Université Cheikh Anta DIOP de Dakar

Faculté des Sciences et Techniques

Département Mathématiques et Informatique

Laboratoire d'Algèbre de Cryptologie de Géométrie Algébrique et Applications

LACGAA

Master Transmission de Données et Sécurité de l'Information

Thème :

Développement d'un Client Mobile Bancaire Sous C#

Présenté et soutenu par:
Abdoulaye DIA

Sous la direction de:
M. El Hadji Ousmane DIALLO

Jury :

Président : Professeur. Ismaila Diouf	UCAD
Membres : Docteur Demba Sow	UCAD
M. El Hadji Ousmane Diallo	ESP
M. Khalifa Ahmedou	UCAD
M. ADJEOUA HAIKREO	UCAD

Année Académique 2016 – 2017

Table des matières

Table des Figures :

Remerciements

Je tiens à présenter mes sincères remerciements au directeur du labo Pr Mamadou Sangharé, le doyen de la faculté des Sciences et Techniques ainsi que le responsable de la M2TDSI Pr Oumar Diankha de m'avoir donné la chance de pouvoir poursuivre mes études au sein du labo.

Je remercie également tous les enseignants de la formation particulièrement ceux de Master pour toutes les connaissances qu'ils m'ont transmises et leur soutien pour me permettre de m'améliorer chaque jour.

Je remercie spécialement Mr El Hadji Ousmane Diallo enseignant en même temps mon professeur encadreur d'avoir consacré de son temps si précieux pour m'aider à réaliser ce projet.

Je présente enfin mes remerciements à tous ceux qui m'ont soutenu mes parents, mes frères, mes sœurs, mes camarades de classes et tous ceux que j'avais oublié de citer.

Dédicaces

Je rends grâce à Dieu, et dédie ce travail à :

Mes parents, mes frères et sœurs ;

Mes amis et camarades de promotion de TDSI ;

Aux enseignants de la TDSI ;

A Tous ceux que je n'ai pas pu citer.

Avant-propos

Le laboratoire LACGAA a pour objectifs : d'une part, la formation à la recherche fondamentale et appliquée dans les domaines de la Cryptographie, de la Théorie des codes, de l'Algèbres et de leurs applications(en logique, en informatique, en sécurité de l'information, en biologie, en robotique etc.) par :des enseignements pour les jeunes doctorants durant leur première année d'inscription en thèse ;l'encadrement des jeunes doctorants durant toute la durée de leur thèse ;la mise en place d'un cadre approprié pour l'épanouissement des jeunes doctorants ;d'autre part, l'organisation de la recherche par la mise en place d'un cadre approprié pour l'épanouissement des chercheurs et le développement de la recherche.

Les principaux domaines de recherche sont l'algèbre et ses différentes applications :

Algèbre commutative, algèbre non commutative, algèbre associative, algèbre non associative ; Géométrie algébrique commutative et non commutative, Homologie et Co homologie, Théorie algébrique et analytique des nombres Cryptographie, Théorie des Codes correcteurs d'erreurs, Théorie du signal Informatique théorique, Sécurité informatique etc.

Directeur du labo : Professeur Mamadou Sangharé

PRESENTATION DE LA FORMATION :

Master Professionnel MTDSI

1. **Théorie de l'information et du codage**
- Théorie du codage algébrique et applications
- Compression des données

2 Informatique appliquée

- Langage de programmation orienté objet : Java
- Administration Oracle et développement d'applications
- Méthodes d'analyses (Merise, UML,..)
- Technologies XML
- Administration réseaux et systèmes sous Linux et Windows avancée
- Développement Web ASP
- Bases de données réparties,
- Ingénierie des systèmes répartis
- Théorie du Signal et Télécoms
- Applications distribués et Java/J2EE

3 Sécurité

- Cryptographie appliquée
- Sécurité des réseaux
- Sécurité des systèmes d'exploitation
- Sécurité des bases de données

4. **Autres :**
- Anglais
- Projet

Master 2 Recherche

1. **Mathématiques fondamentales :**
- Géométrie algébrique et calcul formel
- Théorie des anneaux et des modules
- Calcul différentiel et Géométrie différentiel
- Théorie des catégories et algèbre homologique
2. **Théorie de l'information et du codage**
- Théorie des codes
3. **Sécurité :**
- Cryptanalyse des systèmes cryptographiques
- Sécurité et IPV6
- Sécurité réseaux
- Audit de sécurité technique
- **Sécurité des protocoles**

LISTE DES ACRONYMES ET ABREVIATION

- Acronymes :
  - ✓ UCAD ; Université Cheikh Anta Diop de Dakar
  - ✓ LACGAA : Laboratoire d'Algèbre de Cryptologie de Géométrie Algébrique et Applications
- Abréviations :
  - ✓ Monsieur : M.
  - ✓ Professeur : Pr.
  - ✓ Docteur : Dr.
  - ✓ TDSI : Transmission de Données et Sécurité de l'Information
  - ✓ FST : Faculté des Sciences et Techniques
  - ✓ DAB : Distributeur Automatique de Banque

REFERENCES :

[1] www.bd.enst.fr/dombd/Cours/Applications/3tiers/index.html.

[2] http://developer.android.com/index.html

[3] http://android-developers.blogspot.com/

[5] http://www.w3schools.com/json/default.asp

[6] L'art du développement Android « Mark Murphy ; PEARSON »

Introduction générale :

L'avantage compétitif des banques repose sur leur capacité à répondre rapidement aux changements des besoins du marché et augmenter leurs bénéfices en produisant des produits de biens ou des services qui satisfont le besoin de la clientèle. Dans un monde en constante évolution et 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 M-Banking, constitue une solution possible à cette situation. On rencontre plusieurs définitions et divers acronymes qui s'intéressent aux services bancaires par mobiles. Le terme Mobile Banking désigne une spécificité du Mobile Business adaptée au métier de la Banque. C'est en quelque sorte une adaptation du désormais canal de banque à distance classique de « e-banking » sur terminal. Le e-banking est l'ensemble des services bancaires assurés par voie électronique « electronic banking » et donc par Internet : consultation de comptes, virements, achats de produits financiers. Il est à noter que l'e-banking est un terme générique désignant tous les services assurés par internet sur support portable et non portable (ordinateur de bureau) par contre le M-Banking, *mobile banking* (banque mobile) regroupe toutes les techniques permettant de réaliser des opérations bancaires à partir du support « téléphone portable ». En fonction des capacités techniques (téléphone voix, données, internet, SMS, WAP) et de la diversité des terminaux mobiles (téléphone mobile, Pocket PC, PDA, *Smartphone*) plusieurs banques ont adopté des solutions de Mobile Banking, du fait de la meilleure couverture du réseau mobile par rapport au réseau filaire, et au nombre important de téléphones mobiles utilisés. Les banques ont exploité les potentialités des outils de communication en particulier la téléphonie mobile en un accès mobile comme canal de distribution et comme service afin de concrétiser l'objectif. Pour les clients, les services bancaires par mobile représentent un équilibre difficile entre un concept au potentiel considérable (pouvoir faire des transactions n'importe où, n'importe quand) et des obstacles de nature pratique. Les Smartphones permettent à la clientèle de la banque de simplifier le support clients, d'économiser et d'augmenter la satisfaction client. Dans ce cadre, nous allons essayer de réaliser une application mobile destinée aux banques. Elle sert d'interface entre le client et sa banque et offre un ensemble de services pour la clientèle. Pour atteindre cet objectif, nous commençons par comprendre le fonctionnement du système Mobile et son déploiement pour la mise en place de la banque à distance. Dans la deuxième étape, nous choisissons une architecture à deux niveaux (Client/serveur) afin d'implémenter les différents modules logiciels de l'application. Ensuite, nous procédons à la conception. Et enfin, nous développons l'application. Notre travail se subdivisera en quatre parties : Dans la première partie intitulée ***Aperçu général et cadre théorique*** nous allons présenter notre application, et dans la deuxième partie qui a pour titre ***Analyse et spécification des besoins***, nous commençons par comprendre le contexte du système, déterminons les principaux cas d'utilisation, puis les besoins fonctionnels et les besoins non fonctionnels. Puis, dans le troisième chapitre, nous tenterons d'approfondir la compréhension du système et d'obtenir une spécification, une analyse et une conception détaillées des cas d'utilisation. Au niveau de la quatrième partie intitulée *Réalisation*, nous allons présenter l'architecture de notre application. Finalement, nous terminons par une conclusion générale et quelques perspectives intéressantes concernant ce travail.

I. Le cadre théorique et méthodologique :

Chapitre I : Le cadre théorique :

I.1 : Présentation du projet :

Depuis le milieu des années 2000, les plus grandes banques mondiales commencent à investir dans le Mobile Banking appelé aussi M-Banking. Rappelons que le «M-Banking désigne une spécificité du Mobile Business adaptée au métier de la Banque. Il s'agit du traitement des services bancaires à travers le mobile» C'est dans ce contexte que vient s'inscrire notre projet de fin d'études. Il s'agit en fait d'une application Android M-Banking permettant à la clientèle de bénéficier des services bancaires que lui offre sa banque sur un terminal mobile (Smartphone, tablette..) d'où vient le concept de l'informatique ubiquitaire. «Les systèmes dits ubiquitaires supportent la mobilité de l'utilisateur. Le principal intérêt de ces systèmes est de s'adapter à l'environnement de l'homme (géo localisation, disponibilité des réseaux de communication, identification de l'utilisateur ...). En d'autre termes, c'est le système qui s'adapte à l'Homme, et pour y arriver le système se doit d'être présent partout et en temps réel.» D'où le nom du guichet virtuel dans le secteur bancaire permettant d'assurer ces trois principes :

Any Time : continuité de service, au besoin du client.
Any Where : à la maison, au travail, à l'extérieur, au Sénégal, à l'étranger.
Any Acces : multi canaux.

I.2 : Cadre de projet :

De plus en plus nombreuses sont les applications mobiles qui reposent sur une plate-forme de mobilité en équipant les périphériques mobiles consommateur les plus récents avec la Connectivité en temps réel et l'intégration aux systèmes service client de la banque. C'est pourquoi la clientèle peut bénéficier des services bancaires en ligne. Ces applicatifs réduisent les coûts et augmentent la satisfaction du client. En effet, le client n'a pas besoin de se déplacer à sa banque et attendre aux guichets pour bénéficier de service, c'est sa banque qui lui rend visite. Et au niveau de la banque, abaisse les coûts de transaction et charge de service, en particulier le coût de mise en place et de maintien d'un réseau de distribution pour les banques.

I.3 : Le M-Banking :

Le Mobile Banking désigne une spécificité du Mobile Business adaptée au métier de la Banque. C'est en quelque sorte une adaptation du canal de banque à distance classique de «e banking» sur terminal mobile. D'autre part L'e-banking est l'ensemble des services bancaires assurés par voie électronique « electronic banking » et donc par Internet : demander informations, commande chéquier, service bancaire, consultation de comptes, virements et achats de produits financiers. De nos jours, les réseaux de connexion mobile permettent d'obtenir une connectivité plus sûre et moins coûteuse. En fonction des capacités techniques (téléphone voix, données, internet, SMS, WAP) et de la diversité des terminaux mobiles (téléphone mobile, Pocket PC, PDA, Smartphone).
On peut classifier le Mobile Banking comme suit **:**
Le PDA banking
Le SMS banking, le SMS Payment

Le WAP banking
Le Mobile Internet banking
Le Mobile Embedded Software banking
Le paiement par SMS est classé dans le Mobile Banking et non dans le Mobile Payment, car il ne se substitue par à une carte de paiement bancaire de type carte bleue ou carte VISA. Les solutions de Mobile Banking peuvent servir à consulter les informations personnelles d'un client au travers d'un mobile. Elles ont ensuite permis d'effectuer des transactions bancaires de type virement et achat/revente d'actions. C'est le Mobile Banking transactionnel.
Les applications mobiles « toujours disponibles et toujours connectées » vous fournissent une communication très personnalisée et ciblée par l'envoi automatique des notifications à votre clientèle. Vos applications de gestion de comptes peuvent ainsi transmettre toute la gamme d'informations, du moment d'une panne locale jusqu'aux estimations de temps de résolution. Avec l'accès automatique au support critique, vous gagnez du temps en évitant des appels inutiles.

Chapitre II : Le cadre méthodologique :

II.1 : Les techniques de M-Banking :

Réseaux mobiles et transmission de données les réseaux mobiles disponibles utilisent la technologie (GSM et depuis UMTS). Ces normes internationales permettent aux équipements mobiles de bien sûr téléphoner, mais aussi d'effectuer des transmissions de données. Les techniques disponibles pour la transmission de données sont:
• l'échange de données par SMS : bien adapté à la transmission de données textuelles de petite taille.
• l'établissement d'une liaison TCP/IP qui selon la technologie (GSM, UMTS) et les options d'abonnement peut être à différentes vitesses:
CSD (circuit switched data), proche d'un protocole modem traditionnel: permet l'échange de données mais interdit les appels téléphoniques durant la connexion en mode data.
GPRS (General Packet Radio Service): permet l'échange de données à
64 kbps (dans la pratique plutôt 40 kbps) tout en restant disponible pour recevoir un appel: néanmoins, l'échange simultané de données et la téléphonie n'est pas possible.
UMTS: échange de données à haute vitesse (de l'ordre de 300 kbps).
Sur la liaison TCP/IP ainsi établie, les données peuvent être échangées selon tous types de protocoles:
Le WAP, version simplifiée et plus légère du Web, dont le navigateur Wap est embarqué dans la plupart des portables commercialisés actuellement.
Le Web, embarqué dans les portables haut de gamme.
Email (SMTP), également disponible sur les portables haut de gamme.
échange de messages multimédia MMS permettant la transmission de messages composites intégrant texte, photo, sons, vidéo, etc.
Toute application spécifique, qui peut être soit embarquée sur le portable (par exemple sous forme d'application Android)
Si le portable le supporte ou sur un autre équipement (PC, Palm, etc.) utilisant le portable pour accéder au réseau.
Ces différentes techniques permettent de réaliser de nombreuses applications mobiles professionnelles dans le domaine bancaire offrir les fonctionnalités suivantes en temps réel: Authentification de l'utilisateur, mise à disposition des informations utiles aux clients concernant la banque, services à la clientèle des outils liés à la banque, synthèse des comptes bancaires de type chèque, carte bancaire, placements liste des opérations d'un compte,

virements internes et externes, services professionnels à forte valeur ajoutée, services destinés aux opérateurs télécoms mobiles et intégration au système d'information bancaire : Messagerie, bases de données Les Smartphones puissants actuels vous permettent à étendre vos outils de service client en ligne aux périphériques mobiles. Au fait, plusieurs processus sont rationalisés grâce à la mobilité, car les applications peuvent identifier et localiser automatiquement le client, en lui fournissant toutes autres données pertinentes.

II.2 : Solution retenue :

L'application envisagée, dans le secteur bancaire est une application mobile client riche serveur sur une plateforme Android.
Cette application permet à un client d'accéder à partir d'un terminal mobile pour bénéficier des services informationnels et services transactionnels sur sa banque en toute sécurité après vérification des paramètres d'accès au niveau serveur distant. Le réseau internet via la suite des protocoles TCP/IP permet la connexion entre le terminal mobile et serveur d'application et la transmission de message se fait par le langage JSON qui constitue un moyen de communication. Le terminal mobile envoi les informations afin s'authentifier et une demande chèque de au serveur qui décode ces données et les analyse pour décision. Android est un système d'exploitation open-source pour smartphone conçu par Google. Le SDK fournit les outils et l'API nécessaires pour commencer à développer sur mobile sur la plateforme Android en C#. Le développement mobile ouvre de larges perspectives. En effet, les mobiles possèdent maintenant des accéléromètres, des connexions sans-fil et des GPS. Ce Framework est nouveau et c'était très enrichissant d'apprendre son fonctionnement. Avec ce projet, nous allons apprendre comment utiliser une architecture 3-tier et les web services. Cette architecture divise l'application en trois parties. Le client (téléphone) se connecte à un serveur (Middleware) via des web services et ce serveur interroge la base de données.

Figure 1:Architecture de la solution proposée

II. Le cadre analytique et quelques recommandations :

Chapitre III : analyse et spécification des besoins :

Introduction :

Une étape essentielle de tout cycle de développement logiciel ou conceptuel consiste à Effectuer une étude préalable. Le but de cette phase est de comprendre le contexte du système. Il s'agit d'éclaircir au mieux les besoins fonctionnels et non fonctionnels, apparaitre les acteurs et identifier les cas d'utilisation. Dans ce chapitre, nous allons essayer d'exprimer les besoins sous forme de diagrammes de cas d'utilisation.

III.1 : spécification des besoins :

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins Fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante ainsi de Trouver un accord commun entre les spécialistes et les utilisateurs pour réussir le projet.

III.1.1 : spécification des besoins fonctionnels:

Après une étude détaillée de système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces besoins se regroupent dans les diagrammes des cas d'utilisation. Les besoins utilisateur :

- L'authentification de client.
- Cours devise.
- Change devise.
- Consultation annuaire réseau des agences
- Commande chéquier.
- Consultation solde.
- Retrait à distance.
- Virement
- Contact banque par mail ou par téléphone.

III.1.2 : spécification des besoins non fonctionnels :

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé le besoins suivants :

- La disponibilité : l'application doit être disponible pour être utilisé par n'importe quel utilisateur.
- La sécurité de l'accès aux informations critiques : nous devons prendre en considération la confidentialité des données de clients surtout au niveau de l'authentification. Pour cela nous devons restreindre l'accès à ces informations à l'administrateur.
- La fiabilité : les données fournies par l'application doivent être fiables.

- La convivialité de l'interface graphique : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci on découvrira ses fonctionnalités.
- Une solution ouverte et évoluée : l'application peut être améliorée par l'ajout d'autres Modules pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.
- La possibilité de retourner au menu principal de l'application à partir de n'importe quelle fenêtre de celle-ci.

Chapitre IV : méthodologie et approche adoptée :

Avant de programmer l'application et se lancer dans l'écriture du code : il faut tout d'abord organiser les idées, les documenter, puis organiser la réalisation en définissant les modules et Les étapes de la réalisation. Cette démarche antérieure à l'écriture que l'on appelle modélisation ; son produit et un module.
La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Dans le cadre de notre projet on a utilisé la Méthodologie UML pour la modélisation des différents diagrammes.

IV.1 : présentation d'Uml :

En regardant les objectifs fixés pour la réalisation du projet, nous remarquons que nous sommes face à une application modulaire et qui devra rester ouverte pour les améliorations futures. De ce fait, il est très important d'utiliser un langage universel pour la modélisation afin de clarifier la conception et de faciliter les échanges. Notre choix est porté sur le langage UML puisqu'il convient pour toutes les méthodes objet et se prête bien à la représentation de l'architecture du système UML : UML (Unified Modeling Language) est un langage de modélisation unifié permet de modéliser une application logicielle d'une façon standard dans le cadre de conception orienté objet. UML permet de couvrir le cycle de vie d'un logiciel depuis la spécification des besoins jusqu'au codage en offrant plusieurs moyens de description et de modélisation des acteurs et de l'utilisation système, du comportement des objets, du flot de contrôle internes aux opérations, des composants d'implémentation et leurs relations, de la structure matérielle et de la distribution des objets et des composants indépendamment des techniques d'implémentation et peut être mis à jour selon les besoins.

IV.2 : les avantages d'Uml :

- Universel.
- Adopté par les grandes entreprises.
- Notation unifié
- Facile à comprendre.
- Adopté par plusieurs processus de développement
- Limite les risques d'erreur.
- N'est pas limité au domaine informatique.

IV.3 : les diagrammes de cas d'utilisation :

Le diagramme de cas d'utilisation a pour but de donner une vision globale sur les interfaces de future application. C'est le premier diagramme UML constitué d'un ensemble d'acteurs qui agit sur des cas d'utilisation et qui décrit sous la forme d'actions et des réactions, le comportement d'un système du point de vue utilisateur. Acteur : un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une entité ayant un comportement comme une personne, système ou une entreprise. Système : cet élément fixe les limites du système en relation avec les acteurs qui l'utilisent (en dehors de système) et les fonctions qu'il doit fournir (à l'intérieur du système).Cas d'utilisation : un cas d'utilisation représente un ensemble de séquences d'actions à réaliser par le système et produisant un résultat observable intéressant pour un acteur particulier représenté par des ellipses et limité par un rectangle pour représenter le système.

IV.4 : identification des acteurs :

Acteur	**Rôle**
Client banque (utilisateur mobile)	**- L'authentification de client.** **- Cours devise.** **- Change devise.** **- Consultation annuaire réseau des agences** **- L'envoi de demande chéquier.** **- Consultation solde à distance.** **- Virement.** **- Retrait à distance.** **- Contact banque par mail ou téléphone.**

IV.5 : diagramme de cas d'utilisation globale :

Ci-dessous, nous présentons le diagramme de cas d'utilisation pour la compréhension du fonctionnement du système.

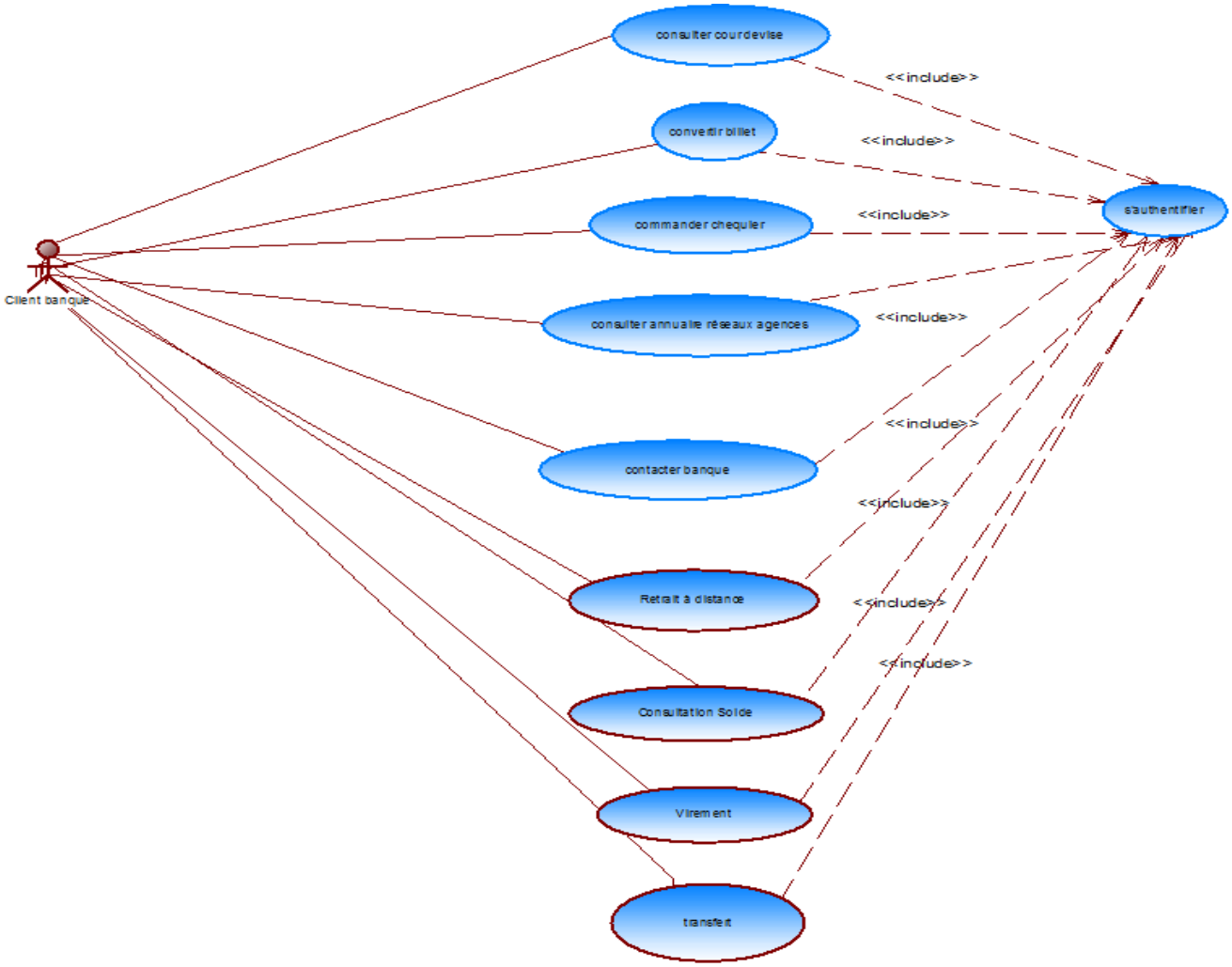


Figure 2:Diagramme de cas d'utilisation globale:

Description du cas d'utilisation << consulter cours devises >>

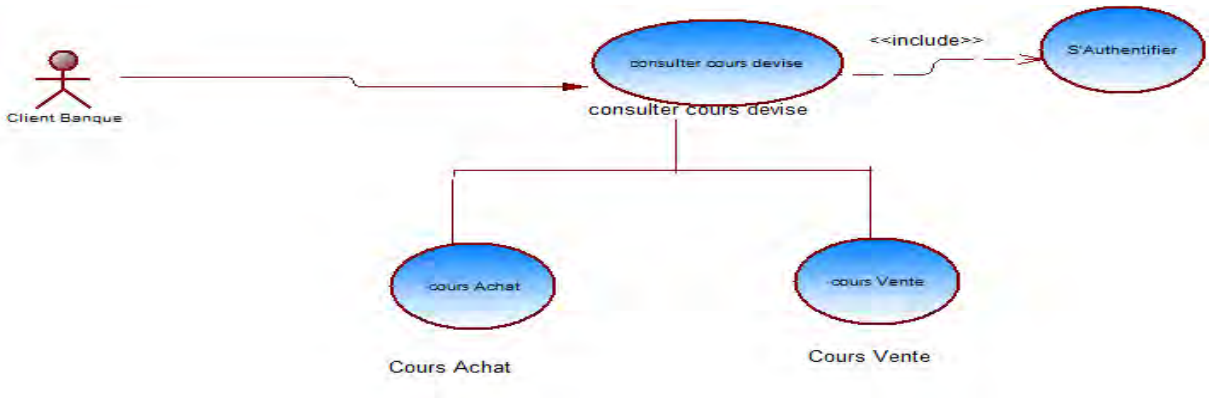


Figure 3:Diagramme de cas d'utilisation «consulter cours devise

SOMMAIRE :	
Titre :	Consulter cours devise
But :	**Lister le cours de chaque devise par type opération Achat ou vente.**
Résumé :	**L'utilisateur clique sur le bouton « cours devise » l'action se déclenche et le système liste cours de vente ou achat.**
Acteur :	**Client banque**
Description Des Enchainements :	
Pré conditions	**Post conditions**
- client est authentifié	**- affichage cours devises par sens opération ACHAT ou VENTE.**
Scénario nominal	
1. Le client se connecte au système de la banque à travers son application mobile par login et un mot de passe. 2. Le client valide la requête d'affichage du cours de change via le bouton correspondant celui de « cours devises » 3. Le système expose l'affichage cours devises correspondants aux opérations d'achat ou vente. 4. Le client à la possibilité de basculer entre les opérations achat/vente.	
Enchainement alternatif	

1. a Client n'a pas rempli champ ou les données sont incorrectes. **1. Le système affiche un message d'erreur.** **2. Retour à l'étape 1 du scénario nominal pour lancer à nouveau la connexion.**

Description du cas d'utilisation convertir billets de banque :

Figure 4:Diagramme de cas d'utilisation « convertir billet de banque

SOMMAIRE :		
Titre :	Convertir billet de	
But :	**Le client de banque peut convertir deux devises différentes. « Change devise »**	
Résumé :	**L'utilisateur choisit la devise à convertir et la devise cible.**	
Acteur :	**Client banque**	
Description Des Enchainements		
Pré conditions		**Post conditions**
- L'utilisateur est authentifié.		**- Affichage le résultat de conversion devise.**
Scénario nominal		

1. L'utilisateur se connecte à l'application à par login et un mot de passe. **2. Le client clique sur l'icône « change devise »** **3. Le système présentera une page permettant de choisir la devise à convertir et la devise cible.** **4. L'utilisateur choisit les deux devises et valide la conversion.** **5. Système appelle un service web distant afin de récupérer le résultat de la fonction de conversion.** **6. Le système affiche résultat conversion.**
Enchainement alternatif
1. a Client n'a pas rempli champ ou les données sont incorrectes. **1. Le système affiche un message d'erreur.** **2. Retour à l'étape 1 du scénario nominal pour lancer à nouveau la connexion.** **6. a Client n'a pas précisé les devises en entrées** **1. Le système affiche un message d'erreur.** **2. Retour à l'étape 1 du scénario nominal pour lancer à nouveau pour choisir les devises et valider la conversion.**

Description du cas d'utilisation << commander chéquier >> :

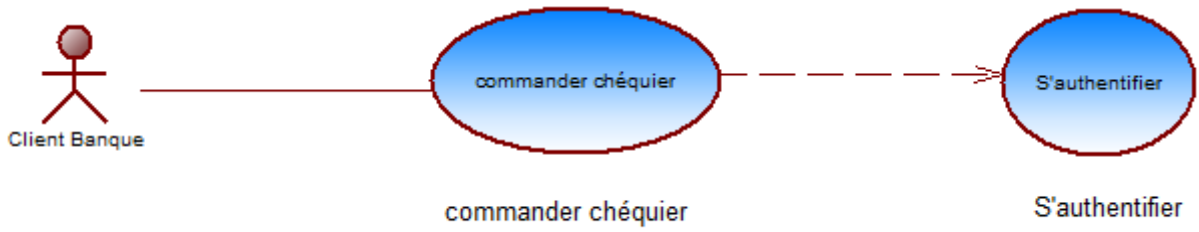


Figure 5:Diagramme de cas d'utilisation « commander chéquiers

SOMMAIRE	
Titre :	Commander chéquiers
But :	**Le client peut commander un chéquier.**
Résumé :	**Commander chéquiers**
Acteur :	**Client banque**
Description Des Enchainements	

Pré conditions	**Post conditions**
- Le client est authentifié. **- Consultation compte utilisateur.**	**- Message de confirmation de prise en charge de la demande client du chéquier.**
Scénario nominal	
1. Le client se connecte à système de la banque à travers son application mobile par login et un mot de passe. **2. Le système présente une page permettant de connecter au menu de l'application.** **3. L'utilisateur lance la commande de chéquiers via l'icône « commander chéquiers »** **4. Système effectue la mise à jour à la base de données du compte utilisateur et affiche message de succès.**	
Enchainement alternatif	
1. a Le client n'a pas rempli champ ou les données sont incorrectes. **1. Le système affiche un message d'erreur.** **2. Retour à l'étape 1 du scénario nominal pour lancer à nouveau la connexion.**	

Description du cas d'utilisation << consulter annuaire réseau agences >> :

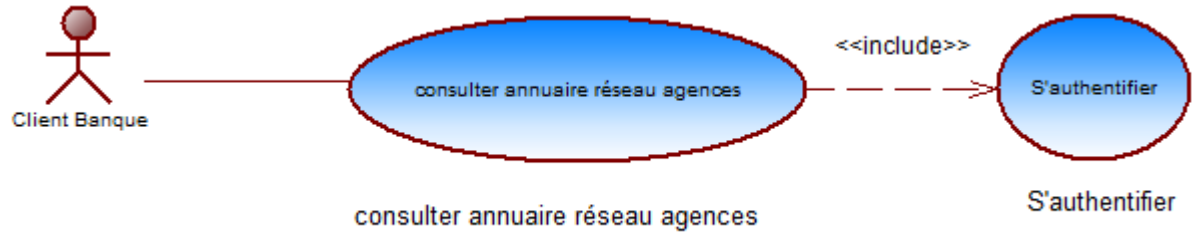


Figure 6:Diagramme de cas d'utilisation « Consulter annuaire réseau agences

SOMMAIRE	
Titre :	Consulter annuaire réseau agences
But :	**Le client peut consulter la liste des agences et leurs coordonnées.**
Résumé :	**Exposer la liste des agences commerciales de la banque.**

Acteur :	**Client Banque.**
Description Des Enchainements	
Pré conditions	**Post conditions**
- Le client est authentifié.	**- Affichage liste des agences avec leurs cordonnées.**
Scénario nominal	
1. Le client se connecte à l'application par login et un mot de passe. **2. Le système affiche le menu de l'application.** **3. Le client clique sur l'icône « Réseau agences »** **4. Le système présente une page permettant de consulter la liste des agences.**	
Enchainement alternatif	
1. a le client n'a pas rempli champ ou les données sont incorrectes. **1. Le système affiche un message d'erreur.** **2. Retour à l'étape 1 du scénario nominal pour lancer à nouveau la connexion.**	

Description du cas d'utilisation contacter banque :

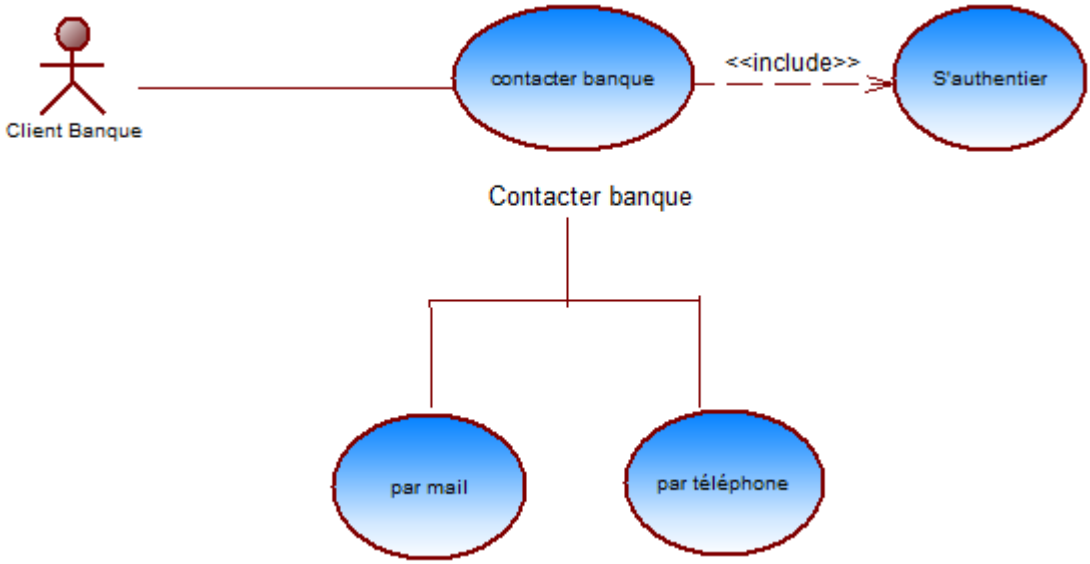


Figure 7:Diagramme de cas d'utilisation «contacter banque

SOMMAIRE	
Titre :	Contacter banque
But :	**Il permet consulter les coordonnées de la banque et contacter la banque par mail ou par téléphone.**
Résumé :	**Consulter coordonnées de la banque.** **Contacter banque par téléphone ou appeler banque.**
Acteur :	**Client banque**

Description Des Enchainements	
Pré conditions	**Post conditions**
- Le client est authentifié.	**- Présenter coordonnées de la banque et possibilité de le joindre par mail ou par téléphone.**

Scénario nominal
1. Le client se connecte à système de la banque à travers son application mobile par login et un mot de passe. **2. Le système affiche le menu de l'application.** **3. Le client clique sur l'icône « Réseau agences »** **4. Le système présente une page permettant de contacter sa banque.** **5. .le client peur envoyer un mail à la banque ou appeler le service clientèle de sa banque.**

Enchainement alternatif

Description du cas d'utilisation retiré de l'argent :

Figure 8:Retiré de l'argent

SOMMAIRE	
Titre :	**Retirer de l'argent**
But :	**il permet de retirer de l'argent à distance.**
Résumé :	**Retirer de l'argent à distance**
Acteur :	**Client Banque**
Description Des Enchainements	
Pré conditions	**Post conditions**
- Le client est authentifié.	**- Affichage menu retiré de l'argent pour permettre au client de retirer de l'argent à distance.**
Scénario nominal	
1. Le client se connecte à système de la banque à travers son application mobile par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le client clique sur l'icône « Retirer de l'argent » 4. Le système présente une page permettant de retiré de l'argent. 5. .le client peut retirer de l'argent via son mobile.	
Enchainement alternatif	

Description du cas d'utilisation consultation solde :

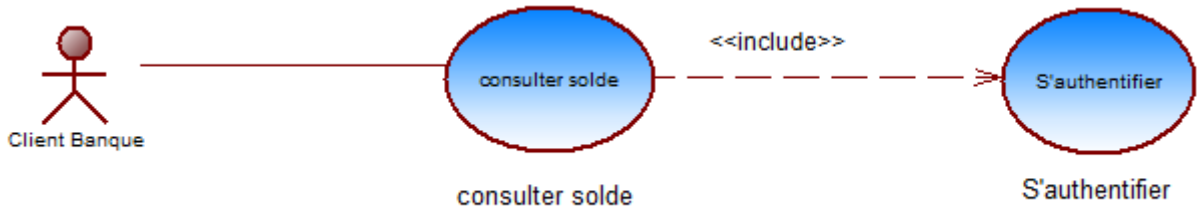


Figure 9:Consultation Solde

SOMMAIRE	
Titre :	**Consulter solde**
But :	**Il permet de consulter son solde.**
Résumé :	**Consulter son solde.**
Acteur :	**Client banque**
Description Des Enchainements	
Pré conditions	**Post conditions**
- Le client est authentifié.	**- Affichage menu consulter solde pour permettre au client de consulter son solde à distance.**
Scénario nominal	
1. Le client se connecte à système de la banque à travers son application mobile par login et un mot de passe. 2. Le système affiche le menu de l'application. 3. Le client clique sur l'icône « consulter solde » 4. Le système présente une page permettant de consulter son solde à distance. 5. .le client peut consulter son solde via son mobile.	
Enchainement alternatif	

Description du cas d'utilisation virement :

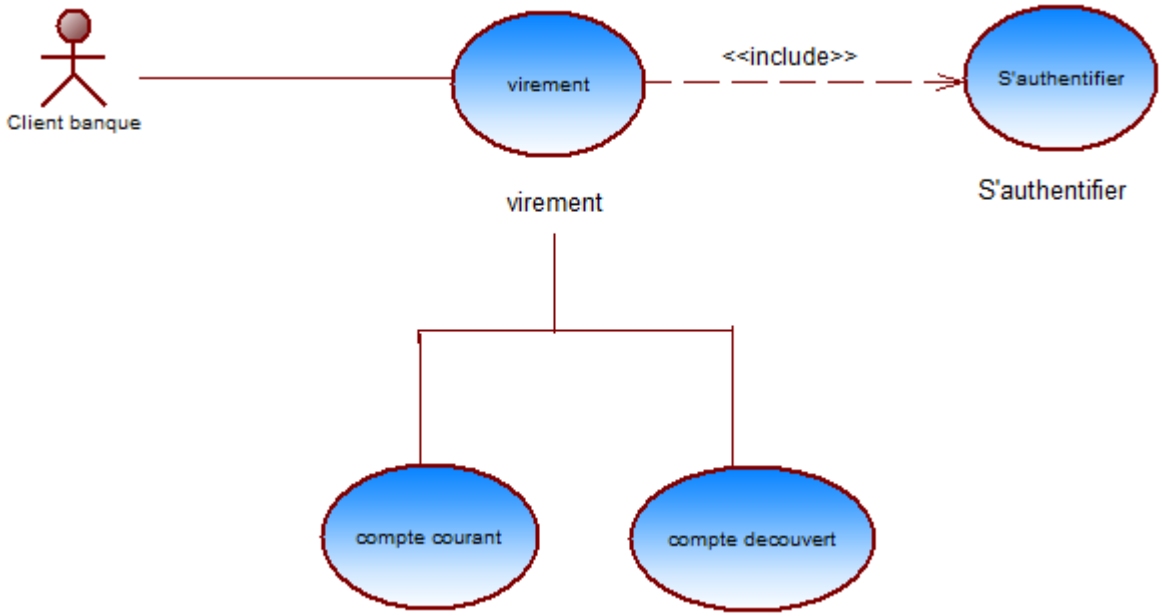


Figure 10:Diagramme de cas d'utilisation virement

SOMMAIRE	
Titre :	**Virement**
But :	**Il permet d'effectuer un virement dans son compte.**
Résumé :	**Effectuer un virement d'un compte vers un autre compte.**
Acteur :	**Client banque**
Description Des Enchainements	
Pré conditions	**Post conditions**
- Le client est authentifié.	**- Affichage menu virement pour permettre au client d'effectuer un virement à distance.**
Scénario nominal	

1. Le client se connecte à système de la banque à travers son application mobile par login et un mot de passe. **2. Le système affiche le menu de l'application.** **3. Le client clique sur l'icône « virement »** **4. Le système présente une page permettant d'effectuer un virement à distance.** **5. .le client peut faire un virement via son mobile.**
Enchainement alternatif

III. La présentation, l'analyse et l'interprétation des résultats :

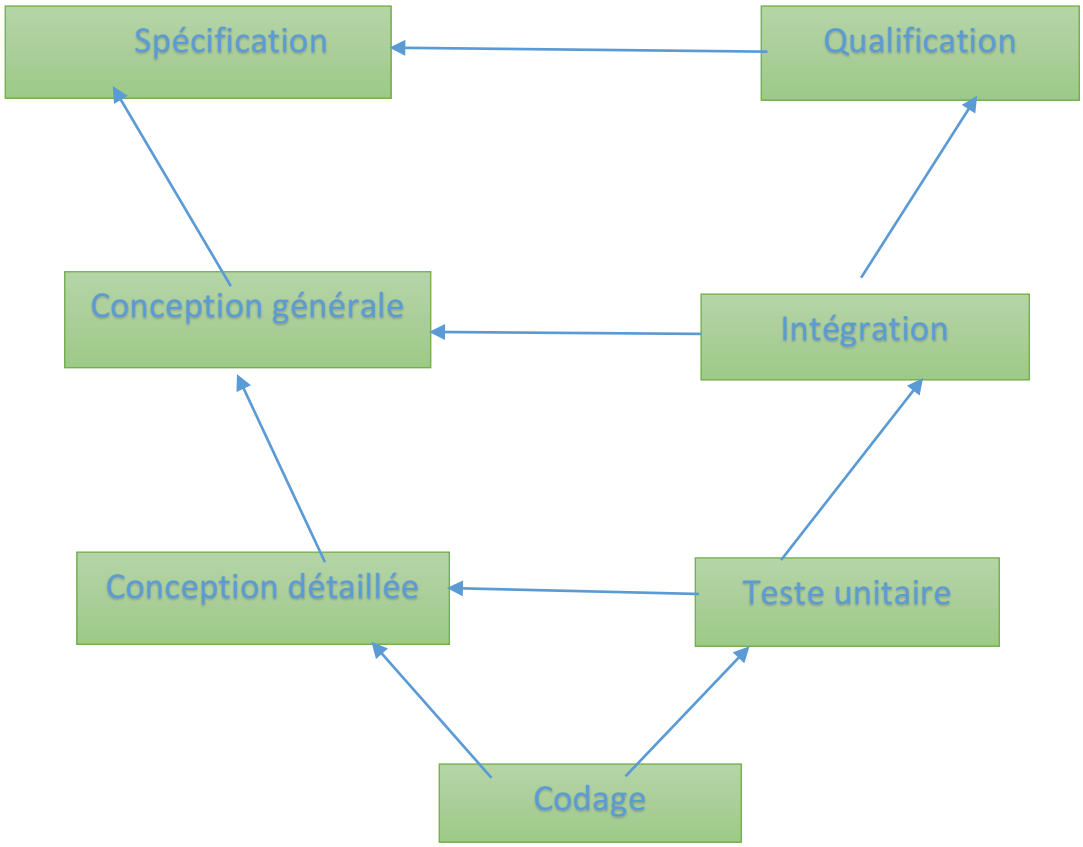
Chapitre V : conception :

La conception consistera à faire une représentation des diagrammes de séquences et d'activités en se basant sur le langage de modélisation UML.
On distingue deux types de conceptions :

V.1 : la conception générale :

V.1.1 : le cycle de développement en v :

De nos jours, la méthodologie adoptée dans l'analyse et la conception des systèmes représente un choix stratégique pour le bureau d'études afin de mener à terme les projets tout en respectant les délais annoncés au client et avec la qualité demandée. Vu l'évolution des besoins des utilisateurs finaux, les applications d'entreprise deviennent de plus en plus complexes et difficiles à concevoir et à développer. Pour la conception, le développement et la réalisation de notre application, nous avons opté pour l'application du processus de développement V qui actuellement le cycle de vie le plus connu et certainement le plus convenable aux projets complexes. Ce processus nous accompagnera du début de projet jusqu'à l'implémentation. Son principe est qu'avec toute décomposition doit être décrite la recomposition, et que toute description d'un composant doit être accompagnée de test qui permettront de s'assurer qu'il correspond à sa description. Ceci rend explicite la préparation des dernières phases (validation-vérification).par les premières (construction de l'application) et on sait progressivement si s'approche de ce que le client désire.
Le schéma ci-dessous représente les différentes phases du modèle en V :



V.2 : la conception détaillée :

La conception détaillée met en œuvre itérativement un microprocessus de construction et c'est en cette phase que l'on génère le plus de volume d'informations. En tant que concepteurs, nous allons élaborer le modèle de conception qui va donner une image « prête à coder » de notre solution. Cette étape se fera par étape afin d'aboutir à un système fonctionnel reflétant une réalité physique. Le diagramme de déploiement : Le diagramme de déploiement définit l'architecture matérielle de l'application. Il présente les périphériques utilisés et la répartition du système sur ces différents éléments. Il montre aussi les liens de communication entre ces diverses entités. Le diagramme de déploiement de notre application est représenté par le diagramme ci-après :

V.2.1 : le diagramme de déploiement :

Le diagramme de déploiement définit l'architecture matérielle de l'application. Il présente les Périphériques utilisés et la répartition du système sur ces différents éléments. Il montre aussi Les liens de communication entre ces diverses entités.
Le diagramme de déploiement de notre application est représenté par le diagramme ci-après :

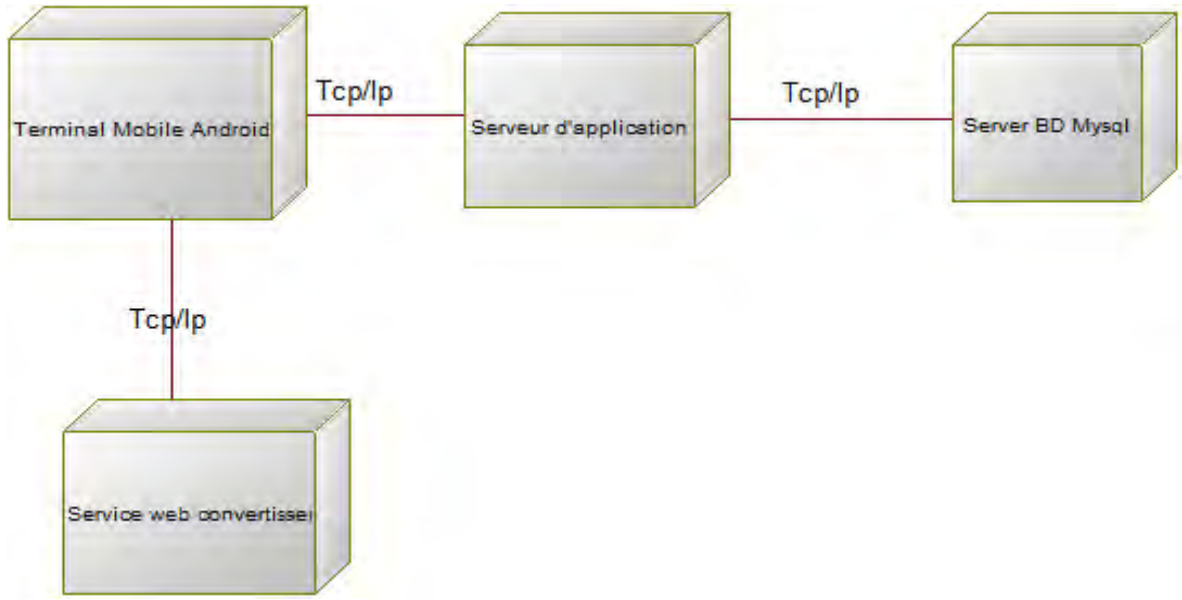


Figure 11:Diagramme de déploiement

V.2.2 : les diagrammes de séquences :

Les diagrammes de séquence peuvent servir à illustrer les cas d'utilisations. Ils permettent de représenter la succession chronologique des opérations réalisées par un acteur et qui font passer d'un objet à un autre pour représenter un scénario. Dans cette partie, nous allons décrire les scénarios les plus importants ainsi que leurs représentations par les diagrammes de séquence

V.2.2.1 : le diagramme de séquence << s'authentifier >>

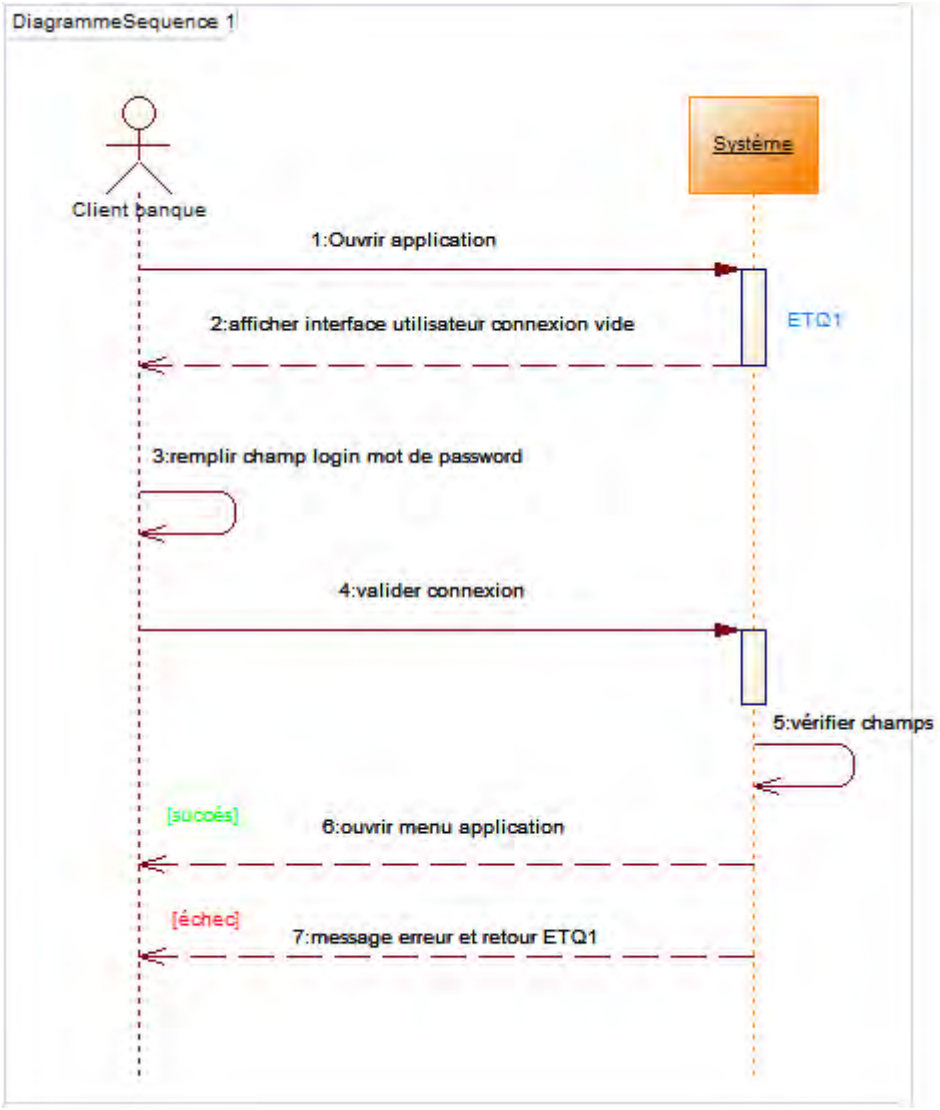


Figure 12:Diagramme de séquence « s'authentifier

V.2.2.1 : le diagramme de séquence << consulter cours devise >>

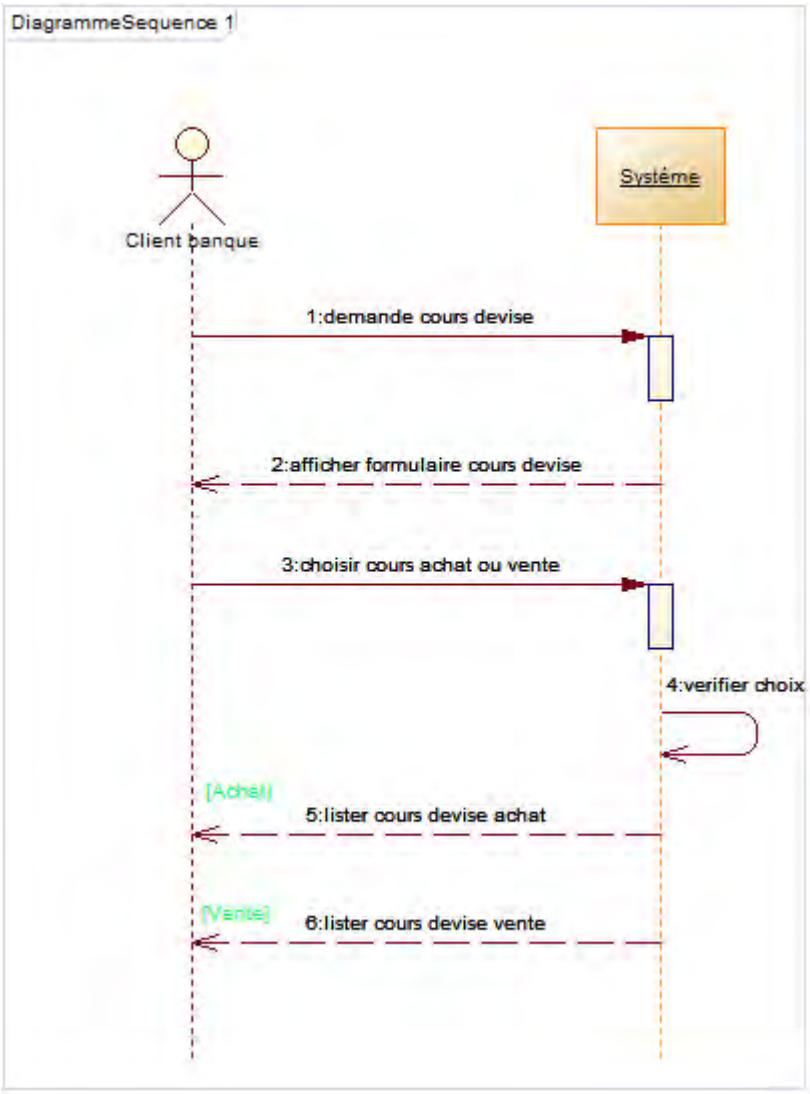


Figure 13:Diagramme de séquence « consulter cours devise

V.2.2.1 : le diagramme de séquence << convertir billet de banque >>

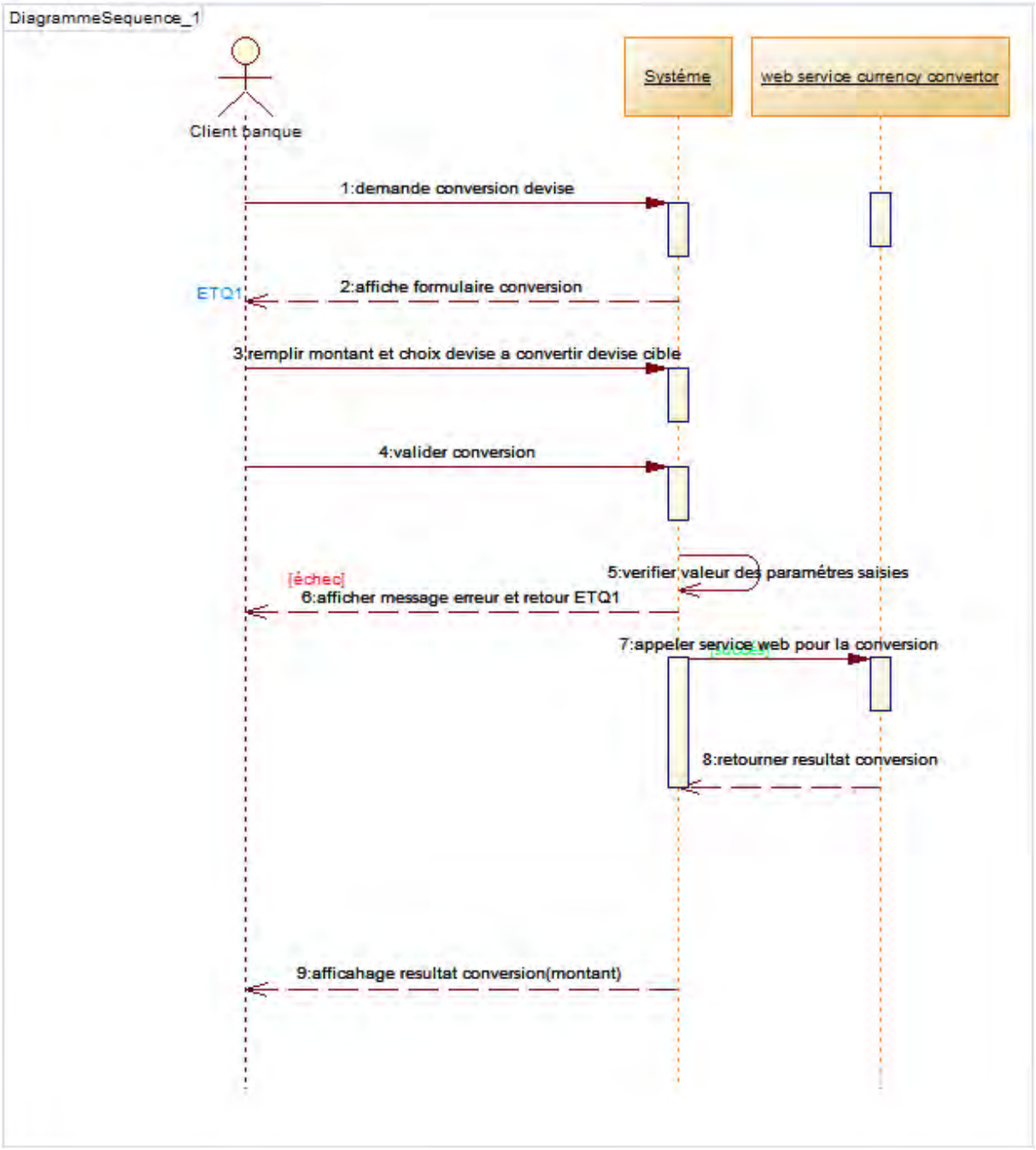


Figure 14:Diagramme de séquence « Convertir billet de banque

V.2.2.1 : le diagramme de séquence << commander chéquiers >>

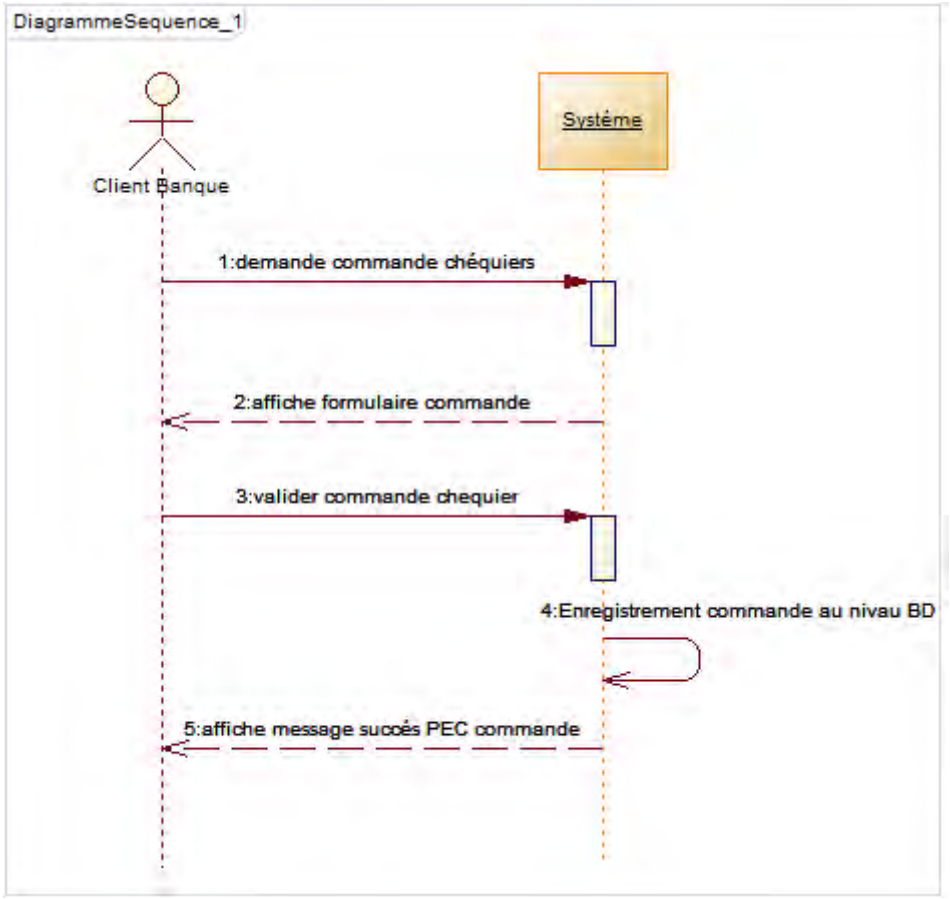


Figure 15:Diagramme de séquence «Commander chéquiers

V.2.2.1 : le diagramme de séquence << consulter annuaire réseau agences >>

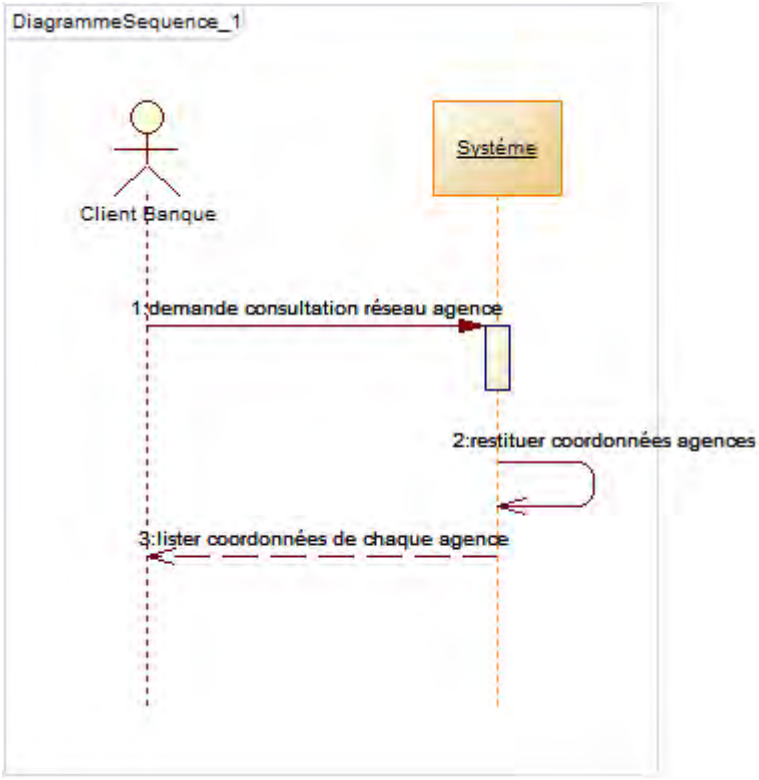


Figure 16:Diagramme de séquence «Consulter annuaire réseau agences

V.2.2.1 : le diagramme de séquence << contacter banque >>

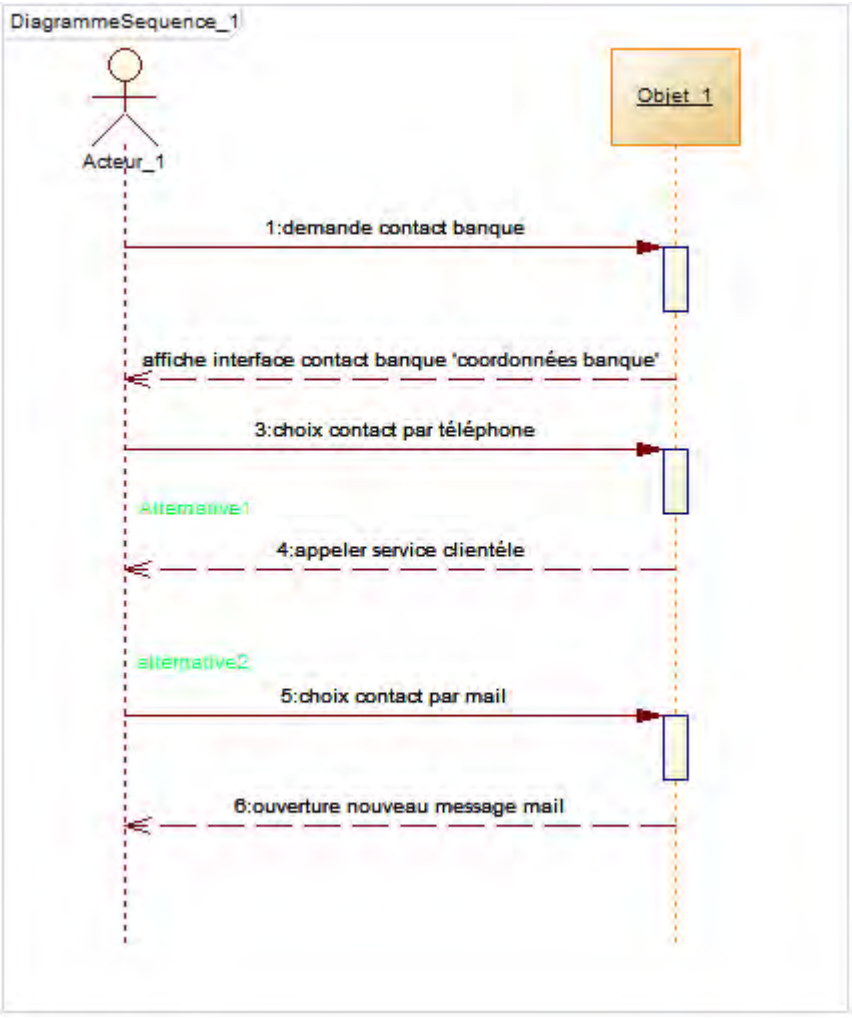


Figure 17:Diagramme de séquence «Contacter banque

V.2.3 : les diagrammes d'activité :

Le diagramme d'activité permet de représenter le déclenchement d'évènements en fonction Des états du système et de modéliser des comportements parallélisables. Il donne une vision Des activités propres à une opération ou à un cas d'utilisation.
Une activité est une opération d'une certaine durée qui peut être interrompue.
Dans ce cas, on va représenter ci-après des diagrammes d'activités qui décrivent :
L'authentification d'un utilisateur :

V.2.2.1 : le diagramme d'activité << s'authentifier >>

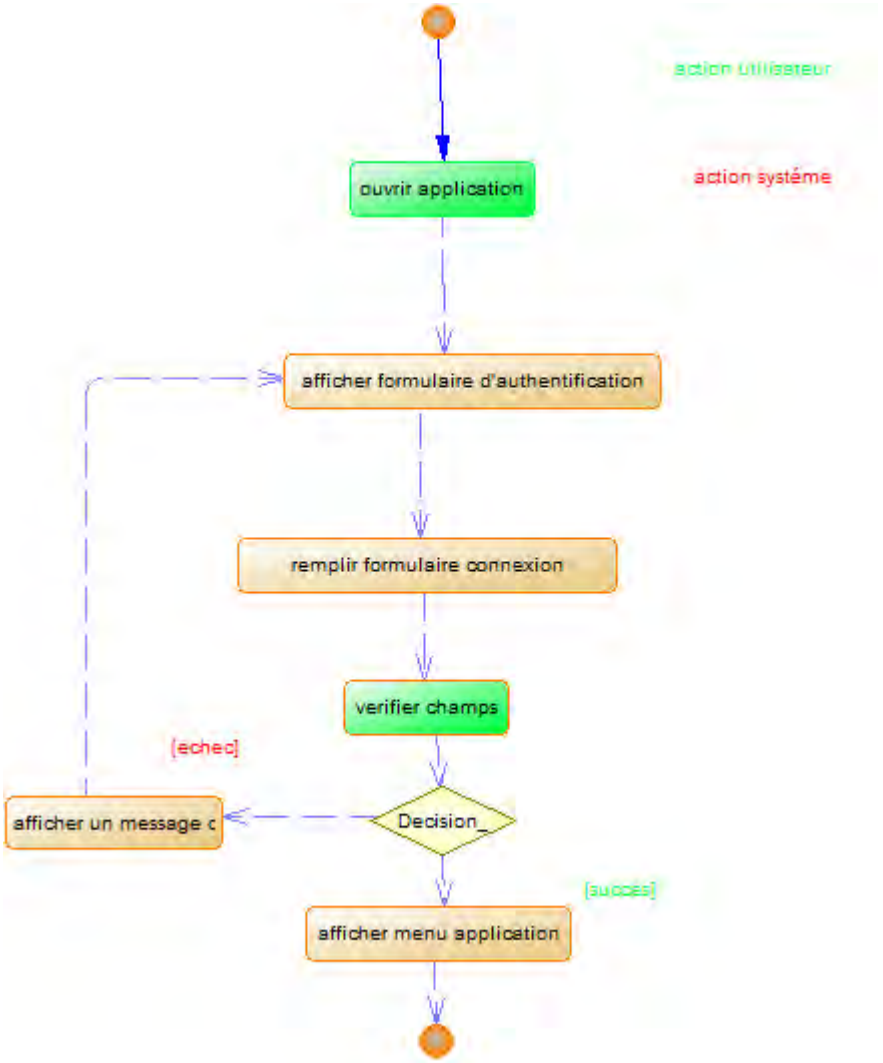


Figure 18:Diagramme d'activité « s'authentifier

V.2.2.1 : le diagramme d'activité << consulter cours devise >>

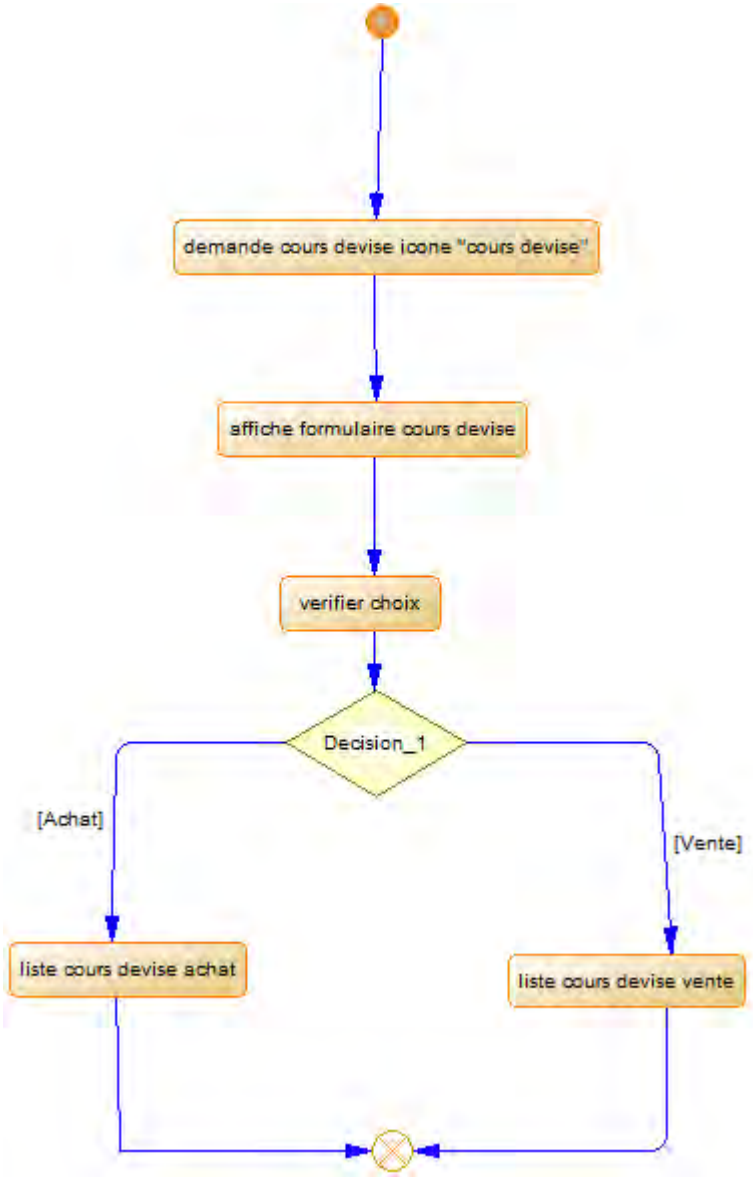


Figure 19:Diagramme d'activité «Consulter cours devise»

V.2.2.1 : le diagramme d'activité << convertir billet de banque >>

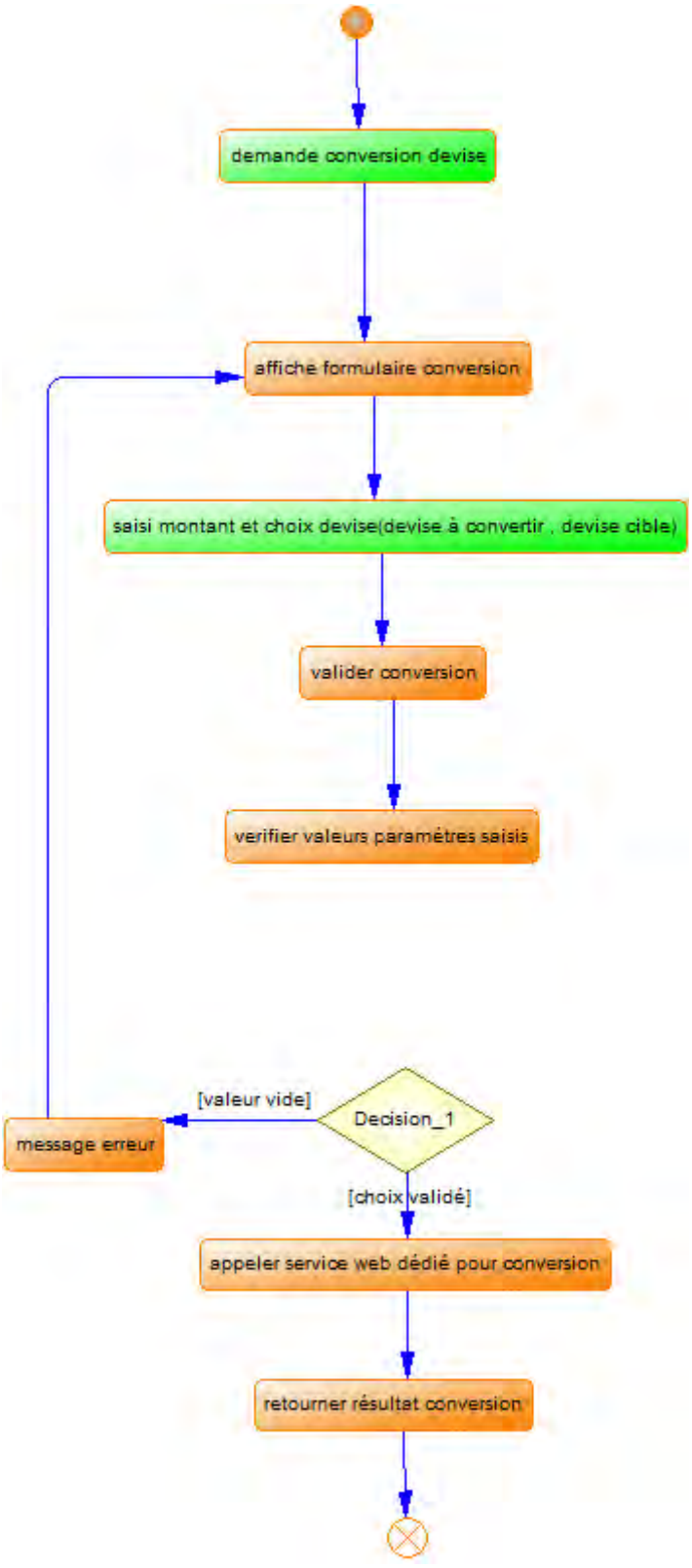


Figure 20:Diagramme d'activité «Convertir billet de banque

V.2.2.1 : le diagramme d'activité << consulter annuaire réseau agences >>

Figure 21:Diagramme d'activité «Commander chéquiers»

V.2.2.1 : le diagramme d'activité << contacter banque >>

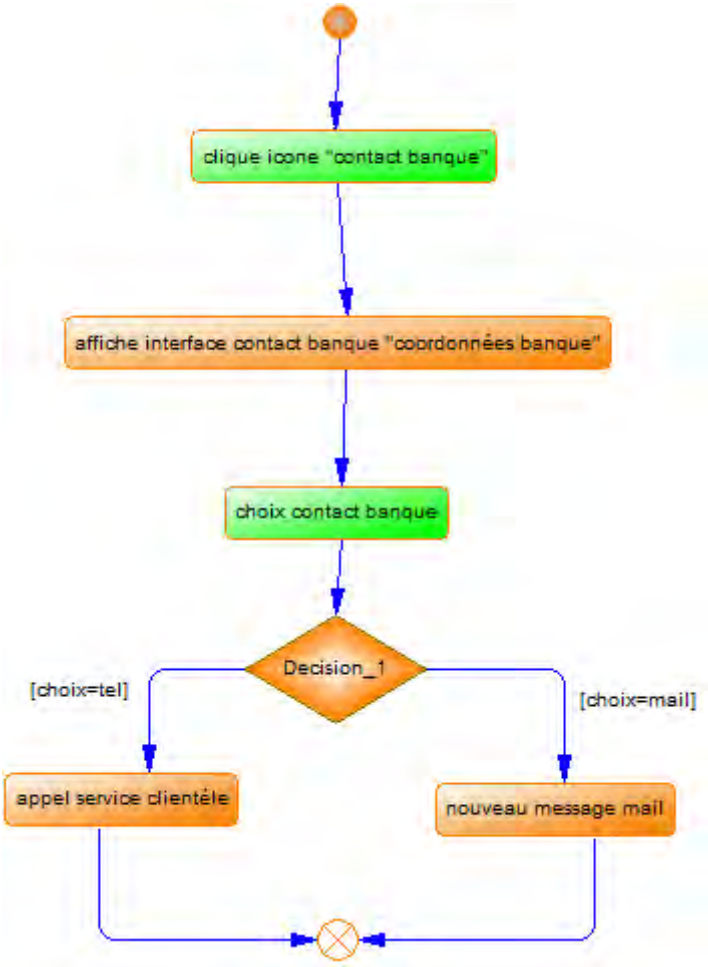


Figure 22:Diagramme d'activité «Contacter banque»

IV : modélisation :

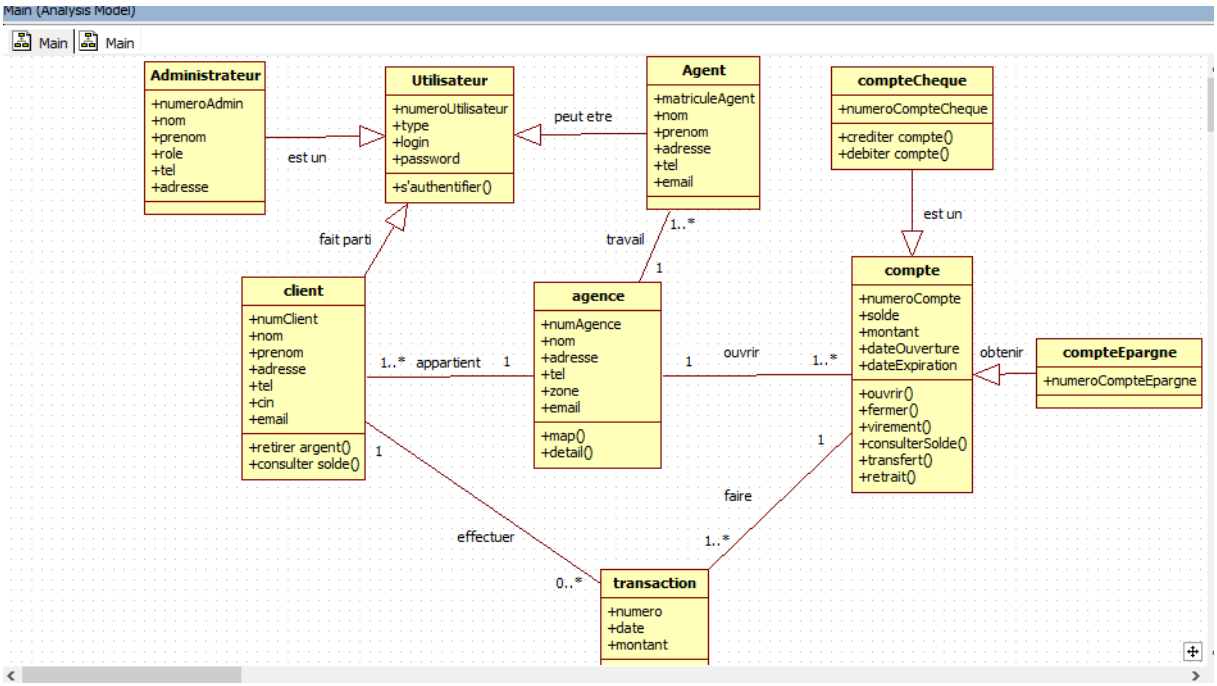


Figure 23:Modelisation

Chapitre VI : Réalisation

Introduction :

Cette partie constitue le dernier volet de ce rapport. Après avoir terminé la phase de Spécification et conception, la solution étant déjà choisie et étudiée, il nous reste que de se Décider dans quel environnement nous allons travailler, exposer les choix techniques utilisés Et le langage adopté, et présenter l'implémentation et les tests réalisés.

VI.1 Choix technique :

VI.1.1 Choix du langage de programmation :

C# :

Le langage C# (C Sharp) est un langage objet crée spécialement pour le Framework Microsoft .NET. L'équipe qui a créé ce langage a été dirigée par *Anders Hejlsberg*, un informaticien danois qui avait également été à l'origine de la conception du langage *Delphi* pour la société Borland (évolution objet du langage Pascal).Le **Framework .NET** est un environnement d'exécution (CLR Common Language Runtime) ainsi qu'une bibliothèque de classes (plus de 2000 classes). L'environnement d'exécution (CLR) de .NET est une machine virtuelle comparable à celle de Java. Le Runtime fournit des services aux programmes qui s'exécutent sous son contrôle : chargement/execution,isolation des programmes, vérification des types, conversion code intermédiaire (IL) vers code natif, accès aux métadonnées (informations sur le code contenu dans les assemblages .NET), vérification des accès mémoire(évite les accès en dehors de la zone allouée au programme), gestion de la mémoire (Garbage Collector), gestion des exceptions, adaptation aux caractéristiques nationales (langue, représentation des

nombres), compatibilité avec les DLL et modules COM qui sont en code natif (code non manage).Les classes .NET peuvent être utilisées par tous les langages prenant en charge l'architecture .NET. Les langages .NET doivent satisfaire certaines spécifications : utiliser les mêmes types CTS (Common Type System),
Les compilateurs doivent générer un même code intermédiaire appelé MSIL (Microsoft Inter médiate Language).Le MSIL (contenu dans un fichier .exe) est pris en charge par le Runtime .NET qui le fait tout d'abord compiler
Par le JIT compiler (Just In Time Compiler). La compilation en code natif a lieu seulement au moment de l'utilisation du programme .NET. Définir un langage .NET revient a fournir un compilateur qui peut générer du langage MSIL. Les spécifications .NET sont publiques (Common Language Spécifications) et n'importe quel éditeur de logiciel peut donc concevoir un langage/un compilateur .NET. Plusieurs compilateurs sont actuellement disponibles : C++.NET (version Managée de C++), VB.NET, C#, Delphi, J# Lors du développement d'applications .NET, la compilation du code source produit du langage MSIL contenu dans un fichier .exe. Lors de la demande d'exécution du fichier .exe, le système d'exploitation reconnait que l'application n'est pas en code natif. Le langage IL est alors pris en charge par le moteur d'exécution du
Framework .NET qui en assure la compilation et le contrôle de l'exécution. Un des points importants étant que le Runtime .NET gère la récupération de mémoire allouée dans le tas (Garbage Collector), a l'instar de la machine virtuelle Java. *«En .NET, les langages ne sont guère plus que des interfaces syntaxiques vers les bibliothèques de classes. «Formation à C#, Tom Archer, Microsoft Press.* On ne peut rien faire sans utiliser des types/classes fournis par le Framework .NET puisque le code MSIL s'appuie également sur les types CTS. L'avantage de cette architecture est que le Framework .NET facilite *l'interopérabilité* (projets constitues de sources en différents langages). Avant cela, faire cohabiter différents langages pour un même exécutable imposait des contraintes de choix de types “communs” aux langages ainsi que de respecter des conventions d'appel de fonctions pour chacun des langages utilises.
Remarque (particularité de C++.NET) : la version .NET de C++ a la particularité de permettre à l'utilisateur de générer soit du code natif soit du code manage. Avec Visual C++ Express, le langage C++ a été complet pour permettre la gestion managée de la mémoire allouée dans le tas.
C# est un langage en perpétuel mouvement. Changeant sans rien remettre en cause, s’améliorant au-delà de ce que bon nombre de développeurs pouvait même imaginer. Cette métamorphose pousse C# vers un langage fonctionnel supportant un style de plus en plus déclaratif. Bien entendu les avancées du langage font intégralement partie du bouillonnement général d’idées qui est celui des équipes Microsoft depuis ce que j’appellerais « l’ère .NET ».
En effet, depuis le lancement de la plateforme .NET (le Framework et C#), ce sont régulièrement des idées plus innovantes les unes que les autres que nous propose Microsoft. Qu’il s’agisse de WPF, WCF, WF, de Silverlight, ou bien de l’Entity Framework et donc aussi de LINQ (et je raccourci volontairement la liste à laquelle on pourrait ajouter Microsoft Ajax, Astoria, …), chacune de ces évolutions pourrait à elle seule passer pour une révolution géniale chez un autre éditeur… Ne nous laissons pas par habitude, tellement le rythme des nouveautés est soutenu et gardons intact notre capacité d’émerveillement ! Le Framework 3.5 regorge d’idées nouvelles, C# 3.0 n’est finalement qu’une partie de cette immense galaxie.
Avec le recul lorsqu’on sait ce que va être Entity Framework 7, lorsqu’on sait ce qu’est devenu C# jusqu’à sa version 6 on mesure encore plus la chance que nous avons d’avoir fait le bon choix !

Apache Cordova :

Apache Cordova est un Framework de développement mobile open-source. Il permet d'exploiter les technologies Web courantes telles que HTML5, CSS3 et JavaScript pour développer des applications multiplateformes, évitant ainsi l'utilisation des langages natifs propres aux différentes plates-formes mobiles. Les applications s'exécutent dans des wrappers ciblés pour chaque plate-forme, elles s'appuient sur des API conformes aux standards permettant l'accès aux capteurs de chaque appareil, aux données ainsi qu'à l'état du réseau.

Apache Cordova a obtenu son diplôme en octobre 2012 comme un projet de niveau supérieur au sein de l'Apache Software Foundation (ASF). Par le biais de l'ASF, développement futur de Cordova assurera intendance ouvert du projet. Il restera toujours gratuit et open source sous Apache License, Version 2.0. Visitez cordova.apache.org pour plus d'informations.

Utiliser Apache Cordova, si vous êtes :

- un développeur mobile et que vous voulez étendre une application à plusieurs plates-formes sans avoir à ré implémenté celle-ci dans chacun des langages et avec chacun des outils propres aux différentes plates-formes.
- un développeur Web et que vous souhaitez déployer une application Web prête à être distribuée dans divers portails de vente d'applications.
- un développeur de mobile intéressé par la combinaison de composants de l'application native avec une *WebView* (fenêtre de navigateur spécial) qui peut accéder aux API de niveau périphérique, ou si vous voulez développer une interface plugin entre les autochtones et les composants WebView.

Universal Windows Platform (UWP) :

Universal Windows Platform (**UWP**) ou **plateforme d'application universelle Windows** est une architecture homogène créée par Microsoft et introduite pour la première fois dans Windows 10. L'objectif de cette plate-forme logicielle est d'aider à développer des applications universelles qui fonctionnent sous Windows 10, Windows 10 Mobile, Xbox One et Hololens sans qu'il y ait le besoin de réécrire un nouveau code source pour chacun de ces systèmes. Il prend en charge le développement d'applications Windows en utilisant C++, C#, VB.NET et XAML. L'API est implémentée en C++ et prise en charge en C++, VB.NET, C#, F# et JavaScript. Conçu comme une extension de la plate-forme Windows Runtime introduite pour la première fois dans Windows Server 2012 et Windows 8, UWP permet aux développeurs de créer des applications susceptibles d'être exécutées sur plusieurs types de périphériques.

Xamarin :

1. Construire une application mobile multiplateformes et native en C#.

2. Historique de Xamarin En 2001, la compagnie Ximian de Miguel De Icaza lance le projet mono, une implémentation des normes de langage ECMA C#.Le but est de pouvoir créer des applications en .NET pour les plateformes UNIX / Linux. En 2004, rachat de Ximian par Novell 2011, Attachmate acquiert Novell et se débarrasse de Mono Miguel De Icaza et une partie de son équipe fondent alors Xamarin pour poursuivre le travail et faire évoluer Mono Décembre 2012 : Xamarin pour Mac Février 2013, Xamarin 2.0 : Xamarin Studio, Xamarin iOS, Xamarin Android

3. Pourquoi utiliser Xamarin ? D'ici 2016, 70% des travailleurs mobiles auront un téléphone intelligent et 90% des entreprises auront au moins 2 plateformes à supporter – Gartner Research Les utilisateurs veulent du natif, ne serait-ce que pour les performances. Créer des composants partagés entre plusieurs plateformes, pour éviter de devoir réécrire 4 apps différentes. Éviter la courbe d'apprentissage de langages comme Objective-C par exemple.

4. Qu'est-ce qu'une app « native » ? Les applications natives sont construites avec des interfaces utilisateurs se conformant parfaitement aux principes de design spécifiques à chaque plateforme.

Elles offrent des performances optimales en utilisant l'accélération matérielle des diverses plateformes. Elles peuvent aussi accéder à toutes les fonctionnalités de la plateforme. Par exemple, elles peuvent accéder aux nouveaux API de reconnaissance d'empreintes digitales d'Apple.

5. Architecture

6. Les composants Xamarin. IOS (ex MonoTouch) Sur iOS, le compilateur AOT (Ahead Of Time) génère directement du code natif assembleur ARM Génère un fichier .app Xamarin. Android (ex Mono for Android) 🗣Ici, le compilateur génère d'abord en IL (Intermediate Language) qui sera compilé en natif au lancement de l'application grâce au compilateur JIT (Just In Time) Génère un fichier .apk

7. Le partage de code selon Xamarin Selon leur documentation, une moyenne de 75% du code

D'une application peut être partagé.

8. Option File Linking Façon la plus répandue de partager du code, on crée un répertoire

Distinct sur le disque et on le partage entre les différents projets Beaucoup de manipulations manuelles sont nécessaires cependant. Pas de façon simple de le faire dans Visual Studio entre autres. Par exemple, si on ajoute un fichier, il faut le faire dans tous les projets qui partage le code parce pas d'option de Link de dossiers.

9. File Linking

10. Autres méthodes de partage de code Fichiers de projets clonés (fonctionne mieux dans Xamarin Studio mais demande d'être vigilant, car il y a de la gestion manuelle) Microsoft Project Linker (fonctionne seulement avec Visual Studio et n'est pas supporté officiellement)

11. IDE Outre l'intégration avec Visual Studio, ils ont leur propre IDE, Xamarin Studio. Tourne sous Windows ou Mac OSX

12. Xamarin en chiffres Plus de 425 000 développeurs au moment d'écrire ces lignes 17 000 consommateurs payants

13. Quelques bémols 75% du code partagé ? Vraiment ? OK, le Model,

la BL, certains appels à des API externes, etc. Mais malgré tout, le code commun comportera son lot de « if » Vous devrez tout de même apprendre les paradigmes de chaque plateforme pour pouvoir développer. Une « vue » est un concept totalement différent sur iOS et sur Android. Le fichier généré tend à être plus lourd vu les références à des librairies externes comme LINQ par exemple.

VI.1.2 Choix de l'architecture de l'application :

Client/ Serveur

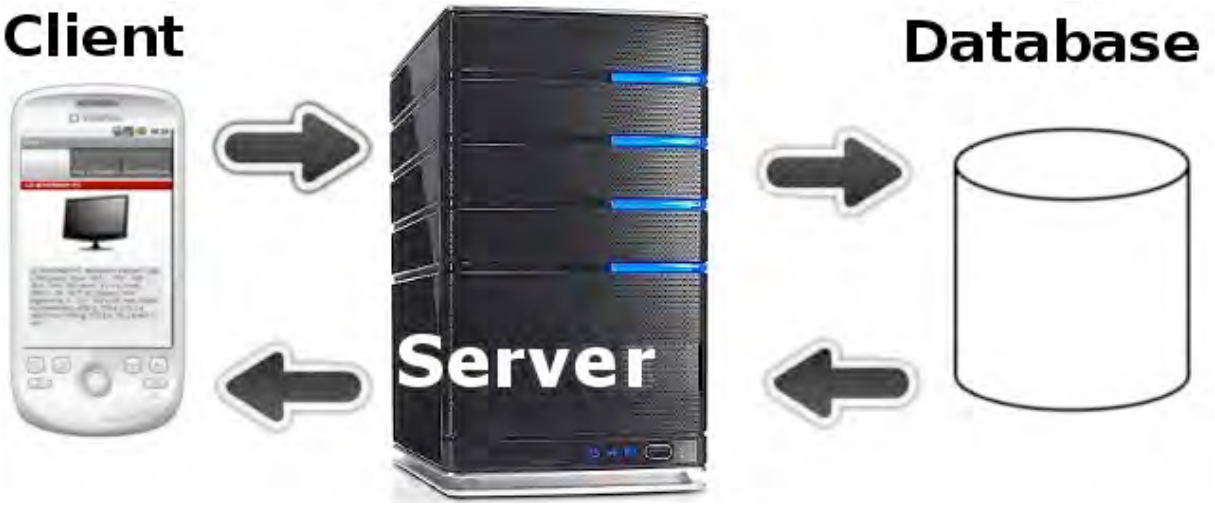


Figure 24:Architecture à 3 niveaux

Dans l'architecture à trois niveaux, les applications au niveau serveur sont délocalisées, c'est à-dire que chaque serveur est spécialisé dans une tâche (serveur web/ serveur de base de Données par exemple). Il permet :
- une plus grande flexibilité/souplesse ;
- une sécurité accrue car la sécurité peur être définie indépendamment pour chaque Service, et à chaque niveau ;
- de meilleures performances, étant donné le partage des tâches entre les différents Serveurs. Cette architecture (appelée 3 tiers) fait intervenir trois parties indépendantes les unes des autres :
1. la couche de données liée au serveur de base de données (SGBD) : stockage et accès Aux données. Le système de stockage des données a pour but de conserver une Quantité plus ou moins importante de données de façon structurée. Nous pouvons Utiliser pour cette partie des systèmes très variés qui peuvent être des systèmes de Fichiers, des mainframes, des systèmes de bases de données relationnelles, etc.
2. la logique applicative : il se compose généralement d'un script ou d'un programme qui

Constitue les traitements métier nécessaires sur l'information afin de le rendre Exploitable par chaque utilisateur.
3. La couche présentation (ou affichage) associé au client qui de fait est dit « léger » Dans la mesure où il n'assume aucune fonction de traitement à la différence du modèle 2-tiers. C'est la partie la plus immédiatement visible pour l'utilisateur. Elle a donc une Importance primordiale pour rendre l'information lisible, compréhensible et Accessible.

VI.2 ENVIRONNEMENT LOGISTIQUE :

IV.2.1 Environnement de développement :

Microsoft Visual Studio est une suite de logiciels de développement pour Windows et macOS conçue par Microsoft. La dernière version s'appelle **Visual Studio 2017**.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.

Visual studio 2015 :

Disponible depuis le 20 juillet 2015, cette nouvelle version apporte:

- la possibilité de se connecter avec plusieurs comptes ;
- le développement multiplateforme mobile (IOS, Android, Windows Phone) ainsi que le débogage ;
- la prise en charge du débogage des applications DirectX 12 pour les diagnostics des graphiques ;
- la possibilité de se connecter à divers services (Azure, Salesforce, Office 365) ;
- l'ajout de l'analyse dynamique ;
- l'amélioration de l'installation d'outils d'extensibilité ;
- l'ajout d'une fonctionnalité pour envoyer des commentaires auprès de Microsoft ;

Cette version marque une fusion entre les éditions *Premium* et *Ultimate* pour simplifier le choix : trois éditions sont disponibles : *Community*, *Professionnel* et *Enterprise*.

Le numéro de version interne de Visual Studio 2015 est 14.0 (le symbole `_MSC_VER` étant défini comme 1900).

VI.3 Travail Réalisé

La conception des interfaces de l'application M-Banking est une étape très importante puisque toutes les interactions avec le couleur de l'application passent à travers ces interfaces, nous devons alors guider l'utilisateur avec les messages d'erreur et avec des notifications si besoin.

VI.3.1 Les jeux de Test :

Dans cette partie, nous allons présenter quelques cas d'utilisations, sous forme d'un guide Utilisateur. Pour accéder à notre application le client banque doit s'authentifier. Comme toute application, La sécurité d'accès est nécessaire. La figure ci-après donne l'interface à travers laquelle L'utilisateur s'identifie. Il saisit son login et son mot de passe puis le serveur vérifie ces informations.

Figure 25:Page d'accueil

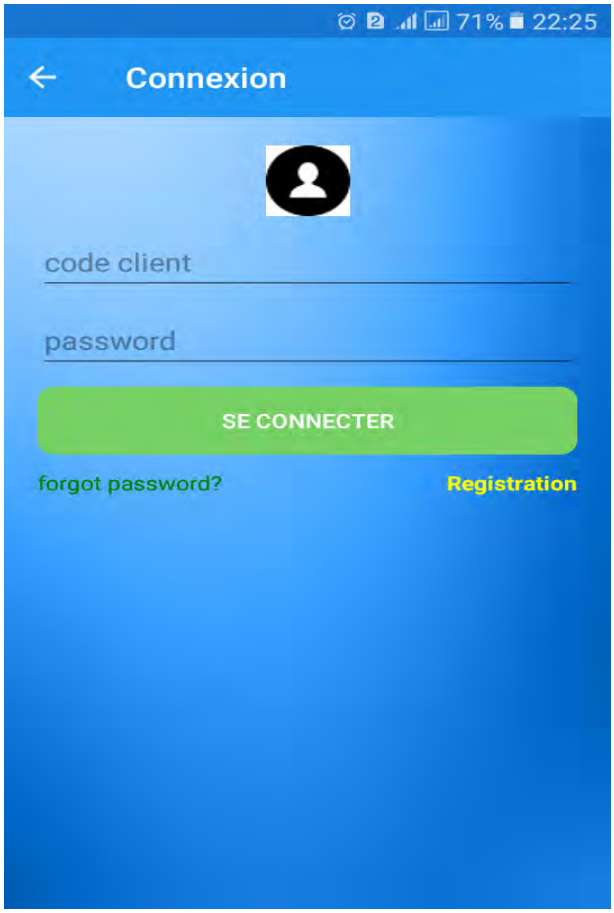


Figure 26:Page de connexion

Si l'utilisateur n'a pas encore de compte sur la plateforme il doit cliquer sur le bouton Registration pour créer un compte en entrant les différentes informations demandées.

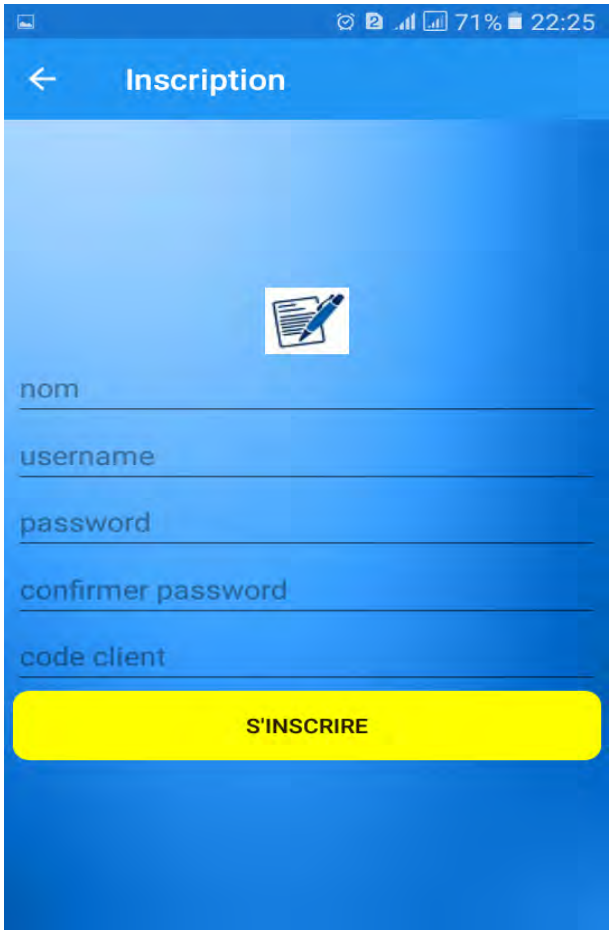
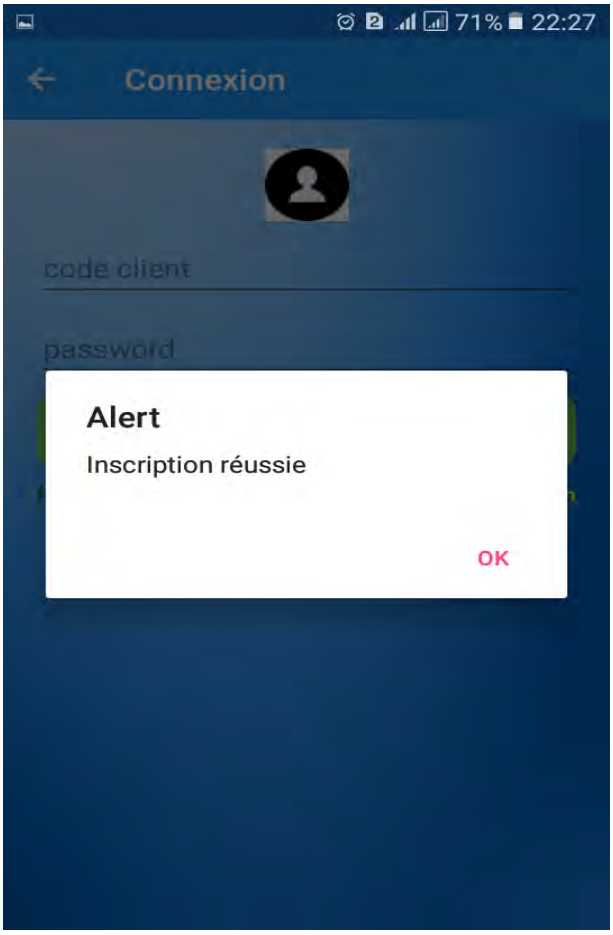
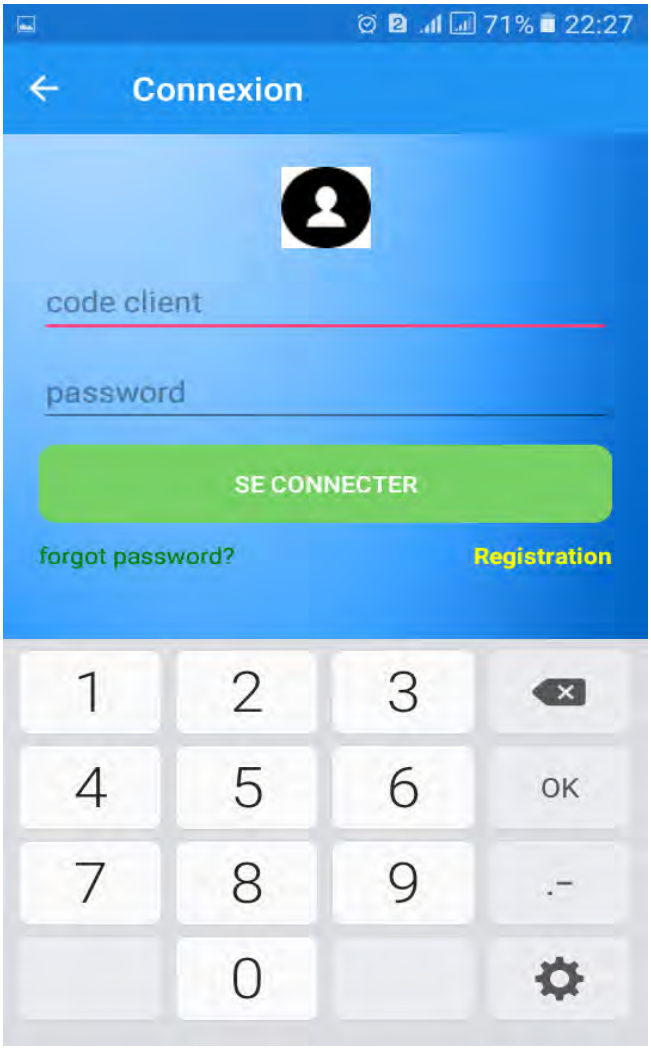


Figure 27:Page d'inscription


71% 22:27
Connexion
code client
password
Alert
Inscription réussie
OK


71% 22:27
Connexion
code client
password
SE CONNECTER
forgot password?
Registration
1
2
3
4
5
6
OK
7
8
9
.-
0

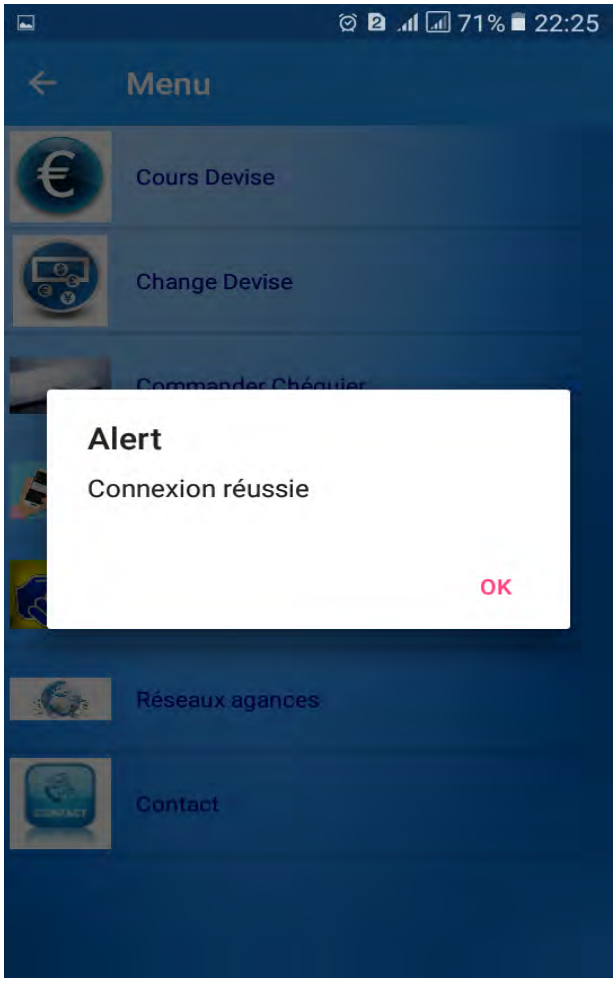


Figure 28:Connexion réussie

Une fois les données sont valides, l'utilisateur accède au menu. Dans cette interface L'utilisateur peut gérer toutes les fonctionnalités de l'application.

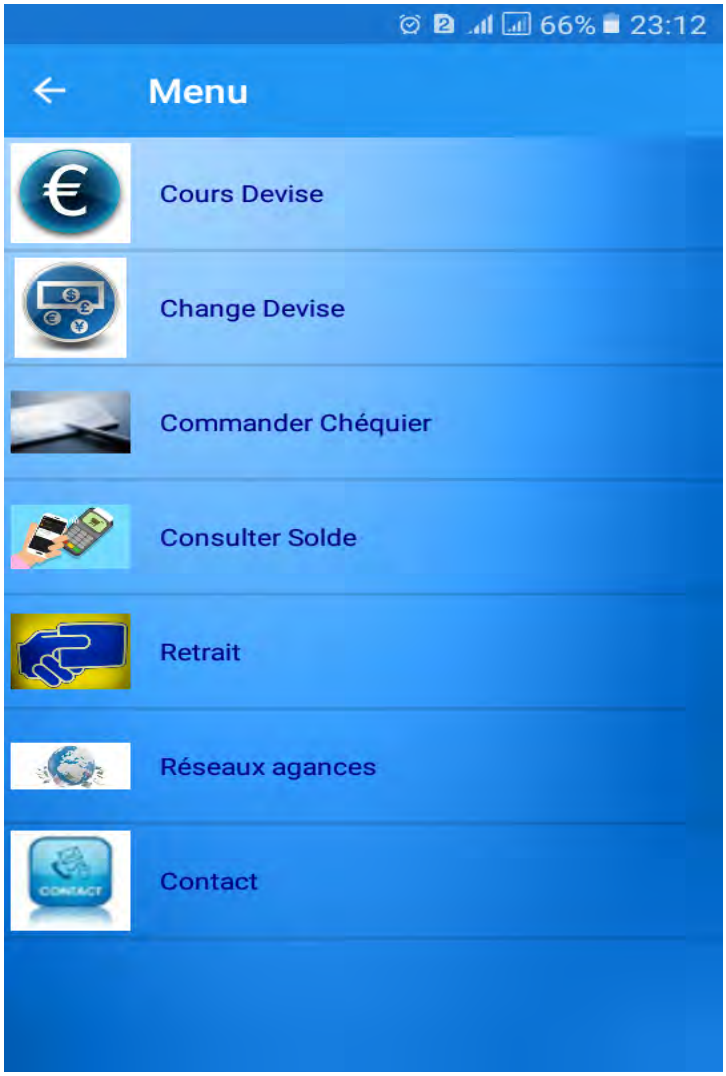


Figure 29:Page de Menu

Si l'utilisateur sélectionne l'icône « cours devises » sur le menu principale. Le système Affiche l'interface relative au cours devises, d'où le client banque peut choisir l'un des Fonctionnalités ; cours achat ou cours vente via les onglets correspondants. Lorsque L'utilisateur choisie onglet achat le système affiche liste cours achat et lorsqu'il choisit cours vente le système affiche cours vente.

Figure 30:Page Cours Devises

Dans l'écran « menu » si l'utilisateur choisit « change devises », le système enchaine à L'interface utilisateur relative au « change devises ».

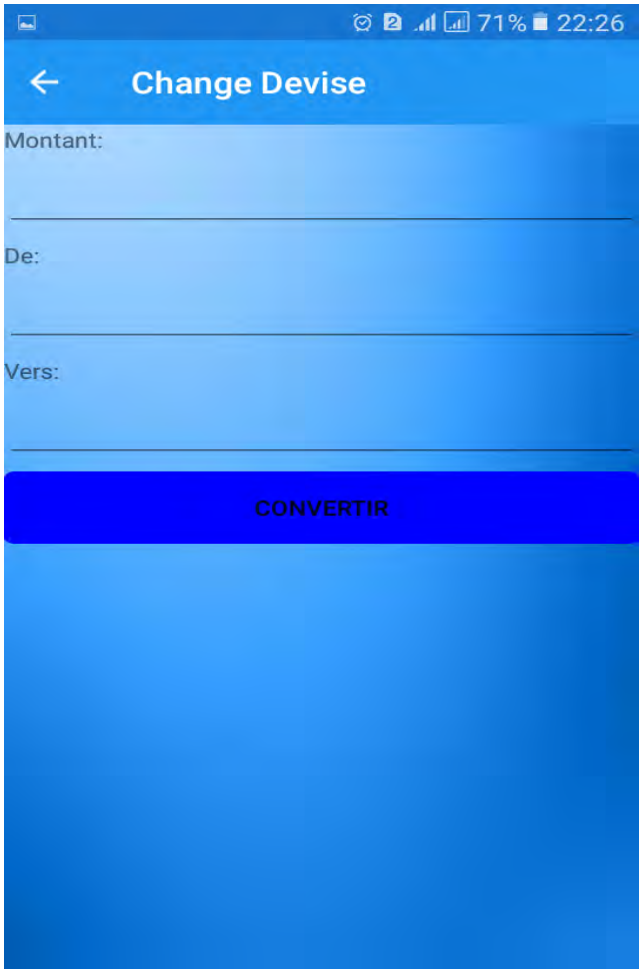


Figure 31:Page Change Devises

L'utilisateur saisit le montant et choisit les devises source et cible du formulaire et valide-la Conversion de billet de banque.

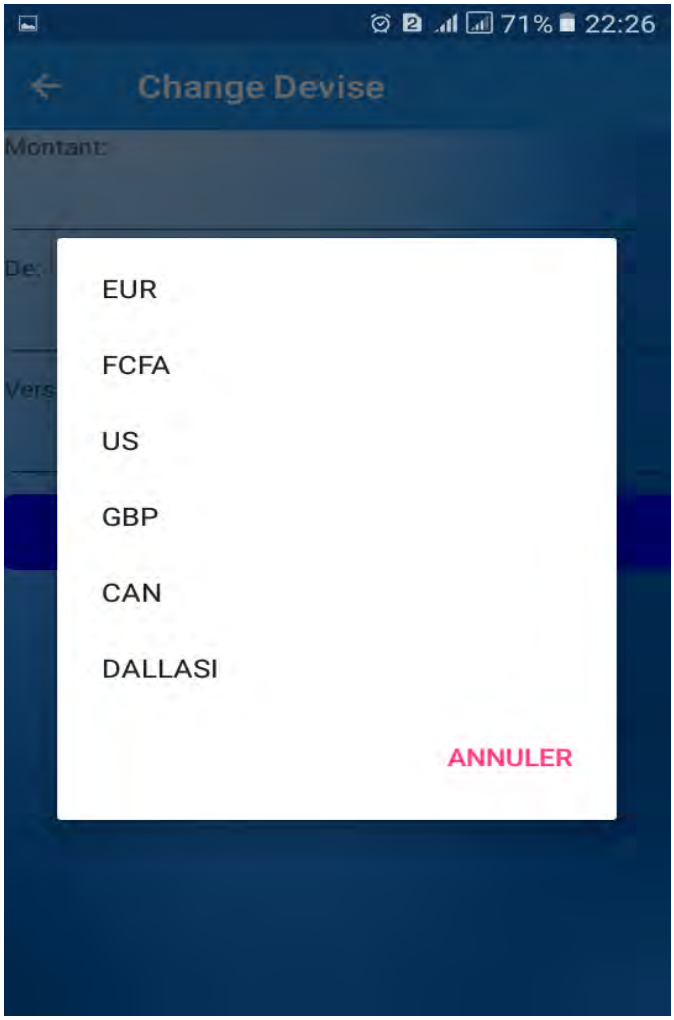


Figure 32:Page Change Devise suite

Enfin, si l'utilisateur choisit de sélectionner la fonctionnalité « *Contact* » l'interface suivante Sera affiché afin de mettre à la disposition du client les coordonnées de la banque et le choix De contacter sa banque par mail ou par téléphone. Donc l'application prend en charge les Actions susvisées.

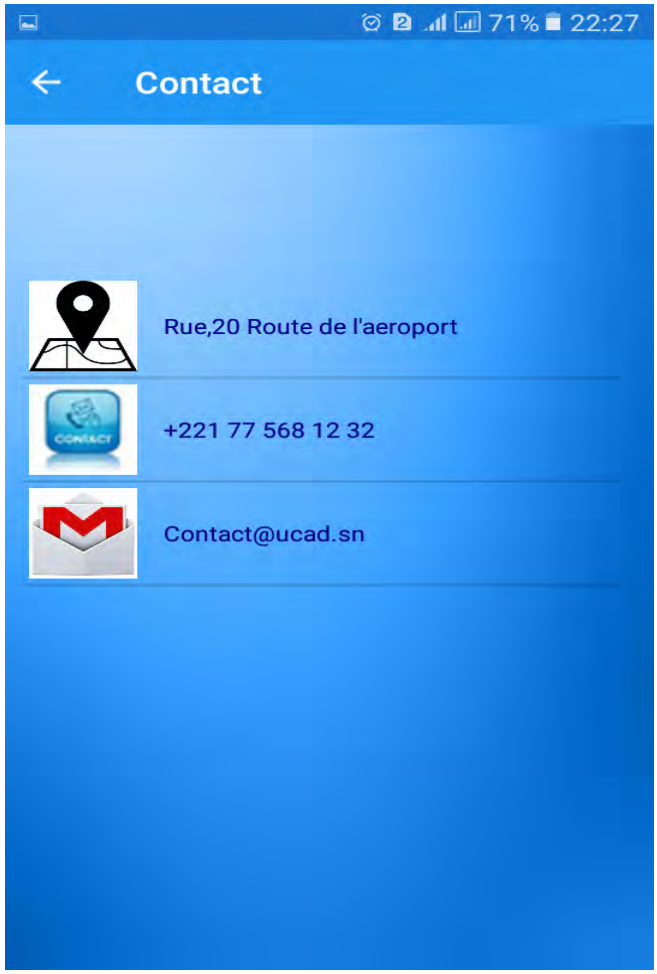


Figure 33:Page Contacts

La figure 34 représente la page d'annuaire des agences. L'interface de cette page est organisée d'une manière lisible pour l'utilisateur. Les informations affichées sont :

- Nom de l'agence,
- Adresse ;
- Numéro de téléphone.

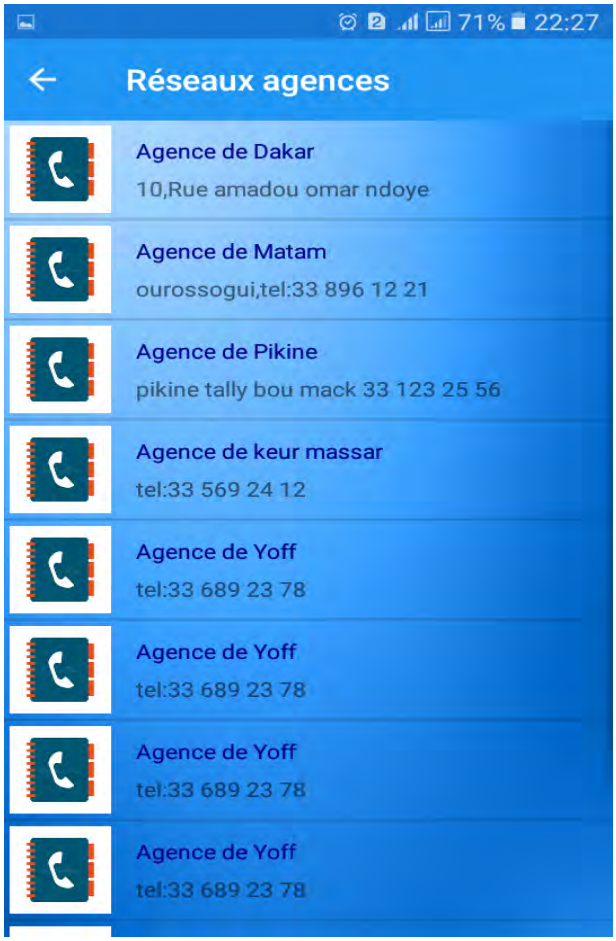


Figure 34:Page annuaire agences

Une fois l'abonné choisit le menu « Consulter compte » la page représentée par la figure 35 sera affiché. Cette page contient le numéro de compte et le solde.

Figure 35:Page Consultation soldes


71% 22:26
solde
Saisir votre n° de compte:
Saisir Votre Mot Passe:
CONSULTER
1
2
3
4
5
6
OK
7
8
9
.-
0

Une fois l'abonné choisit le menu « *Commander chéquier* » la page représentée par la figure 36 sera affichée. À Partir de cette interface, l'utilisateur choisit le nombre de chèques (25, 50 etc.) et finalement, il valide l'opération de commande de chéquier.

Figure 36:Page commande chéquier

Figure 37:commande chéquier suite

Une fois l'abonné choisit le menu *Retrait à distance* la figure 38 sera affichée. On demande au client de saisir son numéro de compte et le montant qu'il veut retirer. Ensuite il va recevoir une notification avec un numéro à quatre chiffres qui lui permet d'aller retirer la somme demandée dans un des distributeurs automatiques de billets (*DAB)* de la banque.

Figure 38:Page Retrait à distance

FIGURE 14 : Page Retrait

VII. Sécurité :

La sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des ressources possèdent uniquement les droits qui leur ont été octroyés.
C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité.
La politique de sécurité est donc l'ensemble des orientations suivies par une organisation en termes de sécurité. A ce titre elle se doit d'être élaborée au niveau de la direction de l'organisation concernée, car elle concerne tous les utilisateurs du systéme.La mise en œuvre se fait selon les quatre étapes suivantes :
Identifier les besoins en termes de sécurité, les risques informatiques pesant sur l'entreprise et leurs éventuelles conséquences ;
Élaborer des règles et des procédures à mettre en œuvre dans les différents services de l'organisation pour les risques identifiés ;
Surveiller et détecter les vulnérabilités du système d'information et se tenir informé des failles sur les applications et matériels utilisés ;
Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une menace ; Cependant on est conscient qu'il n'y a pas de sécurité absolue, il faut une sécurité adaptée au contexte et aux enjeux de l'organisation concernée Ainsi, dans notre étude nous allons établir un plan de sécurité qui structuré ainsi :

VII.1 Sécurité organisationnelle :

Sur le plan organisationnel la politique de sécurité consistera à :
Définir les règles de sécurité
Former et sensibiliser tous les agents et utilisateurs qui ont accès aux données de l'application mobile.
Définir les moyens de contrôles et réaliser les contrôles.

VII.1.2 Sécurité physique :

La sécurité physique s'articulera en trois points bien précis :
Protection des biens matériels et des locaux qui doivent abriter ces biens matériels
Protection contre les incendies, sinistres etc.
Contrôler l'accès aux locaux, aux postes de travail et autres biens matériels

VII.1.2 Plan de secours:

Un plan de secours est un ensemble de mesures permettant d'assurer la continuité de l'activité en cas de sinistre. Dans le plan de secours il faudra établir :
Les mesures de sauvegarde : dupliquer la sauvegarde des données en des endroits différents
Prévoir des solutions d'urgence pour la continuité du service

VII.1.2 Sécurité Logique :

Elle peut s'articuler sur les points suivants :
Sécurité des accès : Elle a pour objectif de protéger contre les intrusions
Sécurité des échanges : elle permet la confidentialité et le non, altérabilité des données qui s'échangent dans l'organisation
Sécurité des systèmes d'exploitation : elle consiste à une définition des règles de configuration et une parfaite maitrise des fonctions de sécurité des systèmes d'exploitation

CONCLUSION GENERALE ET PERSPECTIVES

L'accroissement du marché des mobiles est plus important que ceux des marchés des ordinateurs, cet accroissement est traduit par des millions d'applications qui sont présentes en ligne. (Source Google)Suite à l'émergence des nouvelles technologies de l'information et de la communication, la plupart des secteurs ont connu des changements importants notamment le secteur bancaire, avec l'avènement de Mobile Banking. Les banques vont devoir redoubler d'efforts afin de proposer des applications plus sophistiquées et contenant plus de fonctionnalités pour offrir aux clients des services rapides à distance, en ligne et faciles à utiliser. Notre application mobile bancaire constitue un apport par rapport à ces objectifs. Nous avons eu l'occasion d'utiliser de nouvelles technologies :

- Système d'exploitation Android.
- Le Framework Xamarin
- Le Langage C#

Dans le but d'améliorer encore notre application, nous pensons qu'il est certainement intéressant de compléter les autres services et d'ajouter d'autres fonctionnalités telles que :

- Virement entre les comptes,
- **simulation des crédits, etc.**
- Gestion *BackOffice* qui permet à un administrateur de la banque de répondre aux demandes des abonnés (commandes de chéquier, etc.).

Ces perspectives permettent d'apporter un haut niveau de satisfaction pour les clients.