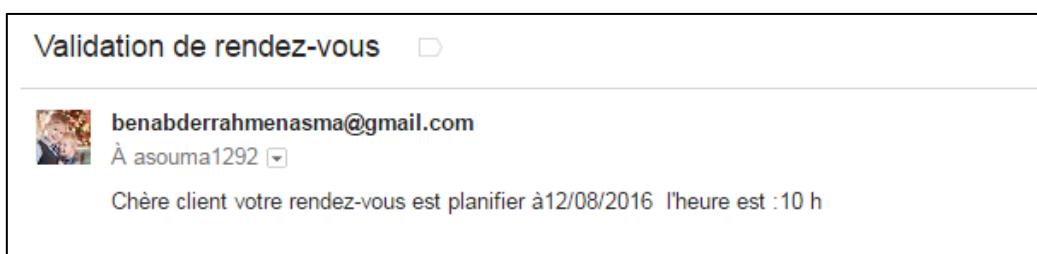


Figure 57. Validation de rendez-vous

Conclusion

Avec ce sprint nous avons livré à notre client le deuxième incrément et par la suite nous avons terminé la réalisation de notre Backlog de produit.

Conclusion général

L'objectif de notre projet de fin d'études est la conception et la réalisation d'une application web et mobile de gestion des opérations bancaires «Attijari Mobile Banking».

Afin d'atteindre cet objectif, nous avons commencé par une étude théorique qui nous a permis de comprendre les concepts relatifs à notre projet. Puis nous avons spécifié les besoins fonctionnels et non fonctionnels auxquels devra répondre notre système. Par la suite grâce à l'application de la méthode Scrum, nous avons pu planifier notre projet en le découplant en 2 releases et chaque release est découpé à des ensembles des sprints. Ensuite, nous avons suivi le même enchainement pour élaborer chaque sprint, en commençant par la phase de spécification des besoins puis l'analyse, la conception et enfin la réalisation.

Au cours de ce stage, on a eu l'occasion d'avoir une bonne expérience enrichissante dont on a profité dans le domaine professionnel et d'être responsable, autonome et productif. On est maintenant capable de faire face aux défis et aux difficultés de la vie professionnelle et on est devenu plus expérimentés.

En plus, ce projet était une occasion pour la mise en pratique de nos connaissances acquises tout au long de notre formation, il nous a donné l'occasion d'utiliser, des nouvelles technologies comme JEE et Android, et Framework de développement comme Hibernate, Spring Security, JSF et PrimeFaces, et d'acquérir des nouvelles connaissances en Java, CSS, JavaScript, PHP.

Cependant, notre travail peut être amélioré et on peut le maintenir sur plusieurs axes, on a proposé quelques perspectives d'amélioration et d'évolution de notre travail, une première amélioration possible pour notre partie mobile est d'ajouter des fonctionnalités qui permettent à l'utilisateur d'ajouter des commentaires, consulter leur historique, ajouter des photos, etc. Une seconde amélioration est d'ajouter un système de gestion des habilitations pour les administrateurs de l'application web. Une autre amélioration au niveau de la compatibilité est de développer une version iOS et Windows Phone.

Nous espérons enfin que le travail que nous avons effectué a été à la hauteur de la confiance qui nous a été donnée.

Webographie

¹ http://www.attijaribank.com.tn/Fr/Nous-conna%C3%AEtre_11_15

Consulter le 15/03/2016

² <https://innovation.orange.tn/publication/le-suivi-davancement-des-projets>

Consulter le 20/03/2016

³ <https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur>

Consulter le 24/03/2016

⁴ <https://fr.wikipedia.org/wiki/NetBeans>

Consulter le 01/07/2016

⁵ <http://portfolio-chatillon.weebly.com/android-studio.html>

Consulter le 01/07/2016

⁶ <http://www.commentcamarche.net/download/telecharger-34100729-genymotion>

Consulter le 01/07/2016

⁷ https://fr.wikipedia.org/wiki/Apache_Tomcat

Consulter le 01/07/2016

⁸ <http://www.prologue-solutions.tn/technologies/65-mysql.html>

Consulter le 01/07/2016

⁹ <http://www.futura-sciences.com/tech/definitions/internet-java-485/>

Consulter le 04/07/2016

¹⁰ <http://www.prologue-solutions.tn/technologies/64-php.html>

Consulter le 04/07/2016

¹¹ <http://www.bases-as3.fr/flex-json>

Consulter le 07/07/2016

¹² https://fr.wikipedia.org/wiki/Service_web

Consulter le 07/07/2016

¹³ https://fr.wikipedia.org/wiki/Java_EE

Consulter le 18/07/2016

¹⁴ http://gardeux-vincent.eu/Documents/ProjetJEE/LLM_Hibernate_JSF/Hibernate.html

Consulter le 20/07/2016

¹⁵ <https://sites.google.com/site/herbinaudrey/veille-technologique/mapping-relationnel/jpa>

Consulter le 22/07/2016

¹⁶ https://fr.wikipedia.org/wiki/JavaServer_Faces

Consulter le 25/07/2016

¹⁷ https://app.assembla.com/spaces/comsoft/wiki/Prime_Faces/print

Consulter le 26/07/2016

تلخيص

تم تفزيذ هذا المشروع في التجاري بنك في نطاق الحصول على شهادة التخرج في الهندسة الإعلامية.

الهدف الرئيسي من هذا الترخيص هو تصميم وانجاز تطبيق واب والتليفون المحمول تمكن من إدارة الخدمات المصرفية. كما أن هذا التطبيق يسمح للحريف من التمتع بخدمات البنك من الهاتف الجوال.

كلمات المفاتيح : الخدمات المصرفية

Résumé

Le présent travail effectué au sein de la société Attijari Bank s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du diplôme national d'ingénieur en informatique.

L'objectif principal de ce travail consiste à concevoir et réaliser une application web sur la plateforme JEE et mobile sur Android pour la gestion des opérations bancaires.

Mots clés : SCRUM, JAVA, JEE, ANDROID, JSF, Hibernate, opération bancaire.

Abstract

We have performed an internship within the company Attijari Bank in order to obtain the national computer engineering degree.

The main objective of this work is to implement a web application on JEE and Android mobile platform for the management of banking.

Keywords: SCRUM, Java, JEE, ANDROID, JSF, Hibernate, transaction banking