République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **CONDUITE DE PROJET INFORMATIQUE**

Présenté par
Ghiles AMARA
Marzouk OUANES

Thème

Conception et réalisation d'une application Android pour un service de restaurant

Mémoire soutenu publiquement le 23/06/ 2016 devant le jury composé de :

Président : Mr M. DAOUI
Encadreur : Mme R. HADAOUI
Examinateur : Mme R. AOUDJIT
Examinateur : Mme M. BELKADI

Remerciements

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et bénit par sa volonté durant toute cette période. Nos remerciements vont à notre promotrice « Mme hadaoui » qui nous a fait l'honneur de diriger notre travail. Comme nous remercions notre amis « karim» pour son aide tout ou long de notre projet.

Nous adressons nos remerciements aux membres du jury, devant qui nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire notre mémoire pour juger son contenu.

Dédicaces

Je dédie ce modeste travail :

A mes parents, ce travail vous doit beaucoup, qu'il soit pour vous le témoignage de ma reconnaissance infinie pour ces années de compréhension et d'efforts communs.

A mon frère karim et ma sœur lydia est surtout notre chouchou dadi en leur souhaitant beaucoup de Chances et de réussites.

A tous mes cousins et cousines, oncles et tantes.

A tous mes amis, qu'elles trouvent ici l'expression de mon estime.

A celle qui ma toujours encouragé est soutenue « ma chère imane » et qui souhaité ma réussite.

Merzouk

Dédicaces

Je dédie ce travail à ma mère celle a qui tous les compliments du monde ne représentent rien à côté d'elle, qui m'a soutenu durant les meilleurs moments et les plus dures aussi Toute la gratitude du monde ne saurait exprimé ce que je ressens envers elle.

A mon petit frère Anis à qui je souhaite une bonne réussite.

A toutes ma famille oncles ,tantes ,cousins ,cousines

A tous mes ami(e)s

A tous ceux qui mon soutenus

Ghiles

Sommaire

Chapitre I
Le développement des applications mobiles

Chapitre II
Android

Chapitre III
Analyse et Conception

Chapitre IV
Réalisation

Liste des figures

Liste des tableaux

Introduction générale

Introduction générale

Dans un monde actif et continuellement évolutif, la motivation d'avoir des moyens performants et efficaces de communication et d'échange d'informations devient de plus en plus fondamentale. Cette motivation donne naissance à une révolution favorisant le travail à distance et l'accès aux besoins en temps réduit à l'aide d'internet qui a bouleversé les habitudes de travail dans de nombreux métiers. D'après beaucoup d'analyses et statistiques effectuées, il s'avère que de plus en plus d'internautes se connectent désormais à internet via leurs téléphones portables. Nous remarquons ces dernières années un développement exponentiel des appareils mobiles qui sont répandus comme une traînée de poudre dans le monde en développant et révolutionnant le domaine des communications notamment dans les zones rurales. Dans ce cadre, les Smartphones apparaissent pour rompre avec nos anciennes idées sur les téléphones portables et donner une autre dimension à cette technologie tout en intégrant de nouveaux apports à la téléphonie mobile et en attirant la clientèle grâce à l'ergonomie exponentielle et révolutionnaire. Grace à l'arrivée de ce miracle de la technologie, les usages des téléphones mobiles vont être modifiés d'une manière significative. Les appareils mobiles joueront le rôle de lien entre le monde virtuel du web et le monde physique qui nous entoure. Ils fournissent aux utilisateurs individuels et aux communautés un accès précieux à toute une série de services d'information à des fins personnelles et commerciales.

C'est dans cette optique que s'inscrit notre travail, qui consiste à développer une application Android pour la gestion d'un service de restauration. Pour l'organisation de notre travail, nous avons adopté la structure suivante :

Le premier chapitre intitulé ‹‹ Le développement des applications mobiles ››, donnera un aperçu général sur les mobiles.

Le second chapitre intitulé ‹‹ Technologie Android ››, présentera cette technologie.

Le troisième chapitre intitulé ‹‹ Analyse et conception ››, sera consacrée à l'analyse et la conception de notre application, en utilisant le langage UML.

Le quatrième chapitre intitulé ‹‹ Réalisation ››, présentera les différents outils utilisés pour le développement de notre application, suivis de quelques captures d'écran illustrant son fonctionnement.

Enfin, une conclusion générale dans laquelle sont récapitulés les points clés de notre travail, les difficultés rencontrées et les améliorations.

Chapitre I

Le développement des applications mobiles

Introduction

Les enjeux du mobile ont considérablement augmenté et font maintenant partie intégrante des stratégies des entreprises. Néanmoins, si l'importance du mobile est un fait globalement partagé, le sujet reste complexe à aborder pour la plupart des entreprises car il demeure nouveau.

Une application mobile se distingue d'une application desktop, ce qui fait la différence de son processus de développement par rapport aux logiciels de PC. Avec les nouveaux facteurs à considérer dans le cycle de vie d'une application mobile les acteurs du domaine ont proposé des méthodes de gestion de projet spécifiques pour le développement mobile.

Nous allons aborder dans ce chapitre les différents concepts et facteurs à prendre en considération lors de développement des applications mobiles.

I. Définition d'application mobile

Les applications mobiles sont des logiciels développés pour les petits appareils portables, comme les téléphones mobiles, Smartphones, PDA et ainsi de suite. Les applications mobiles peuvent être chargées par l'utilisateur sur l'appareil mobile et téléchargées sur les plateformes de téléchargement d'applications (Apps Stores) ou sur Internet.

Ils existent plusieurs systèmes d'exploitation mobiles les plus populaires sont : Windows Mobile, Android, iOS, Symbian, Java ME et Palm [1].

II. Les Types d'application mobile

II.1. Application native

Une application native est une application dédiée à un périphérique mobile (Smartphone, Tablette,… etc). Elle est installée directement sur l'appareil et accessible à travers une icône. Les utilisateurs acquièrent généralement ces applications à travers une boutique en ligne ou sur un store en ligne comme l'App Store ou Android Apps sur Google Play. Une application native est développée pour qu'elle fonctionne sur une plateforme précise, elle peut profiter de toutes les fonctionnalités qui existent dans l'appareil : GPS, Accéléromètre, Appareil photo, la liste des Contacts… Les applications mobiles peuvent aussi interpréter les gestes tactiles sur l'écran de l'appareil, peuvent utiliser le système de notification de l'appareil et peuvent même fonctionner hors ligne.

II.2. Application Web Mobile

Une application web mobile (ou webapp) est une application utilisée via Internet et possédant des fonctionnalités spécifiques pour les appareils mobiles. Elles ne sont pas réellement des applications mais plutôt des sites web adaptés à l'usage mobile. Elles sont accessibles exactement comme les sites web c'est-à-dire via le navigateur Web de l'appareil mobile et n'ont pas besoin d'être téléchargées et installées sur l'appareil pour fonctionner, mais leurs installation reste possible et permet de faciliter l'accès à une application par la création de l'icône sur l'écran d'accueil de l'appareil.

Les applications Web Mobile ne bénéficient pas de de toutes les fonctionnalités de l'appareil mobile. Pour prendre avantage de ces fonctions il faut opter pour une application native ou une solution hybride que nous allons entamer après la comparaison entre site mobile et application mobile.

II.3. Comparatif entre une application mobile et une application native

Voici un tableau comparatif des deux solutions

	Site mobile	**Application native**
Vue par les utilisateurs		
Enjeux	URL connue Réponse rapide	Classification Notoriété marque Cohérence Tops
les utilisateurs reviennent sur le « service » (l'application)	 Favoris ou raccourcis (icone standard ou personnalisée) Connexion internet est nécessaire	 Icones Connexion internet ou hors ligne
Les fonctions	Accès limité aux fonctionnalités et informations : Web, contenus, transactions, paiement par CB, géolocalisation, interface tactile (partiele), en partie mode déconnecté (HTML5)	Accès à toutes les fonctionnalités : Idem + Look & feel premium, paiement in-app., accéléromètre, contacts, vibreur, caméra, flash, d'avantage de puissance, mode full déconnecté, full tactile

Mise à jour	La mise à jour s'effectue sur le serveur web : tous les utilisateurs sont mis à jour	La mise à jour peut être ignorée par les utilisateurs
Point fort	Possèdent une base de code commune sur toutes les plateformes. Les utilisateurs n'ont pas à aller sur un store, ils utilisent l'application depuis le navigateur web. Peut être lancée sous n'importe quelle forme et à n'importe quel moment, aucun tiers (App Store) devant au préalable approuver l'application. Si une application Web existe déjà, elle peut être rendu accessible aux appareils mobiles via une adaptation en Responsive Design	S'exécute généralement plus rapidement qu'une application web mobile. Les App Stores et marketplaces aident les utilisateurs à trouver des applications natives. Le processus d'approbation des applications mis en place par l'App Store peut contribuer à rassurer les utilisateurs sur la qualité et la sécurité de l'application. Les outils, le soutien et la mise à disposition de bonnes pratiques fournies par les fabricants de périphériques peuvent aider à accélérer le développement
Faiblesse	Les applications web mobiles ne peuvent pas accéder à toutes les fonctionnalités de l'appareil. Supporter plusieurs navigateurs Web mobiles peut se traduire par des coûts plus élevés en matière de développement et de maintenance. Les utilisateurs peuvent posséder des navigateurs mobiles différents, ce qui peut rendre la maintenance difficile et ne facilite pas le support utilisateurs. Il peut être difficile de trouver une application web mobile en raison de l'absence d'un store centralisé	Plus coûteuses à développer, surtout si le développement porte sur de multiples appareils mobiles. Prendre en charge plusieurs plates-formes nécessite la gestion de plusieurs bases de code et peut entraîner des coûts plus élevés en matière de développement, de maintenance, de mise à disposition des mises à jour. Les versions présentes chez les utilisateurs peuvent être différentes, ce qui peut rendre la maintenance difficile et ne facilite pas le support utilisateurs. Le processus d'approbation App Store peut retarder le lancement de l'application ou d'empêcher la mise à disposition de l'application pour les utilisateurs.

Tableau 1.1 Comparatif entre une application native et une application Web mobile

II.4. Solution intermédiaire : application hybride

Application hybride est une partie application natives, et une partie application web.

Comme des applications natives :

- Elles existent sur les Apps Stores
- Peuvent profiter des nombreuses fonctionnalités du périphérique.

Comme les applications web :

- Elles dépendent de HTML sachant que le navigateur est intégré dans l'application.
- Accessibles via une url et installée automatiquement.
- Evolutives sans mise à jour à effectuer via les Stores.

Une application hybride est Consultable par Smartphone, Tablette, PC… Et elle est adaptée aux écrans mobiles. Construite en une seule page afin de concentrer le temps de chargement au départ et d'offrir des transitions rapides sans chargement lorsque l'utilisateur change d'écran. Elle fonctionne hors connexion sur la base des données stockées localement.

Une application hybride possède :

- Icône d'application
- Mode plein écran

II.5. Synthèse des possibilités fonctionnelles et techniques des trois approches

Possibilités	Application Native	Application Hybride	Application Web
Appareil photo numérique, gyroscope, carnet de contacts, calendrier, vibreur, compas, état du réseau, push de notification	Oui	Non	Non
Géolocalisation	Oui	Oui	Oui
Fonctionnement hors connexion	Oui	Oui	Non
Stockage de grandes quantités de données (système de fichiers)	Oui	Non	Non
Accès aux applications natives (liées à l'OS comme par exemple la galerie de photo sur iPhone)	Oui	Non	Non
Interfaces natives (boutons, listes, etc…)	Oui	Oui	Oui
Performances et fluidité	Oui	Non	Non
Animations, graphismes avancés	Oui	Oui	Oui
Vidéo, son, images	Oui	Oui	Oui
Accès via URL	Non	Oui	Oui
Accès via stores	Oui	Non	Non
Chargement en une seule fois	Oui	Oui	Non
Mise à jour type web	Non	Oui	Oui
Mise à jour via les stores	Oui	Non	Non

Tableau 1.2 Les possibilités fonctionnelles et techniques des trois approches [2]

III. Cycle de vie application mobile

Le cycle de vie d'une application désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement.

Ils existent plusieurs modèles de cycles de vie d'un logiciel tels que : *Modèle en cascade*, *en V*, en *spiral*, par *incrément*, etc. Parmi tous ces modèles qui existent aucun est spécifiquement visé pour le développement des logiciels mobiles**.**

Certains acteurs du domaine ont essayé de mettre en place quelques approches de cycle de vie d'une application mobile :

3.1. Mobile Application LifeCycle Management « MALM »

Les sept phases de Cycle de Vie d'Application Mobile « CVAM » (Mobile Application LifeCycle Management « MALM ») de Aberdeen Group sont le résultat aspiré des expériences des meilleures pratiques des organisations les plus performantes :

1) Identifier

- Définir les principaux futurs utilisateurs de l'application (client, partenaire d'affaires, et / ou employés) ;
- Déterminer les besoins de l'entreprise ;
- Etayer l'analyse de la rentabilisation.

2) Précisez

- Définir les exigences du produit à réaliser ;
- Décider d'en développer, acheter, ou d'intégrer ;
- Identifier les plates-formes mobiles cibles.

3) Acquérir / Développer

- Identifier l'environnement de développement cible ;
- Développer, et itérer application, intégrer et synchroniser les données d'arrière-plan ;
- Tester et contrôler la qualité (en répétition).

4) Sécuriser l'accès : authentification des utilisateurs ;

- Sécuriser le stockage de l'application qu'il en soit en interne ou externe ;
- Sécuriser l'application lors de la transmission de données.

5) Déployer

- Sélectionner les canaux de distribution, et suivre le produit

6) Gérer

- Gérer les versions des applications ;
- Etablir des politiques pour l'application ;
- Surveiller, analyser et rapporter l'utilisation de l'application.

7) Fin du cycle

- Identifier les applications adaptées à la mise hors service ;
- Suivre et rapporter les applications qui doivent être enlevée ;
- Archiver les applications qui ne sont plus en cours de fonction.

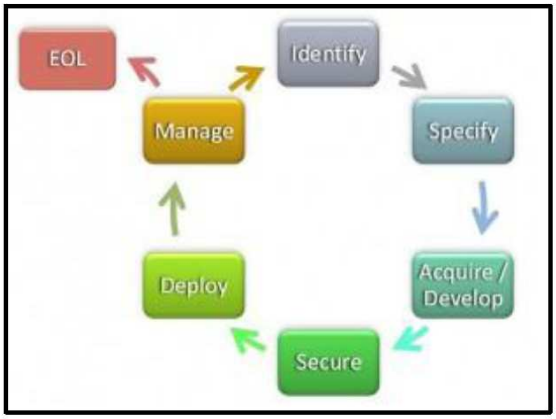


Figure 1.1 Cycle de vie d'une application mobile par : Aberdeen Group,

Lorsque cette approche est combinée avec les principes du développement agile de logiciels elle devient un processus itératif persistant. En conséquence, il n'y a pas une seule et unique approche de développement d'applications mobiles, et nous pouvons toujours apprendre des meilleures pratiques des organisations les plus performantes.

III.2. Mobile-D

III.2.1. Définition

Mobile-D est une approche agile pour l'équipement mobile qui est basée sur XP Extreme Programming (pratique), méthodologie Crystal (scalability) et Rational Unified Process (assurance de cycle de vie). Elle est conçue pour rencontrer les caractéristiques spécifiques de développement de l'application mobile et le standard de qualité de l'industrie. Le cas idéal est d'avoir moins de dix développeurs travaillant dans un espace de bureau.

III.2.2. Pourquoi Mobile-D est adapté à mobile application

En faisant la combinaison des avantages de trois méthodes agiles XP, Crystal et RUP, Mobile-D satisfait les caractéristiques du développement de l'application mobile :

- Changement élevé d'environnement
- Petite équipe de développement
- Client identifiable
- Environnement de développement d'objet orienté
- Pas de sécurité critique de logiciel
- Logiciel de niveau d'application [3]

III.2.3. Les Etapes de Mobile-D

Figure 1.2 Les étapes itératives de La méthodologie Mobile-D [3]

L'approche mobile-D est divisée en cinq itérations. Ces phases sont ***set-up***, ***core***, ***core2***, ***stabilize*** and ***wrap-up***. Chacune de ces phases est constituée de trois types de jours de développement: Planning Day (Jour de planification), Working Day (Jour de travail) and Release Day (jour de délivrance). Le mobile-D est organisée dans un cadre qui relie le principal processus (Planifier, concevoir, mettre en oeuvre, tester, Livrer) avec le processus de support (Gestion de projet, Gestion de configuration de logiciel, amélioration processus logiciel)

1) Explore

Etablissement des groupes de Stakeholders (intervenants) nécessaires dans la planification et la surveillance du projet.

- Définition du porté du projet.
- Planification du projet en considérant des problèmes concernant l'environnement, le personnel, et le processus.

2) Initialize

- Description de L'architecture logicielle
- Préparation des ressources physiques, techniques, humaines et la communication avec client ;
- Préparation du plan de projet et des solutions pour tous les problèmes critiques du développement.

3) Productionize

- Implémentation en commençant par les fonctionnalités prioritaires par le client.
- Se Concentrer sur la fonctionnalité cruciale du produit.

4) Stabilize

- Finissage de l'implémentation du produit
- Amélioration et assurance de la qualité du produit
- Finissage de la documentation du produit

5) System test & fix

- Tester de système en se basant sur la documentation produite dans le projet ;
- Fournir les informations des défauts trouvés ;
- Permettre à l'équipe de projet de planifier et corriger les défauts trouvés ;
- Corriger les défauts.

III.2.4. Evaluation de la méthode Mobile-D

Avantages

- Mobile-D est bien adapté au développement de logiciel mobile réalisé par une petite équipe ;
- Capable de réagir rapidement au changement des besoins des utilisateurs grâce aux courtes itérations (les phases *initialize, productionize, stabilize*).
- Programmation en binôme dans chaque itération est très efficace.

Inconvénients

- Mobile-D est conçue pour adapter au petit système développé par de petite équipe
- Il faut mentionner que cette approche n'est pas complètement définie pour être littéralement utilisée dans la pratique [3].

III.3. MASAM : Mobile Application Software Agile Methodology

Jeong et al. ont proposé Mobile Application Software Agile Methodology (Masam) qui fournit un processus de développement des applications mobiles rapidement en utilisant une approche agile. Il est basé sur l'extrême Programmation (XP), Unified Process Agile, RUP et Software and Systems Process Engineering Metamodel (SPEM **[1]**).Masam et la méthodologie Mobile-D ont une relation étroite, Masam introduit de légères variations par rapport au Mobile-D.

Phase	Activité	Tache
Phase de préparation	Saisir le produit	Description du produit
		Pré-planification
	Partage des concepts de produits	Définition de l'utilisateur
		L'analyse initiale du produit
	La mise en place du projet	La coordination du processus de développement
		Coordination des ressources de développement pré-étude
Phase de concrétisation	La vue utilisateur	Etude des points de vue des membres de l'équipe
		Conception de l'interface utilisateur
	Architecture	Analyse des besoins fonctionnels
		Définition de l'architecture
		Gestion du patron d'architecture
Phase de développement	Mise en œuvre & Préparation	Configuration de l'environnement
		Planification du développement
	Cycle de mise à jour	Planning de mise à jour
		Itération du cycle de mise à jour
		Mise à jour
Phase de commercialisation	Test du système	Test d'acceptante
		Test des utilisateurs
	Vente du produit	Lancement des tests
		Lancement de produit

Tableau 1.3 Les taches des différentes phases de MASAM

Masam propose un cycle de développement d'application mobile cycle qui comprend quatre phases :

La Phase de préparation :

Définit un résumé et une première notion du produit, et attribue les rôles et les responsabilités.

La phase de concrétisation :

Se concentre sur la compréhension des besoins des utilisateurs et définition de l'architecture du produit logiciel.

Phase de développement :

L'implémentation du produit est menée par « le développement piloté par les tests **[2]** » « la Programmation par pair **[3]** », « le Refactoring**[4]** », avec une relation étroite avec les activités de tests itératifs.

Phase commercialisation :

Se concentre sur les produits activités de lancement et la vente de produits.

III.4. Rapid7

Dooms et al. ont proposé une méthode appelée « Rapid7 ' (Rapid Production of Documentation, 7 étapes) qui améliore le travail de la documentation. Rapid7 décrit comment l'interaction humaine est prévue dans les projets logiciels et comment les documents doivent être créés en menant des séances d'atelier animées.

Rapid7 fournit une structure à trois couches : Projet, Cas et des couches d'atelier.

Couches	Description
Couche projet	Décrit comment s'effectue l'interaction humaine et la prise de décision conjointement pour les projets de logiciels.
Couche des cas	Décrit comment les cas sélectionnés, tels que les documents doivent être créés dans des ateliers consécutifs.
Couche atelier	Décrit comment le travail proprement dit est réalisé sous la forme d'atelier, en utilisant sept étapes du procédé.

Tableau 1.4 : Description des différentes couches de structure de la méthode Rapid7 [4]

Les ateliers :

Ateliers Rapid7 comprennent sept étapes. Les étapes visent à fournir des informations sur la façon d'organiser des ateliers efficaces dans les projets logiciels. Les étapes de Rapid7 sont présentées dans la *Figure 1.3*.

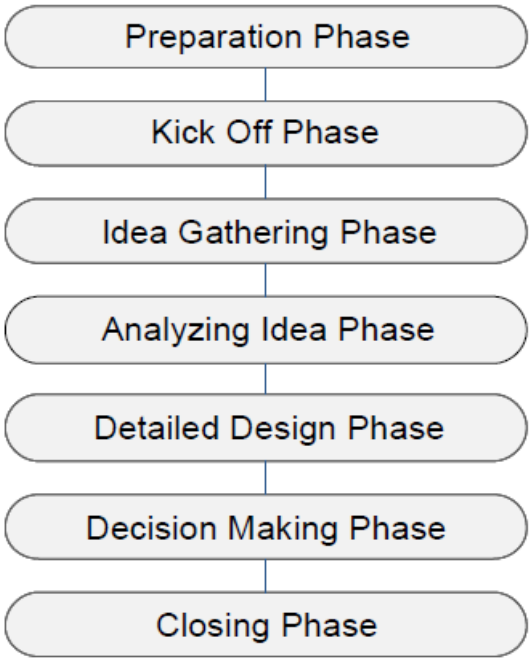


Figure 1.3 Représentation des sept étapes de Rapid7 [5]

1. Phase de préparation : planifier les ateliers en détail.
2. Phase de démarrage : explication des objectifs de l'atelier.
3. Phase de collecte d'idées : rassembler le maximum des idées avec la participation de toute l'équipe.
4. Phase d'analyse des idées : travailler sur les idées collectées.
5. Phase de conception détaillée : détailler chaque idée à un certain niveau.
6. Phase de décision : concerne chaque travail sur chaque idée.
7. Phase de clôture : soumettre le travail réalisé.

Avantages

- Plus d'interaction entre les parties prenantes du projet.
- Mettre l'accent sur la rédaction que sur la documentation essentielle.
- Créer une compréhension commune; un partage plus efficace de l'information.
- Mettre davantage les points, qui devraient aboutir à des résultats plus rapides
- Mieux répondre aux changements.

Mais engendre de fortes pressions en particulier pour :

- Raccourcir la durée d'arriver au marché
- Réagir rapidement à l'évolution de la spécification du produit au cours du développement
- Dépenser moins d'argent.

IV. Les Systèmes d'exploitation mobiles

Un système d'exploitation mobile est un système d'exploitation conçu pour fonctionner sur un appareil mobile. Il existe plusieurs systèmes d'exploitation mobiles (OS) dont les plus répandus sont les suivants :

- iOS (Apple) utilisé sur iPhone et iPad,
- Android (Google) qui anime un grand nombre de smartphones tels que Samsung, HTC, LG, Motorola…
- Blackberry OS,
- Windows Phone (Microsoft),
- Symbian (Nokia),
- Bada (Samsung).

A chaque OS mobile correspond une technologie et un « store » où les applications peuvent être téléchargées, de manière gratuite ou payante :

AppStore pour les app/lications iPhone et iPad,

- Android Market,
- Blackberry App World,
- Marketplace de Windows,
- Samsung Apps Store,
- Ovi de Nokia.

Parmi tous ces systèmes d'exploitation nous allons présenter les plus dominants du marché des applications mobiles (Android, iOS et Windows Phone).

IV.1 . Android

IV.1.1. Définition

Android est un système d'exploitation Open Source pour smartphones, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 15 novembre 2007. Afin de promouvoir ce système d'exploitation ouvert, Google a su fédérer autour de lui une trentaine de partenaires réunis au sein de l'Open Handset Alliance. Ils ont comme principaux concurrents Apple avec iPhone OS qui équipe l'iPhone, Research In Motion avec BlackBerry OS, Palm avec Nova ou webOS, Microsoft et son Windows Mobile, Nokia avec Symbian OS, libéré en 2008, et bien sûr OpenMoko, le projet dont les spécifications logicielles et matérielles sont ouvertes [6].

IV.1.2. Historique

Android doit son nom à la startup éponyme spécialisée dans le développement d'applications mobiles rachetée par Google en août 2005, nom venant lui-même d'« androïde» qui désigne un robot construit à l'image d'un être humain. Le logiciel, qui avait été surnommé gPhone par les rumeurs de marchés et qui selon un de ses concepteurs Andy Rubin était initialement prévu pour être un système d'exploitation pour appareil photo, est proposé de façon gratuite et librement modifiable aux fabricants de téléphones mobiles, ce qui facilite son adoption. Le gPhone a été lancé en octobre 2008 aux États-Unis dans un partenariat de distribution exclusif entre Google et T-Mobile. Anticipant les annonces officielles, les marchés financiers se ruent massivement sur les actions Google les faisant monter jusqu'au plus haut historique de 724 dollars le 5 novembre 2007 (Le vendredi 18 octobre 2013 les actions Google franchissent les 1000 dollars). En 2004, le prix du cours d'introduction du moteur de recherche était de 85 dollars l'action.

IV.1.3. Versions

Depuis le rachat de la startup Android, Google a lancé plus de 18 versions d'Android jusqu'à maintenant et toutes ces version ont des noms de desserts (en anglais) depuis la sortie de la version 1.5 et le comparatif suivant liste les principales différences entre chaque version du système Android.

Version	Nom de Version	Caractéristiques majeures ajoutées	Date de sortie	Part de marché Android
Android 4.4.x	KitKat	□ Nouvelle interface translucide □ Enregistrement séquence vidéo de l'écran □ Amélioration du système de notification □ Gestion système des sous-titres □ Amélioration des performances	31 oct. 2013	1,1 % (4.4 - 4.4.2)
4.3	Jelly Bean	□ Ajout de l'auto-completion sur le pavé de numérotation □ Amélioration de Photo Sphere □ Modification de l'interface de l'application Appareil photo □ Ajout du support pour la résolution 4K □ Possibilité de créer des profils restreints sur les tablettes □ Support de l'hébreu et de l'arabe RTL (écriture de droite à gauche) □ Support de Bluetooth basse consommation (BLE)	24 juil. 2013	4,2 %
4.1.x	Jelly Bean	□ Google Now □ Android Beam (Transfert de données rapide : NFC + Bluetooth) □ Assistant Vocale □ Amélioration de la vitesse d'exécution □ Amélioration gestion appareil photo □ Accessibilité : mode gestuel, prise en charge clavier externe "Braille".	9 juil. 2012	37,4 % (4.1 - 4.1.2)
Android 4.0.x	Ice Cream Sandwich	□ New lock screen actions □ Improved text input and spell-checking □ Control over network data □ Email app supports EAS v14 □ WI-FI direct □ BlueTooth Health Device Profile	19 oct. 2011	18,6 % (4.0.3 - 4.0.4

Tableau 1.5 : Description des dernières versions d'Android [7].

IV.2.ios (Apple)

IV.2.1. Définition

iOS, anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch et l'iPad. Il est dérivé de OS X dont il partage les fondations (le kernel hybride XNU basé sur le micro-noyau Mach, les services Unix et Cocoa, etc.). iOS comporte quatre couches d'abstraction, similaires à celles de Mac OS X : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ».

IV.2.2. Versions

Le tableau suivant présente les différentes versions d'IOS, Il débute avec la sortie de l'iPhone aux États-Unis, le 29 juin 2007.

Version	Nom de version	Date de sortie	Caractéristique	Appareil
5.x	iOS 5	12 octobre 2011	□ Siri □ Notification Center □ PC-free □ iTunes Wi-Fi Sync □ iMessage □ iCloud	iPhone 4 iPhone 4S iPhone 3GS iPad iPad2 iPod Touch 4G iPod Touch 3G
6.x	iOS 6	11 juin 2012	□ Intégration de Facebook, Passbook. □ Amélioration majeure de Siri. □ Nouvelle icône pour les services Twitter. □ Amélioration du Centre de notifications : possibilité d'y poster instantanément ses messages Twitter ou Facebook. □ Amélioration de l'application des Photos : flux de photos partagés. □ Musique sur iPod et iPhone : un changement radical d'interface. □ De nouveaux réglages ont vu le jour, ajout d'une fonctionnalité "Ne pas déranger". □ Amélioration de l'application téléphonique : interface pour passer des appels modifiée et nouvelles options. □ Service de cartographie signé Apple. □ Bouton "Un problème ?" plus en évidence dans Plans	iPhone 5 iPhone 4S iPhone 4 iPhone 3GS iPad mini iPad 4G iPad 3G iPad 2 iPod Touch 4G iPod Touch 5G

			Nouveau design du lecteur de musique sur l'écran verrouillé, le même design que l'app Musique Amélioration de l'application Plans	
7.0.x	iOS 7	10 juin 2013	Visual overhaul Control Center AirDrop Refreshed core apps iTunes Radios FaceTime Audio	iPhone 5s iPhone 5c iPhone 5 iPhone 4S iPhone 4 iPad mini iPad 4G iPad 3G iPad 2 iPod Touch 5G

Tableau 1.6 Description des dernières versions d'iOS [8].

IV.3. Windows Phone

IV.3.1. Définition

Le Windows Phone est un système d'exploitation spécialement développé pour un usage mobile. Développé par Microsoft, le Windows Phone (anciennement Windows Mobile) offre donc une interface principalement conçue pour les Smartphones qui leur donne en outre accès à la plate-forme d'applications de Microsoft, MarketPlace.

IV.3.2. Historique

Des premiers travaux sur une mise à jour majeure de Windows Mobile ont pu avoir été commencés au début 2004 sous le nom de code « Photon », mais le développement avance lentement et le projet est finalement annulé.

En 2008 Microsoft réorganise le groupe Windows Mobile et commence à travailler sur un nouveau système d'exploitation mobile. Le produit devait être publié en 2009 comme Windows Phone, mais plusieurs retards ont incité Microsoft à développer Windows Mobile 6.5 comme version intermédiaire.

Finalement Windows Phone a été développé rapidement sans chercher à créer une compatibilité ascendante, ainsi Terry Myerson, ingénieur Windows Phone, a déclaré : « nous avons choisi le changement vers les écrans tactiles capacitifs, loin du stylet, et du matériel requis pour avoir une meilleure expérience Windows Phone, nous devions casser la compatibilité des applications avec Windows Mobile 6.5 ».

IV.3.3. Versions

Microsoft a l'intention de diffuser des mises à jour mineures qui proposent de nouvelles fonctionnalités plusieurs fois par an avec une mise à jour plus importante chaque année49. Malheureusement, cet effet d'annonce a été classé sans suite puisque dès le départ en 2010, le développeur savait qu'il ne pourrait pas faire tourner Windows Phone 8 sur les mobiles disposant de Windows Phone.

Le tableau suivant présente les différentes versions et mises à jour de Windows Phone :

Version	Nom de version	Date de sortie	Caractéristiques
8.0.10211.204	Portico (General Distribution Release 1)	Décembre 2012	Quelques améliorations dans l'application Messages Une connectivité Bluetooth plus efficace, ainsi qu'un paramètre permettant de garder la connexion Wi-Fi active même lorsque l'écran du téléphone est verrouillé. La mise à jour a aussi apporté son lot de corrections de bugs
8.0.10327.77	General Distribution Release 2	Juillet 2013	L'ajout de la Radio FM La possibilité d'utiliser Lens comme application photographique par défaut, la supervision de la consommation des données cellulaires Data Sense Des corrections de bugs
	General Distribution Release 3	Août 2013	Le support des écrans Full HD Une gestion du multitâche améliorée Plus de compatibilité sur le Bluetooth et un partage de connexion plus rapide avec Windows 8.1 Apporte un mode Véhicule Des options et applications d'accessibilité pour les malvoyants Apporte le verrouillage de l'orientation de l'écran La possibilité de fermer une application dans le menu multitâche ou de supprimer certains fichiers temporaires

Tableau 1.7 : Description des dernières versions de Windows Phone

IV.4. Synthèse

Malgré l'évolution et la ressemblance des systèmes d'exploitation qui sont cités précédemment, il y a beaucoup de distinctions vue de côté développeur et dans le tableau suivant nous allons montrer les différences entre les trois premiers systèmes d'exploitation mobiles :

Informations générale			
Propriétaire	Google Inc	Apple Inc	Microsoft
Version courante	4.4 (KitKat)	7.0.4	8
Date de sortie	31 Octobre 2013	14 Novembre 2013	14 Octobre 2013
Part de marché	81%	12.9%	3.7%
Licence	Open Source	Source fermé	Source fermé
Détails technique			
OS Famille	Linux	Darwin	Windows CE 7 / Windows NT
Architecture CPU supportée	ARM, MIPS, x86, i.MX	ARM	ARM
Langage de programmation	Java	Objective C	C# .Net, VB .Net
des packages	APK	iTunes	Zune Software
Multiutilisateurs	v4.2+	Non	Non
GPU accélérations GUI	v3+	Oui	Oui
Platform de SDK	Windows, Linux et Mac OS	Mac OS	Windows
Multitâches	Oui	Oui	Oui
Serveur Proxy	v3+	Oui	Oui
IDE de développement	Eclipse NetBeans Intellijit	XCode	Visual Studio
Market			
Place de marché	Google Play	App Store	Windows Phone Marketplace
Nombre d'application	800000+	1000000+	200000+

Tableau 1.8 Comparatif entre les systèmes d'exploitation mobiles Android, iOS, Windows phone

IV.5. Le marché des systèmes d'exploitation mobile

La bataille entre les constructeurs de Smartphones passe aussi par la présence de leurs OS respectifs sur le marché. Après des chiffres par constructeur, en cette fin d'année, nombreux sont les cabinets d'étude à livrer leurs derniers résultats. C'est au tour d'International Data Corporation (IDC), spécialiste des études de marché dans le domaine IT, de livrer ses chiffres concernant les parts de marché par système d'exploitation mobile pour le troisième trimestre.

Top Four Operating Systems, Shipments, and Market Share, Q3 2013 (Units in Millions)

Operating System	3Q13 Shipment Volumes	3Q13 Market Share	3Q12 Shipment Volumes	3Q12 Market Share	Year-Over-Year Change
Android	211.6	81.0%	139.9	74.9%	51.3%
iOS	33.8	12.9%	26.9	14.4%	25.6%
Windows Phone	9.5	3.6%	3.7	2.0%	156.0%
BlackBerry	4.5	1.7%	7.7	4.1%	-41.6%
Others	1.7	0.6%	8.4	4.5%	-80.1%
Total	**261.1**	**100.0%**	**186.7**	**100.0%**	**39.9%**

Tableau 1.9 Top 4 des systèmes d'exploitation selon la part du marché [7].

Le système d'exploitation mobile de Google serait ainsi présent sur 81,3% des Smartphones de la planète. Android règne donc sans surprise, bien loin devant les quelques 13,4% d'iOS. Windows Phone, lui, s'adjuge 4,1%, et BlackBerry 1%.

Autres chiffres intéressants, la progression par rapport à l'année précédente. Android gagne plus de 6 points là où Apple en perd plus de 2. Microsoft double ses parts de marché (seulement 2,1% en 2012) et BlackBerry dévisse encore un peu plus.

V. Le cross-platform

Une application mobile doit être compatible avec les différents OS disponibles sur le marché. Sur terminaux mobiles, à chaque OS correspond un langage, des règles de compatibilité, un SDK (kit de développeur) et un processus de distribution spécifique… Les solutions cross-platform open source (aussi appelées multi-plateformes) telles que Titanium ou PhoneGap, visent à répondre à la problématique de l'absence de standards de développement dans le monde du mobile [2].

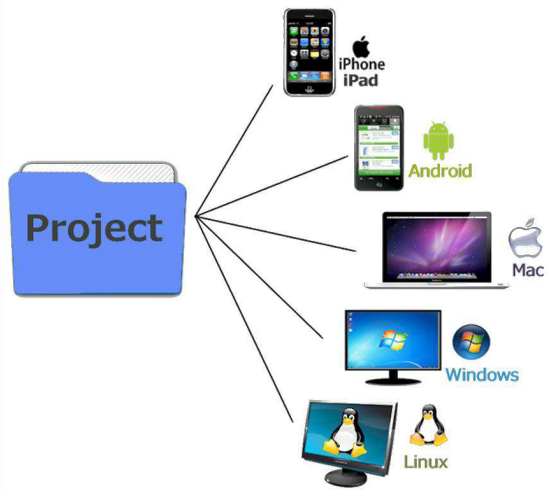


Figure 1.4 Un seul projet pour plusieurs plateformes [9]

Solution cross-plateforme

- Un seul code
- « n » plateformes cibles
- Simple à maintenir et faire évoluer

En pratique :

- Des technos web : faciles à prendre en main
- Un tronc commun (80% du code)
- Des adaptations à faire pour chaque plateforme cible : gestion du bouton retour pour Android et Windows Phone, Push de notifications…

Estimation de charge de développement:

- Si le développement d'une appli native nécessite « Y », le développement de la même appli en natif pour « n » OS coutera approximativement « n x Y ».
- Le même développement en cross-platform coutera entre « (n x Y) / n » et « (n x Y) / (n / 2) » selon les spécificités des fonctionnalités utilisées.

- Autrement dit, le cross-platform est financièrement intéressant au-delà du deuxième OS, mais dès le départ il assure une bonne pérennité et maintenabilité du code [2].

Conclusion

Pour développer une application mobile de nombreux facteurs sont à considérer.
Il y a d'abord le choix de type d'application : Web mobile ou application native ou bien une solution intermédiaire entre les deux.

Puis, il faut considérer la plateforme (le système d'exploitation mobile) dont l'application mobile va fonctionner, et songer au cross-platform : une solution qui permet de réaliser des applications sous plusieurs OS (système d'exploitation) avec un unique code.

Il faut aussi tenir compte des différents appareils mobiles (Smartphones, Tablettes..) dans lesquels l'application va être installée.

Et cela sans oublier les facteurs principaux pour n'importe quelle gestion de projet : Objectifs de l'application, le public ciblé, le modèle de cycle de vie, les exigences techniques, le cout de l'application.

Chapitre II

Android

Introduction

Android est une plate-forme ouverte, un ensemble de librairies de composants logiciels pour les systèmes embarqués et mobiles.

C’est la plate-forme mobile la plus utilisée au monde, plus d’un milliard de téléphones et de tablettes à travers le monde sont équipés d’Android, une plateforme à la fois personnalisable et simple d’utilisation conçue par Google, elle fonctionne parfaitement avec toutes les applications
Google.

Dans ce chapitre nous allons revenir sur l’architecture d’Android, ses caractéristiques dans le but de mieux comprendre ce qui fait aujourd’hui de ce système d’exploitation un système performant et très utilisé dans le monde.

I. Définition d’Android [10] [11]

Android est un système d'exploitation mobile pour Smartphones, tablettes tactiles, PDA (Personal Digital Assistant), Smartwatches et terminaux mobiles. C'est un système Open Source utilisant le noyau Linux. Il est basé essentiellement sur la simplicité d'utilisation et surtout sur une capacité de customisation importante présentant un argument commercial de poids.

II. Caractéristiques

Android est une pile applicative pour les appareils mobiles qui comprend un système d'exploitation, middleware et des applications clés.
Le SDK Android fournit les outils et les API nécessaires pour commencer à développer des applications sur la plate-forme Android en utilisant le langage de programmation Java.

III. Architecture d’Android [12]

Le schéma suivant illustre les principaux composants du système d'exploitation Android.

Chaque section est décrite plus en détail ci-dessous.

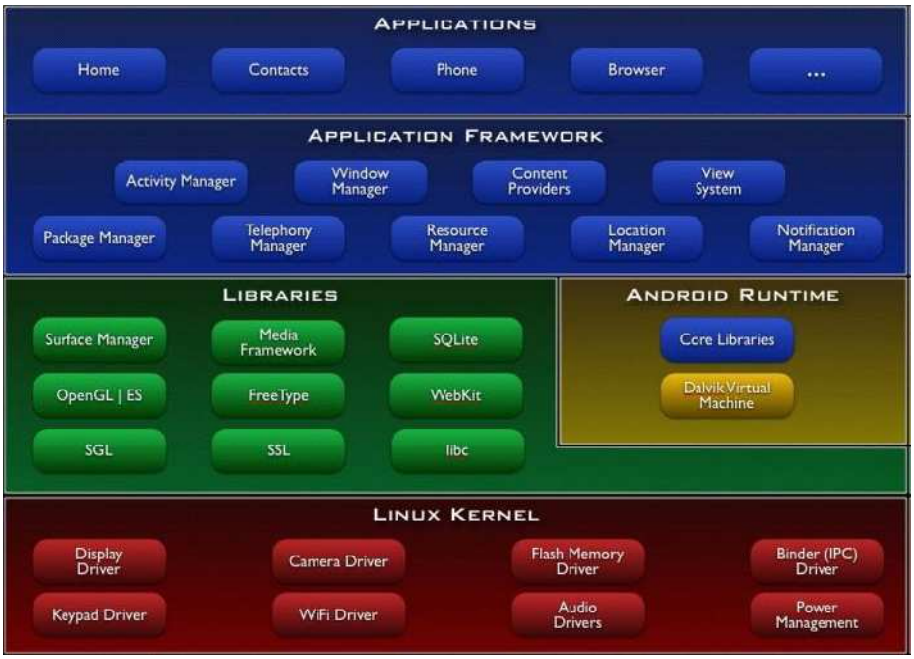


Figure 2.1. Architecture d'Android

III.1. Linux Kernel (niveau 1)

Android repose sur un noyau Linux 2.6 qui gère les services du système, comme la sécurité, la gestion de la mémoire et des processus, la pile réseau et les pilotes. Il agit également comme une couche d'abstraction entre le matériel et la pile logicielle.

Figure 2.2 : Le noyau Linux

III.2. Libraries et Android Runtime (niveau2)

a. Libraries

En interne, Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants de la plateforme Android. Ces bibliothèques sont en réalité accessibles au développeur par l'intermédiaire du framework Android.

En effet, le framework Android effectue, de façon interne, des appels à des fonctions C/C++ beaucoup plus rapides à exécuter que des méthodes Java standard.

La technologie Java Native Interface (JNI) permet d'effectuer des échanges entre le code Java et le code C et C++. La liste ci-dessous énumère quelques-unes des bibliothèques disponibles dans Android :

- **Bibliothèque système C**. Implémentation (dérivée de BSD) de la bibliothèque standard C (libc), optimisée pour les systèmes Linux embarqués.
- **Bibliothèques multimédias**. Basées sur StageFright, elles permettent le support de nombreux formats audio et vidéo, tels que MPEG4, H.264, MP3, AAC, AMR, JPG et PNG
- **SurfaceFlinger**. Permet l'accès au sous-système d'affichage.
- **LibWebCore**. Moteur de rendu de pages Internet basé sur Webkit. Cette bibliothèque est donc principalement utilisée dans le navigateur et dans les vues web embarquées
- (WebView).
- **Skia**. Moteur graphique 2D.
- **Bibliothèques 3D**. Implémentation basée sur OpenGL ES 1.0 API et plus récemment
- OpenGL ES 2.0.
- **FreeType**. Rendu des polices de caractères.
- **SQLite**. Base de données légère et puissante.

Figure 2.3 : Librairies

b. Android Runtime

Android inclut un ensemble de **bibliothèques** qui fournit la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java.

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de **machine virtuelle Dalvik**. Dalvik VM est une implémentation de machine virtuelle ayant été conçue pour optimiser l'exécution multiple de machines virtuelles. Elle exécute du bytecode qui lui est dédié : le bytecode .dex (format qui est optimisé pour une empreinte mémoire minimale).

Cette particularité d'Android en fait un système unique, loin des systèmes Linux traditionnels que beaucoup avaient pu rencontrer auparavant.

Figure 2.4 : Android Runtime

III.3. Application Framework (niveau 3)

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes.

Les développeurs sont libres de profiter du matériel périphérique, les informations de localisation d'accès, exécuter les services d'arrière-plan, définir des alarmes, ajouter des notifications de la barre d'état et beaucoup plus.

Figure 2.5 : Application Framework

Core Plateform Services

Android introduit la notion de services. Un service est une application qui n'a aucune interaction avec l'utilisateur et qui tourne en arrière plan pendant un temps indéfini.

Les services coeurs de la plateforme (Core Plateform Services) fournissent des services essentiels au fonctionnement de la plateforme :

Activity Manager : gère le cycle de vie des applications et maintient une "pile de navigation" (navigation backstack) permettant d'aller d'une application à une autre et de revenir à la précédente quand la dernière application ouverte est fermée.

Package Manager : utilisé par l'Activity Manager pour charger les informations provenant des fichiers .apk (android package file)

Window Manager : juste au dessus du Surface Flinger (lien), il gère les fenêtres des applications quelle fenêtre doit être affichée devant une autre à l'écran.

Ressource Manage : gère tous ce qui n'est pas du code, toutes les ressources images, fichier audio, etc.

Content Provider : gère le partage de données entre applications, comme par exemple la base de données de contact, qui peut être consultée par d'autres applications que l'application Contact. Les Données peuvent partager à travers une base de données
(SQLite), des fichiers, le réseau, etc.

View System : fournit tous les composants graphiques : listes, grille, text box, buttons et même un navigateur web embarqué.

Hardware Services

Les services matériels (Hardware Services) fournissent un accès vers les API matérielles de bas niveau :

- **Telephony Service** : permet d'accéder aux interfaces "téléphoniques" (gsm, 3G, etc.)
- **Location Service** : permet d'accéder au GPS.
- **Bluetooth Service** : permet d'accéder à l'interface bluetooth.
- **WiFi Service** : permet d'accéder à l'interface Wifi.
- **USB Service** : permet d'accéder aux interfaces USB.
- **Sensor Service** : permet d'accéder aux détecteurs (détecteurs de luminosité, etc.)

III.4. Applications (niveau4)

Android est fourni avec un ensemble de programmes de base (également nommés applications natives) permettant d'accéder à des fonctionnalités comme les courriels, les SMS, le téléphone, le calendrier, les photos, les cartes géographiques, le Web, pour ne citer que quelques exemples.

Ces applications sont développées à l'aide du langage de programmation Java. Pour l'utilisateur final, c'est la seule couche accessible et visible.

.

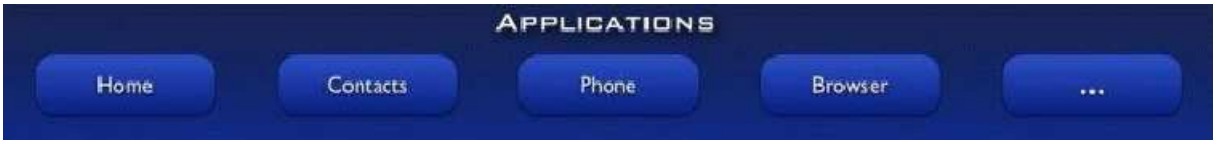


Figure 2.6 : Applications

IV. SDK Android

VI.1. Développement [13]

Le kit de développement (SDK) d'Android est un ensemble complet d'outils de développement utilisés pour développer des applications pour la plateforme Android**.**

Il inclut un débogueur, des bibliothèques logicielles, un émulateur, de la documentation, des exemples de code et des tutoriaux.

Les plateformes de développement prises en charge par ce kit sont les distributions sous Noyau Linux, Mac OS X 10.5.8 ou plus, Windows XP ou version ultérieure.

L'IDE officiellement supporté est Eclipse combiné au plugin d'outils de développement d'Android (ADT).

Les développeurs peuvent utiliser n'importe quel éditeur de texte pour modifier les fichiers Java et XML, puis utiliser les outils en ligne de commande (Java Development Kit et Apache Ant sont obligatoires) pour créer, construire et déboguer des applications Android ainsi que contrôler des périphériques Android .

IV.2. Quelques outils fournit par le SDK

1. l'émulateur [13]

Le SDK comprend un émulateur appelé aussi AVD (Android Virtual Device*)*, qui permet de simuler les différentes versions d'Android permettant ainsi aux développeurs de tester leurs applications ou de tester les fonctionnalités d'Android.

Figure 2.7 : Emulateur

2. Android Debug Bridge (ADB) [13]

L'Android Debug Bridge (ADB) est un outil inclus dans le package Android SDK. Il se compose d'un programme client et d'un programme serveur qui communiquent entre eux. Les fonctions principales de l'ADB sont :

- Copie de fichier ;
- Accès à la console android ;
- Sauvegarde de la mémoire rom ;
- Installation de logiciel.

3. DDMS (Dalvik Debug Monitor Service) [14]

Le Dalvik Debug Monitor Service (DDMS) est un outil de débogage plus avancé du SDK qui a également été intégré dans Android Studio.

Le DDMS est capable de surveiller à la fois un dispositif réel et l'émulateur. C'est un outil puissant qui permet de voir ce qui se passe en profondeur, en effet grâce à ce dernier il est possible d'interroger les processus actifs, examiner la pile , surveiller et mettre en pause les threads actifs et explorer le système de fichier de n'importe quel matériel Android connecté.

La perspective DDMS sous Eclipse fournit également un accès simplifié aux captures d'écran de l'émulateur et aux journaux générés par LogCat.

Le DDMS est complètement intégré à Eclipse via le plugin ADT et peut être lancé depuis la ligne de commande : il se connectera automatiquement à tout appareil ou émulateur actif.

V. Concepts

V.1. Le bureau virtuel [15]

Le bureau virtuel est l'écran d'accueil du Smartphone, il est composé de 5 parties (ou plus).Chacune est personnalisable, il est possible d'y mettre des raccourcis (vers des applications, des fichiers, des dossiers, des contacts, etc..) ou des widgets (calendrier, horloge, notes, etc..).

L'image de fond s'étend aussi sur tout le bureau et bouge dés qu'on passe d'une partie à une autre ce qui donne l'impression que le contenu fait partie du décor.

Cependant et dans le but de faire la différence les constructeurs ajoutent leurs touches personnelles par exemple : le constructeur HTC possédant l'interface Sence qui est dotée de 7 bureaux.

Le problème avec cette personnalisation est que lorsqu'une version nouvelle d'Android sort, chaque constructeur doit procéder aux adaptations dans sa propre version. Donc soit la mise à jour n'est jamais effectuée, soit elle sort plusieurs semaines après la version de base.

V.2. Widgets [16]

Le mot widget recouvre deux notions distinctes, chacune en relation avec les **interfaces graphiques**. Il peut alors être considéré comme étant la contraction des termes window (fenêtre) et gadget. Il peut désigner :

- un **composant d'interface graphique**, un élément visuel d'une interface graphique (bouton, ascenseur, liste déroulante, etc.) ;
- un **widget interactif**, qui est une petite application qui peut être affichée sur l'écran d'accueil du téléphone ou de la tablette.

Il permet d'afficher des informations mises à jour à intervalle régulier (météo, actualité, dictionnaire, carte routière, post-it.) et offre des interactions simples avec l'utilisateur.

Le **widget** a la plupart du temps une dimension réduite afin de ne pas prendre trop de place sur l'écran, ce qui permet de le laisser affiché en permanence.

Figure 2.8 : Widget

Figure 2.9 : Widget

V.3. Android market [17]

Il a été remplacé en mars 2012 (pour la France) par Google Play.

Android Market est la place de marché où il est possible de télécharger les applications pour les Smartphones opérant sous le système d'exploitation Android racheté et développé par Google.

Les applications obtenues sur Android Market peuvent être gratuites ou payantes. Pour les applications payantes, Google prélève une commission de 30 % sur le prix de l'application.

L'éditeur obtient donc 70 % du prix HT.

Android Market peut être accessible par un accès web classique ou le plus souvent directement à partir du Smartphone.

VI. Composant d'android :[18]

Une application Android est composée de quatre composants suivants:

- L'activité (Activity)
- Service (Service)
- Fournisseur de contenu (Content provider)
- Receveurs de diffusion (Broadcast receivers)

VI.1. L'activité (Activity)

- C'est la brique de base de l'interface utilisateur
- C'est l'équivalent Android de la fenêtre ou d'une boite de dialogue d'une application classique
- En pratique, une activité correspond à un écran de l'interface de l'application. Par exemple, une application email peut avoir une activité qui affiche la liste des nouveaux emails, une autre activité pour écrire un nouvel email et une autre activité pour lire un email particulier, …etc
- Chaque activité est indépendante et peut constituer un point d'entrée de l'application. Une application peut faire appel à une activité particulière d'une autre application. Par exemple, l'appareil photo peut lancer l'activité de création de nouvel email pour envoyer une photo

VI.2. Service

Un service est un composant qui s'exécute en arrière plan, sans interface graphique

- Un service à une durée de vie plus longue que l'activité

Un service s'exécute indépendamment de toute application

- Par exemple, un service peut jouer de la musique en arrière plan pendant que l'utilisateur utilise un autre programme ou télécharge des informations du réseau

VI.3. Fournisseurs de contenu (Content providers)

- Les fournisseurs de contenu offrent un moyen de partager des données entre les applications
- Les données d'une application peuvent être stockées sur le système de fichier local, sur une base de données SQLite ou le réseau (FTP, Web, ..etc.). Les autres applications peuvent accéder à ces données (ou les modifier si l'application le permet) en utilisant un content provider
- Par exemple, le système Android fournit un content provider fournissant la liste des contacts téléphoniques qu'une autre application peut utiliser

VI.4. Receveurs de diffusion (Broadcast receivers) :

Un receveur de diffusion est un composant qui répond à un message diffusé par le système Android ou une autre application

- Le message diffusé est appelé Intent (ou intention) et signale un événement particulier
- L'écran est passé en mode veille
- Le niveau de la batterie est faible
- l'appareil photo à pris une photo,

- …etc

Les applications peuvent émettre leurs propres intents qui seront captés par les Boradcast receivers d'autre applications

VII. Les services offerts par Android [11]

Les services offerts par Android grâce à une quinzaine d'applications incorporées facilitent notamment l'exploitation des réseaux de télécommunications GSM, Bluetooth, Wi-Fi et UMTS la manipulation de médias, notamment de la vidéo H.264, de l'audio 123MP3 et des images JPEG ainsi que d'autres formats, l'exploitation des senseurs tels que les capteurs de mouvements, la caméra, la boussole et le récepteur GPS, l'utilisation de l'écran tactile, le stockage en 39 bases de données, le rendu d'images en 2D ou 3D en utilisant le processeur graphique, l'affichage de page web, l'exécution 32 multitâches des applications.

Conclusion

Dans ce chapitre nous avons vu l'architecture d'Android, les différents niveaux de la plate-forme Android, quelques outils fournis par le SDK et les composants d'Android.

Nous allons par la suite présenter le chapitre concernant l'analyse et la conception de notre application.

Chapitre III

Analyse et Conception

Introduction :

La conception de toute solution informatique doit être traitée avec rigueur et précision, car elle constitue la base du système à développer. Avant de s'engager dans la conception, il est impératif de passer par la phase d'analyse qui permet d'identifier les différents acteurs qui interagissent avec le système ainsi que leurs besoins. Puis on passera à la conception qui utilisera les résultats de la phase d'analyse, on donnera aussi la description détaillée du système cible et les objectifs à atteindre. Pour ce faire, notre démarche va s'appuyer sur le langage UML, qui offre une bonne représentation des aspects statique et dynamique d'une application.

I. Le langage de modélisation UML :[18]

UML est un langage de modélisation fondé sur les concepts objet standard conçu pour l'écriture de plans d'élaboration de logiciels. L'objectif d'UML est de fournir une notation standard utilisable dans le développement de systèmes informatiques basés sur l'objet.

UML est adapté à la modélisation de systèmes, depuis les systèmes informatiques d'entreprise jusqu'aux applications distribuées basées sur le Web, c'est un langage très expressif qui couvre toutes les perspectives nécessaires au développement et au déploiement de tels systèmes. Pour apprendre à s'en servir efficacement, il faut d'abord s'appuyer sur une représentation conceptuelle de ce langage, ce qui nécessite l'assimilation de trois éléments fondamentaux qui sont :

Les briques de base d'UML, (Des éléments. Des relations. Des diagrammes) et les règles qui déterminent la manière de les assembler et quelques mécanismes généraux qui s'appliquent à ce langage.

UML s'articule autour de neuf diagrammes différents, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Par ailleurs, UML modélise le système suivant deux modes de représentation : l'un concerne la structure du système pris "au repos" modèle statiques (diagramme de cas d'utilisation, de classe…etc.) l'autre concerne sa dynamique de fonctionnement (diagramme de séquence, d'états transitions, d'activités…etc.). Les deux représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composantes fonctionnent entre elles.

I.1. Analyse :

Cette partie a pour objectif la spécification d'une manière claire de notre application. Pour ce faire, il est nécessaire de déterminer globalement ce que ce trouve dans le champ de

l'application en s'intéressant à la définition des besoins ainsi que les interactions entre les différents acteurs impliqués.

I.1.1 Identification des besoins :

Notre projet consiste à mettre en œuvre une application, qui a pour objectif d'assurer le service d un restaurant via un Smartphone. Dans le but d'une meilleure organisation et une bonne maîtrise du travail, tout processus de développement d'applications ou systèmes informatiques doit suivre une méthode ou une démarche bien définie. La figure ci-dessous montre la représentation graphique de la démarche de modélisation choisie pour concevoir notre application :

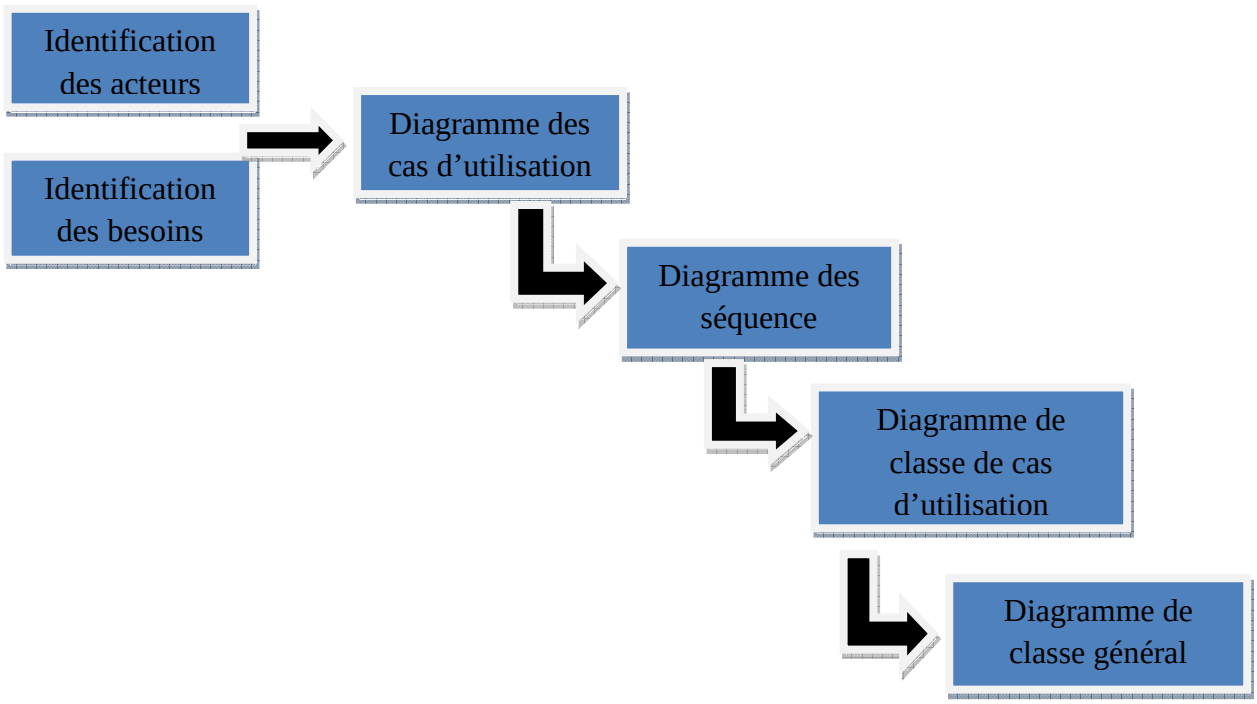


Figure 3.1 : La démarche de modélisation de l'application.

I.1.2 Identification des acteurs :

Après avoir étudié les besoins de notre système, nous avons procédé à l'identification de ces principaux acteurs :

Administrateur : le gérant (caissier) disposant d'un compte qui peut changer le menu, payer sur le client, c.à.d. il a comme tâches :

- S'authentifier par un identifiant et un mot de passe.
- Gérer le menu (ajouter, supprimer, modifier).
- Payer sur les clients.

Cuisinier : le cuisinier reçois la commande du client ils gèrent le menu et prépare la commande, c'est-à-dire, il a comme taches :

- Gérer le menu.
- Prépare la commande.

Client : visualiser le menu et passé une commande pour le cuisinier c'est-à-dire il a comme tache :

- Visualiser le menu.
- Cocher le plats et la boisson et le desserre choisi.
- Appuyé sur valider.

I.1.3 Diagramme de contexte :

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une vision globale des interactions entre le système et l'environnement extérieur, notre diagramme de contexte est donné par la figure suivante :

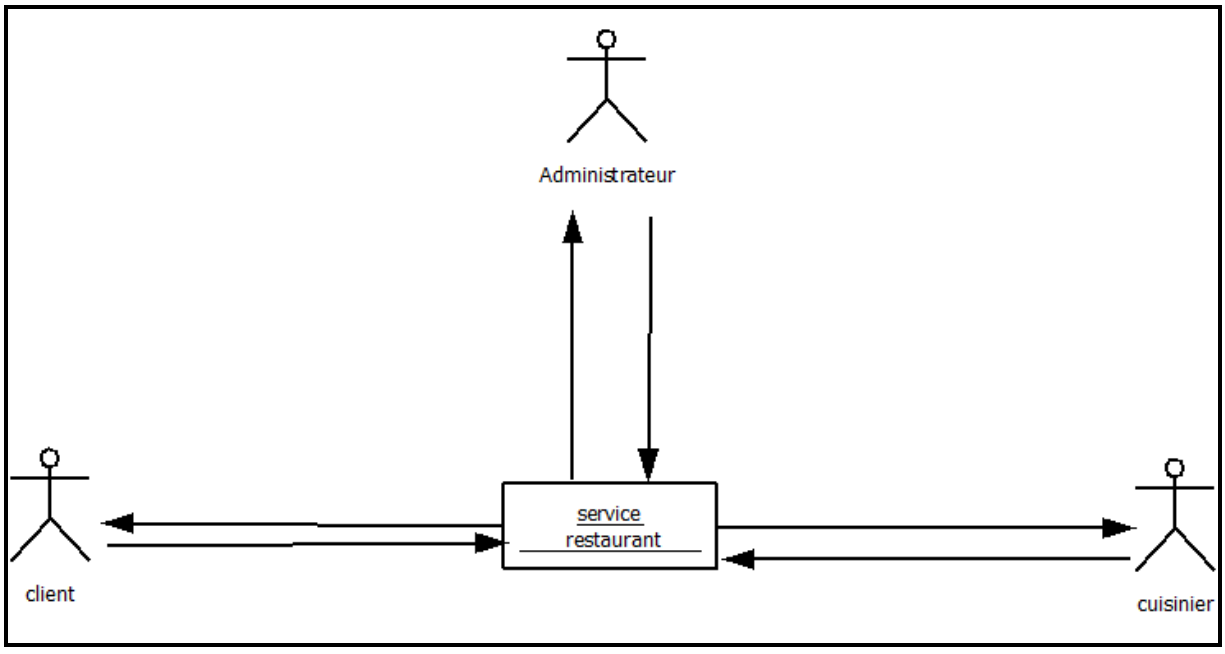


Figure 3.2 : Diagramme de contexte de l'application

I.1.4 Les cas d'utilisation :

Un cas d'utilisation représente un ensemble de séquence d'actions qui sont réalisées par les systèmes et qui produit un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le système devra faire, sans spécifier comment le faire. Dans notre cas nous distinguons les cas d'utilisation suivants :

I.1.4.1 Cas d'utilisation relatifs à l'administrateur :

La figure suivante montre le cas d'utilisation général de l'administrateur

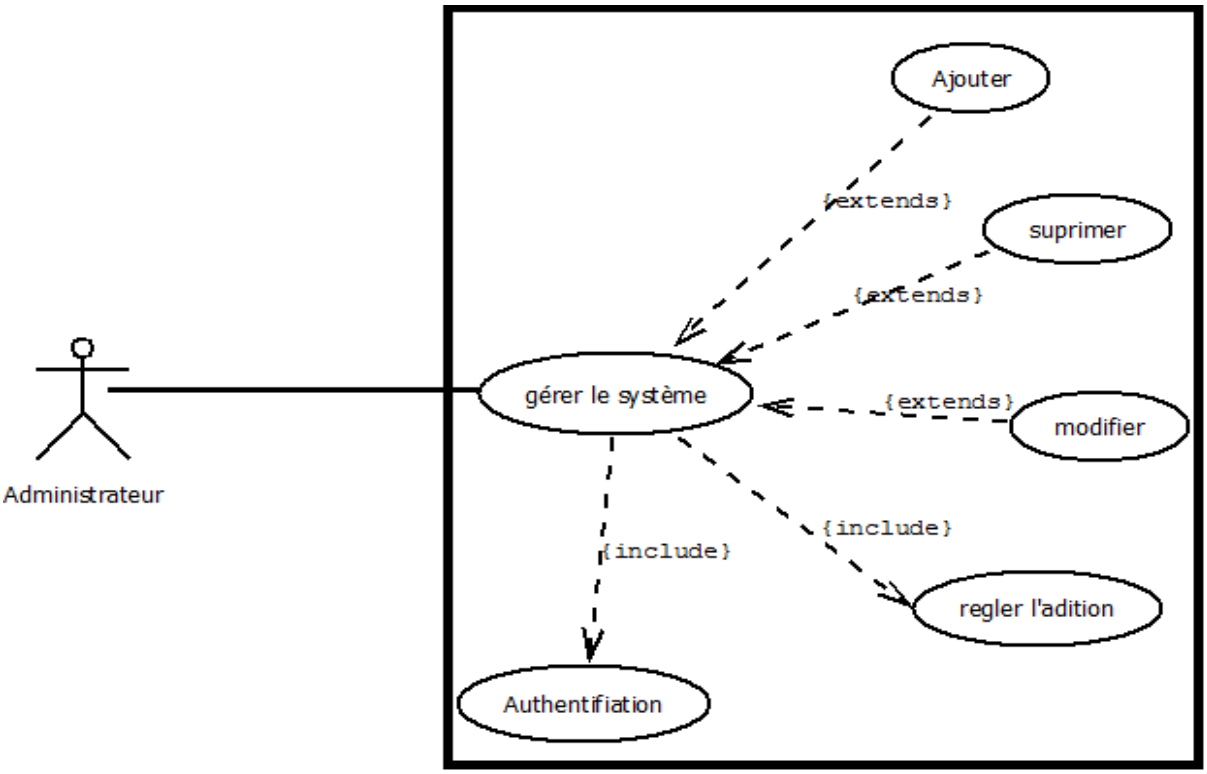


Figure 3.3 : Diagramme de cas d'utilisation de l'administrateur

I.1.4.2 Cas d'utilisation relatifs aux clients :

La figure suivante montre le cas d'utilisation du client.

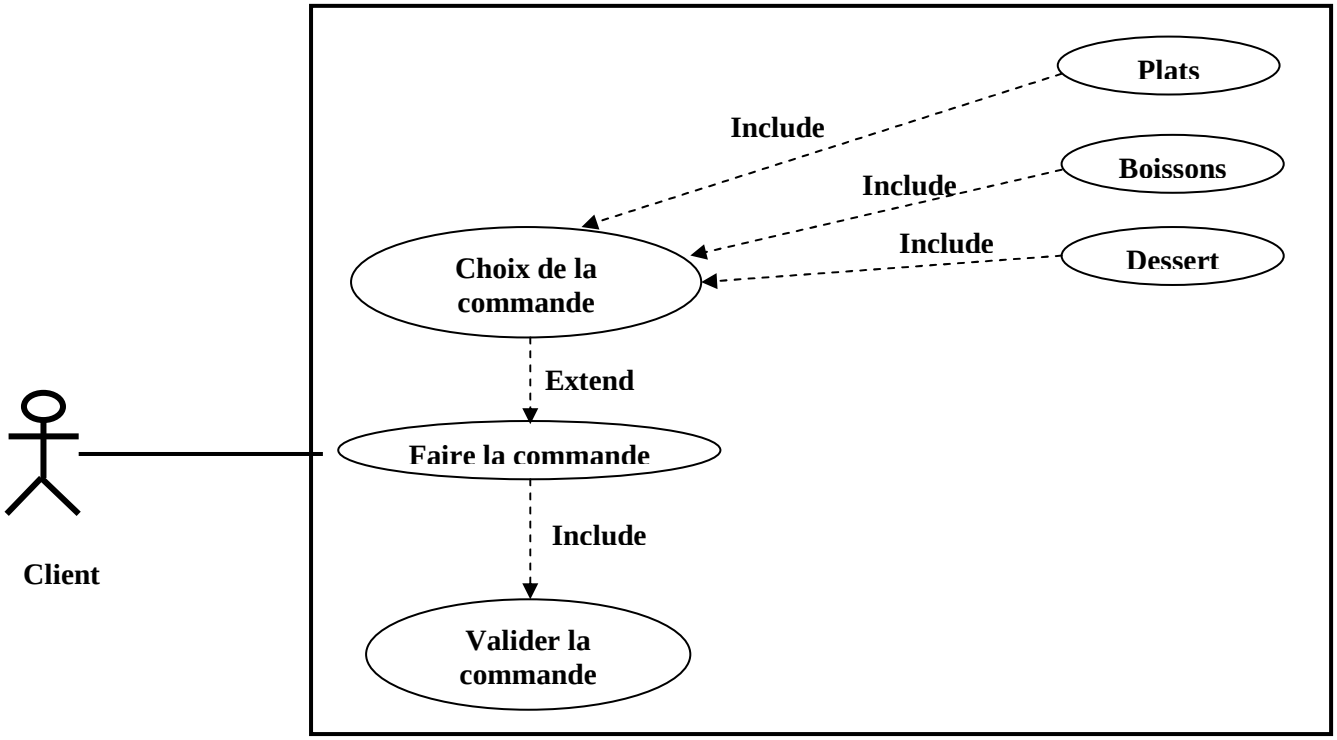


Figure 3.4 : Diagramme de cas d'utilisation de clients .

I.1.4.3 Cas d'utilisation relatifs aux cuisiniers :

La figure suivante montre le cas d'utilisation du cuisinier.

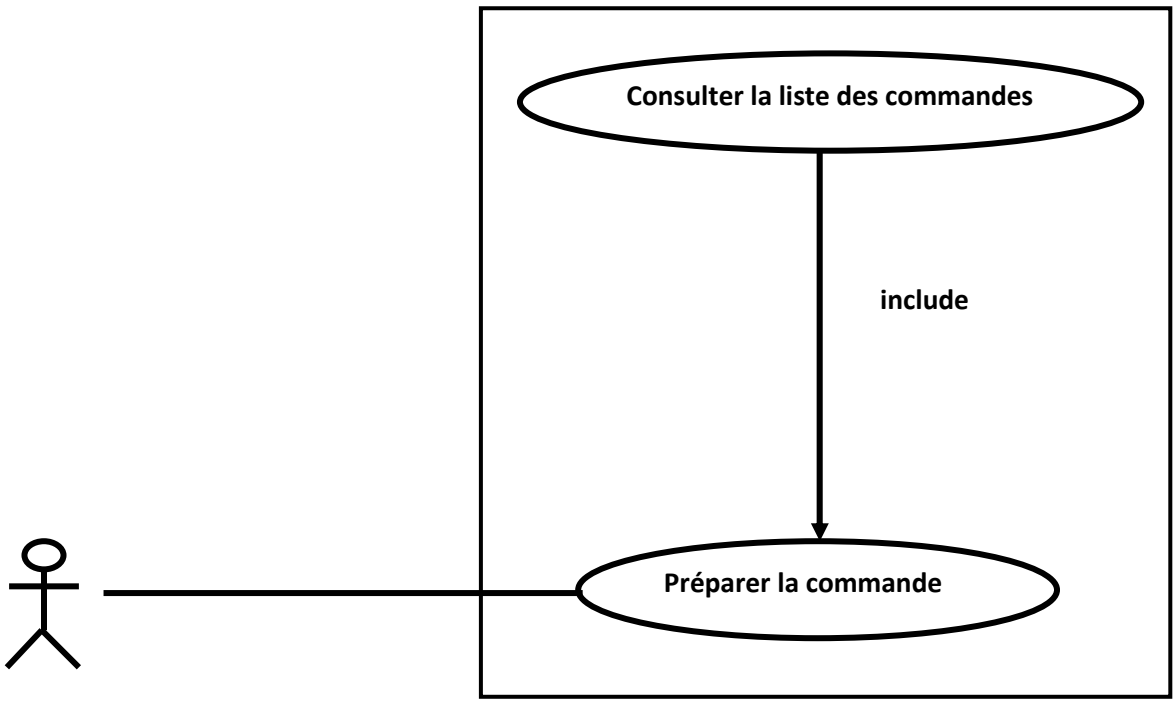


Figure 3.4 : Diagramme de cas d'utilisation de cuisinier .

I.1.5. Description des cas d'utilisation avec des scénarios :

Pour détailler le déroulement d'un cas d'utilisation, la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système. Dans ce qui suit nous décrivons donc quelques cas d'utilisation de notre système.

I.1.5.1Cas d'utilisation «Authentification Administrateur » :

Use Case : Authentification Acteur : Administrateur Description : 1. Atteindre la page «Authentification » 2. Saisie les informations d'identification (Pseudo, mot de passe) 3. Le système affiche un message d'erreur d'identification si les informations saisies sont incorrectes ou si elles ne correspondent pas à l'administrateur 4. Affichage de la page principal de l'espace administrateur

I.1.5.2 Cas d'utilisation « Ajouter un plat coté administrateur » :

Use Case : Ajouter un plat. Acteur : Administrateur Description 1. Cliquer sur le bouton «plat» 2. Remplir le formulaire d'ajout et valider 3. Le system affiche une erreur si les champs obligatoire n'est pas remplis. 4. le système affiche le plat ajouté dans la liste des plats.

I.1.5.3 Cas d'utilisation « passer une commande coté client » :

Use Case : passer une commande.
Acteur : client
Description :

1. Atteindre la page «Accueil ».
2. le client sélectionne le bouton « plat, boisson, dessert »
3. Le système affiche la page de choix des menus.
4. le client coche son choix (plats, boisson, dessert).
5. le client appui sur le bouton « valider ».
6. le système envoi la commande pour le cuisinier.
7. le système envoi l'adition pour l'administrateur.

I.1.5.4 Cas d'utilisation supprimer une boisson :

Use Case : supprimer une boisson.
Acteur : Administrateur.
Description

1. Cliquer sur le bouton «boisson »
2. le système affiche la liste des boissons.
3. cliquer sur le bouton supprimer de la boisons voulu.
4. le système affiche la liste de la boisson sans la boisson supprimé.

I.1.5.5 Cas d'utilisation modifier dessert :

Use Case : modifier un dessert.
Acteur : Administrateur.
Description

1. Cliquer sur le « dessert »
2. le système affiche la liste des desserts.
3. cliquer sur le bouton « mise a jour » de dessert voulu.
4. le système affiche le formulaire a modifier .
5. inséré les informations voulu et appui sur le bouton « valider ».

I.1.5.6 Cas d'utilisation recevoir la commande et la préparer

Use Case : recevoir la commande et la préparer. Acteur : cuisinier Description 1. accéder a sont application. 2. Recevoir la liste des commandes dans linter fasce principale. 3. prépare la commande.

I.2. Conception :

Le processus de conception de notre application repose sur l'organisation conceptuelle, logique et physique des données collectées durant la phase analyse. En effet, elle s'appuie essentiellement sur quelques diagrammes du langage de modélisation UML.

I.2.1.Les diagrammes de séquence :

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée, …, il met en évidence les objets manipulés ainsi que les opérations qui font passer d'un objet à l'autre. Dans notre cas on s'intéresse seulement à effectuer la représentation des diagrammes de séquence pour les cas d'utilisation déjà présentés auparavant.

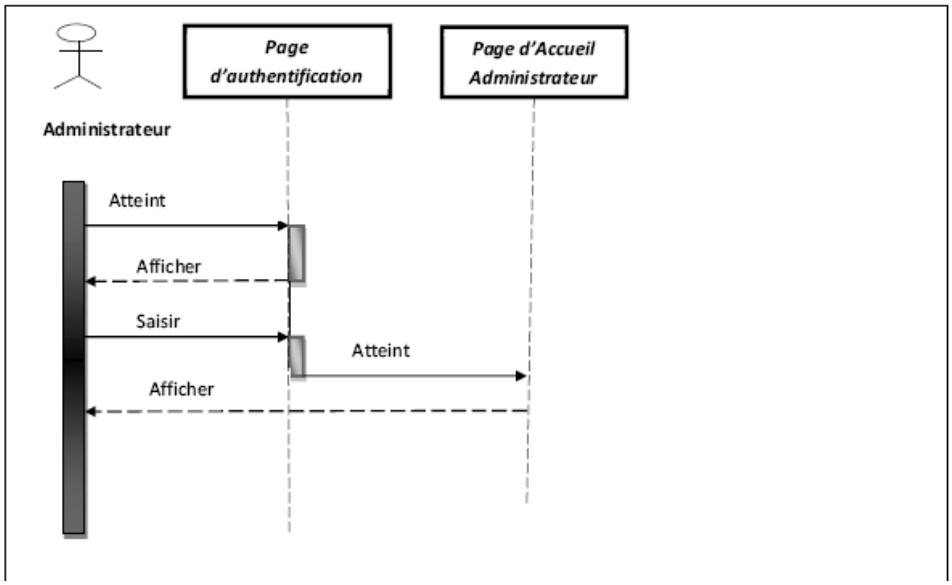


Figure 3.5 : Diagramme de séquence de cas utilisation : « Authentification utilisateur ».

I.2.2. Diagramme de séquence «Ajouter un plat par l'administrateur » :

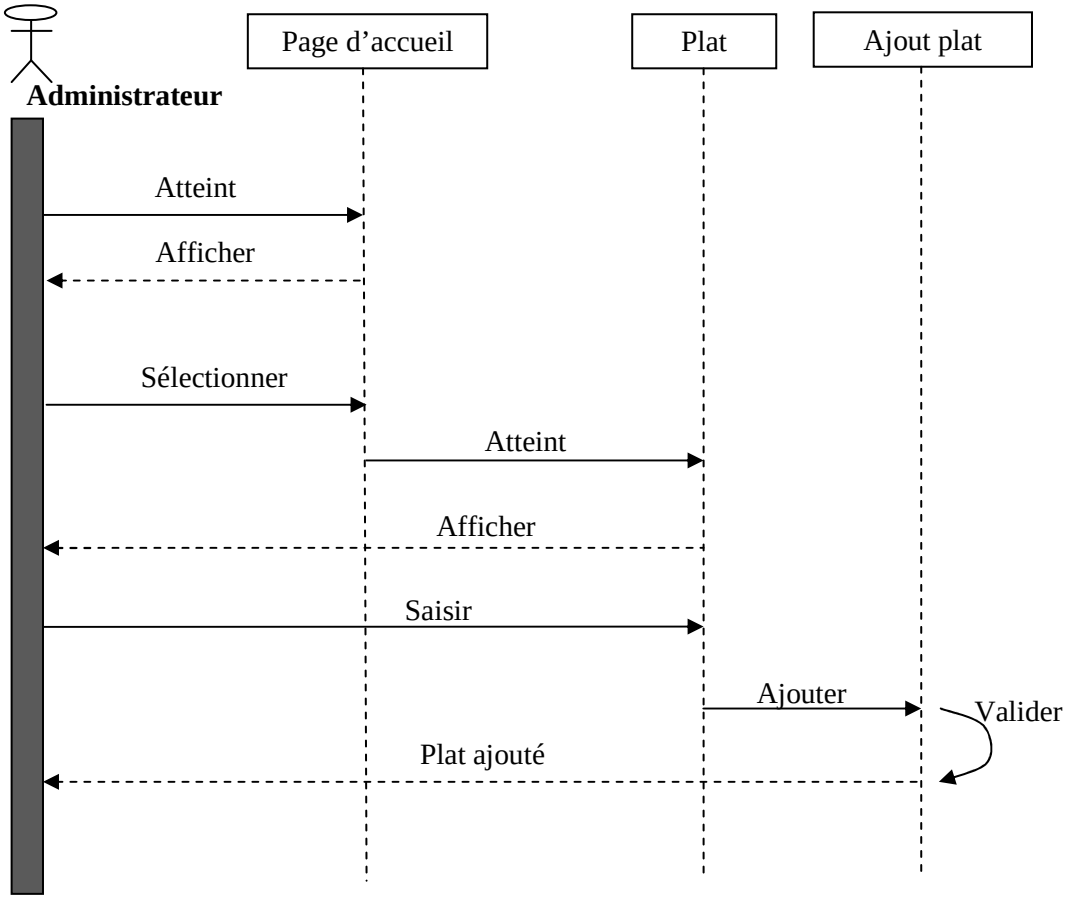


Figure 3.6: Diagramme de séquence du cas d'utilisation « Ajouter un plat »

I.2.3. Diagramme de séquence «Passer une commande par le client » :

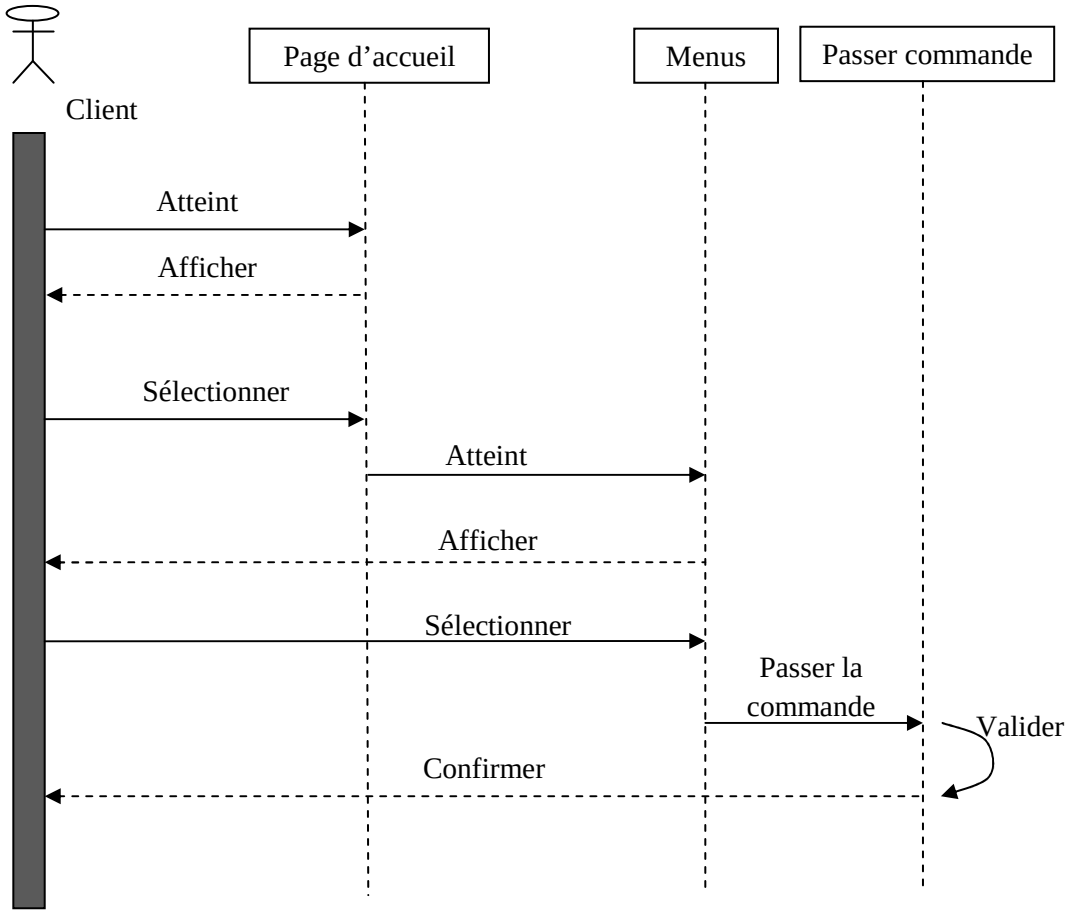


Figure 3.7 : Diagramme de séquence du cas d'utilisation « passer une commande »

I.2.4. Diagrammes de Classes (Class Diagram):

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il s'agit donc d'une vue purement statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Il permet aussi de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

I.2.5. Diagramme de classe du cas d'utilisation «Préparer la commande»

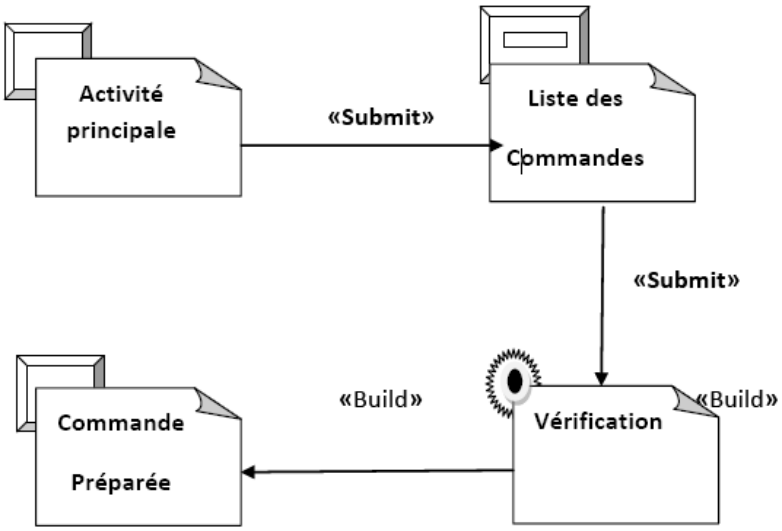


Figure 3.8 : **Diagramme de classe du cas d'utilisation «Préparer la commande»**

I.2.6. Diagramme de classe du cas d'utilisation «Ajouter une boisson »

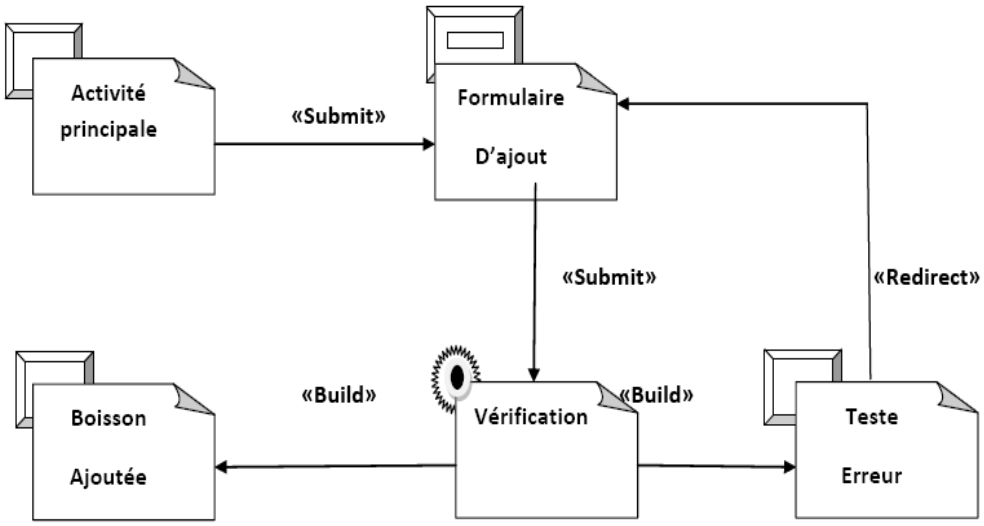


Figure 3.9 : **Diagramme de classe du cas d'utilisation «Ajouter une boisson »**

I.2.7. Diagramme de classe du cas d'utilisation «Régler l'addition»

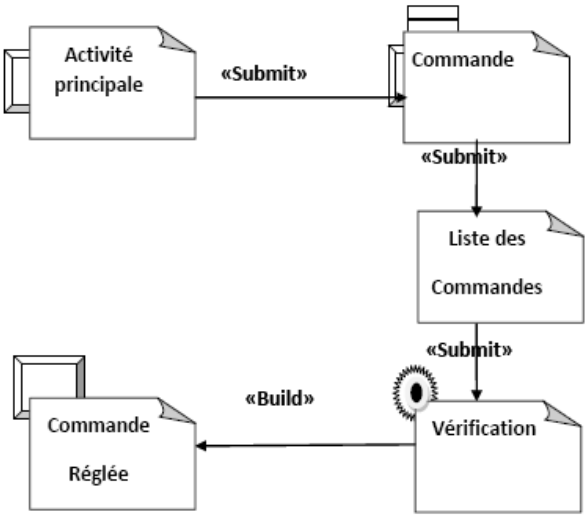


Figure 3.10 : **Diagramme de classe du cas d'utilisation «Régler l'addition»**

II. Conception de la base de données

II.1 Présentation des tables de la base de données

- **Table itemcmd**

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	id_cmd	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	id_plat	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	qnt	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

- **Table cmd**

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	date	bigint(20)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	table	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	etat	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

- **Table login**

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	Num	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	pseudo	text	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	passe	text	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

- **Table plat**

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	nom	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	description	text	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	prix	varchar(10)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
5	type	int(1)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
6	image	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

III. Diagramme de classe :

Le diagramme de classes représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens :

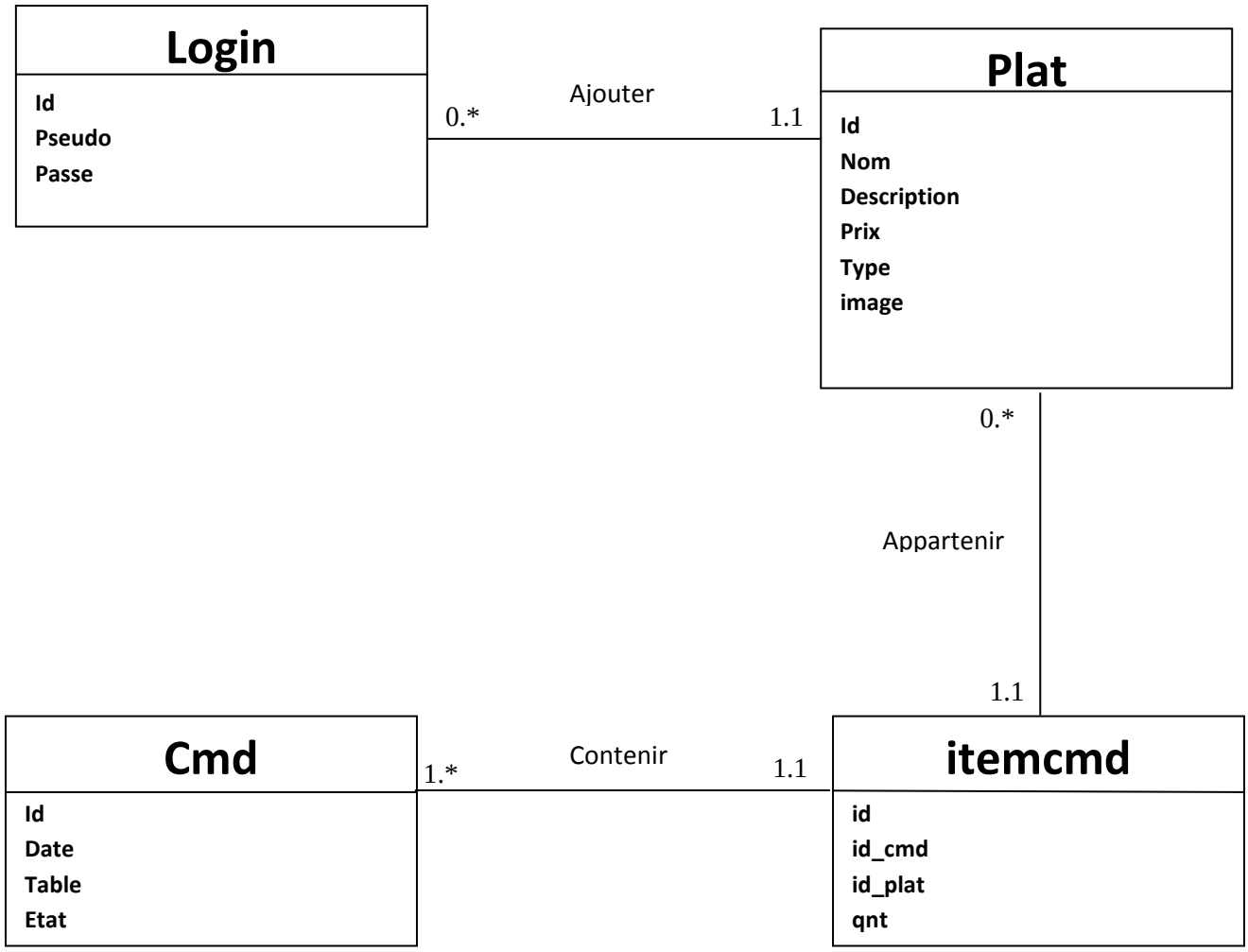


Figure 3.11 : Diagramme de classe générale

Table login

Id : identifiant de la table

Pseudo : le pseudo de l'administrateur

Passe : mot de passe de l'administrateur

Table plat

Id : identifiant de la table

Nom : nom du plat

Description : description du plat

Prix : prix du plat

Type : type du plat (boisson, plat, dessert)

Image : l'image du plat

Table cmd (commande)

Id : identifiant de la commande

Date : la date de la commande

Table : identifiant de la table

Etat :l'état de la commande (payer ou non)

Itemcmd (item commande)

Id=identifiant de la table

Id_cmd : identifiant de la commande

Id_plat : identifiant du plat

Qnt : quantité du plat commandé

Conclusion :

Dans ce chapitre nous avons mené une conception détaillée, selon une approche objet afin de garantir la fiabilité et l'efficacité de la phase de réalisation de l'application. Nous avons défini les acteurs constituant le système en exprimant leur besoins avec le diagramme de cas d'utilisation, puis nous l'avons détaillé en précisant comment les objets et les acteurs doivent collaborer ensemble selon une dimension temporelle par l'utilisation des diagrammes de séquence. Finalement nous avons décrit l'aspect statique avec les diagrammes de classes. Le chapitre suivant sera consacré à la réalisation de notre application.

Chapitre IV

Réalisation

Introduction :

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir compléter l'étude conceptuelle dans le chapitre précédent, nous allons aborder la partie implémentation dans ce qui suit. Nous commençons par présenter l'environnement matériel et logiciel, ensuite nous présenterons les différentes interfaces de l'application.

I. Environnement matériel et logiciel[19]

I.1. Environnement matériel

I.1.1. Architecture matérielle

L'architecture de notre application est à 3 niveaux (architecture 3-tiers), elle est partagée entre :

Le client Android : Conteneur d'application et demandeur de ressources.

Le serveur Web : Vue que les données serons communiquées entre deux environnements hétérogènes, le rôle principale du serveur web est de gérer la communication entre le client Android et le serveur de base de données, Le serveur de base de données fournis les données au serveur web.

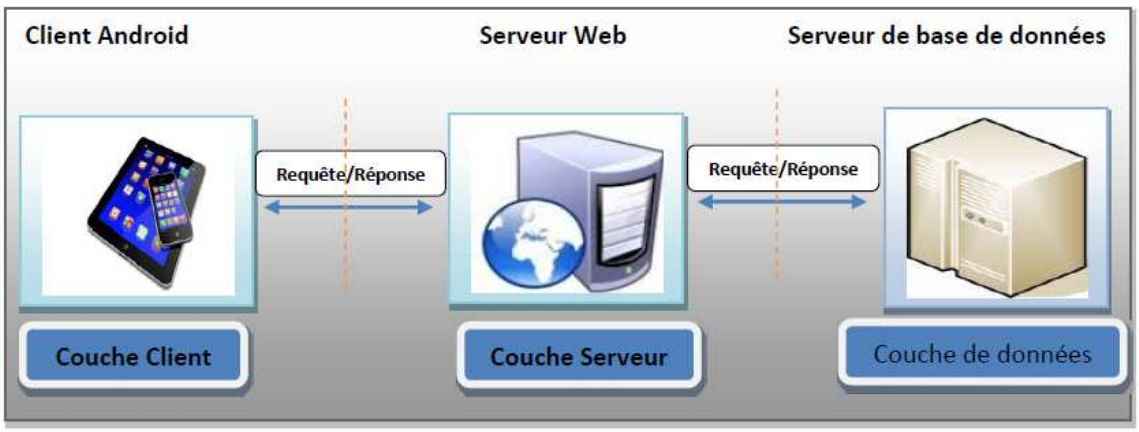


Figure 4.1:Architecture matériel du système

I.1.2. Matériels utilisés :

Pour la réalisation de l'application :

Un pc portable pour le développement ayant les caractéristiques suivantes :

- 8 Go de mémoire vive.
- Windows 7 - 64 bits.

Un Smartphone pour réaliser les tests.

Technologie

Ci-dessous un tableau représentant les différentes technologies utilisées dans notre application :

	Android Système d'exploitation open source pour Smartphones, PDA et terminaux mobiles.
	PHP Langage de scripts libre principalement utilisé pour produire des pages Web dynamiques.
	MySQL Système de gestion de base de données (SGBD).
	JSON (JavaScript Object Notation) Format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript.

Figure 4.2 : Les différentes technologies utilisées dans l'application.

La méthode la plus répandue de se connecter à une base de données MySQL à distance à partir d'un appareil Android, est de l'effectuer à l'intermédiaire d'un serveur web. Le SGBD MySQL est habituellement utilisé avec le langage PHP, donc la façon la plus simple et la plus évidente est d'écrire des scripts PHP pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP du système Android. Nous avons codé les données dans le format JSON.

L'architecture 3-tiers du point de vue technologique, le client est la plateforme Android, le serveur web est le PHP et le serveur de bases de données est le MySQL.

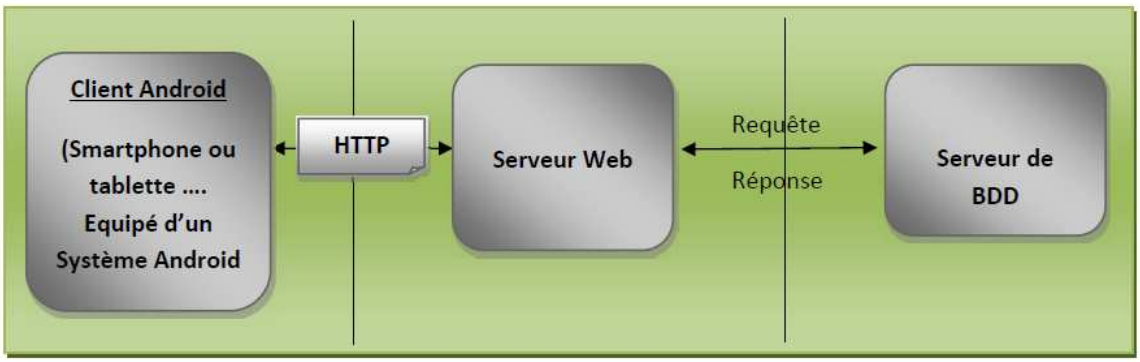


Figure 4.3 :Architecture3-tiers du point de vue technologique

I.2. Environnement logiciel

Pour pouvoir réaliser une application dans des bonnes conditions il est de bien entendu de choisir son environnement de travail selon les besoins. Nous avons opté pour la réalisation d'une application mobile sous le système Android ce qui nous impose de travailler sous Android Studio avec le langage JAVA et le SDK Android pour son développement, afin d'obtenir un fichier à extension .apk qui sera par la suite installé sur des terminaux mobiles de différents types fonctionnant avec le système Android 3.0 ou plus.

Le développement d'applications pour Android se fait entièrement en Java, ce dernier est un langage puissant orienté objet, utilisé très largement dans le monde du développement, en utilisant des IDE tels que : Eclipse, NetBeans et Android Studio.

I.2.1 JAVA [20]

Est un langage de programmation informatique orienté objet, La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows,
Mac OS ou Linux, avec peu ou pas du tout de modifications. Pour cela, diverses plateformes associées visent à garantir la portabilité des applications développées en Java.

I.2.2 XML[21]

XML (eXtensible Markup Language, soit « Langage de balisage extensible ») est un langage de balisage définissant un format universel de représentation des données, permettant de décrire la structure hiérarchique d'un document. Le fichier à extension .xml contient à la fois les données et les indications sur le rôle que jouent ces données, ces indications (ou balises) permettent de déterminer la structure des documents.

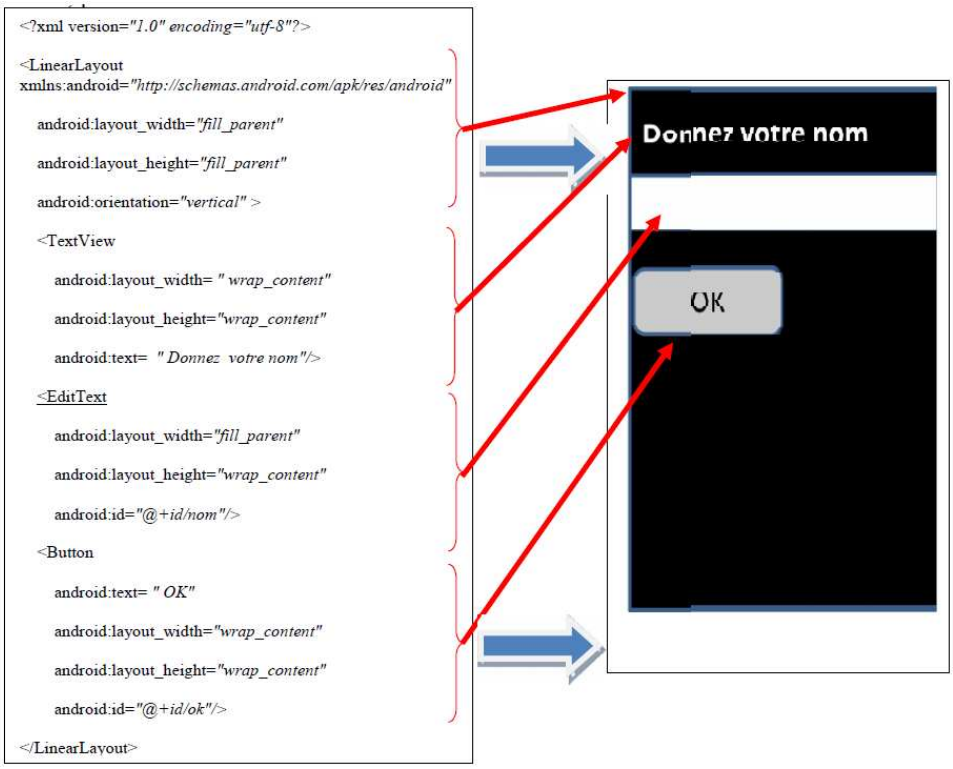


Figure 4.4 : exemple d un fichier xml

I.2.3 Android Studio [22]

Android Studio est un environnement de développement des applications Android. Il est basé sur la version IntelliJ.

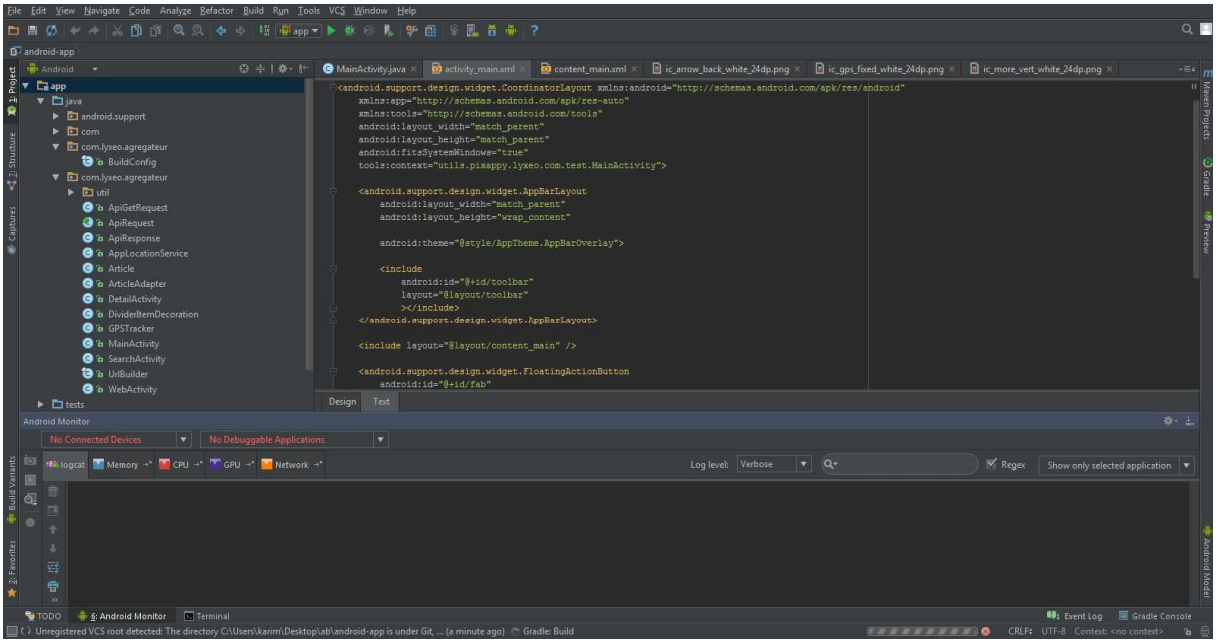


Figure 4.4:Interface de l'environnement Android Studio 1.1.0

I.3. Protocole et format de données

I.3.1 Protocole de communication

Dans notre projet, nous avons utilisé le protocole HTTP, afin de communiquer les données entre le client et le cuisinier et le serveur web. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client (facilite le dispatch des fonctions).

I.3.2 Format de données communiquées

JSON (JavaScript Object Notation) est un format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter des informations structurées.

Un document JSON ne comprend que deux éléments structurels :

- des ensembles de paires nom / valeur.
- des listes ordonnées de valeurs.

Ces mêmes éléments représentent 3 types de données :

- des objets.
- des tableaux.
- des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou nulle.

Le principal objectif du choix d'utilisation du JSON, dans notre application, est qu'il est simple à mettre en ouvre. Comme il présente les avantages suivants :

Facile à apprendre, car sa syntaxe est réduite et non-extensible;

Ses types de données sont connus et simples à décrire ;

Peu verbeux et léger, ce qui le rend bien adapté aux terminaux mobiles contrairement au XML qui est très verbeux.

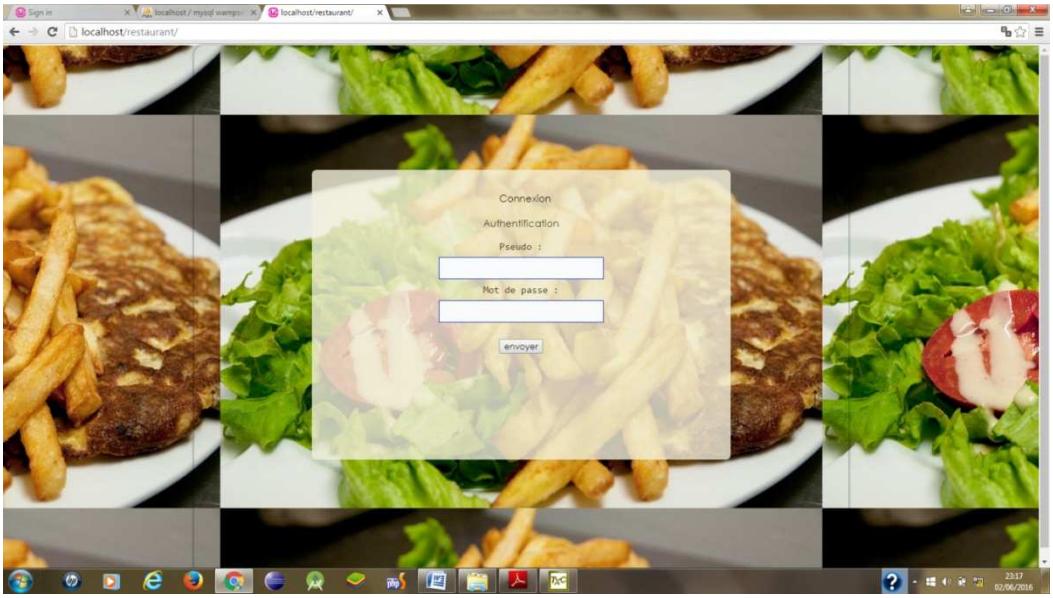
II. Interface Homme/Machine de l'application

Dans ce qui suit nous présenterons les différentes interfaces de l'application en citant les détails de chaque chaque interface ..

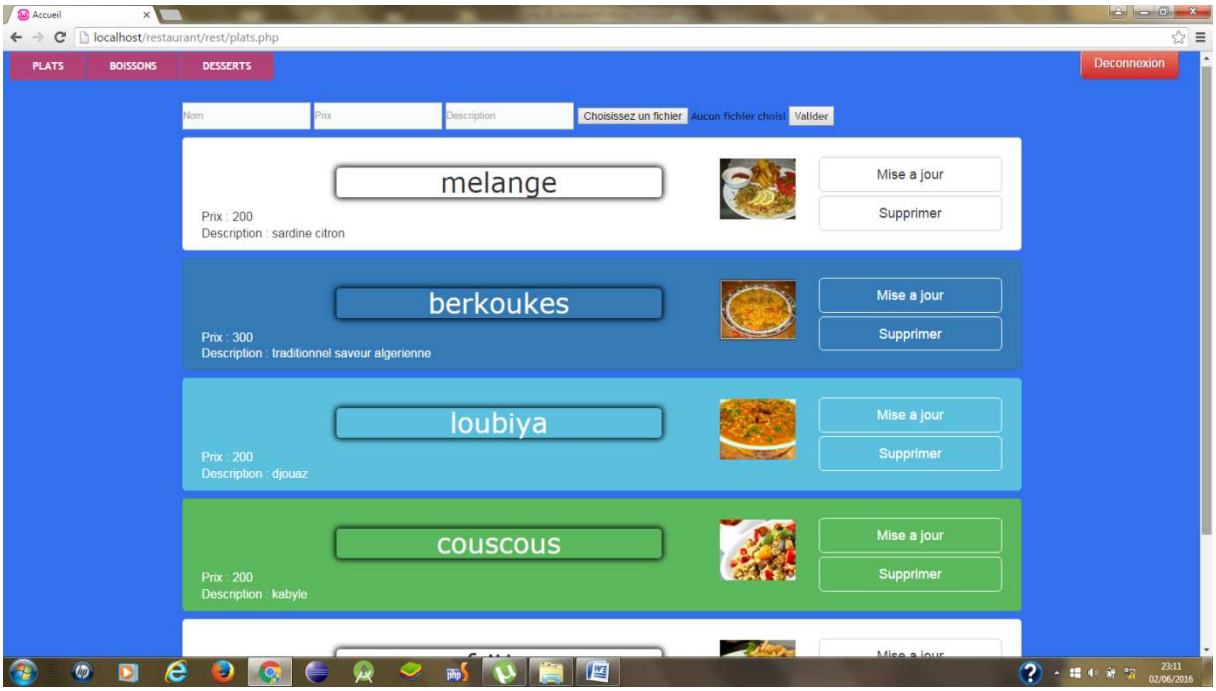
Interface « Administrateur (caissier) »

Au premier lancement de l'application c'est l'interface principale, elle s'affiche lors du lancement de notre application coté caissier, l’authentification.

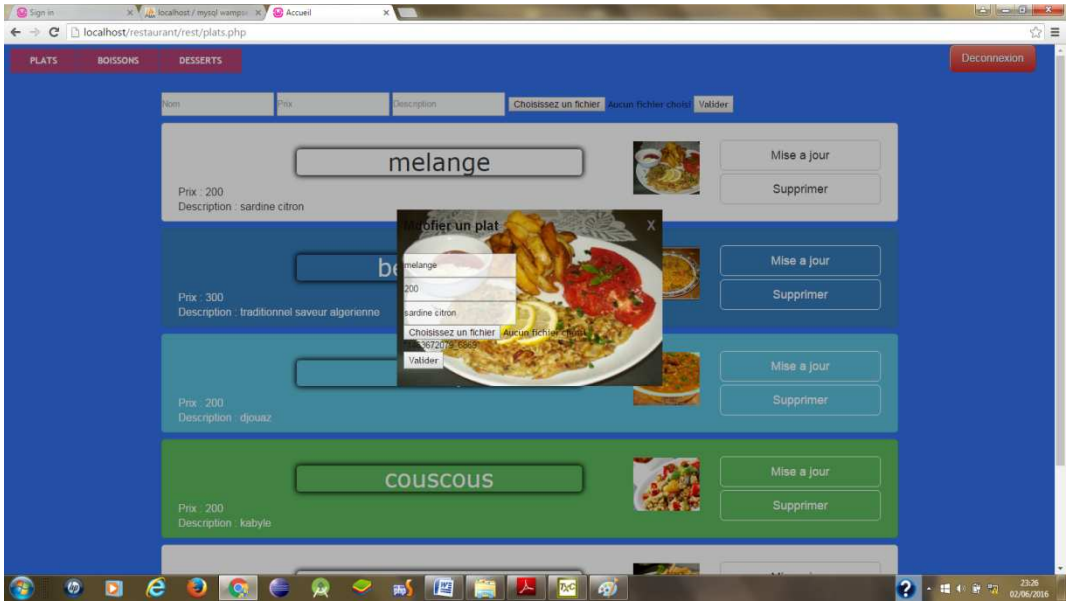
II.1. Interface authentification caissier :



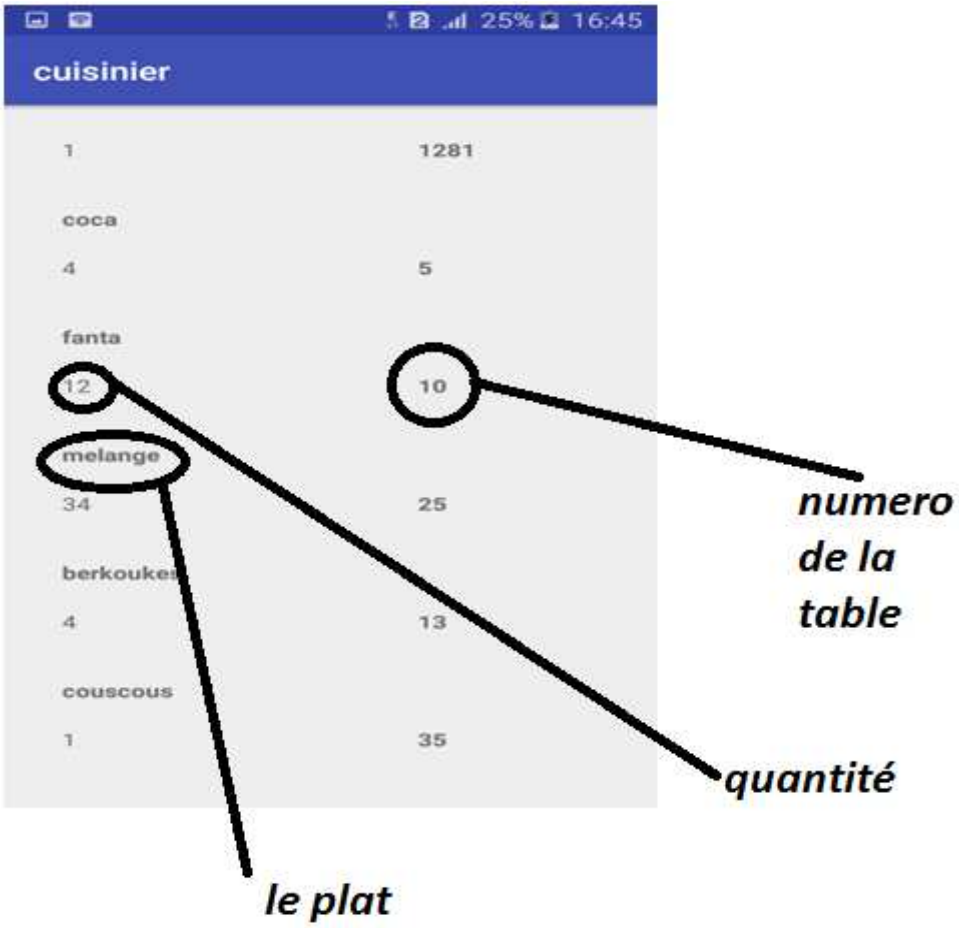
II.2. Interface plat



II.3 Interface mise à jour plat



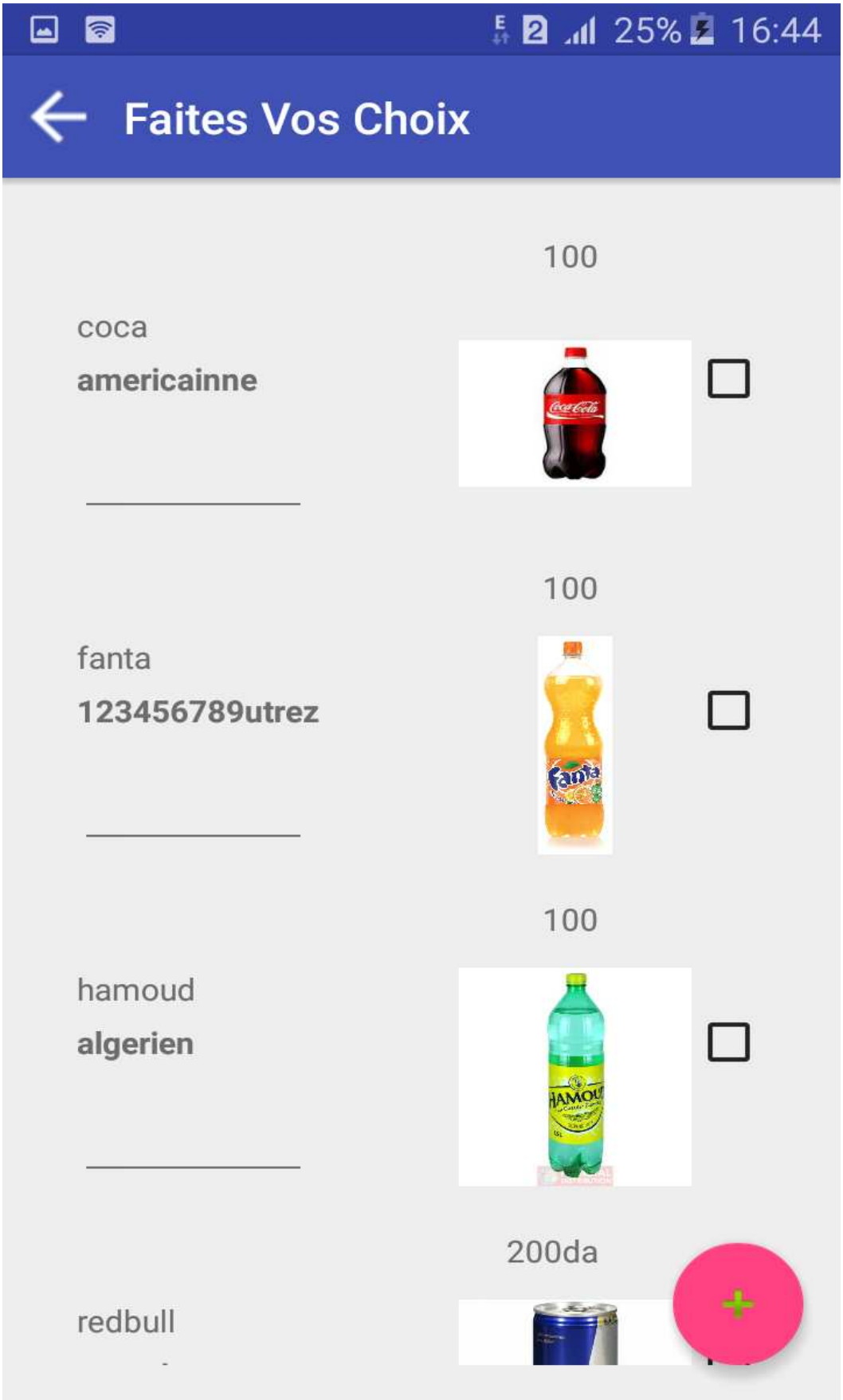
II.4 Interface cuisinier



II.5 Interface écran d'accueil client

II.6 Interface écran menu plats

II.7 Interface écran menu boissons



Conclusion

Dans ce chapitre de réalisation, nous avons présentée les plates-formes matérielles et logicielles sur lesquelles nous avons développé notre projet, ainsi que les technologies employées. Nous avons, par la suite, présenté les interfaces les plus significatives de notre application.

Conclusion générale

Conclusion générale

Ce projet nous a permis d'avoir une idée plus claire et plus précise sur le développement sous Android ; ce système d'exploitation qui a envahit le marché des terminaux mobiles.

Durant notre étude nous avons vu de façon détaillée le système, son architecture, ses dernières versions ainsi que les outils utilisés afin de réaliser ce projet, d'ailleurs nous avons opté pour l'utilisation des dernières technologies d'où notre travail s'est reposé sur une implémentation sous Android Studio avec le langage JAVA, le SDK Android et une base de données SQL.

Pour arriver à un résultat nous avons suivie une démarche de modélisation en commençant par la spécification du cas d'utilisation dans un premier temps, suivi d'une étude de conception précise et détaillée.

Enfin il y a une la mise en œuvre de notre application, l'obtention des résultats et l'objectif est atteint, comme l'amélioration de notre travail reste envisageable et comme perspective.

Webographie

Références webographiques

[1] Priya Viswanathan. What is a Mobile Application?. Disponible sur http://mobiledevices.about.com/od/glossary/g/What-Is-A-Mobile-Application.htm .

[6]Android-france.fr. C'est quoi Android. Disponible sur http://android-france.fr/android-cest-quoi/

[7] Socialcompare.com. Comparaison de versions Android. Disponible sur http://socialcompare.com/fr/comparison/android-versions-comparison.

[8] Theverge.com. ios iphone ipad history. Disponible sur http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad

[9] Digitalknob.com. Disponible sur http://digitalknob.com/

[11] http://fr.wikipedia.org/wiki/Android

[12] http://blog.erlem.fr/android/31-architecture-generale-d-android

[13] http://fr.wikipedia.org/wiki/Android_SDK.

[16] http://fr.wikipedia.org/wiki/Widget.

[17] http://www.definitions-webmarketing.com/Definition-Android-Market.

[19] http://www.devlopement android .com

[20] http://fr.wikipedia.org/wiki/Java_(langage)

Bibliographie

Références Bibliographiques

[2] Michael brard, Le guide du chef de projet mobile, Smile – Open Source Solutions. 2012.

[3] NGUYEN Van Tien. « Outils et méthodes pour le développement des applications embarquées ». Juillet 2007.

[4] Ko Dooms, Roope Kylmäkoski, “Comprehensive documentation made agile – experiments with RaPiD7 in Philips”. 2005.

[5] Harleen K. Flora, Dr. Swati V. “Chande «a review and anaysis on mobile application development processes using agile methodlogies”. 2013.

[10]Nouha KHYARI, « Rapport de stage sur le projet ‘’Locate my car’’- google map android », Ecole nationale des sciences de l’informatique Tunisie, 2010.

[14]Reto Meier , « Android 4: développement d'applications avancées» , Pearson Education France, 4 sept. 2012 - 836 pages.

[15]Béatrice Bertrand, «ANDROID», CDDP de l’Eure , Académie de Rouen.

[18] Chantal Morly, Jean Hugues, Berland le blanc , «UML2 pour l’analyse d’un système d’information» , 2002- 2006

[21] Ludovic Roland , «Structurez vos données avec XML» , Mis à jour en 2015

[22] AndroWiiid et Frédéric Espiau , «Créez des applications pour Android»