

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



*Mémoire de fin d'étude En vue de l'obtention de diplôme de Master en
Informatique Option : Administration et Sécurité des Réseaux
Thème*

Conception et Réalisation d'une Application Réseau pour
le Système ANDROID
Cas : Gestion du stock de Cevital.

Présenté par

M^r *ZOURANE* Akli

M^{lle} *MAMMERI* Lydia

Soutenu devant le jury composé de :

President M^r *ALOU* Abdelouhab

Promoteur M^{me} *ALOU* Soraya

Examineurs M^r *KHENOUS* Lachemi

M^r *BAADACHE* Abderrahmane

Promotion 2013

Dédicaces

Je dédie ce modeste travail à :

A mes parents ,
A mes soeurs,
Aux membres de l'association Focus d'Ouzellaguen ,
Aux membres de l'association Afriscience de Bejaia ,
A mes amis et camarades, et tous ceux qui m'ont aidé.

ZOURANE Akli

Je dédie ce modeste travail à :

A mes chères parents ,
A mon frère et mes adorables soeurs,
A la mémoire de mes grand parents,
A mes adorables amis Farid, Meriem, Mehani ,Salah ,
A Tous mes amis, tous les étudiants de notre promotion et à tous ceux qui ont contribué à la réussite de ce projet.

MAMMERI Lydia

Remerciements

Nous tenons à remercier le tout puissant de nous avoir donné le courage et la patience jusqu'à l'achèvement de ce modeste travail.

Nous tenons à remercier aussi M^{me} ALOUI Soraya, notre encadreur, M^r BOUKHARCHA Amirouche notre encadreur à l'entreprise, ainsi que toute l'équipe du service informatique de Cevital pour leur aide précieuse qui nous ont permis de mener à bien ce modeste travail.

Nous tenons à exprimer notre gratitude aux membres de jury pour avoir accepté de juger ce travail.

Un grand merci à tous nos collègues en Master 2 et tous les enseignants du département Informatique, spécialement D^r BAADACHE et M^r AKILAL qui nous ont offert un environnement de travail extrêmement agréable.

Un grand merci à notre famille, pour son soutien qui nous a poussé à chercher au fond de nous la volonté de faire toujours beaucoup plus, à nos amis et tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

TABLE DES MATIÈRES

Table des Matières	i
Table des figures	vi
Liste des tableaux	v
Introduction Générale	1
1 les plates-formes mobiles	5
1.1 Introduction	5
1.2 L'origine de l'informatique mobile	5
1.2.1 Presentation	5
1.3 Windows mobile, IOS Apple, Android, BlackBerry : Que choisir ?	6
1.3.1 Windows Mobile	6
1.3.2 Ios Apple	7
1.3.3 BlackBerry	8
1.3.4 Android	8
1.4 Conclusion	10
2 Le système d'exploitation Android	11
2.1 Introduction	11
2.2 Historique	11
2.2.1 Open Handset Alliance	12
2.2.2 Licence Utilisée	12
2.3 Système Android	13
2.3.1 Définition	13
2.3.2 Différentes versions d'Android	13

2.4	Architecture d'Android	16
2.4.1	Linux Kernel	17
2.4.2	Moteur d'exécution d'Android (Android Runtime)	17
2.4.3	Bibliothèques [library]	18
2.5	Framework de développement	19
2.6	Applications	19
2.6.1	Les composants principaux d'une application	19
2.6.1.1	Composants applicatifs	20
2.6.1.2	les différents éléments d'interaction	21
2.6.1.3	Le fichier de configuration " Android Manifest "	21
2.7	Cycle de vie d'une application Android	22
2.8	Conclusion	23
3	Analyse et conception	24
I	Présentation de l'organisme d'accueil	25
3.1	Introduction	26
3.2	Présentation de Cevital	26
3.2.1	Historique de Cevital	26
3.2.1.1	Etapes historiques du Groupe industriel	26
3.2.2	Les facteurs clés du succès de Cevital	27
3.2.3	Organigramme générale de CEVITAL	28
3.2.4	Les objectifs	28
3.2.5	Les différentes activités des directions de l'entreprise	29
3.2.5.1	Direction des ressources humaines	29
3.2.5.2	Direction finances et comptabilité	29
3.2.5.3	Direction du contrôle de qualité	29
3.2.5.4	Direction logistique	30
3.2.5.5	Direction système d'information	30
3.2.5.6	Direction Marketing	30
3.2.5.7	Direction Energie et Utilités	30
3.2.5.8	Direction Maintenance et travaux neufs (DMTNF)	31
3.2.5.9	Direction Approvisionnements	31
3.2.5.10	Direction des Silos	31
3.2.5.11	Direction Corps Gras	31
3.2.5.12	Direction Pôle Sucre	31
3.2.5.13	Direction des Boissons	32

3.2.5.14	présentation de la direction commerciale	32
3.2.6	Conclusion	34
II	Présentation du projet	35
3.3	Introduction	36
3.4	Objectifs de l'étude	36
3.5	Définition du processus unifié(UP)	37
3.6	Définition du processus unifié(UP)	37
3.6.1	Besoins Fonctionnels	37
3.6.2	Besoins Non-Fonctionnels	37
3.6.3	Etude des acteurs du système	38
3.6.4	Diagramme de cas d'utilisation	40
3.6.5	Description des cas d'utilisation	40
3.6.6	Analyse dynamique	47
3.6.7	Analyse statique	53
3.6.7.1	Diagrammes de classes	53
3.7	Conception	56
3.7.1	Le schéma relationnel	57
3.8	Conclusion	57
4	Implementation	58
4.1	Introduction	58
4.2	Environnement de développement	58
4.2.1	Architecture matérielle	58
4.2.2	Matériels utilisés	59
4.2.3	Le modèle de développement adopté	59
4.2.3.1	Architecture MVC	60
4.2.3.2	Les avantages apportés par l'architecture MVC	60
4.2.4	TECHNOLOGIES	61
4.2.4.1	COMMENT JSON VA ÊTRE UTILISÉ DANS NOTRE AP- PLICATION ?	62
4.2.5	Implémentation de la base de données	63
4.3	Développement de l'application	64
4.3.1	Création d'un AVD	65
4.3.2	Vérification de l'environnement de développement Android	66
4.3.2.1	Explication des paramètres du projet	66
4.3.3	Emulateur	66

4.4	Interface graphique de l'application	67
4.4.1	Présentation détaillée	69
4.4.1.1	Depuis un Smartphone Android	70
4.4.1.2	Depuis un ordinateur	77
4.5	Chronogramme	82
4.6	Conclusion	82

TABLE DES FIGURES

2.1	Logo d'OHA [13]	12
2.2	Ouverture de l'Android	13
2.3	Les différentes versions d'Android	14
2.4	Utilisation des versions android. [19]	16
2.5	Architecture d'Android [20]	17
2.6	Conversation d'un fichier .java à un fichier .dex [M1]	18
2.7	Architecture de ces médiathèques [M3]	18
2.8	les composants d'une application Android [23]	20
2.9	Cycle de vie d'une activité.[25]	22
3.1	Organigramme général de Cevital.	28
3.2	Organigramme général de Cevital.	33
3.3	Diagramme de cas d'utilisation de toutes les taches de l'application.	40
3.4	Diagramme de séquence " Authentification "	48
3.5	Diagramme de séquence " gestion des clients".	49
3.6	Diagramme de séquence " gestion des fournisseurs".	50
3.7	Diagramme de séquence " gestion des produits".	51
3.8	Diagramme de séquence " Gestion des commandes clients/fournisseurs "	52
3.9	Diagramme de séquence " Gestion des mouvements (entrées/sorties) "	53
3.10	le diagramme de classe	56
4.1	Architecture matérielle du système	59
4.2	Architecture MVC	60
4.3	Architecture 3-tiers du point de vue	62
4.4	Protocole et format de données	63
4.5	Implémentation de la base de données	63

4.6	Eclipse 3.8	64
4.7	création d'un AVD	65
4.8	création d'un projet	66
4.9	Emulateur	67
4.10	Page d'accueil	68
4.11	Page d'accueil en 3D	69
4.12	Imbrication de l'ensemble des composants de l'interface sous forme d'un arbre .	69
4.13	Authentification	70
4.14	Menu principal	71
4.15	Gestion des clients	72
4.16	Gestion des fournisseurs	72
4.17	Gestion des produits	73
4.18	Gestion des commandes des clients	74
4.19	Gestion des entrées/sorties	75
4.20	Page administration	75
4.21	Page de messagerie	76
4.22	Arborescence de l'interface administrateur	77
4.23	Authentification	78
4.24	Gestion des clients	78
4.25	Gestion des fournisseurs	79
4.26	Gestion des catégories	79
4.27	Gestion des produits	80
4.28	gestion des entrées	80
4.29	gestion des sorties	81
4.30	gestion de la messagerie	81
4.31	Chronogramme	82

LISTE DES TABLEAUX

2.1	les différentes versions d'Android à partir de 2007	16
2.2	Les différents composants applicatifs android et classes associées [23]	21
2.3	les différents éléments d'interaction avec leurs classes associées [23]	21
3.1	La liste des cas d'utilisation.	39
3.2	La description textuelle de l'authentification du magasinier.	41
3.3	La description textuelle de l'authentification de l'administrateur.	42
3.4	La description textuelle de la gestion des clients.	43
3.5	La description textuelle de la gestion des fournisseurs.	44
3.6	La description textuelle de gestion des produits/articles.	45
3.7	La description textuelle de la gestion des mouvements Entrées et Sorties.	46
3.8	La description textuelle de la gestion des commandes clients et fournisseurs.	47
3.9	le dictionnaire de donnée.	55
4.1	Les différents langages et outils	61

LISTE DES ABRÉVIATIONS

API	A pplication P rogramming I nterface
BSD	B erkeley S oftware D istribution
CSS	C ascading S tyle S heets
CPU	C entral P rocessing U nit R egion
GPL	G eneral P ublic L icense
GPS	G lobal P ositioning S ystem
HTTP	H yper T ext T ransfer P rotocol
HTTPS	H yper T ext T ransfer P rotocol S ecure
HTML	H yper T ext M arkup L anguage
IOS	I Phone O perating S ystem
LLK	L A L A K hadija
MySQL	M y S tructured Q uery L anguage
MMS	M ultimedia en M essaging en S ervice
OHA	O pen H andset A lliance S leeping
PDA	P ersonal D igital A ssistant
Pc	P ersonal C omputer
RAM	R andom A ccess M emory
RIM	R esearch I n M otion
RTC	R éseau T éléphonique C ommuté
SMS	S hort M essage S ervice
SSL	S ecure S ockets L ayer
WIFI	W ireless F idelity
TVA	T axe V aleur A joutée
USB	U niversal S erial B us
URL	U niform R esource L ocator

VPN	V irtual P rivate N etwork
NFC	N ear F ield C ommunication

INTRODUCTION GÉNÉRALE

Les entreprises ont pris conscience des avantages stratégiques de l'informatique mobile. Compte tenu de la diversité des moyens d'accès, de la baisse du prix des équipements et du coût des télécommunications, la portabilité constitue désormais un atout compétitif tangible pour un nombre d'entreprises qui peuvent aujourd'hui s'équiper d'ordinateurs portables, de PDA, de tablette tactiles ou de téléphones évolués avec l'assurance d'un retour sur investissement rapide.

D'après CARPENTIER Gilles [L1] Différentes études ont montré que les entreprises équipées d'un parc d'assistants personnels numériques connectés à distance aux systèmes d'information de l'entreprise réalisent un chiffre d'affaires supérieur à celui de leurs concurrents.

Pour les entreprises, l'informatique nomade constitue un extraordinaire moyen de renforcer leur productivité tout en rassurant aux collaborateurs une efficacité accrue par la mise à disposition à tout moment et en tout lieu des mêmes applications qu'à leur bureau. C'est le paradigme du " Anywhere, Anytime ".

Aujourd'hui ces périphériques mobiles ont atteint une puissance de calcul et une taille mémoire permettant de faire tourner des applications aussi diverses que variées en termes de technologies et de services offerts. Ces applications vont de l'Outlook mobile jusqu'aux applications de navigation GPS. Dès lors pourquoi ne pas tirer partie de cette puissance pour y déposer des applications de gestion de stock et faire en sorte que des collaborateurs aient toujours à portée de main les dernières informations sur les clients, les commandes et autres à tout moment et partout. [L1].

Au vue de ce qui précède, il nous a semblé très important de consacrer ce travail à la conception et la réalisation d'une application web pour la gestion du stock sous la plate-forme ANDROID pour l'entreprise Cevital, en particulier dans le département commercial. Nous avons choisi de nous focaliser sur ANDROID étant donné que celui-ci est manifestement parmi les plates-formes les plus commercialisé au monde.

Notre travail est structuré en quatre chapitres :

1. Le premier chapitre présente une étude sur les plates-formes mobiles.
2. Le deuxième chapitre se focalise sur l'étude du système exploitation Andoid.
3. Le troisième chapitre présente une analyse de notre projet, Pour se faire, nous avons envisagé deux parties : dans la première partie une présentation de l'entreprise de Cevital avec ces différentes activités. Et pour la deuxième partie recueillir les besoins fonctionnels et techniques ainsi le développement d'un modèle dynamique en suivant la démarche UP.
4. Le quatrième chapitre illustre l'implémentation et la réalisation de l'application.

Enfin, nous concluons notre mémoire par une conclusion générales et quelques perspectives.

CHAPITRE 1

LES PLATES-FORMES MOBILES

1.1 Introduction

Le marché des téléphones intelligents connaît actuellement une croissance qui donne une variété de choix aux consommateurs désirant de se munir de leur propre appareil.

Toutefois, parmi cette multitude d'alternatives :

- Comment choisir la plate-forme qui puisse répondre le plus adéquatement à nos besoins ?
- Quels sont les avantages et inconvénients de chacun des plates-formes ? [1]

Dans ce chapitre nous allons faire un léger survol sur l'origine de l'informatique mobile, ainsi les avantages et les inconvénients des plates-formes mobiles les plus populaires.

1.2 L'origine de l'informatique mobile

1.2.1 Présentation

A l'origine, l'informatique mobile constituée essentiellement des PC portable avait pour dessein principale de :

- Permettre aux commerciaux de présenter des produits.
- Faire des démonstrations.
- Se connecter à une messagerie interne.
- Remonter certaines données a une application spécifique à travers d'un équipement composé d'un modem et d'une connexion filaire au réseau téléphonique(RTC).

Nombre des opérations d'échange avec le système d'information de l'entreprise pouvaient aussi s'effectuer par minitel, par des services et des émulateurs auxquels les entreprises de moyenne et grande taille ont eu recours, en particulier dans la grande distribution.

L'informatique mobile est entrée dans une phase de maturité grâce à une floraison de produits permettant d'accéder à distance à Internet ou à des données d'entreprise.

Ordinateur de poche communicants, téléphones multifonctions, ordinateurs portable ultralégers ou tablette tactiles doté de connexion sans fils a la norme WIFI ou Bluetooth, ces équipements dédiés à la mobilité informatique ont envahi notre quotidien et constituent désormais une réalité incontournable. [2]

1.3 Windows mobile, IOS Apple, Android, BlackBerry : Que choisir ?

1.3.1 Windows Mobile

Première plate-forme de Microsoft pour les appareils portables, y compris les PDA, smartphones, lecteurs multimédia portables et les automobiles. Le système d'exploitation Windows CE paru en novembre 1996 est le fondement de Windows Mobile dont la première véritable version (Windows Mobile 2003) fut créée en novembre 2003. [3]

On constate que Windows Mobile possède une API extrêmement riche. Ainsi, il est possible de développer tous types de services multimédias. A l'heure actuelle, à partir d'une idée de service, Windows Mobile s'avère être la plateforme sur laquelle on peut le déployer le plus rapidement. [4]

Avantages

- Réception des courriels en temps réel, ce qui fait de Windows mobile un concurrent direct du BlackBerry de RIM.
- Capacité de visionner et de télécharger des vidéos.
- Intégration de diverses applications de communication (Messenger, Windows live) pour fonctionnée sur un téléphone.
- Utilise fast-pull pour la messagerie fiable [5]
- Une bonne intégration des réseaux sociaux.

- Une synchronisation sans fil fiable. [1]

Inconvénients

- Pas de comptabilité avec le format Flash.
- Pas de multitâches.
- Pas de transfert de fichiers via la technologie sans fil Bluetooth.[1]
- Moins de sécurité par rapport au Black Berry.

Public intéressé :

Cette plate-forme mobile " Windows Mobile " retient l'attention non seulement des consommateurs désireux d'avoir les applications professionnelles sur leur Smartphone et leur ordinateur, mais aussi d'une vaste communauté de développeurs. [1]

1.3.2 Ios Apple

iOS est le système d'exploitation utilisé dans les iPhone, iPad, iPod touch et Apple TV. Il est conçu exclusivement pour les produits vendus par la société Apple iOS depuis 2007. iOS s'appelait à l'origine l'iPhone OS, mais a été rebaptisé en 2010 afin de refléter l'évolution du support du système d'exploitation pour les appareils Apple supplémentaires.[6]

Avantages

- Un système de messagerie propriétaire.
- Un système de notifications push performant.
- Disponible uniquement dans les produits Apple.
- Une très grande quantité d'application sur l'App Store
- Un système de navigation très performant
- Appareil aux fonctionnalités très ergonomiques.
- Une bonne flexibilité entre l'utilisation personnelle et professionnelle. [1]

Inconvénients

- Durée de vie de la pile plutôt courte.
- Appareil fragile, sujet aux bris et dommages.
- Problèmes avec le système de gestion des courriels.[1]
- contrôle systématique de toutes les applications présentes dans son store.[7]

Public intéressé

Ce système d'exploitation vise a son début de création les hommes d'affaire mais après la concurrence qui a surgie, un vaste publique est visé par les concepteur de cette plate-forme. [8]

1.3.3 BlackBerry

Blackberry OS est le système d'exploitation développé par la société canadienne RIM (Research In Motion) pour les smartphones vendus sous le nom de BlackBerry.

Présent sur le marché depuis le début des années 2000, les téléphones équipés de Blackberry OS étaient avant tout réservés aux professionnels. Depuis le milieu des années 2000, et poussé par la concurrence, différentes gammes de smartphones pour le grand public ont vu le jour.

Le système Blackberry OS a subi de nombreuses évolutions depuis sa première version.

Avantages

- Un système de messagerie propriétaire.
- Une pile à l'excellente durée de vie.
- Un système de notifications push performant. [M1]
- Appareil solide d'une excellente durabilité. [1]
- Synchronisation de la messagerie d'entreprise comme Microsoft exchange lotus domino. [9]

Inconvénients

- Une quantité d'applications limitée sur le BlackBerry App World.
- Une approche exclusivement corporative (manque d'applications plus ludiques).
- Une navigation qui manque d'ergonomie et d'accessibilité. [1]

Public intéressé

Ce système d'exploitation vise a son début de création les hommes d'affaire avec un abonnement très chers pour l'utilisations mais après la concurrence qui a surgie, un vaste publique est visé par les concepteur de BlackBerry [10]

1.3.4 Android

Avec l'augmentation du nombre de smartphones qui se vendent sur le marché mondial mobile, le nombre de plates-formes de système d'exploitation pour développer des applications pour les smartphones a également augmenté. Selon les rapports publiés vers le début de l'année

2012, Android remplace progressivement iOS en termes de système d'exploitation le plus populaire pour le développement des applications pour les smartphones. Bien que le système d'exploitation de Google fasse face à la rude concurrence des autres grandes plates-formes contemporaines, il est apparu que le système d'exploitation le plus vendu de 2011 et 2012. Voici pourquoi Android est populaire parmi certains et pas si populaire parmi d'autres développeurs : [11]

Avantages

- Un langage de programmation simple.
- Une plate-forme de développement d'application ouverte.
- Un système d'exploitation polyvalent :
 - il permet aux concepteurs de construire des applications dynamiques mobiles qui servent à des fins multiples après avoir été installé dans l'appareil.
 - il permet également aux développeurs d'en savoir plus à partir du processus de conception.
- Un des meilleurs environnements de test application par rapport aux principaux rivaux systèmes d'exploitation. [11]

Inconvénients

- Durée de vie limitée de la pile
- Le confondre des caractéristiques multitâches. Ceci dit un développeur amateur a du mal à apprendre et comprendre les différentes fonctions du système d'exploitation afin d'être en mesure de l'utiliser efficacement pour le processus de développement.
- La limitation matérielle. ceci dit si les applications Android viennent avec des fonctionnalités multiples, elles ne sont pas utilisées efficacement que les appareils mobiles dans lesquels ils sont installés qui ne prennent pas en charge de nombreuses fonctionnalités. [11]

Public intéressé

Android s'adresse à un public large, mais les utilisateurs expérimentés apprécieront particulièrement la flexibilité du système, permettant une personnalisation très poussée. [12]

1.4 Conclusion

Dans ce chapitre nous avons passé en revue les différents types de système exploitation mobile, tout en présentant un bref historique de chaque plateforme.

La connaissance de ces systèmes d'exploitation est un préalable essentiel pour le projet informatique en termes de développement d'une application gestion de stock.

Le chapitre suivant sera consacré pour la plateforme Android a fin de faire une étude approfondie ainsi le développer une application simple.

CHAPITRE 2

LE SYSTÈME D'EXPLOITATION ANDROID

2.1 Introduction

A l'ère actuelle, la mobilité a pris le dessus sur les technologies informatiques, venant du fait que les gens ont toujours cherché à être continuellement connectés avec le monde extérieur quelque soient leurs emplacements.

Plusieurs fabricants ont adopté ces nouvelles technologies mobiles, en mettant à disposition pour chacun de leurs produits des plateformes de développement spécifiques comme android, permettant de mettre en valeur les capacités de leurs Smartphones [M2].

Dans ce chapitre une etude consacrée à l'exploration de la plate-forme android a savoir, son l'historique, ses différentes versions, son architecture, et finalement le cycle de vie d'une activités d'une application.

2.2 Historique

La naissance d'Android a commencé en Aout 2005, lorsque Google a acquis Android INC, une startup qui développait des applications pour téléphones mobiles.

Depuis, Andy Rubin, étant un ancien d'Android INC, a entamé son travail sur un système d'exploitation basé sur un noyau linux dédié aux appareils mobiles.

En 2007, le 5 novembre, l'Open Handset Alliance fut officiellement annoncé. C'est un consortium qui réunit un ensemble de sociétés, ayant pour but de développer des standards open sources pour terminaux mobiles. Le 15 novembre, le premier standard a été annoncé. Il s'agis-

sait d'Android .En effet, ce système d'exploitation est basé essentiellement sur la simplicité d'utilisation et surtout sur une capacité de personnalisation importante présentant un argument commercial de poids.

Pour le promouvoir, Google lui a conféré des alliés puissants réunis au sein de l'Open Handset Alliance.

En octobre 2008, apparait la première version d'Android qui n'avait pas reçu de nom. Cette version s'est avérée être la 1.beta du système.

Le 9 décembre 2008, fut annoncé que 14 nouveaux membres allaient rejoindre le projet Android, ceux-ci sont ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp et Vodafone Group Plc. [11]

2.2.1 Open Handset Alliance

L'OHA est une alliance internationale de compagnies qui se compose :

- d'opérateurs mobiles (Vodafone, Telefonica, Telecom Italia, China Mobile, etc.).
- de fabricants de téléphones mobiles (Asus, HTC, LG, Motorola, etc.).
- de fabricants de semi-conducteurs (Intel, Nvidia, ARM, etc.).
- d'éditeurs logiciels (Ebay, Google, PacketVideo, etc.).
- de distributeurs (Aplix corporation, Borqs, TAT). [12]



FIGURE 2.1 – Logo d'OHA [13]

2.2.2 Licence Utilisée

A l'exception de brèves périodes de mise à jour, les sources d'Android sont disponibles grâce à Google depuis le 21 Octobre 2008 sous la licence Apache qui permet une redistribution

du code sous forme libre. [14]

2.3 Système Android

2.3.1 Définition

Android n'est pas un robot comme on pourrait le penser, c'est une plateforme complète dédiée aux appareils mobiles. Il est basé sur un kernel linux, les bibliothèques "middleware" qui le compose sont écrites en C/C++, alors que le Framework est quant à lui écrit en java. Comme illustrée dans la **figure 2.5**. C'est une nouvelle plateforme open source. De plus, selon Google qui est un majeur distributeur, Android est une plateforme puissante, moderne, sûre et ouverte. Grâce à son ouverture, les développeurs obtiennent la permission d'intégrer, d'agrandir et de remplacer les composants existants. Ceci permettra aux utilisateurs d'adapter les applications à leur besoin.

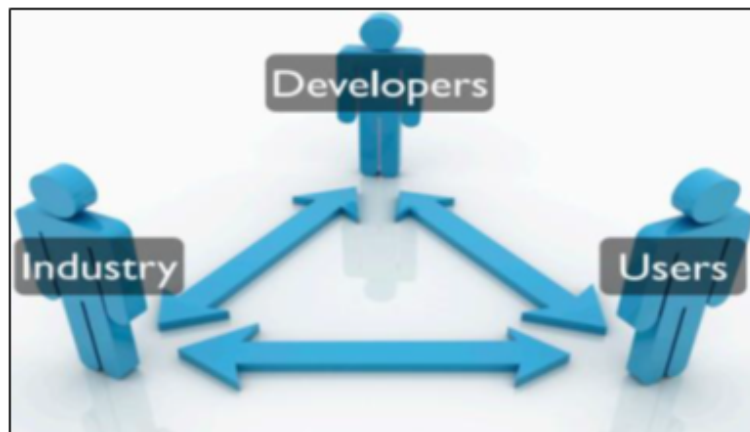


FIGURE 2.2 – Ouverture de l'Android

Android a plusieurs avantages comme la gestion mémoire, la gestion des processus, le modèle de sécurité, le soutien de bibliothèque partagée, etc. Son SDK les APIs pour développer l'application sur Android. [M1]

2.3.2 Différentes versions d'Android

Android bénéficie d'une architecture en couches complète faisant de lui une plateforme riche et évolutive dans le temps, comme l'indique cette figure 2.3 et le tableau 2.1 [L3] :

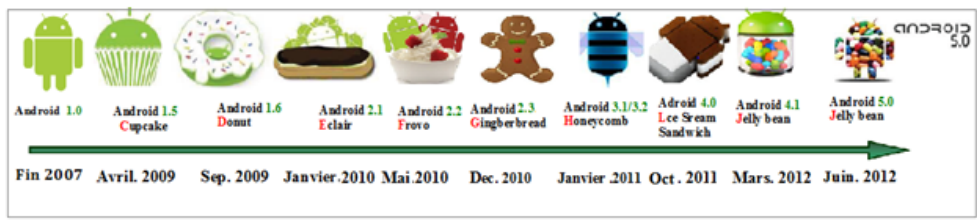


FIGURE 2.3 – Les différentes versions d’Android

Version	Caractéristique
Android 1.0 Fin 2007	<ul style="list-style-type: none"> • Début de l’aventure Android. • Recherche sur internet avec le moteur de recherche Google. • Service de messagerie instantanée avec Google Talk. • Envoi de SMS et de MMS. • Lecteur multimédia pour lire des fichiers audio et vidéo. • Notifications dans la barre de statuts, possibilité de personnaliser les sonneries, le vibreur. • Autres applications incluses: alarme, calculatrice, menu d’appel, écran d’accueil, galerie photo et menu paramètres. • Support du Wi-Fi et du Bluetooth [15].
Android 1.1 Février 2008	<ul style="list-style-type: none"> • Beaucoup de corrections de bug (alarme, Gmail...). • Amélioration des fonctionnalités MMS. • Introduction des applications payantes sur l’Android Market [L3]. • Mise à jour de Maps. • Changement dans l’API. [16].
Android 1.5 Avril 2009	<ul style="list-style-type: none"> • Possibilité d’enregistrer et lire des vidéos. • Envoi de vidéos sur YouTube. • Nouveau clavier avec dictionnaire d’aide. • Support Bluetooth. • Fonction copier/coller. • Ajout des transitions animées [16].

<p>Android 2.2 Mai 2010</p>	<ul style="list-style-type: none"> • Optimisation générale du code, meilleures performances. • Lanceur d'applications amélioré avec des raccourcis utiles. • Partage d'internet par HotSpot WiFi ou USB Tethering. • Option de mise à jour automatique dans l'Android Market. • Support de l'installation d'applications sur carte mémoire. • Support d'Adobe Flash 10.1. [16] • Vitesse améliorée. • Mémoire RAM supplémentaires • Activation de la radio FM. [18]
<p>Android 2.3 Décembre 2010</p>	<ul style="list-style-type: none"> • Mise à jour graphique de l'interface. • Amélioration du copier-coller. • Nouveau clavier virtuel. • Nouveau gestionnaire de téléchargements. • Support natif de caméra frontale. [16]
<p>Android 3.0 Janvier 2011</p>	<ul style="list-style-type: none"> • Version spécialement optimisée pour les tablettes tactiles. • Navigateur spécial avec onglets, synchronisation avec Chrome. • Support du chat vidéo. [16].
<p>Android 3.3</p>	<ul style="list-style-type: none"> • Evolution de l'interface utilisateur. • Support NFC. • Vitesse améliorée. • Support vidéo amélioré. [17]. • Reconnaissance vocale plus performante. • Appareil photo plus rapide. • Fonction Wi-Fi Direct (transfert Wi-Fi entre deux appareils). [15].
<p>Android 4.0 Octobre 2011</p>	<ul style="list-style-type: none"> • Suppression des boutons physiques. • Interface graphique plus ergonomique. • Déverrouillage par reconnaissance faciale. • Navigateur Web plus performant. [17].
	<ul style="list-style-type: none"> • Nouvelles fonctionnalités pour les notifications. • Ajout du mode hors-ligne pour la reconnaissance vocale.

<p>Android 4.1 Jelly Bean Mars 2012</p>	<ul style="list-style-type: none"> • Amélioration de la fluidité des animations dans l'interface. • Nouvelle application pour l'appareil photo. [16] • Transfert de données rapide avec le Bluetooth. • Accessibilité : mode gestuel, prise en charge clavier externe "Braille". [15].
<p>Android 4.2.2</p>	<ul style="list-style-type: none"> • Permet d'activer/désactiver le Wi-Fi et Bluetooth avec un appui long dans le menu rapide de paramètres. • Affichage du pourcentage et le temps restant estimés des téléchargements dans les notifications. • Nouveaux sons pour le chargement sans-fil.

TABLE 2.1 – les différentes versions d'Android à partir de 2007

Le taux d'utilisation des versions android :

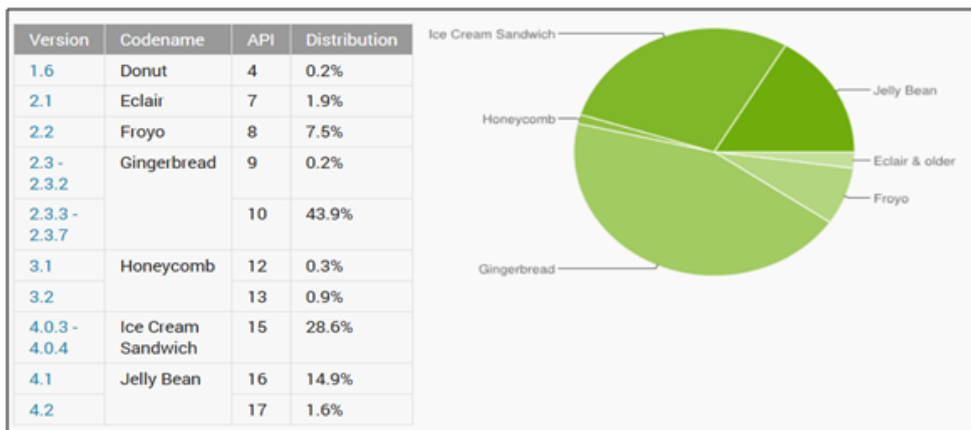


FIGURE 2.4 – Utilisation des versions android. [19]

2.4 Architecture d'Android

Le diagramme suivant illustre les composants principaux du système d'exploitation Android à savoir le Linux Kernel, Android Runtime, Librairies, Application Framework et Applications .

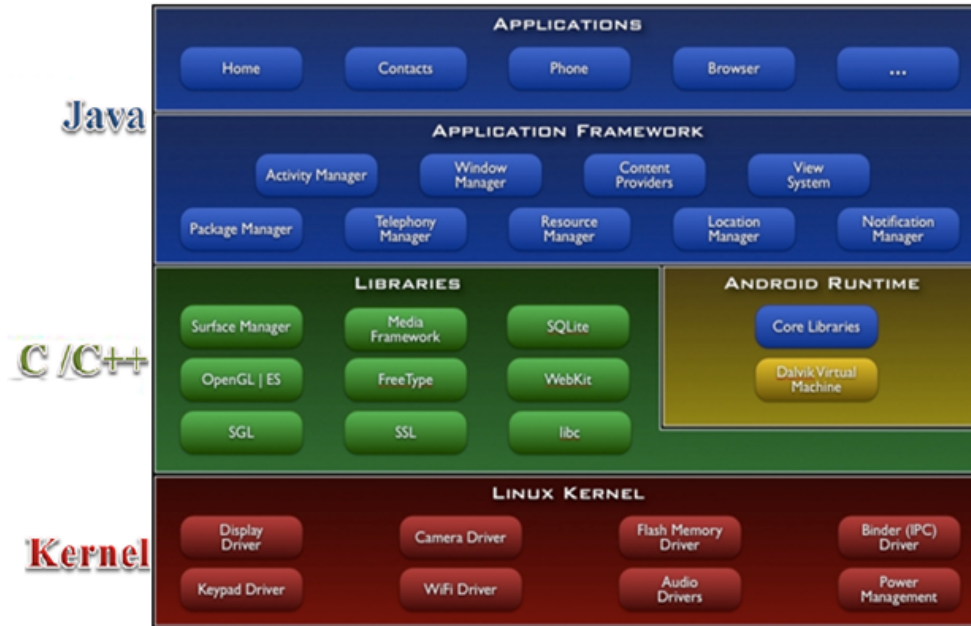


FIGURE 2.5 – Architecture d'Android [20]

2.4.1 Linux Kernel

Android repose sur la version Linux 2.6 pour les services système de base tels que la sécurité, la gestion mémoire, la gestion de processus pile réseau, et le modèle de pilote. Le noyau agit également comme une couche intermédiaire entre le matériel et le reste de la pile logicielle. [21]

2.4.2 Moteur d'exécution d'Android (Android Runtime)

Le moteur d'exécution d'Android se compose des bibliothèques du cur, la machine virtuelle **Dalvik**[22]

- Les bibliothèques du cur pour Java fournissent des puissances simples pour développer la plateforme.
- La machine virtuelle **Dalvik** fournit un environnement qui peut fonctionner sur une CPU limitée et dans les situations de faible énergie. Il utilise le format du fichier **.DEX** pour optimiser les ressources comme **CPU**, **RAM**, et pour l'application portable. Pour les fichiers **.jar**, le système utilise un outil intégré pour convertir "**byte code**" .jar en .dex. (à bien comprendre) [M1].



FIGURE 2.6 – Conversation d'un fichier .java à un fichier .dex [M1]

2.4.3 Bibliothèques [library]

Android dispose d'un ensemble de bibliothèques C/C++ utilisées par les différents composants du système. Elles sont offertes aux développeurs à travers le framework Android. En voici quelques-unes :

- **Système de bibliothèque C** : une mise en uvre dérivée de BSD de la bibliothèque C standard du système (libc), à l'écoute pour les systèmes embarqués basés sur Linux.
- **Médiathèques** : basée sur **PacketVideo** de **OpenCore**, ce sont des bibliothèques permettant la lecture et l'enregistrement audio et vidéo, ainsi que la gestion des fichiers image, y compris **MPEG4**, **H.264**, **MP3**, **AAC**, **AMR**, **JPG** et **PNG** [20].

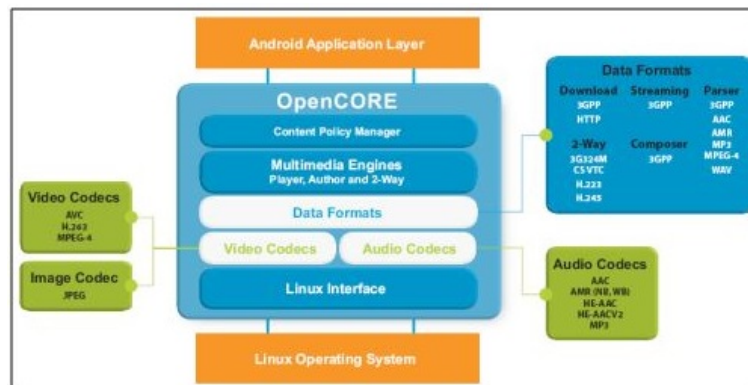


FIGURE 2.7 – Architecture de ces médiathèques [M3]

- **Surface Manager** : gère l'accès au sous-système d'affichage et de façon transparente.
- **LibWebCore** : un moteur de navigateur web moderne qui fait tourner, à la fois le navigateur Android et une vue web intégrable.
- **WebKit** est moteur de rendu, qui fournit une "fondation" sur laquelle on peut développer un navigateur web. WebKit est composé de deux bibliothèques : WebCore et JavascriptCore qui sont disponibles sous licence GPL.
- **SGL** : le moteur graphique 2D.
- **Bibliothèques 3D** : une implémentation basée sur OpenGL ES 1.0 API, les bibliothèques utilisent l'accélération 3D matérielle selon la disponibilité.

- **FreeType** : bitmap et vectoriel du rendu de police.
- **SQLite** : un moteur de base de données relationnelle puissant et léger, disponible pour toutes les applications.
- **SQLiteSSL** gérant le protocole SSL.

2.5 Framework de développement

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.
- Accéder aux informations sur la localisation d'accès.
- Exécuter des services d'arrière-plan.
- Définir des alarmes,
- Ajouter des notifications à la barre d'état...

L'architecture d'application est conçue pour simplifier la réutilisation des composants ; elle offre un ensemble de services et de systèmes, y compris :

- **Un jeu extensible de vues** : La **vue** est une couche graphique, intégrant les boutons, les zones de textes, les images, les layouts, elles peuvent être utilisées pour construire une application.
- **Les fournisseurs de contenu.**
- **Un gestionnaire de ressources.**
- **Un gestionnaire de notification.**
- **Un gestionnaire d'activité** qui gère le **cycle de vie des applications** et propose une navigation commune [20].

2.6 Applications

C'est la Couche la plus haute de l'architecture .Android est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, les contacts, un service de cartographie, un navigateur, etc. toutes écrites en JAVA. [20]

2.6.1 Les composants principaux d'une application

Une application Android se compose des composants principaux suivants :

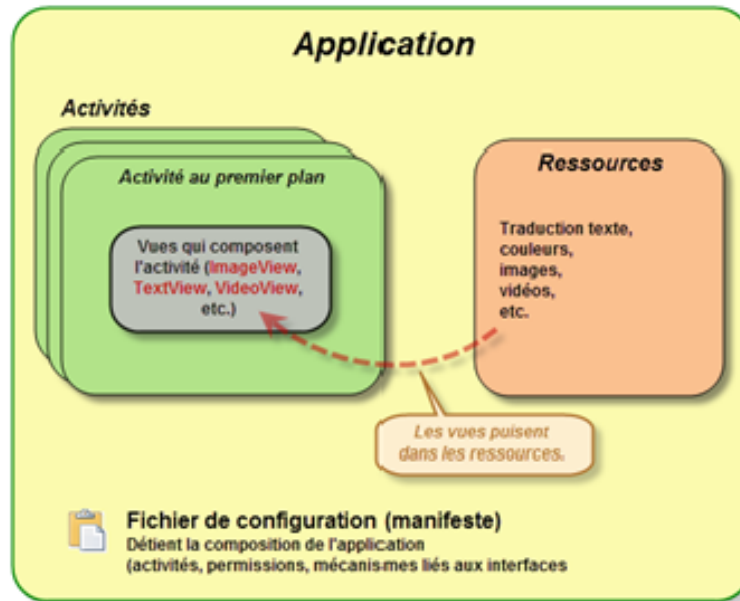


FIGURE 2.8 – les composants d'une application Android [23]

2.6.1.1 Composants applicatifs

- **Activité** : elle représente le bloc de base d'une application .elle correspond à la partie présentation de l'application et fonctionne par le bais de vue qui affiche des interfaces graphique et répond aux actions utilisateur. [23]
- **Service** : Un Service est utilisé pour réaliser l'application en arrière-plan, de manière invisible. C'est-à-dire, cette application peut marcher quand d'autres sont en train de s'exécuter comme les services de lecture de musique. [M1]
- **Content Provider(Le fournisseur de contenu)** : Ils offrent un niveau d'abstraction pour l'accès à toutes les données stocké sur le terminal ainsi le partage des données entre les applications via les fichiers, ou une base de données de **SQLite**. [M1]. les données sont identifiées au moyen d'URI (Unified Ressources Identifier). [23]
- **Le gadget (widget)** : Le gadget est un composant graphique qui s'installe sur le bureau android. Le calendrier qui affiche de l'information ou le lecteur audio qui permet de contrôler la lecture de fichiers. [23]

Les différents composants applicatifs android et classe associées :

Nom	Classe ou paquetage concerné
Activité	Android.app.Activity
Service	Android.app.Service
Fournisseurs de contenu (content provider)	Android.content.ContentProvider
Gadgets (widget)	Android.app.widget.*

TABLE 2.2 – Les différents composants applicatifs android et classes associées [23]

2.6.1.2 les différents éléments d'interaction

- **Broadcast Intent Receiver (récepteur d'intent)** : Ce sont les messages système qui servent de support événementiel pour permettre le dialogue entre application. Permet de répondre et d'envoyé des intentions qui lancent ou communiquent avec des activités. [23][24]
- **Notification** : Une notification signal une information à l'utilisateur sans interrompre ses actions en cours [23].

les différents éléments d'interaction avec leurs classe associées :

Nom	Classe concernée
Intent	Android.content.Intent
Récepteur d'intent (broadcast Receiver)	Android.content.BroadcastReceiver
Notification	Android.app.Notification

TABLE 2.3 – les différents éléments d'interaction avec leurs classes associées [23]

2.6.1.3 Le fichier de configuration " Android Manifest "

C'est un fichier XML indispensable à chaque application qui décrit entre autres :

- Le code qui doit être exécuté au démarrage de l'application
- Quels composants constituent ce programme
- Les permissions nécessaire à l'exécution du programme (accès à l'internet, accès à l'appareil photo...) [23]

2.7 Cycle de vie d'une application Android

Une application android est composée d'une ou de plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faciliter l'interactivité de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité. Le cycle de vie d'une activité est exprimé par la figure 2.9 [L4]

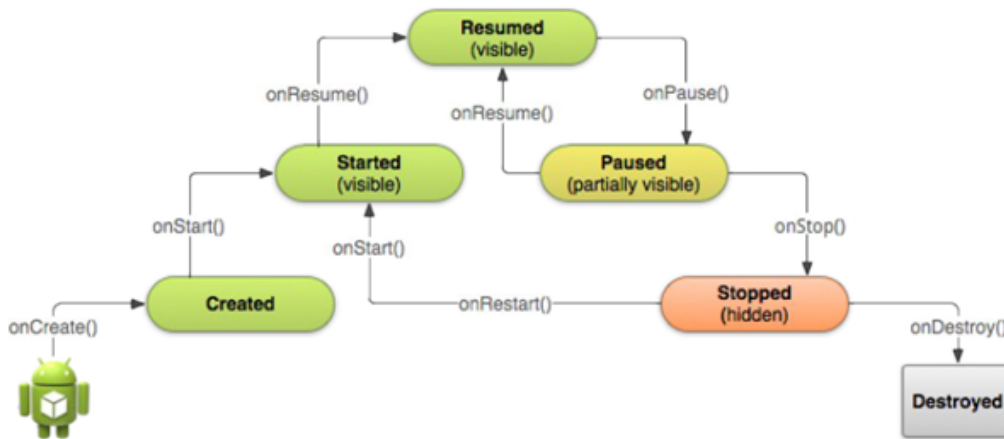


FIGURE 2.9 – Cycle de vie d'une activité.[25]

1. **L'activité démarre** : la méthode `onCreate()` est appelée.
2. **L'activité est active (démarrée)** : Activité visible qui détient le focus et attend les entrées utilisateur. C'est l'appel à la méthode `onResume()`, à la création ou à la reprise après la pause qui permet à l'activité d'être dans cet état. Elle est ensuite mise en pause quand une autre activité devient active grâce à la méthode `onPause()`.
3. **Activité suspendue (en pause)** : Activité au moins en partie visible à l'écran mais qui ne détient pas le focus. La méthode `onPause()` est invoquée pour entrer dans cet état et les méthodes `onResume()` ou `onStop()` permettent d'en sortir.
4. **Activité arrêtée** : Pendant l'utilisation d'une activité, l'utilisateur presse la touche Accueil, ou bien l'application téléphone, qualifiée comme prioritaire, interrompt l'activité en cours et prend en compte l'appel entrant. L'activité est arrêtée par l'appel de la méthode `onStop()`. Le développeur détermine l'impact sur l'interface utilisateur, par exemple la mise en pause d'une animation puisque l'activité n'est plus visible.
5. **L'activité redémarre** une fois l'appel téléphonique terminé, le système réveille l'activité précédemment mise en pause en appelant successivement les méthodes `onRestart()` et `onStart()`.
6. **Destruction de l'activité** : si l'activité reste trop longtemps en pause, le système a besoin de mémoire, il détruit l'activité au moyen de la méthode `onDestroy()`.

7. **Activité partiellement visible :**

- les méthodes **onPause()** et **onResume()** rajoutent un état à l'activité, puisqu'ils interviennent dans le cas d'activité partiellement visibles, mais qui n'ont pas le focus.
- La méthode **onPause()** implique également que la vie de cette application n'est plus une priorité du système. Donc, si celui-ci a besoin de mémoire, l'activité peut être fermée. Ainsi, il est préférable, lorsque nous utilisons cette méthode, de sauvegarder l'état de l'activité dans le cas où l'utilisateur souhaiterait y revenir avec la touche Accueil. [25]

2.8 Conclusion

Dans ce chapitre, nous avons fait une étude sur Android tout en présentant un bref historique, les versions qui existent, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et l'architecture qui lui correspondant, à savoir ces principaux composants. Enfin pour terminer nous avons donné le cycle de vie d'une activité d'une application android .ceci nous a permis de comprendre les capacités de ce système et les exploiter pour notre application .

Dans le chapitre suivant, nous allons faire une analyse de notre projet, Pour se faire, nous avons envisagé deux parties : dans la première partie une présentation de l'entreprise de Cevital avec ces différentes activité. Et pour la deuxième partie recueillir les besoins fonctionnels et techniques ainsi le développement d'un modèle dynamique en suivant la démarche UP.

CHAPITRE 3

ANALYSE ET CONCEPTION

Première partie

Présentation de l'organisme d'accueil

3.1 Introduction

La gestion de stock au sein d'une entreprise est une opération rigoureuse, qui mérite d'être perfectionnée et analysée soigneusement.

Pour cela il nous a été demandé de concevoir et de réaliser une application réseaux de gestion de stock sous android pour CEVITAL , mais avant d'essayer de porter une solution informatique pour ce processus, la présentation de l'organisme d'accueil en général et le service qui gère les mouvements de stock au sein de CEVITAL en particulier est nécessaire, et c'est ce qui est conseillé d'ailleurs dans toute démarche informatique de Génie Logiciel.

3.2 Présentation de Cevital

3.2.1 Historique de Cevital

CEVITAL est un ensemble industriel intégré, concentré en première partie dans le secteur de l'agroalimentaire : raffinage d'huile et de sucre, produits dérivés, négoce de céréales, distribution de produits destinés à l'alimentation humaine et animale.

L'ensemble industriel a traversé d'importantes étapes historiques pour atteindre la taille et la notoriété dont il dispose aujourd'hui en continuant à uvrer dans la création d'emplois et de richesse.

3.2.1.1 Etapes historiques du Groupe industriel

- **1998** : Création de CEVITAL SPA : Industries agroalimentaires.
- **2000** : Création de NOLIS : Transport maritime.
- **2001** : l'entrée en production de l'unité de margarinerie.
- **2002** :
 - l'entrée en production de la deuxième raffinerie d'huile brute.
 - la mise en place d'une raffinerie de sucre.
- **2005** :
 - Acquisition de Lalla Khedidja : Unité d'eau minérale plate et gazeuse et de sodas.
 - Création de CEVICO : Fabrication de bâtiment préfabriqué en béton.
- **2006** :
 - Acquisition de COJEK, filiale d'ENAJUC : Jus et conserves.

- Création de Numidis : Grande distribution (UNO) et (Unocity).
- **2007** :
 - inauguration de l'unité d'eau minérale au cur des montagnes du Djurdjura.
 - Création de MFG : Industrie du verre.
 - Acquisition de BATICOMPOS : Industrie de fabrication d'éléments de construction préfabriqués.
 - Création de SAMHA : Assemblage et distribution de produits électroniques et électroménagers de marque SAMSUNG Electronics en Algérie.
 - Création du Groupe Cevital.
- **2008** :
 - Création de MFG Europe : Commercialisation de verre plat en Europe.
 - Création de COGETP : Engins de travaux publics VOLVO.
 - Création de CEVIAGRO : Agriculture.
- **2010** : Création de Sodi Automotive.
- **2011** : Création de PCA - Création de Sierra Cevital.

3.2.2 Les facteurs clés du succès de Cevital

- **Capacité à manager des projets** : dans la production et la distribution de grandes envergures.
- **Maîtrise de la technologie** : les unités industrielles utilisent les dernières innovations en matière d'automatisation des procès.
- **Jeunesse des salariés (moyenne d'âge : 35 ans)** : encadrement à fort potentiel pour assurer une gestion pérenne de l'entreprise.
- **Choix du site** : l'avantage de la localisation est un facteur clé du succès, car il représente un avantage compétitif de taille sur le plan logistique, c'est-à-dire la proximité des installations portuaires réduisant les coûts des matières importées et des produits finis exportés.
- **Force de négociation** : la taille de l'entreprise, les parts de marchés investis, une comparaison taux entreprises qui évolue dans les mêmes secteurs d'activité.
- **Présence d'un réseau de distribution** : couvrant l'ensemble du territoire national.

3.2.3 Organigramme générale de CEVITAL

Le diagramme suivant donne une vue générale sur les différents organes constituant le complexe Cevital :

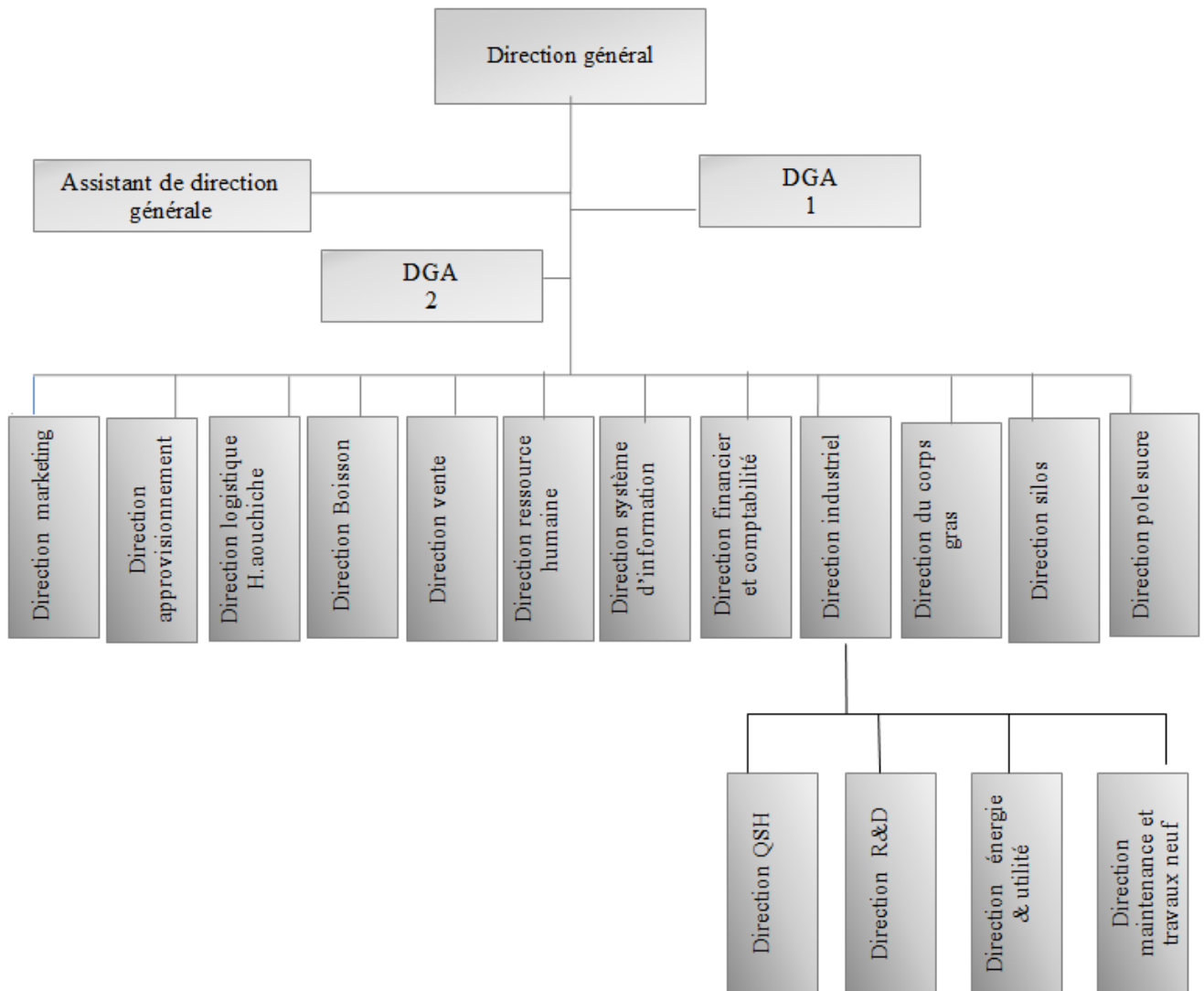


FIGURE 3.1 – Organigramme général de Cevital.

3.2.4 Les objectifs

Les objectifs visés par Cevital peuvent se présenter comme suit :

- L'Extension de ses produits sur tout le territoire national.
- L'Importation des graines oléagineuses pour l'extraction directe des huiles brutes.
- L'Encouragement des agriculteurs par des aides financières pour la production de ces graines localement.

- L'Optimisation de ses offres d'emploi sur le marché du travail.
- La modernisation des installations en termes de marché étranger et leur exportation.
- La Réponse aux besoins du client.
- L'Accroissement du chiffre d'affaire.
- L'Amélioration de ses relations de partenariat étrangères.

3.2.5 Les différentes activités des directions de l'entreprise

L'ensemble des activités de Cevital est concentré sur la production qui se présente comme suit :

3.2.5.1 Direction des ressources humaines

- Assurer la gestion administrative du personnel.
- Suivre l'effectif de l'entreprise en termes de rémunération et de formation.
- Diriger et contrôler les activités de chaque hiérarchie.
- Gérer les plans de carrière des employés et améliorer leurs conditions de travail.
- Assurer le recrutement en matière d'emploi.
- Mettre en place des structures sociales nécessaires.

3.2.5.2 Direction finances et comptabilité

- Participation aux prévisions et à l'élaboration des budgets annuels.
- Assurer la réalisation et la distribution des budgets financiers.
- Analyser les résultats et mesurer les écarts.
- Comptabiliser les différentes opérations afin de réaliser les soldes comptables par rapport aux autres directions.
- Assurer la tenue des comptes comptables et arrêter les comptes de résultats de l'entreprise.
- Traiter l'ensemble des opérations des trésoreries.

3.2.5.3 Direction du contrôle de qualité

- Le contrôle et le suivi de la qualité des produits tout au long du processus de production, pour cela des bilans sont effectués chaque 15 minute.
- Le respect des règles nécessaires pour la fabrication d'un produit conforme aux spécialisations préétablies.
- La coordination et le suivi de qualité au conditionnement et au raffinage.
- Le respect des spécifications des produits et les paramètres de fabrication.

3.2.5.4 Direction logistique

- Acheter et contrôler des matières premières.
- Assurer les approvisionnements, en les suivant de façon à éviter les ruptures de Stock.
- Elaborer une base de données fournisseurs.
- Rechercher les meilleurs prix d'achat.
- Minimiser les décalages des délais de livraison.

3.2.5.5 Direction système d'information

- assure la mise en place des moyens des technologies de l'information nécessaires pour supporter et améliorer l'activité, la stratégie et la performance de l'entreprise.
- veille à la cohérence des moyens informatiques et de communication mises à la disposition des utilisateurs, à leur mise à niveau, à leur maîtrise technique et à leur disponibilité et opérationnalité permanente et en toute sécurité.
- Définit dans le cadre des plans pluriannuels les évolutions nécessaires en fonction des objectifs de l'entreprise et des nouvelles technologies également.

3.2.5.6 Direction Marketing

Pour atteindre les objectifs de l'entreprise, le Marketing de CEVITAL varie les marques et les gammes de produits. Son principal levier est la connaissance des consommateurs, leurs besoins, leurs usages, ainsi que la veille sur les marchés internationaux et sur la concurrence.

Les équipes marketing produisent des recommandations d'innovation, de rénovation, d'animation public-promotionnelle sur les marques et métiers de CEVITAL. Ces recommandations validées sont mises en uvre par des groupes de projets pluridisciplinaires (Développement, Industriel, Approvisionnement, commercial, finances) coordonnés par le Marketing, jusqu'au lancement proprement dit et à son évaluation.

3.2.5.7 Direction Energie et Utilités

C'est la production et la distribution pour les différentes unités, avec en prime une qualité propre à chaque Procès : d'environ 450 m³/h d'eau (brute, osmosée, adoucie et ultra pure) , de la vapeur Ultra haute pression 300T/H et basse pression 500T/H. de l'électricité Haute Tension, Moyenne Tension et Basse Tension, avec une capacité de 50MW.

3.2.5.8 Direction Maintenance et travaux neufs (DMTNF)

- Met en place et intègre de nouveaux équipements industriels et procédés.
- Planifie et assure la Maintenance pour l'ensemble des installations.
- Gère et déploie avec le Directeur Industriel et les Directeurs de Pôles les projets d'investissement relatifs aux lignes de production, bâtiments et énergie/utilité.
 - Rédige les cahiers des charges en interne.
 - Négocie avec les fournisseurs et les intervenants extérieurs.

3.2.5.9 Direction Approvisionnements

Dans le cadre de la stratégie globale d'approvisionnement et des budgets alloués, elle met en place les mécanismes permettant de satisfaire les besoins de matières et services dans les meilleurs délais, avec la meilleure qualité et au moindre coût afin de permettre la réalisation des objectifs de production et de vente.

3.2.5.10 Direction des Silos

- décharge les matières premières arrivées par navire ou camions vers les points de stockage.
- stocke dans les conditions optimales les matières premières.
- Expédie et transfère vers les différents utilisateurs de ces produits dont l'alimentation de la raffinerie de sucre et les futures unités de trituration.
- entretient et maintient en état de services les installations des unités silos.

3.2.5.11 Direction Corps Gras

Le pôle corps gras est constitué des unités de production suivantes :

Raffinerie d'huile, un conditionnement de l'huile, une margarinerie qui sont toutes opérationnelles et elle a pour mission de raffiner et de conditionner les différentes huiles végétales ainsi que la production de différents types de margarines et beurre. Tous les produits sont destinés à la consommation d'où la préoccupation pour la satisfaction du marché local et de l'export qualitativement et quantitativement.

3.2.5.12 Direction Pôle Sucre

Le pôle sucre est constitué de quatre unités de production :

- une raffinerie de sucre solide 2000T/J.
- une raffinerie de sucre solide 3000T/J.
- une unité de sucre liquide 600T/J.
- une unité de conditionnement de sucre 2000 T/J 1K et 5K.

Sa vocation est de produire du sucre solide et liquide dans le respect des normes de qualité, de la préservation du milieu naturel et de la sécurité des personnes. Tous ces produits sont destinés aux industriels et aux particuliers et ce pour le marché local et à l'export.

3.2.5.13 Direction des Boissons

Le Pôle Boissons et plastiques comprend trois unités industrielles situées en dehors du site de Bejaia :

- Unité LALLA KHEDIDJA domiciliée à Agouni-gueghrane (Wilaya de TIZI OUZOU) a pour vocation principale la production d'eau minérale et de boissons carbonatées à partir de la célèbre source de LLK (lala Khadija).
- Unité plastique installée dans la même localité, assure la production des besoins en emballages pour les produits de Margarine et les Huiles et à terme des palettes, des étiquettes etc.
- Unité COJEK, implantée dans la zone industrielle d'El Kseur, qui est une SPA filiale de Cevital et qui a pour vocation la transformation de fruits et légumes frais en Jus, Nectars et Conserves. Le groupe ambitionne d'être Leader dans cette activité après la mise en uvre d'un important plan de développement.

3.2.5.14 présentation de la direction commerciale

C'est une direction des ventes qui s'occupe de la distribution de toutes les ventes effectuées par l'entreprise, elle est dirigée par un directeur des ventes qui est chargé d'appliquer la politique commerciale au niveau du réseau de vente , il fixe les objectifs régionaux ainsi que les objectifs nationaux et il négocie avec les clients importants de l'entreprise. Alors l'objectif principal de la direction commerciale est l'accroissement du volume des ventes en vue d'obtenir un chiffre d'affaire élevé.

La figure suivante illustre les différentes tâches de la direction commerciale :

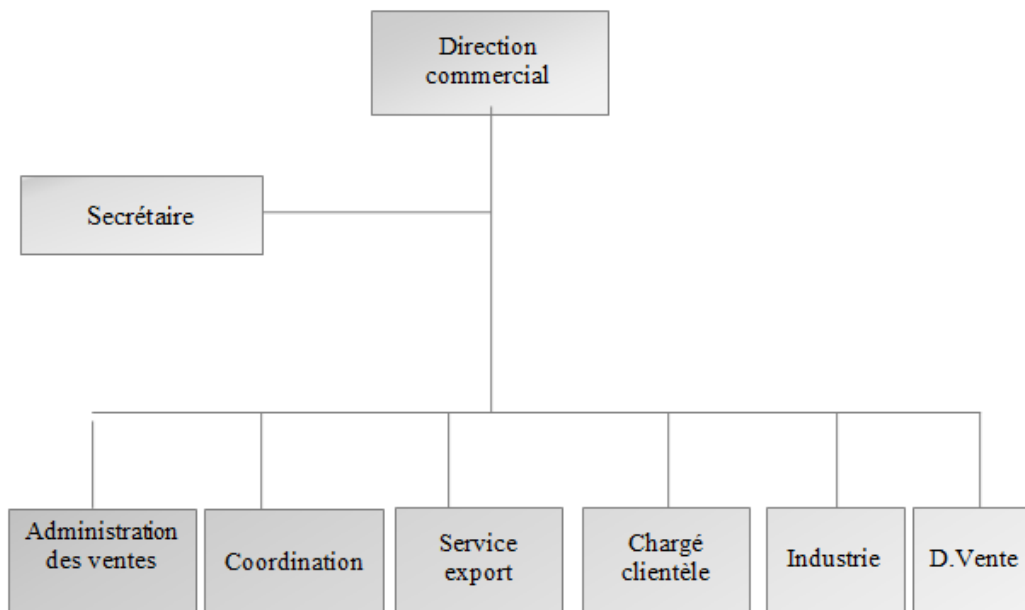


FIGURE 3.2 – Organigramme général de Cevital.

Elle se compose de :

- **Administration des ventes** : elle s’occupe des ventes au niveau national, elle est composée d’un chef de service, d’un chargé de recouvrement des créances, d’un facturier, de plusieurs agents commerciaux, d’un caissier et d’un chauffeur.
- **Service export** : elle s’occupe des ventes à l’export, de la prospection export vers plusieurs pays comme la : Tunisie, certains pays du moyen orient (Liban, Syrie.).
- **Coordination des opérations de vente** : elle s’occupe de la gestion des stocks et elle a une relation avec la logistique.
- **Service industriel** : il s’occupe de la vente aux grandes entreprises privées et industriels qui sont divisées en 3 catégories :
 1. Limonaderie : Coca, Pepsi.
 2. produits laitiers et fromagerie : Soummam, Danone.
 3. chocolaterie et biscuiterie.
- **le chargé clientèle** : Dans le cadre des activités de ventes, le chargé clientèle :
 - prend en charge la gestion globale des comptes affectés quotidiennement.
 - prend les commandes du portefeuille clients des comptes affectés.
 - assure le suivi des commandes et règlements du portefeuille clients des comptes affectés.
 - suivi de la P.A.O du portefeuille clients des comptes affectés.

- Etablissement de ristournes de transport du portefeuille clients des comptes affectés.
- Coordonner les opérations de ventes avec les chefs de ventes régionaux et les agents commerciaux de coordination.
- Tenir à jour le portefeuille clients des comptes affectés et le rapprochement permanent (hebdomadaire et mensuel).
- Communication avec le client des comptes affectés.
- **Services des ventes** : Au niveau de ce service, en premier lieu on a un chef de service qui effectue des résumés journaliers et mensuels avec la collaboration du chef de service de la gestion des stocks, en second lieu il a comme charge le suivi des clients, sa tâche consiste au suivi du fichier client et sa mise à jour, le traitement et l'étude des dossiers clients ainsi que l'établissement des bons d'affectation. En troisième lieu on a un agent de saisie qui établit les factures et les états journaliers et mensuels des ventes et des transferts. Enfin un caissier et un chauffeur.

3.2.6 Conclusion

Ce chapitre nous a permis de dégager les services déployés actuellement par la société CEVITAL. Ce qui prépare le terrain à une capture des futurs besoins qui seront étudiés et modélisés dans le chapitre suivant.

Deuxième partie

Présentation du projet

3.3 Introduction

L'objectif sous-jacent que poursuit une entreprise n'est pas évidemment de posséder un bon système informatique, mais d'exploiter les processus métiers (ou les systèmes intégrés) pour répondre au mieux à la demande du marché et d'accélérer la production de biens et de services de qualité à un prix raisonnable défiant toute concurrence [L4].

Pour cela, nous nous sommes basée dans la conception de notre application Cevitalistock, sur la méthode UP, cette dernière utilise UML comme langage de modélisation et offre une souplesse remarquable qui est exprimée par la possibilité d'obtenir des modèles de systèmes reflétant la réalité en utilisant des diagrammes.

3.4 Objectifs de l'étude

Afin d'améliorer les fonctions et les processus de travail au niveau de CEVITAL, nous allons concevoir et réaliser une application réseaux, permettant de répondre aux besoins perçus par les gestionnaires de l'entreprise.

Le système à réaliser apportera des solutions pour les problèmes cités ci-dessus et à toutes les raisons des dysfonctionnements existants.

Les objectifs du système à réaliser sont :

- Faciliter la gestion du stock avec une application réseau mobile.
- Assurer la disponibilité et la sécurité des informations.
- Exploiter les documents archivés pour délivrer des documents administratifs.
- Avoir un site évolutif pour la prise en charge des besoins futurs.
- Faciliter la récupération de n'importe quelle information concernant n'importe quels produits, clients, fournisseurs.
- Minimiser le temps de traitement des différentes procédures en éliminant les opérations qui se répètent et les redondances de documents.
- Sécuriser les informations internes contre les pertes et les modifications erronées.

3.5 Définition du processus unifié(UP)

Le processus unifié est un processus de développement logiciel permettant de spécifier les différentes phases d'un projet à mener pour transformer les besoins d'un utilisateur en un système logiciel [L5]. Le processus unifié d'une application passe par les étapes suivantes :

1. Analyse et spécification des besoins.
2. Conception.
3. Implémentation.
4. Test.

3.6 Définition du processus unifié(UP)

UP propose d'appréhender l'expression des besoins en se fondant sur une bonne compréhension du domaine concerné pour le système à développer, ainsi qu'une analyse qui se concrétise par l'élaboration de tous les diagrammes donnant une représentation du système tant statique (diagrammes de classe principalement) que dynamique (diagrammes des cas d'utilisation, de séquence) [L6].

3.6.1 Besoins Fonctionnels

L'application Cevitalistock doit satisfaire les exigences fonctionnelles suivantes :

- Authentification
- La gestion des clients.
- La Gestion des fournisseurs.
- La Gestion des produits/articles.
- La Gestion des catégories des produits/articles.
- La Gestion des mouvements des (entrées/sorties du stock).
- La Gestion des alertes.
- Etablissement du bilan des entrées et sorties (jours, mois et année).

3.6.2 Besoins Non-Fonctionnels

A part les besoins fondamentaux, notre future application doit répondre aux critères suivants :

- **La rapidité de traitement** : En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.
- **La performance** : Un logiciel doit être avant tout performant, c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences d'une manière optimale.
- **La convivialité** : l'application doit offrir une interface conviviale et ergonomique, exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support tenu.
- **Existence d'un réseau Wifi (Réseau sans fil)**.

3.6.3 Étude des acteurs du système

Un acteur est un ensemble cohérent de rôles joués par les entités externes (personnes, logiciels, matériels, utilisateurs) qui interagissent avec le système [L7].

Les acteurs qui interviennent pour la conformité de la gestion de l'application sont : **Le Magasinier** et **L'administrateur**.

Voici le tableau illustrant la liste des rôles des acteurs :

N°	Rôle	Acteur / Rôle		
1	Authentification	Administrateur, magasinier		
2	gestion des clients	Administrateur		magasinier
		ajout	modification	Suppression
3	Gestion des fournisseurs.	Administrateur		magasinier
		ajout	modification	Suppression
4	Gestion des produits/articles.	Administrateur		magasinier
		ajout	modification	suppression
5	Gestion des catégories des produits/articles.	Administrateur		magasinier
		ajout	modification	suppression
6	gestion des mouvements (entrés/sortie)	Administrateur		magasinier
		Suppression	modification	ajout
7	Gestion des commandes clients	Administrateur		magasinier
		Suppression		ajout
8	Gestion des commandes fournisseur	Administrateur		magasinier
		Suppression		ajout
9	Génération des commandes format PDF	Administrateur		
10	génération d'un catalogue de produit au format PDF	Administrateur		

TABLE 3.1 – La liste des cas d'utilisation.

3.6.4 Diagramme de cas d'utilisation

Est un formalisme qui permet de modéliser le fonctionnement d'un système par un découpage de celui-ci en fonctionnalités. Il illustre de plus la nature des interactions avec ces fonctionnalités offertes à titre de services à des acteurs externes au système. Le découpage peut être global ou très éclaté. La nature des interactions peut être décrite de manière sommaire ou détaillée selon le niveau de détail recherché par le modélisateur. Chaque fonctionnalité est appelé un cas d'utilisation [L8].

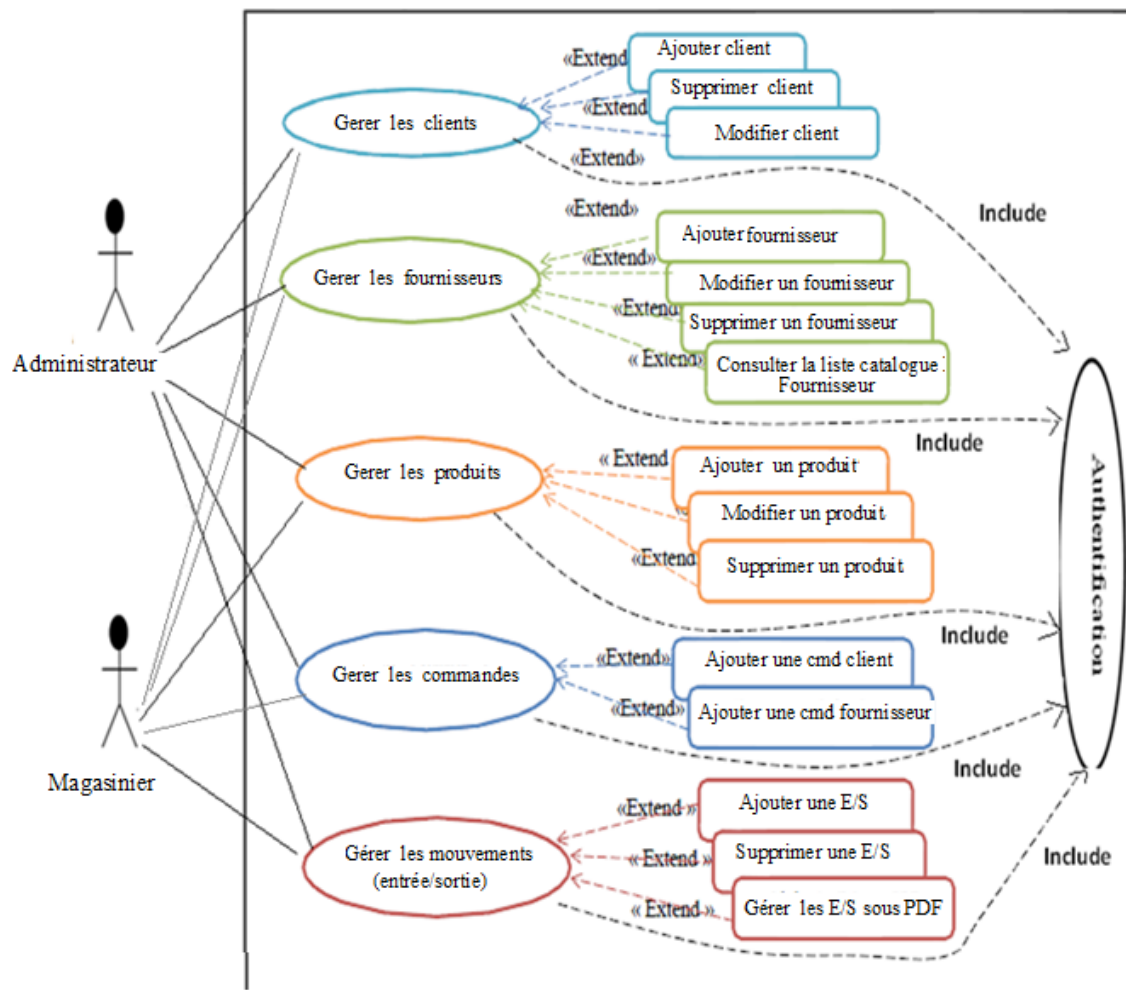


FIGURE 3.3 – Diagramme de cas d'utilisation de toutes les taches de l'application.

3.6.5 Description des cas d'utilisation

Chaque cas d'utilisation d'un système doit être défini textuellement, cela consiste à :

- **Identifier le cas** : résumé de son objectif, les acteurs impliqués.

- **Décrire un scénario nominal** : un ensemble de messages échangés entre les acteurs et le système [L9].
- **Un scénario alternatif** : un ensemble d’actions qui s’exécutent si les conditions dans le scénario nominal ne sont pas validées.

Nous allons donner une description textuelle pour chaque cas d’utilisation :

Cas d’utilisation : Authentification (magasinier).

Description du sommaire	
Titre	Authentification.
But	limiter l'accès au système juste aux acteurs concernés.
Acteurs	Le magasinier
Description des enchaînements	
Scénario nominal	<ol style="list-style-type: none"> 1. Après le lancement de l’application, un espace s’affiche pour introduire un login et un mot de passe. 2. L’utilisateur saisit l’identifiant (login) et le mot de passe chaque fois qu’il veut accéder au système.
Scénario alternatif	<ul style="list-style-type: none"> ➤ L’application réalise un premier contrôle sur la validité des champs <ol style="list-style-type: none"> a. S’il y a un champ vide, l’application affiche un message d’erreur et on revient vers 2. b. Si le format du mot de passe n’est pas valide (après 5 essais). ➤ L’application réalise un deuxième contrôle sur la validité du login/mot de passe <ol style="list-style-type: none"> a. Si le login et/ou mot de passe sont erronés, l’application affiche un message d’erreur et on revient à l’étape 2. b. Sinon (le login et mot de passe sont valides), L’utilisateur se connecte au système et peut ainsi accéder aux rubriques correspondantes à son profil de magasinier. c. si le nombre d’essais autorisés du login et/ou du mot de passe est dépassé, l’application s’éteint.

TABLE 3.2 – La description textuelle de l’authentification du magasinier.

Cas d'utilisation : Authentification (administrateur).

Description du sommaire	
Titre	Authentification.
But	Limiter l'accès au système juste aux acteurs concernés.
Acteurs	Administrateur/magasinier.
Description des enchaînements	
Scénario nominal	<ol style="list-style-type: none"> 1. Après le lancement de l'application, un espace s'affiche pour s'authentifier. 2. L'acteur choisit Connexion, Administrateur et saisit son login et son mot de passe.
Scénario alternatif	<ul style="list-style-type: none"> ➤ L'application réalise un premier contrôle sur la validité des champs. <ol style="list-style-type: none"> a. S'il y a un champ vide, l'application affiche un message d'erreur et on revient vers 2. b. Si le format du mot de passe n'est pas valide (après 5 essais). ➤ L'application réalise un deuxième contrôle sur la validité du login/mot de passe <ol style="list-style-type: none"> a. Si le login et/ou mot de passe sont erronés, l'application affiche un message d'erreur et on revient à l'étape 2. b. Sinon (le login et mot de passe sont valides). L'utilisateur se connecte au système et peut ainsi accéder aux rubriques correspondantes à l'administration. c. si le nombre d'essais autorisés du login et/ou du mot de passe est dépassé, l'application s'éteint.

TABLE 3.3 – La description textuelle de l'authentification de l'administrateur.

Cas d'utilisation : Gestion des clients.

Description du sommaire	
Titre	Gestion des clients.
But	Gérer et sauvegarder dans la base de données tous les clients de CEVITAL.
Acteurs	Administrateur/magasinier.
Description des enchaînements	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède au système après authentification. 2. Il choisit le menu gestion des clients. 3. Le système affiche la liste des clients récupérés depuis la base de données. <p><i>Cas 1 :</i></p> <ol style="list-style-type: none"> a. L'utilisateur choisit d'ajouter un client. b. Le système affiche le formulaire à remplir. c. L'utilisateur remplit et valide le formulaire. d. Le système ajoute les informations dans la base. e. Le système actualise la liste des clients et l'affiche. <p><i>Cas 2 :</i></p> <ol style="list-style-type: none"> f. L'utilisateur choisit le client à supprimer. g. Le système demande une confirmation. h. L'utilisateur soit confirme la suppression ou l'annule. i. Le système supprime le client de la base. j. Le système actualise la liste des clients et l'affiche. <p><i>Cas 3 :</i></p> <ol style="list-style-type: none"> k. L'utilisateur choisit le client à modifier. l. Il modifie les champs voulus. m. Le système met à jour les informations dans la base. n. Le système actualise la liste des clients et affiche un signal du succès de l'opération.
Scénario alternatif	➤ si un signal d'échec s'affiche, le système demande à l'utilisateur de refaire l'opération.

TABLE 3.4 – La description textuelle de la gestion des clients.

Cas d'utilisation : Gestion des fournisseurs.

Description du sommaire	
Titre	Gestion des fournisseurs.
But	Gérer et sauvegarder dans la base de données tous les fournisseurs de CEVITAL.
Acteurs	Administrateur/magasinier.
Description des enchaînements	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède au système après authentification. 2. Il choisit le menu gestion des fournisseurs. 3. Le système affiche la liste des fournisseurs récupérés depuis la base de données. <p><i>Cas 1 :</i></p> <ol style="list-style-type: none"> 1. L'utilisateur choisit d'ajouter un fournisseur. 2. Le système affiche le formulaire à remplir. 3. L'utilisateur remplit et valide le formulaire. 4. Le système ajoute les informations dans la base. 5. Le système actualise la liste des fournisseurs et l'affiche. <p><i>Cas 2 :</i></p> <ol style="list-style-type: none"> 6. L'utilisateur choisit le fournisseur à supprimer. 7. Le système demande une confirmation. 8. L'utilisateur soit confirme la suppression ou l'annule. 9. Le système supprime le fournisseur de la base. 10. Le système actualise la liste des fournisseurs et l'affiche. <p><i>Cas 3 :</i></p> <ol style="list-style-type: none"> 11. L'utilisateur choisit le fournisseur à modifier. 12. Il modifie les champs voulus. 13. Le système met à jour les informations dans la base. 14. Le système actualise la liste des fournisseurs et affiche un signal de succès de l'opération.
Scénario alternatif	➤ si un signal d'échec s'affiche, le système demande à l'utilisateur de refaire l'opération.

TABLE 3.5 – La description textuelle de la gestion des fournisseurs.

Cas d'utilisation : Gestion des produits/articles.

Description sommaire	
Titre	Gestion des produits /articles.
But	gérer les produits de l'entreprise selon le stock.
Acteurs	Administrateur/magasinier.
Description des enchaînements	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page principale après authentification. 2. Il choisit le menu gestion des produits. 3. Le système affiche deux listes « produits » ; « catégorie » récupérés depuis la base de données. 4. les étapes à suivre pour chacune d'elles sont : <p>Cas 1 :</p> <ol style="list-style-type: none"> a. L'utilisateur choisit d'ajouter un produit. b. Le système affiche le formulaire à remplir. c. L'utilisateur remplit et valide le formulaire. d. le système ajoute les informations dans la base. e. Le système actualise la liste des produits et l'affiche. <p>Cas 2 :</p> <ol style="list-style-type: none"> f. L'utilisateur choisit le produit à supprimer. g. Le système demande une confirmation. h. L'utilisateur soit confirme la suppression ou l'annule. i. Le système supprime le produit de la base. j. Le système actualise la liste des produits et l'affiche. <p>Cas 3 :</p> <ol style="list-style-type: none"> k. L'utilisateur choisit le produit à modifier. l. Il modifie les champs voulus. m. Le système met à jour les informations dans la base. n. Le système actualise la liste des produits et affiche un signal de succès de l'opération.
Scénario alternatif	➤ Si l'enregistrement échoue, le système demande à l'utilisateur de refaire l'opération.

TABLE 3.6 – La description textuelle de gestion des produits/articles.

Cas d'utilisation : Gestion des mouvements des entrées/sorties du stock.

Description du sommaire	
Titre	Gestion des mouvements des entrées/sorties du stock.
But	gérer les entrées et les sorties du stock.
Acteurs	Administrateur/magasinier.
Description des enchaînements	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page principale après authentification. 2. Il choisit le menu «gestion des mouvements Entrées/Sorties». 3. Le système affiche la liste des mouvements Entrées/Sorties récupérés depuis la base de données. <p><i>Cas 1 :</i></p> <ol style="list-style-type: none"> 1. L'utilisateur choisit d'ajouter une E/S. 2. Le système affiche le formulaire à remplir. 3. L'utilisateur remplit et valide le formulaire. 4. le système ajoute les informations dans la base. 5. Le système actualise la liste des E/S et l'affiche. <p><i>Cas 2 :</i></p> <ol style="list-style-type: none"> 1. L'utilisateur choisi une l'E/S à supprimer. 2. Le système demande une confirmation. 3. L'utilisateur soit confirme la suppression ou l'annule. 4. Le système supprime l'E/S de la base. 5. Le système actualise la liste des E/S et l'affiche. <p><i>Cas 3 :</i></p> <ol style="list-style-type: none"> 1. L'utilisateur choisit l'E/S à modifier. 2. Il modifie les champs voulus. 3. Le système met à jour les informations dans la base. 4. Le système actualise la liste des E/S et affiche un signal de succès de l'opération.
Scénario alternatif	si l'enregistrement échoue, le système demande à l'utilisateur de refaire l'opération.

TABLE 3.7 – La description textuelle de la gestion des mouvements Entrées et Sorties.

Cas d'utilisation : Gestion des commandes des fournisseurs/clients.

Description du sommaire	
Titre	Gestion des commandes des fournisseurs/clients.
But	gérer les commandes Clients/fournisseurs.
Acteurs	Administrateur/magasinier.
Description des enchaînements	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page principale après authentification. 2. Il choisit le menu « Gestion des commandes des fournisseurs/clients». 3. Le système affiche la liste « Gestion des commandes des fournisseurs/clients» récupérés depuis la base de données. 4. L'utilisateur a le choix de gérer les commandes du client ou bien celle du fournisseur. <p>Cas 1 :</p> <ol style="list-style-type: none"> 1. L'utilisateur choisit d'ajouter un cmd client. 2. Il fait une recherche du client. 3. Le système affiche le formulaire à remplir. 4. L'utilisateur remplit et valide le formulaire. 5. le système ajoute les informations dans la base et la met à jour. <p>Cas 2 :</p> <ol style="list-style-type: none"> 1. L'utilisateur choisit d'ajouter un cmd fournisseur. 2. Il fait une recherche du fournisseur. 3. Le système affiche le formulaire à remplir. 4. L'utilisateur remplit et valide le formulaire. 5. le système ajoute les informations dans la base et la met à jour. <p>5. Le système actualise la liste des E/S et affiche un signal de succès de l'opération.</p>
Scénario alternatif	si l'enregistrement échoue, le système demande à l'utilisateur de refaire l'opération.

TABLE 3.8 – La description textuelle de la gestion des commandes clients et fournisseurs.

3.6.6 Analyse dynamique

Cette phase a pour but de préciser la modélisation dynamique du système en se basant sur les diagrammes de séquence, ces derniers permettent de représenter des collaborations entre objets impliqués dans les scénarios (présentés dans les cas d'utilisation) selon un point de vue temporel, on y met l'accent sur la chronologie d'échange de messages [L10].

Dans ce qui suit, nous présentons le diagramme de séquence pour chaque cas d'utilisation dans notre système.

Opt : Fragment parcouru si une condition est vérifiée.

Loop : Répétition du fragment tant que la condition est vérifiée.

Diagramme de séquence "Authentification" :

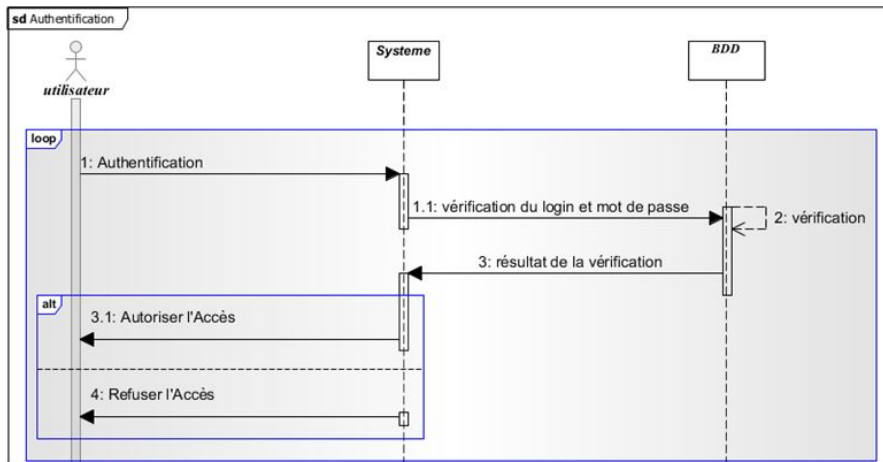


FIGURE 3.4 – Diagramme de séquence "Authentification".

Diagramme de séquence "gestion des clients" :

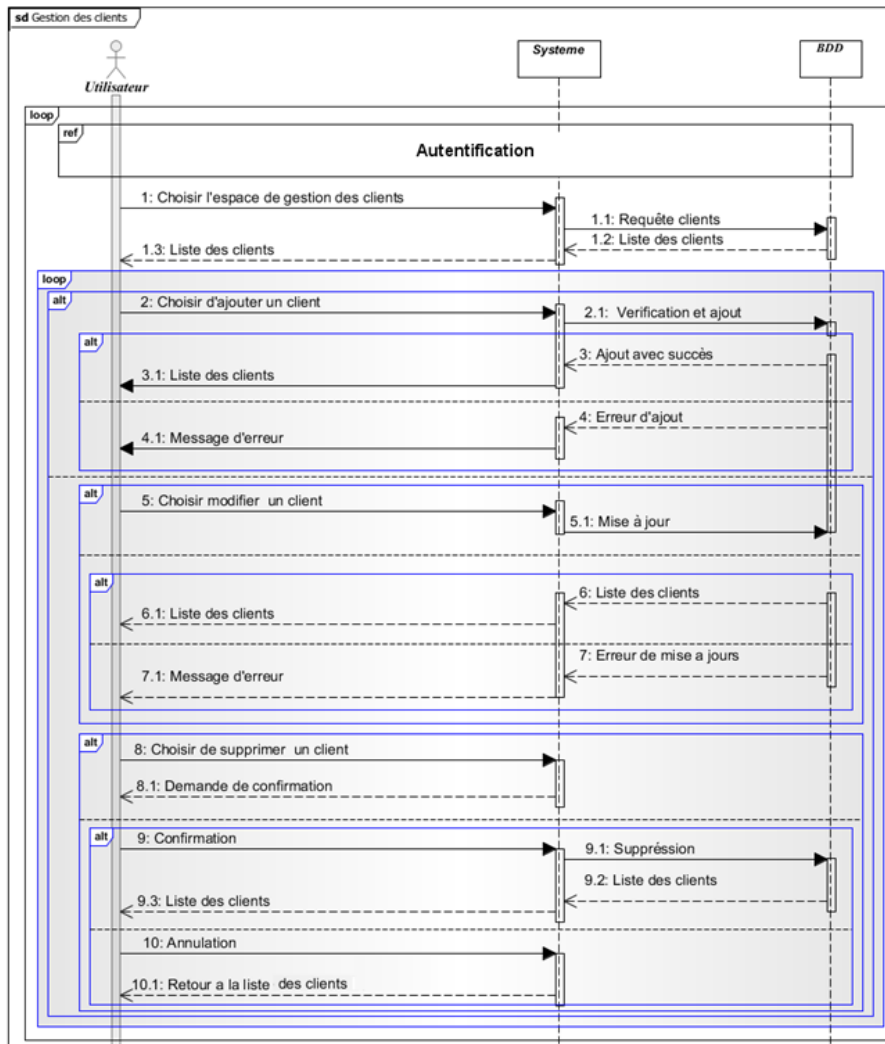


FIGURE 3.5 – Diagramme de séquence " gestion des clients".

Diagramme de séquence "gestion des fournisseurs" :

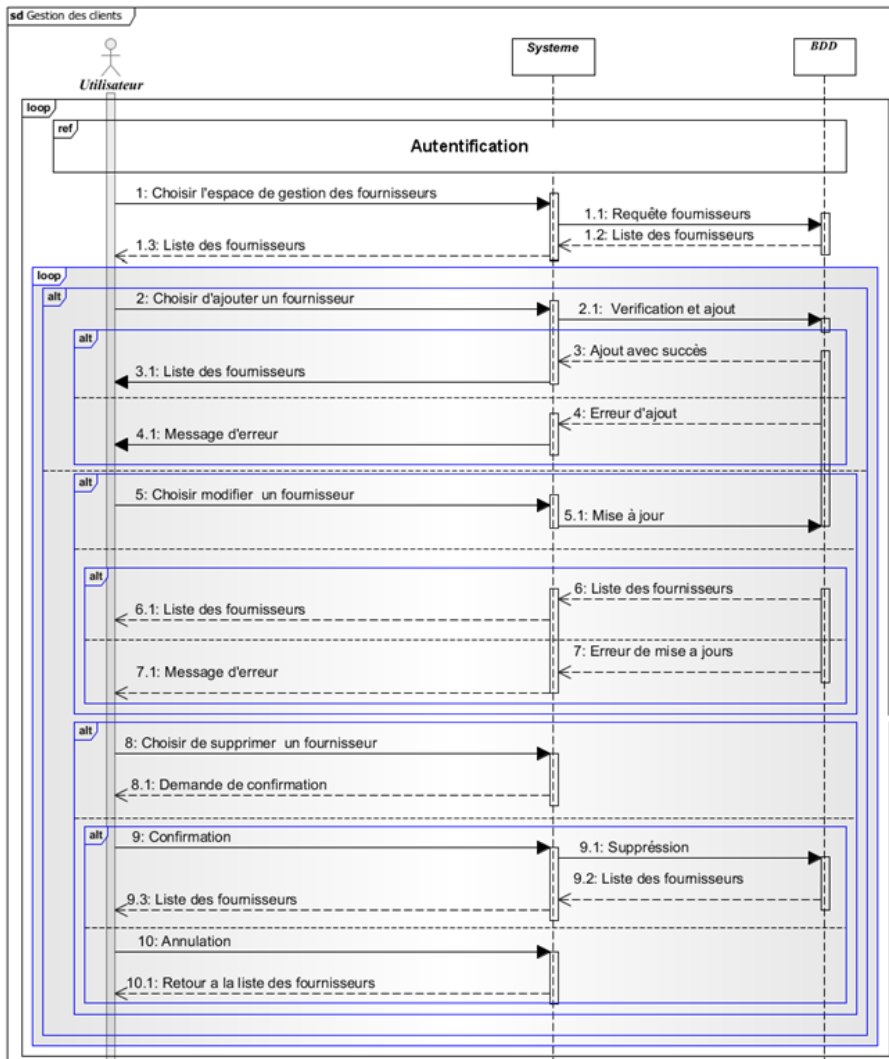


FIGURE 3.6 – Diagramme de séquence " gestion des fournisseurs".

Diagramme de séquence "gestion des produits" :

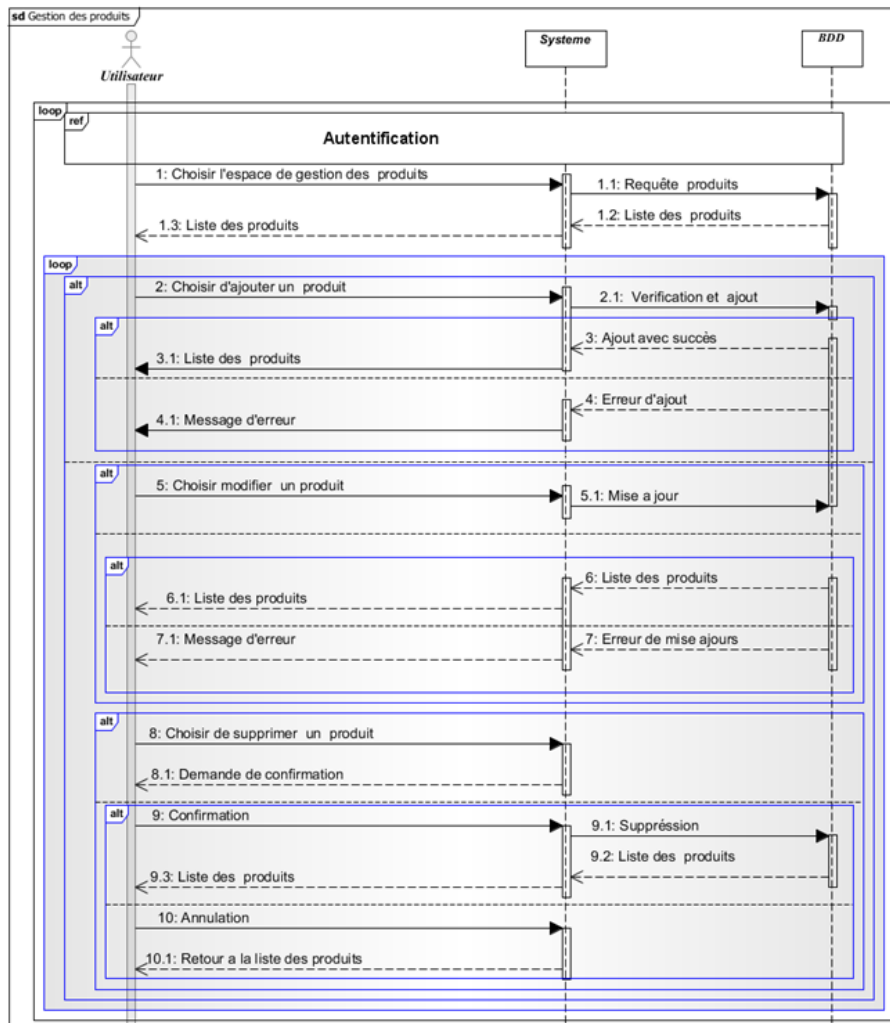


FIGURE 3.7 – Diagramme de séquence " gestion des produits".

Diagramme de séquence "gestion des commandes clients/fournisseurs" :

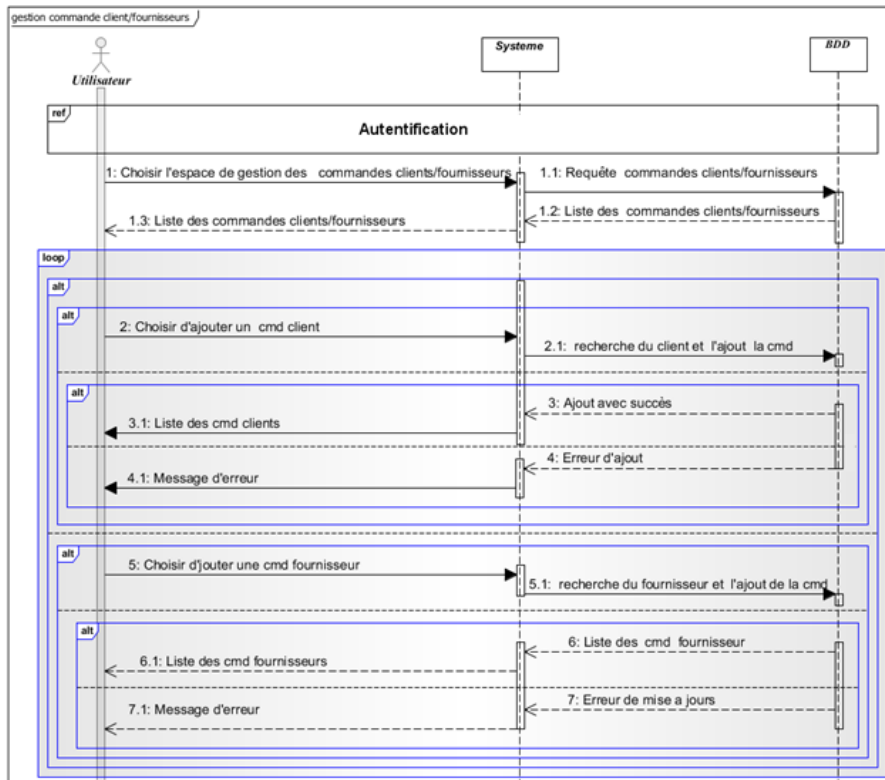


FIGURE 3.8 – Diagramme de séquence " Gestion des commandes clients/fournisseurs ".

Diagramme de séquence "Gestion des mouvements (entrées/sorties)" :

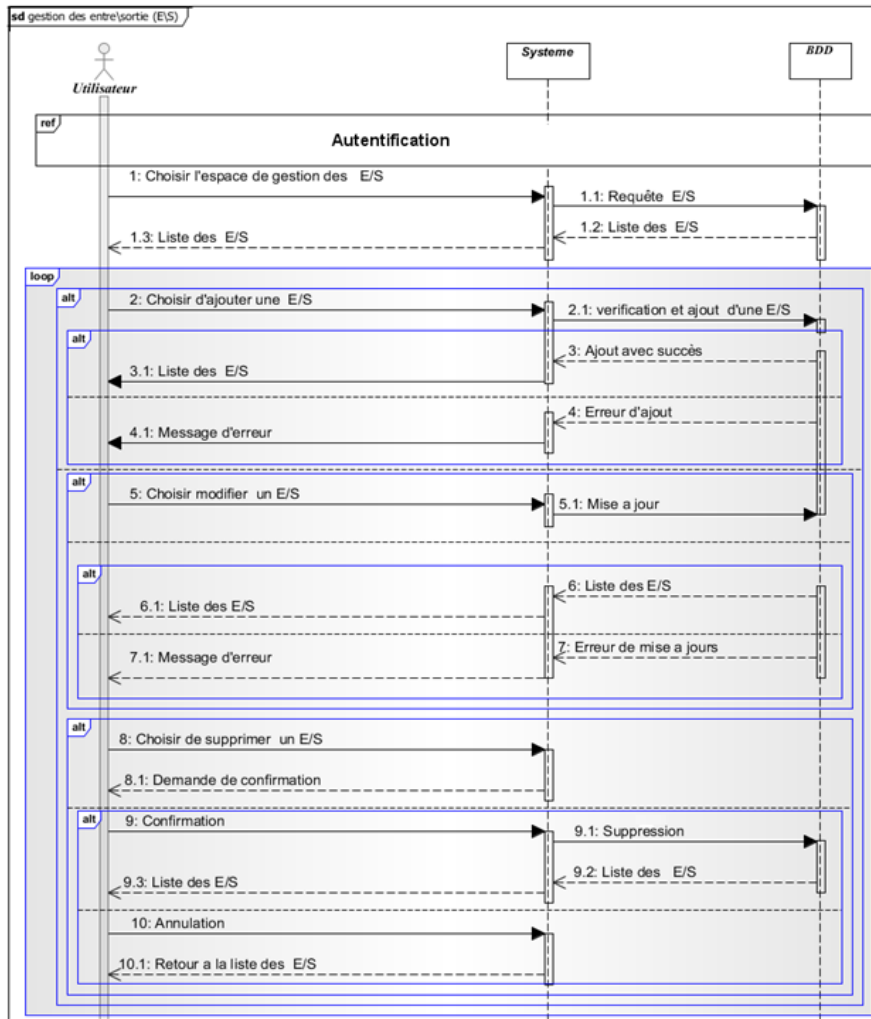


FIGURE 3.9 – Diagramme de séquence " Gestion des mouvements (entrées/sorties) ".

3.6.7 Analyse statique

3.6.7.1 Diagrammes de classes

Le diagramme de classes représente le concept le plus important dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme traduit les besoins en spécifiant comment l'application pourra les satisfaire avant de procéder à sa réalisation. En effet, dans cette étape on va représenter l'architecture de l'application et le diagramme de classes.[L11]

Le dictionnaire de données :

Classe	Attribut	Type	Description /Exemple
Client	ID_client	Int(11)	1
	Reference_Client	Int(11)	00-1
	Nom_Client	varchar(50)	ifri
	Adresse_client	varchar(50)	Ighzer Amokrane Bejaia
	Adresse Liv_client	varchar(50)	Taharachth Akbou Bejaia
	Tel_client	varchar(30)	034352219
	Fax_client	varchar(30)	034351919
	Email_client	varchar(50)	Ifri-dz@gmail.com
Fournisseur	ID_fournisseur	Int(11)	2
	Reference_fournisseur	varchar(50)	00-2
	Nom_fournisseur	varchar(50)	direx
	Adresse_fournisseur	varchar(50)	Marseille France
	Adresse Liv_fournisseur	varchar(120)	Cevital Bejaia
	Tel_fournisseur	varchar(30)	00336344455
	Fax_fournisseur	varchar(30)	00336598471
	Email_fournisseur	varchar(50)	Direx-fr@hotmail.com
Cmdclient	ID_commande_client	Int(11)	1
	Num	Int(11)	2
	Code	varchar(50)	RET1
	Date_cmd client	Int(11)	2013-06-11
	Date_liv	varchar(50)	2013-06-16
	Mode_paie	varchar(50)	Payment par chèque
	Etat	varchar(50)	Enregistré
Cmdfournisseur	ID_commande fournisseur	Int(11)	3
	Num	Int(11)	1
	Code	varchar(50)	RET1
	Date_cmd fournisseur	Int(11)	2012-06-12
	Date_liv	varchar(50)	2013 06 17
Entrée	ID_Entrée	date	2

	Date_entrée	date	2013-02-14	
	Qt_entrée	Int(11)	10	
	Prix_unitaire	Int(11)	40	
	Total	Int(11)	400	
	Description	Int(11)	RAS	
Sortie	ID_sortie	date	1	
	Date_sortie	date	2013-12-23	
	Qt_sortie	Int(11)	10	
	Prix_unitaire	Int(11)	40	
	Total	Int(11)	400	
	Description	Int(11)	RAS	
Lignecmdclient	ID_Ligne cmd client	Int(11)	1	
	Total_TTC	Int(11)	800	
	Prix_unitaire	Int(11)	40	
	Qt	Int(11)	20	
	Total_HT	Int(11)	12	
	TVA	Int(11)	17	
	Total_TTC	Int(11)	936	
	Prix_unitaire	Int(11)	70	
	Qt	Int(11)	7	
	Total_HT	Int(11)	600	
	TVA	Int(11)	77	
	Total_TTC	Int(11)	1062	
	Produit	ID_produit	Int(11)	1
		Reference_produit	Int(11)	Ser247
Nom_Produit		varchar(50)	Matina	
Prix_achat		Int(11)	40	
Prix_vente		Int(11)	100	
Qt_stock		Int(11)	67	
Qt_stockmin		Int(11)	10	
Code_barre		varchar(50)	12457856987	
Emplacement		varchar(50)	Entrepôt 1	
Catégorie	ID_catégorie	Int(11)	1	
	Nom_catégorie	varchar(50)	Sucre	

TABLE 3.9 – le dictionnaire de donnée.

Diagramme de classe :

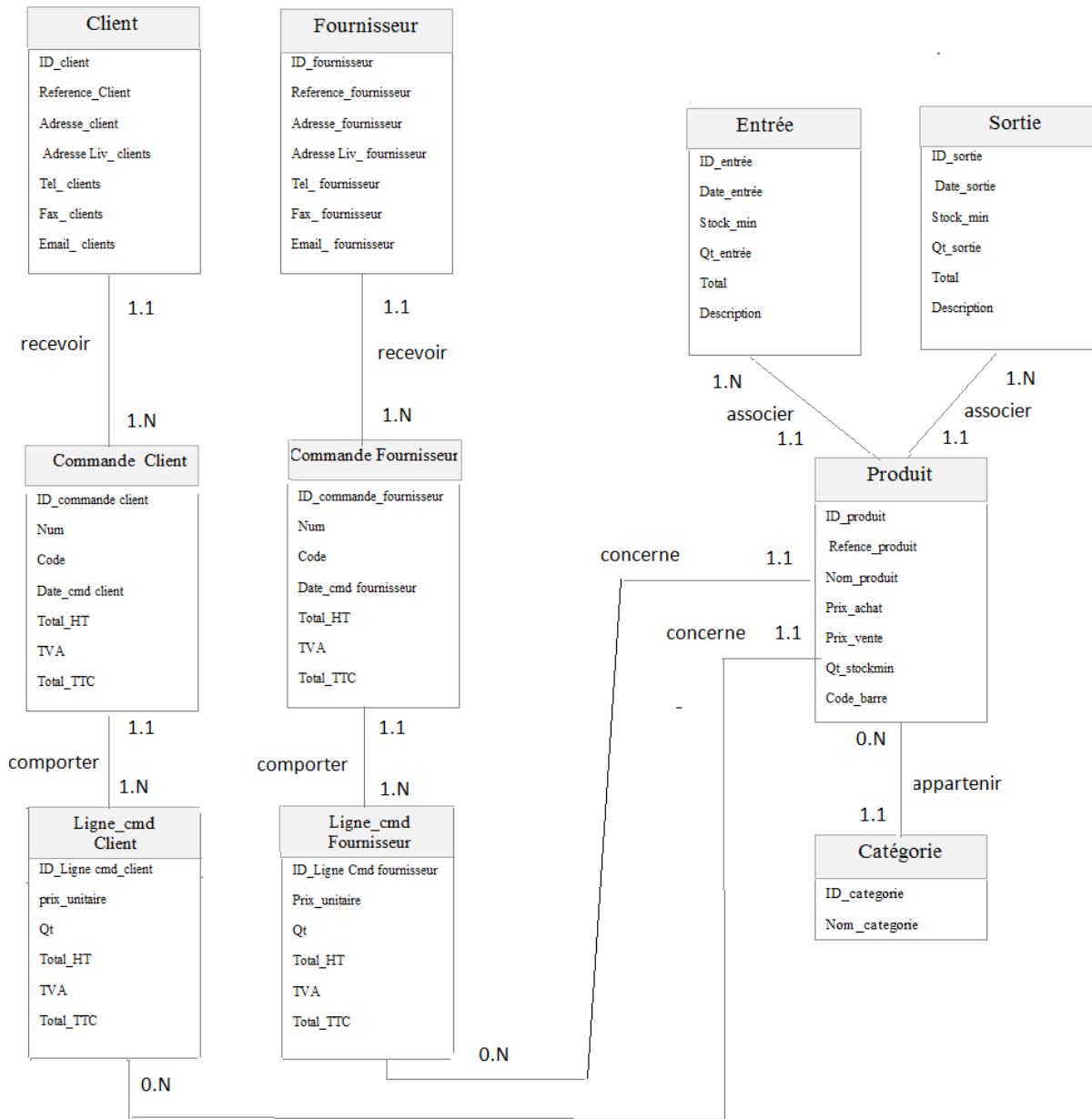


FIGURE 3.10 – le diagramme de classe

3.7 Conception

Dans la démarche du Processus Unifié, la phase de conception suit immédiatement la phase d'Analyse, elle concerne l'interprétation des besoins et la méthode adaptée pour les satisfaire dans l'application, en d'autres termes cette étape consiste à enlever toute abstraction apparue dans l'étape précédente et à donner une vision générale et simple sur le système avant l'implémentation.

3.7.1 Le schéma relationnel

Après le passage du diagramme de classe au modèle relationnel, nous présentons ci-dessous toutes les relations obtenues :

Client (ID_client, Reference_client, Nom_client, Adresse_client, AdresseLiv_client, Tel_client, Fax_client, Email_client).

Fournisseur (ID_fournisseur, Reference_fournisseur, Nom_fournisseur, Adresse_fournisseur, AdresseLiv_fournisseur, Tel_fournisseur, Fax_fournisseur, Email_fournisseur).

Catégorie (ID_catégorie, Nom_catégorie).

Cmdclient (ID_commandeclient, Num, Code, Date_cmdclient, Date_liv, Mode_paie, Etat, #ID_client).

Cmdfournisseur (ID_commandefournisseur, Num, Code, Date_cmdfournisseur, Date_liv, Mode_paie, Etat, #ID_fournisseur).

Entrée (ID_Entrée, Prix_unitaire, Date_entrée, Qt_entrée, Stock_min, Total, Description, #ID_produit).

Sortie (ID_sortie, Date_sortie, Stock_min, Qt_sortie, Prix_unitaire, Total, Description, #ID_produit).

Lignecmdclient (ID_Lignecmdclient, Prix_unitaire, Total_TTC, Qt, Total_HT, TVA, Total_TTC, #ID_produit, #ID_commandeclient).

Lignecmdfournisseur (ID_LigneCmdfournisseur, Prix_unitaire, Qt, Total_HT, TVA, Total_TTC, #ID_produit, #ID_commandefournisseur).

Produit (ID_produit, Reference_produit, Nom_Produit, Prix_achat, Prix_vente, Qt_stock, Qt_stockmin, Code_barre, emplacement, #ID_catégorie).

3.8 Conclusion

Dans ce chapitre nous avons détaillé deux phases essentielles dans l'élaboration d'un système informatique qui concerne l'analyse et la conception, où nous avons eu recours à différents diagrammes UML pour associer la modélisation adéquate à chacune d'elles.

CHAPITRE 4

IMPLEMENTATION

4.1 Introduction

Ce chapitre est dédié à la partie réalisation de l'application Cevitalistock. En premier nous présentons l'architecture sur laquelle nous avons développé notre application, les différents outils utilisés pour l'implimentation ainsi que les outils de mise en oeuvre de notre projet accompagné de quelques interfaces illustrant les fonctionnalités de l'application développée.

4.2 Environnement de développement

4.2.1 Architecture matérielle

Cevitalistock est une application embarquée qui se connecte à un serveur de bases de données distant, via le réseau wifi local de CEVITAL, afin de récupérer et mettre à jour les données. Ce qui nécessite aussi l'intégration d'un serveur entre l'application client et le serveur de bases de données. D'où 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 Cevitalistock Mobile : Vu que les données seront communiquées entre deux environnements hétérogènes, le rôle principale du serveur est de gérer la communication entre les clients Android et le serveur de base de données,
- Le serveur de base de données fournis les données au serveur Cevitalistock Mobile.

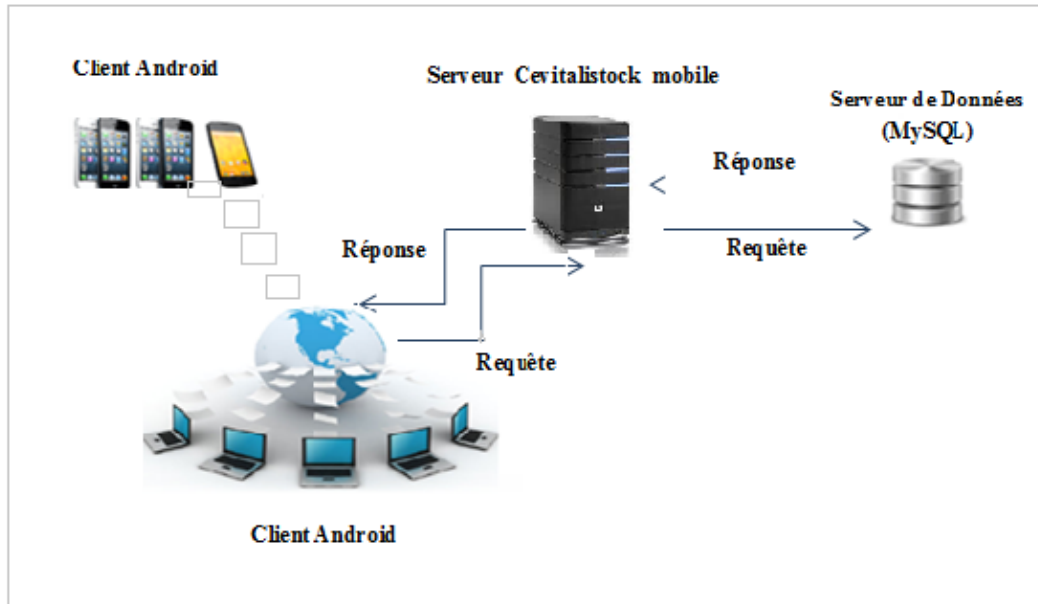


FIGURE 4.1 – Architecture matérielle du système

4.2.2 Matériels utilisés

Nous avons élaboré ce projet sur deux PC dont leurs configurations sont les suivantes :

1. PC Portable : HP - RAM : 4 GO ; windows 7.
 - Disque dur : 320 GO
 - Microprocesseur : Intel(R) Core(TM)2 Duo CPU T6670 @2.30
 - Un Système d'exploitation windows 7
2. PC Portable SONY - RAM : 4 GO windows 7
 - Disque dur : 580 GO
 - Microprocesseur : Intel(R) Core(TM) i3-62350M CPU @2.30
 - Un Système d'exploitation windows 7

4.2.3 Le modèle de développement adopté

Pour tout projet informatique, il est nécessaire d'adopter une méthodologie de travail qui soit rigoureuse. Dans notre projet, nous avons opté pour le model MVC tel qu'il est illustré et décrit dans ce qui suit :

4.2.3.1 Architecture MVC

L'architecture MVC (modèle, vue et contrôleur) est une architecture à trois couches, éprouvées comme un vrai motif de conception logiciel, qui transforme une application en un ensemble maintenable, modulaire et rapidement développé. Façonner les tâches applicatives dans des modèles, vues et contrôleurs séparés. L'architecture modulaire et séparée offre également aux développeurs et designers, la possibilité de travailler en parallèle, avec la capacité de créer rapidement un prototype. [26]

- **Modèle** : correspond aux données stockées généralement dans une base de données. Dans un langage orientée objet ces données sont exploitées sous forme de classes. Le Modèle peut aussi agir sur la vue en mettant à jour ses données. (Activity, fragment). [27]
- **Vue** : a pour rôle d'affiché l'interface graphique de l'application, gérer les interactions avec l'utilisateur et la logique de navigation (layout(XML)). [27]
- **Contrôleur** : a pour rôle de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue, puis renvoyer les données à la vue afin d'être affichées. POJO, DB. [27]

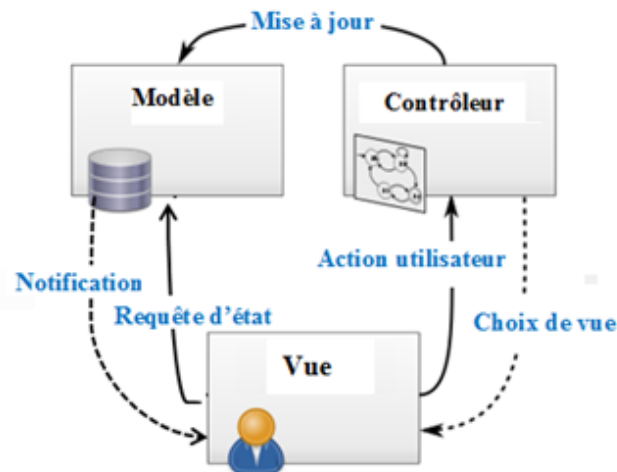


FIGURE 4.2 – Architecture MVC

4.2.3.2 Les avantages apportés par l'architecture MVC

- **L'évolutivité** : ce découpage offre la garantie de la meilleure évolutivité possible pour chaque couche. La mise à jour d'une couche n'affecte pas les autres couches, dans la mesure de ses relations avec les couches voisines. [M4]
- **La sécurité** : les données sont traitées sans qu'elles soient visibles par les clients. Pour

répondre à des exigences de sécurité en intégrer des niveaux d'isolation entre les couches comme des protocoles d'authentification dans le des achats ou transaction bancaires depuis le domicile. [M4]

- **L'interopérabilité** : la communication de divers services réalisé avec différents moyens (protocoles, langage). [M4]
- **La réutilisabilité** : grâce à la séparation logique l'entreprise peut réutiliser des composants de son système dans d'autre application. [M4]

4.2.4 TECHNOLOGIES

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






Outils						
	Latex	MySQL	PHP my admin	Notepad++	apache	Photoshop
Langage						
	Java	CSS	XML	PHP		
Bibliothèque						
	JSON	jQuery				
Plate -forme						
	android	wampserver				

TABLE 4.1 – Les différents langages et outils

La méthode la plus répandue de se connecter à une base de données MySQL à distance à partir d'un appareil Android, est de mettre une sorte de service dans le milieu. MySQL est habituellement utilisé avec PHP, donc la façon la plus simple et le plus évident 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, afin de communiquer les données entre PHP et Android, en exploitant les options faciles à utiliser construit dans les fonctions JSON dans les deux langages.

Si nous parlons de l'architecture 3-tiers de point de vue technologie, 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 – Architecture 3-tiers du point de vue

4.2.4.1 COMMENT JSON VA ÊTRE UTILISÉ DANS NOTRE APPLICATION ?

Lorsque l'application Android Cevitalistock s'exécute, elle se connectera au script PHP. Le script PHP va récupérer les données depuis la base de données MySQL. Ensuite les données seront encodées au format JSON et envoyées au système Android. Ensuite, l'application Android va obtenir ces données codées. Elle les analysera et les affichera sur le Smartphone. Le schéma suivant illustre bien la façon d'échanger les données entre le client Android et la partie des serveurs (Web/SGBD) :

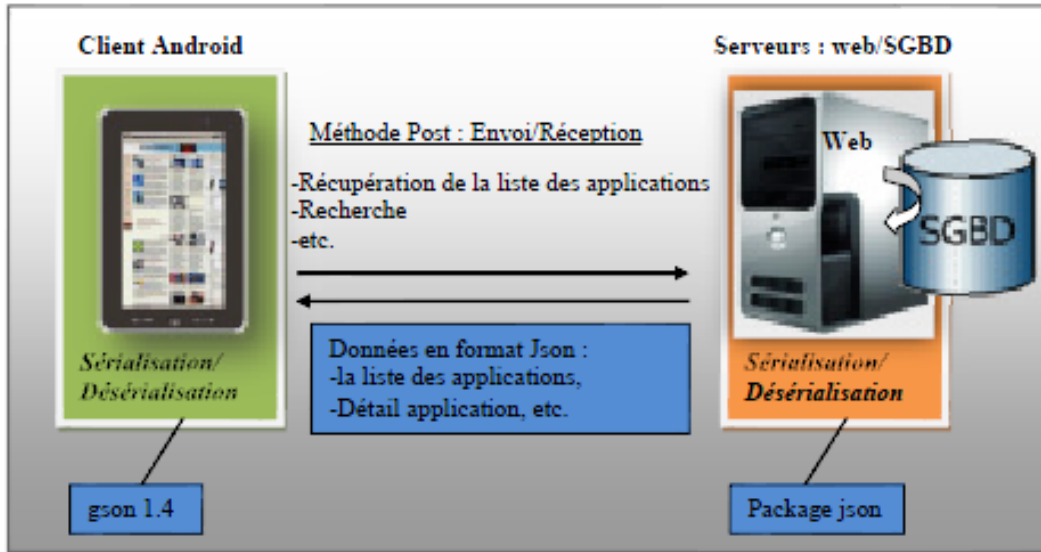


FIGURE 4.4 – Protocole et format de données

4.2.5 Implémentation de la base de données

Pour implémenter notre base des données "gestock", nous avons utilisé l'environnement de création de base des données PhpMyAdmin et le système de gestion de base des données MySQL. La figure ci-dessous présente notre base de données :

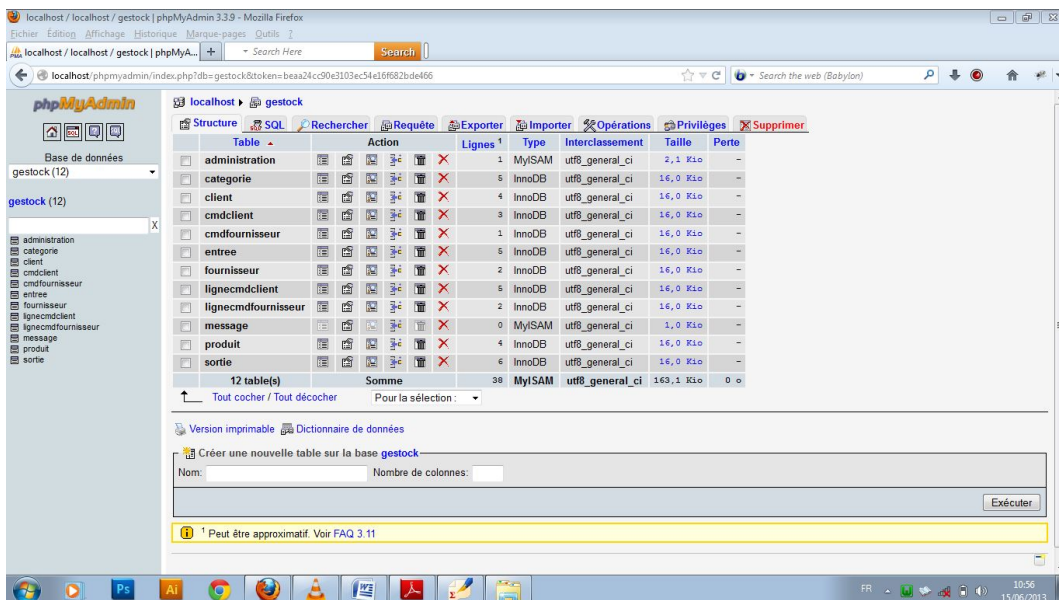


FIGURE 4.5 – Implémentation de la base de données

Après avoir présenté l'architecture matérielle ainsi que les technologies employées, il est

Clair que notre projet est divisé en deux parties :

- Ecriture des scripts PHP,
- Développement de l'application client : **Cevitalistock**

4.3 Développement de l'application

Pour pouvoir développer une application dans de bonnes conditions, il faut bien savoir choisir son environnement de développement selon les besoins, de ce fait, on a développé une application mobile en JAVA, qui a nécessité :

- Installation Eclipse
- Installation SDK Android
- Intégration SDK Android sous éclipse pour pouvoir créer des projets Android.



FIGURE 4.6 – Eclipse 3.8

Le SDK qui permet d'avoir un environnement de développement facilitant la tâche du développeur

Le kit de développement donne accès à des exemples, de la documentation mais surtout à l'API de programmation du système et à un émulateur pour tester ses applications. Stratégi-

quement, Google utilise la licence Apache pour Android ce qui permet la redistribution du code sous forme libre ou non et d'en faire un usage commercial. **Le plugin Android Development Tool** permet d'intégrer les fonctionnalités du SDK à Eclipse. Il faut l'installer comme un plugin classique en précisant l'URL du plugin. Ensuite, il faut renseigner l'emplacement du SDK dans les préférences du plugin ADT. [M5]

4.3.1 Création d'un AVD

Afin de tester les applications, il faut utiliser l'émulateur Android, il faudra donc créer un Android Virtual Device (AVD). Un AVD décrit les paramètres systèmes et les composants de notre émulateur.

Depuis Eclipse 3.8, dans le menu Windows (ou sur la barre d'outil), on peut lancer "Android SDK and AVD Manager", puis ajouter un nouveau AVD en attribuant un nom et en choisissant la cible (the target), éventuellement on peut préciser si le périphérique émulé dispose d'une carte SD en indiquant sa capacité de stockage.

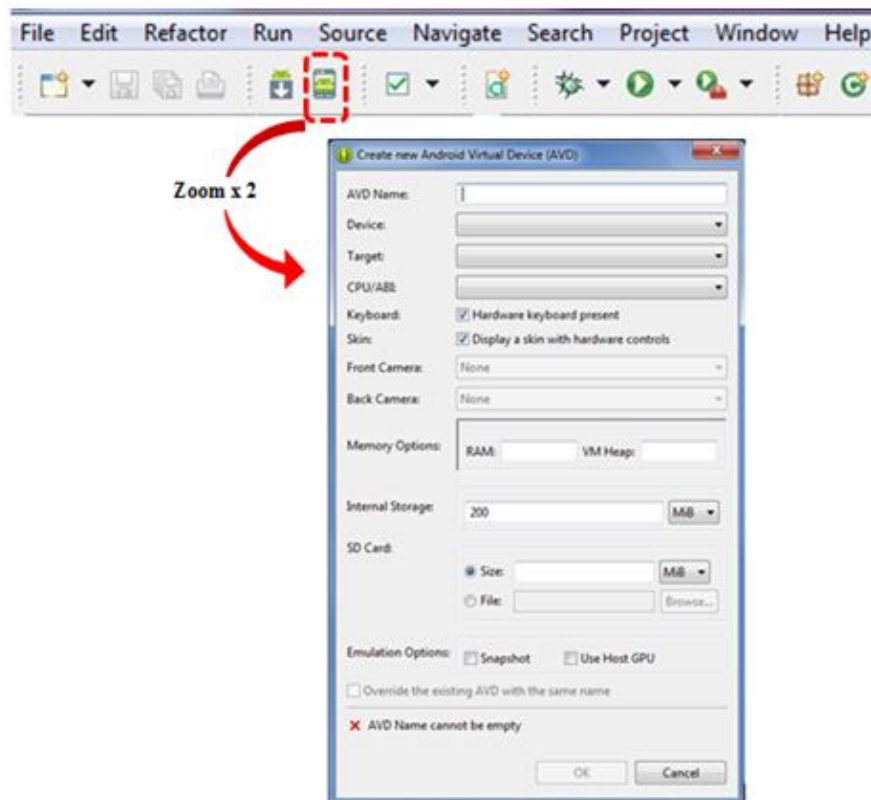


FIGURE 4.7 – création d'un AVD

4.3.2 Vérification de l'environnement de développement Android

Après avoir installé un émulateur Android, on va le tester et créer un nouveau projet sous Eclipse qui est " Cevitalistock "

4.3.2.1 Explication des paramètres du projet

- **Project Name** : C'est le nom du projet Eclipse. Tous les fichiers seront créés sous un dossier portant le même nom.
- **Application Name** : C'est le nom de l'application tel qu'il va apparaître sur le smartphone Android.
- **Package Name** : C'est le package namespace (suivant les mêmes règles de programmation Java) qui regroupera tout le code source qu'on va écrire. D'une manière générale, le nom du package doit être unique. Dans notre projet, on a utilisé `com.android.cevitalistock`.
- **Create Activity** : C'est le nom du stub class qui va être générée par le plugin. Elle va être une sous-classe de la classe Activity d'Android. [M3]

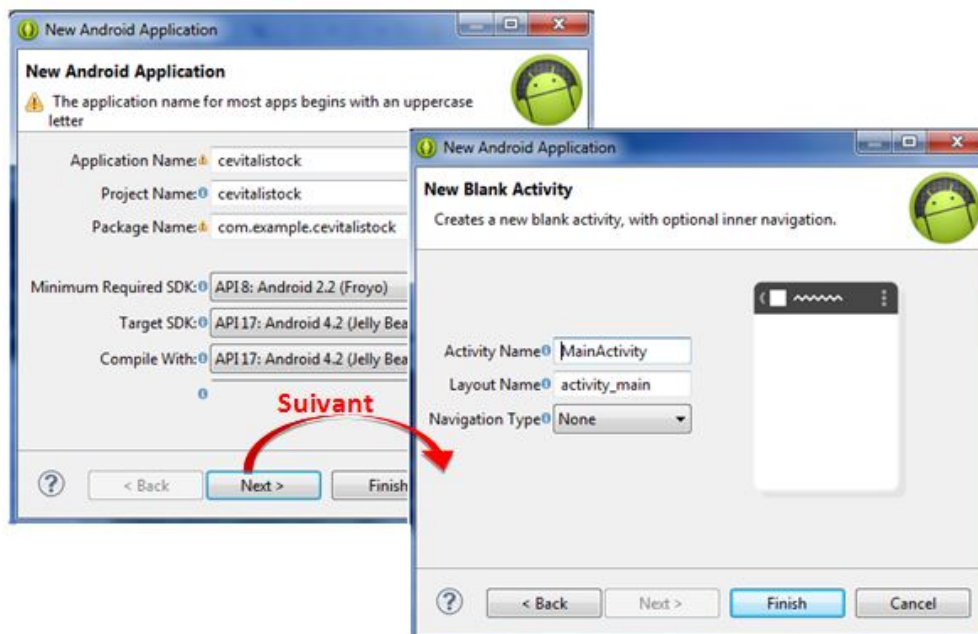


FIGURE 4.8 – création d'un projet

4.3.3 Emulateur

Comme nous l'avons évoqué précédemment, le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran

un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile. À chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel.

Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth.

Dans ce qui suit, nous donnons un aperçu général de notre application, l'interface est illustrée dans cette figure :



FIGURE 4.9 – Emulateur

4.4 Interface graphique de l'application

L'interface graphique n'est pas une image statique mais un ensemble de composants graphiques, qui peuvent être des boutons, du texte, mais aussi des groupements d'autres composants graphiques, pour lesquels nous pouvons définir des attributs communs (taille, couleur, positionnement, etc.). Ainsi l'interface illustrée par la figure 4.10 peut être vu par le

développeur comme un assemblage dans la figure 4.11

La représentation effective par le développeur de l'ensemble des composants graphiques se fait sous forme d'arbre, en une structure hiérarchique illustré dans la figure 4.12. Il peut n'y avoir qu'un composant, comme il peut y avoir des milliers, selon l'interface que l'utilisateur souhaite représenter. Dans l'arbre ci-après figure 4.12, par exemple, le composant ont été organisé 3 parties (haut, milieu et bas)

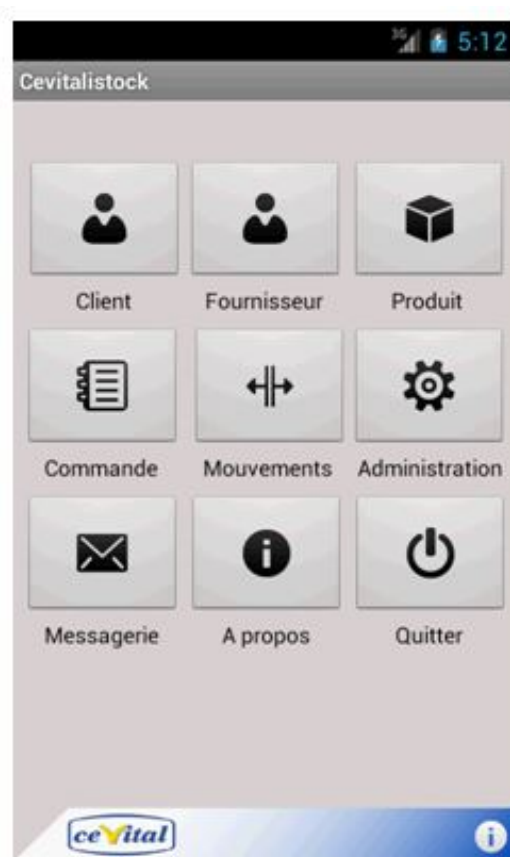


FIGURE 4.10 – Page d'accueil



FIGURE 4.11 – Page d’accueil en 3D

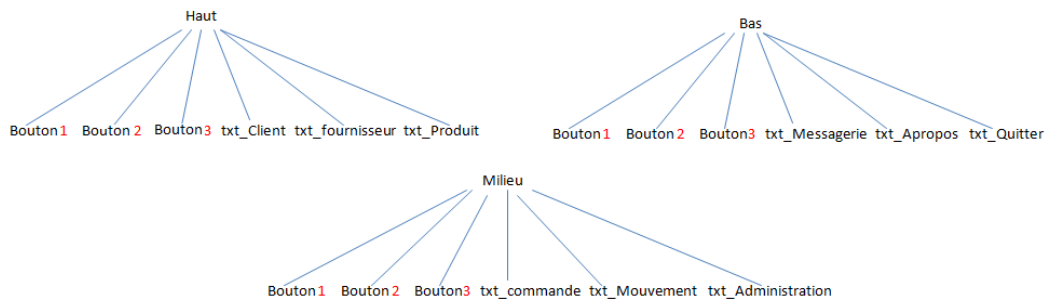


FIGURE 4.12 – Imbrication de l’ensemble des composants de l’interface sous forme d’un arbre

4.4.1 Présentation détaillée

Notre idée initiale était de ne pas imposer à l’utilisateur de se focalisé juste sur l’ordinateur. Nous pensons donc pouvoir gérer le stock de Cevital sur n’importe quel ordinateur portable. Après de nombreux tests, nous nous sommes rendu compte que cela n’était pas évident .Donc on a opté pour l’administration de l’application sous un site web qui sera gérer par l’administrateur depuis un ordinateur. Ainsi une application introduite dans un Smartphone

android sera g rer par le magasinier et l'administrateur dans un quelconque endroit.

4.4.1.1 Depuis un Smartphone Android

– Page Authentication

L'application d bute par le lancement de la fen tre d'authentification qui consiste   la s curisation de notre application pour les acc s non autoris s, une fois l'identifiant et le mot de passe v rifi s, l'utilisateur peut alors acc der   toutes les fonctionnalit s de l'application.

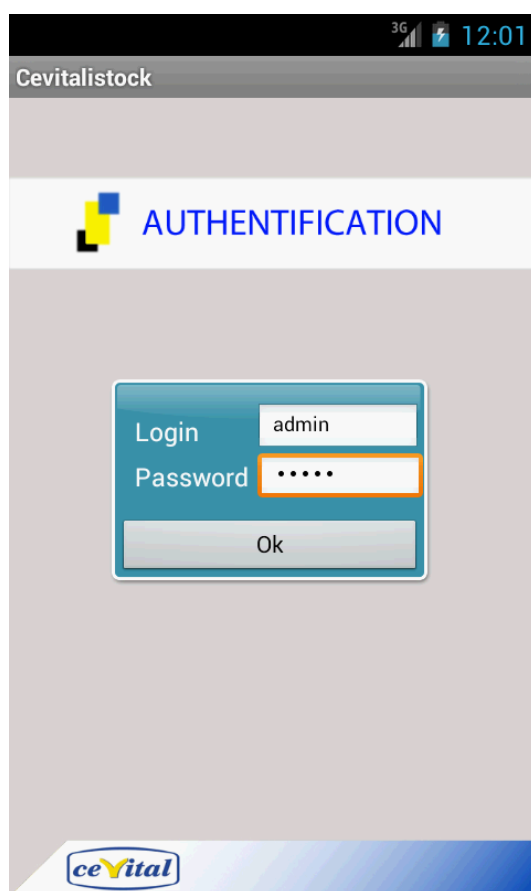


FIGURE 4.13 – Authentication

- **Menu principal** Une fois connecté, l'utilisateur peut accéder à chaque fonctionnalité de l'application grâce aux différents onglets ci -dessous :

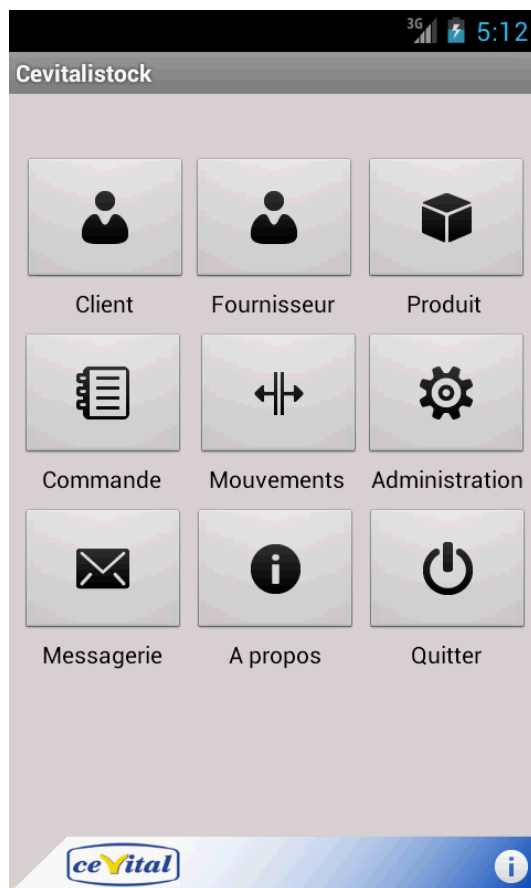


FIGURE 4.14 – Menu principal

- **Gestion des clients** L'onglet " Client" affiche les listes des clients avec leurs coordonnées personnelles. Elle assure bien-évidement la navigation entre les différents onglets (ajout, recherche).

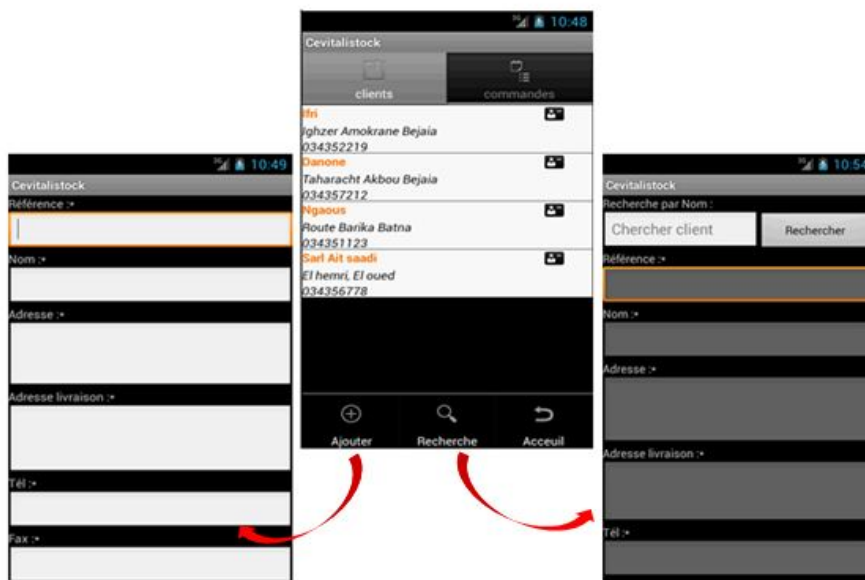


FIGURE 4.15 – Gestion des clients

- **Gestion des fournisseurs** L'onglet "fournisseur " affiche les listes des fournisseurs avec leurs coordonnées personnelles. Il assure bien-évidement la navigation entre les différents onglets (ajout, recherche).



FIGURE 4.16 – Gestion des fournisseurs

- **Gestion des produits** L'onglet " produit " affiche la liste des produits où l'on peut ajouté des produits, affiche la catégorie de chaque produit avec la possibilité d'en ajouter une nouvelle et assure bien-évidemment les alertes.

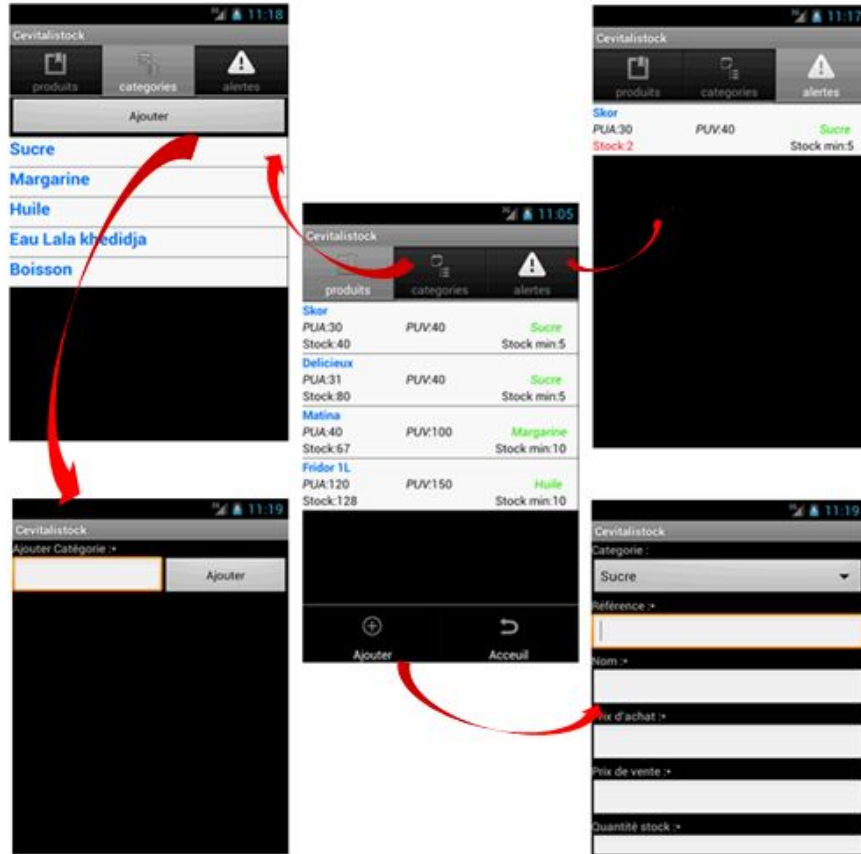


FIGURE 4.17 – Gestion des produits

- **Gestion des commandes clients** L'onglet "commandes des clients" affiche la liste des commandes des clients où l'on peut ajouter une nouvelle commande, on cliquant sur l'onglet détail on pourra voir les différentes caractéristiques de cette nouvelle commande. L'onglet " commandes des fournisseurs " a les mêmes fonctionnalités que l'onglet" commandes des clients précédent "

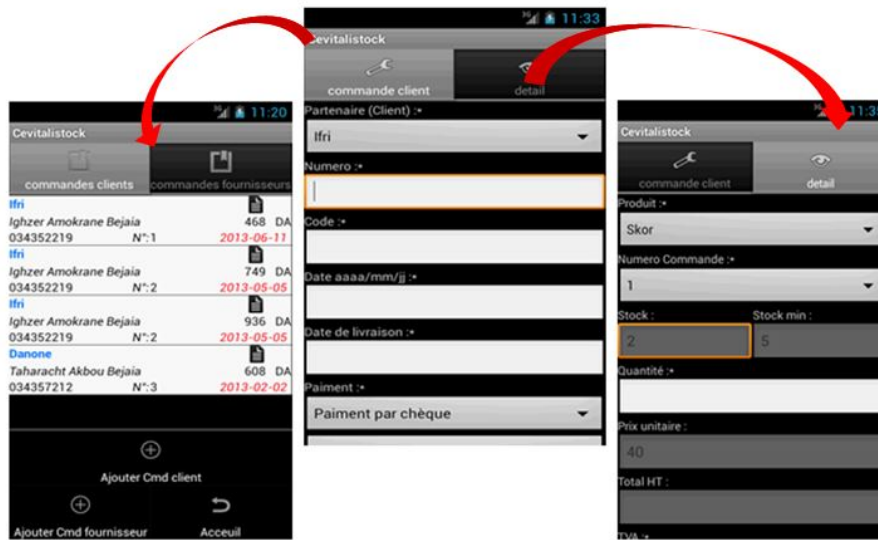


FIGURE 4.18 – Gestion des commandes des clients

- **Gestion des entrées/sorties** L'onglet "entrées" affiche la liste des entrées s où l'on peut ajouter une nouvelle entrées, on cliquant sur Ajouter entrée et on pourra voir les différentes caractéristiques de cette nouvelle entrée. L'onglet " sortie " a les mêmes fonctionnalités que l'onglet" entrée precedement "

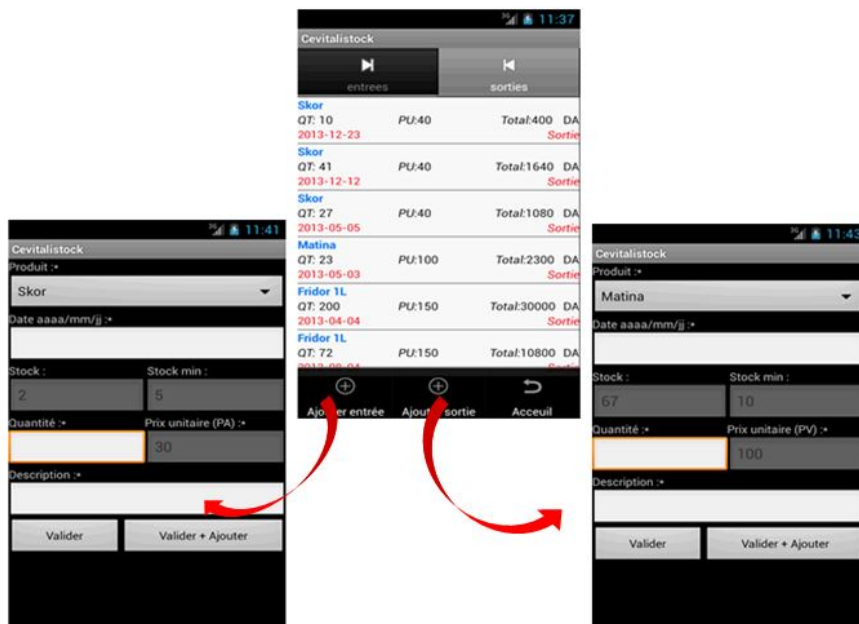


FIGURE 4.19 – Gestion des entrées/sorties

- **Page administration** L'onglet " administration " affiche les coordonnées de l'entreprise ainsi les paramètres

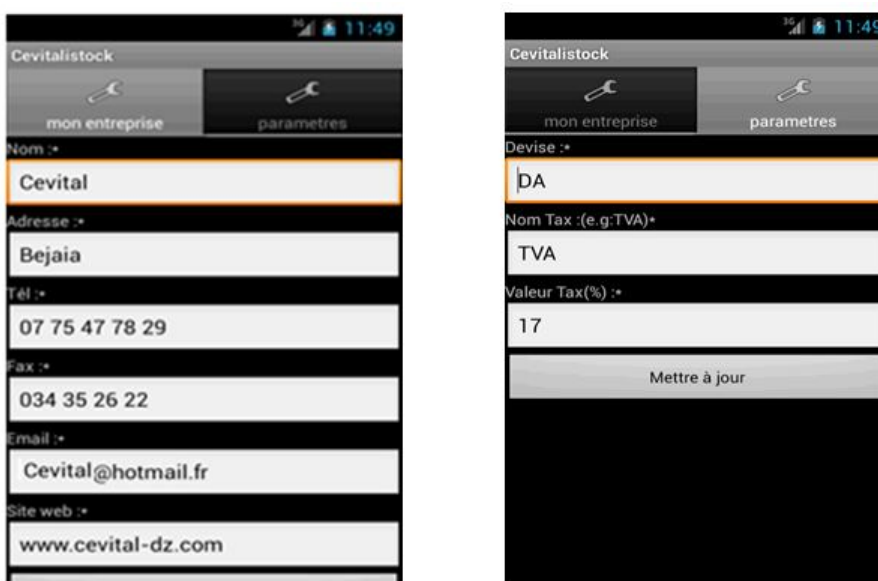


FIGURE 4.20 – Page administration

- **Page de messagerie** L'onglet " messagerie " affiche l'espace pour introduire un message a l'administrateur.

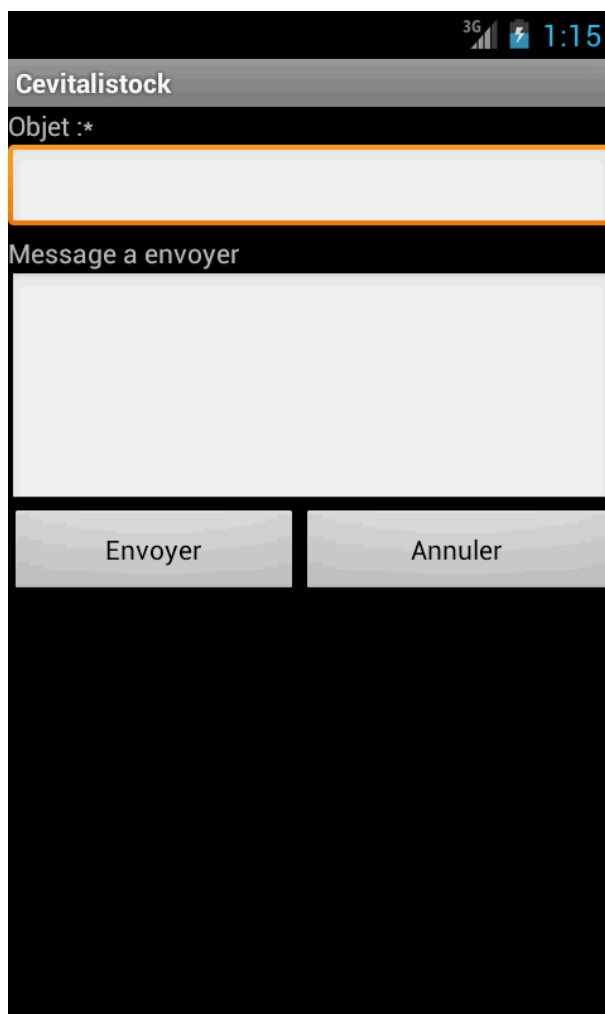


FIGURE 4.21 – Page de messagerie

4.4.1.2 Depuis un ordinateur

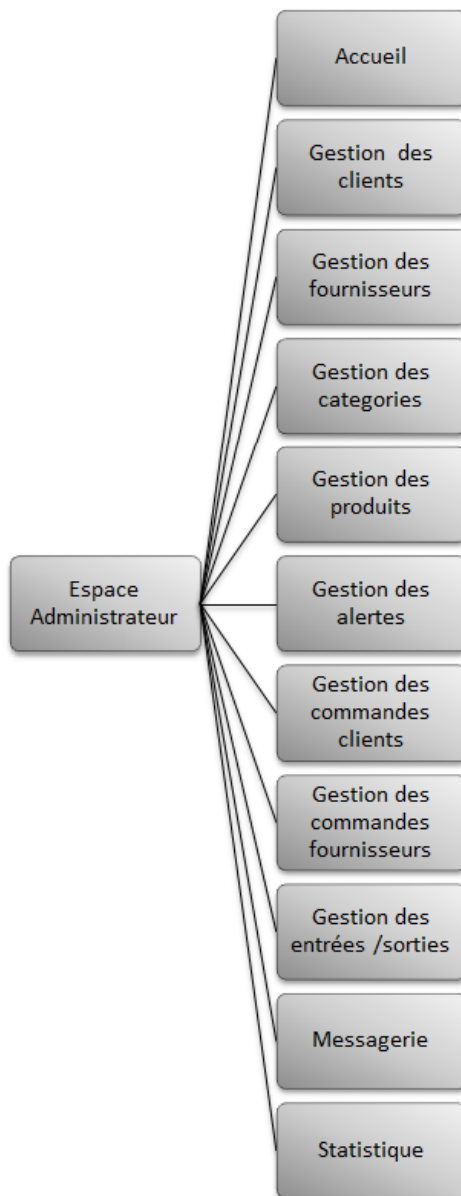


FIGURE 4.22 – Arborescence de l'interface administrateur

L'administrateur accède à notre site web via un ordinateur, les différentes étapes se déroulant dans cette partie sont décrites comme suit :

– Page Authentification

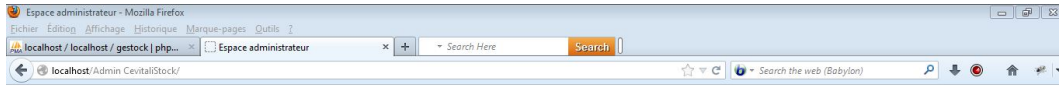


FIGURE 4.23 – Authentification

– **Gestion des clients** Sur cette page, l'administrateur a un aperçu général sur les clients de Cevital, il peut faire une recherche selon la référence, modifier, supprimer ou imprimer sous format PDF.

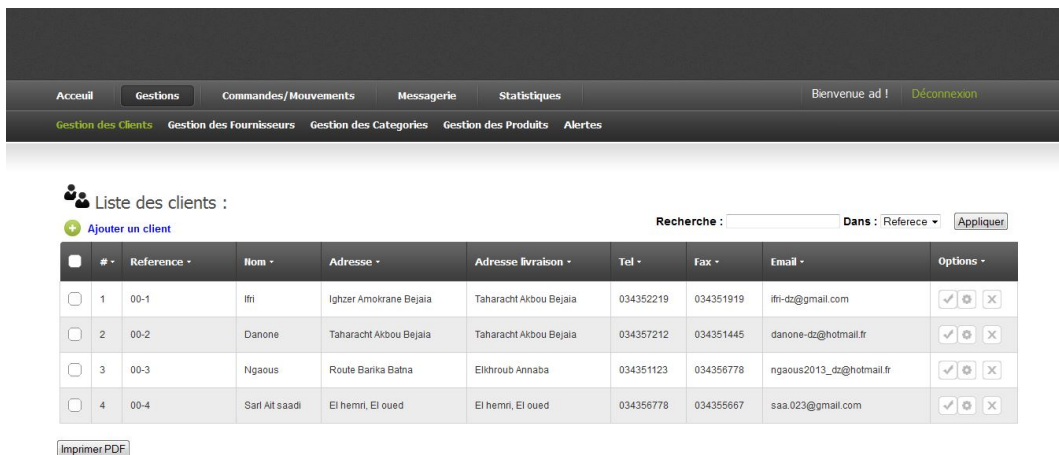


FIGURE 4.24 – Gestion des clients

- **Gestion des fournisseurs** Ici l'administrateur a un aperçu général sur les fournisseurs de Cevital, il peut faire une recherche selon la référence, modifier, supprimer ou imprimer sous format PDF.

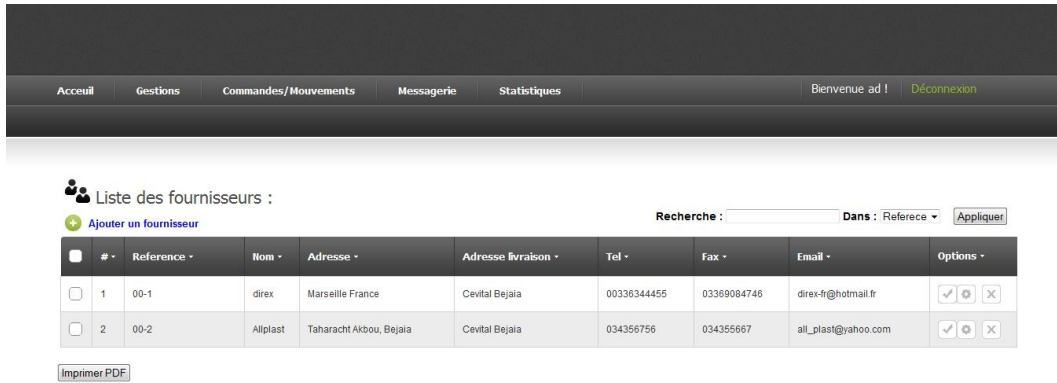


FIGURE 4.25 – Gestion des fournisseurs

- **Gestion des catégories** Sur cette page il y a toute la liste des catégories de produits, l'administrateur peut faire une recherche par nom et aussi supprimer une catégorie.

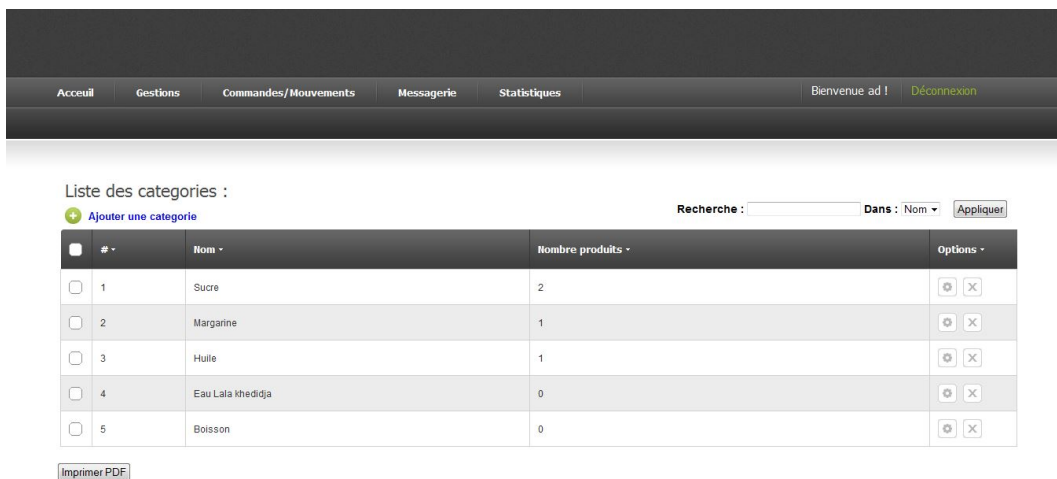


FIGURE 4.26 – Gestion des catégories

- **Gestion des produits** Sur cette page l'administrateur peut ajouter, modifier, supprimer ou faire une recherche trié selon la catégorie.

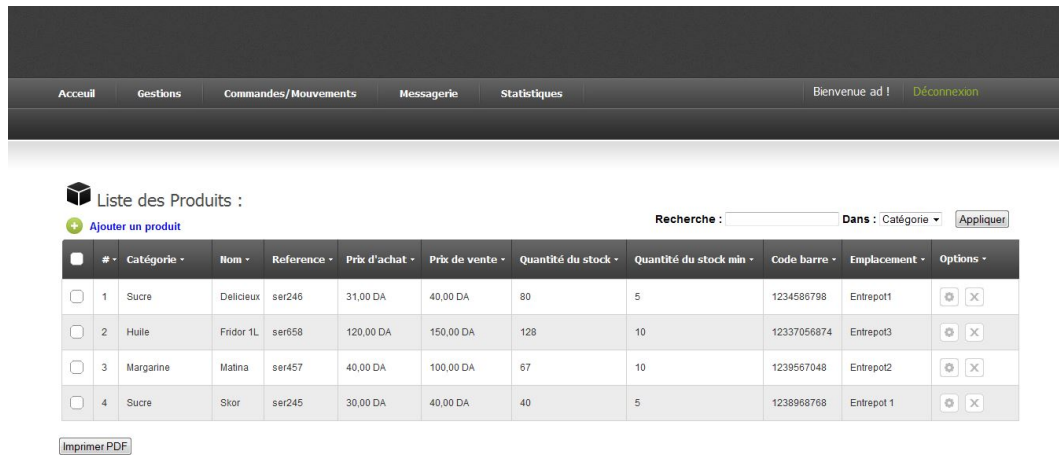


FIGURE 4.27 – Gestion des produits

- **Gestion des entrées** Sur cette page l'administrateur peut ajouter, modifier ou supprimer une entrée.

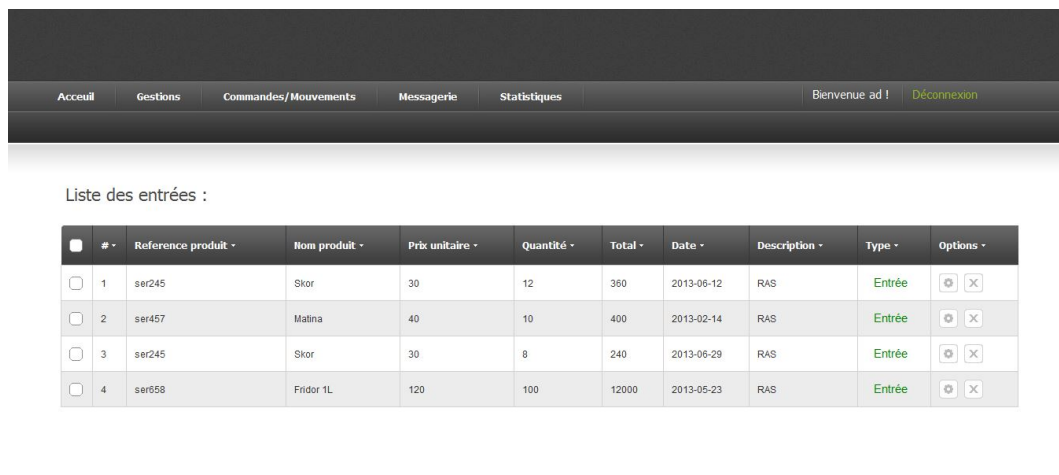


FIGURE 4.28 – gestion des entrées

- **Gestion des sorties** Sur cette page l'administrateur peut ajouter, modifier ou supprimer une sortie.

#	Reference produit	Nom produit	Prix unitaire	Quantité	Total	Date	Description	Type	Options
1	ser245	Skor	40	10	400	2013-12-23	RAS	Sortie	[Icon] [X]
2	ser457	Malina	100	23	2300	2013-05-03	RAS	Sortie	[Icon] [X]
3	ser658	Fridor 1L	150	200	30000	2013-04-04	RAS	Sortie	[Icon] [X]
4	ser658	Fridor 1L	150	72	10800	2013-08-04	RAS	Sortie	[Icon] [X]

FIGURE 4.29 – gestion des sorties

- **Gestion de la messagerie** À partir de cette page l'administrateur peut lire ou supprimer des messages.

#	Lire	Nom	Objet	Date et Heure	Répondre	Options
1	[Icon]	magasinier	salut	10-06-13	[Icon]	[X]
2	[Icon]	magasinier	biba	02-06-13	[Icon]	[X]
3	[Icon]	magasinier	sllt	28-05-13	[Icon]	[X]
4	[Icon]	magasinier	salut	28-05-13	[Icon]	[X]
5	[Icon]	magasinier	hello	26-05-13	[Icon]	[X]

FIGURE 4.30 – gestion de la messagerie

4.5 Chronogramme

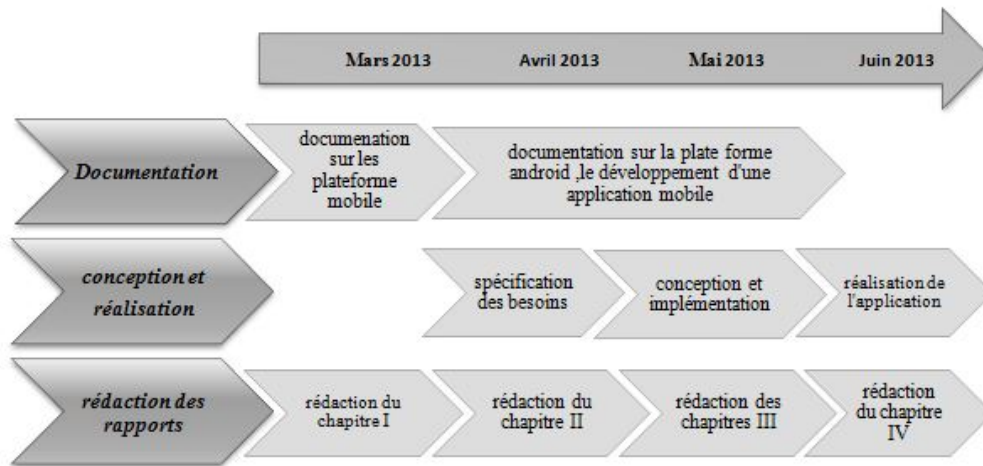


FIGURE 4.31 – Chronogramme

4.6 Conclusion

Dans ce chapitre, nous avons présenté l'environnement de développement matériel et logiciel avec lesquels ce projet a été réalisé. Nous avons présenté aussi une vue de l'application et le site web finale via quelques imprimés d'écrans ainsi que le chronogramme des tâches accomplies durant ce travail.

CONCLUSION ET PERSPECTIVES

Notre projet a consisté en la conception, le développement et la réalisation d'une application de gestion de stock " Cevitalistock " sous la plate-forme Android au sein de l'entreprise Cevital afin d'apporter une valeur ajoutée et un meilleur service aux clients de cette entreprise.

Nous sommes arrivés à développer toutes les fonctionnalités du système dans les temps, l'intégration a été réalisée avec succès, sur ce dans un premier temps, on a utilisé l'outil latex pour la rédaction de ce rapport, MySQL pour la gestion de la base de données et phpMyAdmin pour gérer la base de données sur un serveur PHP, en outre on a utilisé Notepad++ concernant l'édition du code source, Apache pour prendre en charge les nombreux modules lui donnant des fonctionnalités supplémentaires et Photoshop pour le traitement d'image.

Dans un deuxième temps, on a utilisé le Java vu la portabilité des logiciels écrit sous ce langage sur les systèmes d'exploitation notamment android, CSS pour décrire la présentation des documents HTML et XML ainsi que PHP pour produire des pages Web dynamiques.

JSON pour faire communiquer l'application dans un environnement hétérogène, jQuery qui sert à interagir JavaScript avec HTML, enfin on a utilisé wampserver qui nous permet de développer une application Web dynamique.

Ce stage nous a permis d'approfondir nos connaissances théoriques, acquises tout au long de notre formation, par la pratique des nouvelles technologies. Cette expérience nous a permis de maîtriser le langage de modélisation UML, les outils de développement Android à savoir le SDK Android, sous lequel, le développement n'a pas été une tâche facile, mais nous n'avons pas hésité à y participer, malgré le fait qu'il y ait eu peu de supports puisque c'est un langage récent.

Il nous a également permis de découvrir comment se passe l'intégration d'une application sur un serveur web distant ainsi que l'utilisation du langage JSON pour gérer la communication des données entre deux environnements hétérogènes qui sont le client Android et le serveur de bases de données.

Le stage quotidien au sein de la société a aussi été pour nous une occasion unique pour épanouir nos capacités de communication dans un environnement professionnel. C'est une expérience très enrichissante sur tous les domaines.

Par ailleurs, la partie administrateur de l'application développée sous un site web permet de gérer le site de façon dynamique, afin de le mettre à jour, ou de consulter les bases de données pour avoir des informations sur les clients, les fournisseurs etc.

Comme d'autres applications Android, notre application peut être aisément améliorée. En effet, grâce à son aspect ouvert, Android offre l'opportunité de créer des logiciels mobiles innovants et révolutionnaires en encourageant les développeurs à puiser dans leur imagination et à mobiliser toutes leurs compétences pour un meilleur de cette plateforme.

Ouvrages

[L1] CARPENTIER Gilles, COUSTENOBLE Tomas, CROMBE Benjamin, "Solution mobiles avec les logiciels IBM, lotus, DB2, WebSphere, Tivoli et Rational ", Edition DUNOD, Belgique, p.1, Octobre2003.

[L2] CARPENTIER Gilles, COUSTENOBLE Tomas, CROMBE Benjamin, "Solution Mobiles avec les logiciels IBM, lotus, DB2, WebSphere, Tivoli et Rational ", Edition DUNOD, Belgique, p8.9, Octobre2003.

[L3] GUIGNARD Damien ; CHABLE Julien, ROBLES Emmanuel, SOREL Nicolas, COCHODON Vanessa " Programmation Android de la conception aux déploiements avec SDK Google Android " Edition EROLLES, page 3.

L4 JACOSON .I, BOUCH. G, RUMBAUGH.J " Le processus unifié de développement Traduit par ZAIM.V de the unifies software développement process " EYROLLES, IBSN 2-212-09142-7. 2000.

[L5] ROQUES , Pascal " UML 2 par la pratique ", 5 éme Edition EYROLLES, sept 2006.

[L6] GABAYE Joseph, GABAY David " UML2 analyse et conception ". Edition DUNOD, ISBN : 978-2-10-053567-5, 2008.

[L7] GUILBERT Olivier " Analyse et Conception des Systèmes d'Information - Méthodes Objet Le langage de modélisation objet UML " Département Informatique de l'Institut Universitaire de Technologie de l'Université Bordeaux, Page 44, 6/4/10

[L8] LAURIER Boul. " Conception de bases de données avec UML GILLES ROY Presss de l'Université du Québec " Presses de l'Université du Québec, 2009.

[L9] GRAÏNE.S. " UML2 pour une modélisation orientée objet ".Edition l'Abeille, ISBN : 978-9961-723-74-6,2008.

[L10] " Uml2-apprentissage à la pratique (cours et exercices) "FNAC amazon.fr Edition Eclipse 06/05/13.

[L11] GRAÏNE. S " UML2pour modélisation orientée objet et des exercices corrigés " Edition l'Abeille, 2009]

[L12] ETIEVANT Hugo " Cours complet de MySQL " Editions Micro Application ,20 juillet 2003

[L13] MARCEAU Lauranne, BOULANGER Émilie, MARTIN Josée " Adobe, Photoshop CS Notions de base " Carrefour de l'information Université de Sherbrooke, Sherbrooke, Hiver 2009.

[L14] ULMAN Lary " E-commerce sans peine avec PHP et MySQL " first, juin 2011.

[L15] ENGELS .J " XHTML et CSS cours et exercices " Edition EYROLLES, 2006.

[L16] THAHE Serge " Apprentissage du langage JAVA ", ISTIA - Université d'Angers, septembre 98- révision Juin 2002.

[L17] Garnir Henri-Pierre, STRIVAY David, BASTIN Thierry " Le GPS et la physique " Université de Liège, I.P.N.A.S. Belgique, décembre 2002.

Memoires

[M1] NGUYEN Tien Thinh " Système d'exploitation pour les mobiles " Informatique , Rapport final Travail Personne de Institut de la Francophonie pour l'informatique, Juillet 2009.

[M2] Yassine BOULMANE , Imad-Eddine GHAIABI " Développement d'un générateur de code pour les Smartphones mobiles " Génie Logiciel Systèmes Embarqués et Mobiles ,D'Ingénieur d'État en Informatique, 2009-2010.

[M3] MRABET Khaoula , TRABELSI Nessrine " Application de messagerie simple sur Android " :informatique ,Rapport de projet de VAP RSM Projet, : 2010/2011.

[M4] STUCK Philippe " Les architectures 3-tiers, état de l'art " Filière Informatique ,Conception Développement oral probatoire ,conservatoire national des arts et metiers centre regional associe de Strasbourg ,soutenu le 21 novembre 2003.

[M5] CHABCHOUB Halima , ABDELHEDI Rim " Conception et réalisation d'une plateforme d'E-learning " d'ingénieur Spécialité : Informatique, Technologie Web Multimédia, TUNISIENNE, Soutenu le 14 juin 2012.

[M6] MOKHTARI Achraf " Conception et implémentation application sous système android pour télégestion de flotte de véhicules " Projet fin d'étude en Telecom, Année universitaire : 2011/2012.

Sites

- [1] www.tienregion.com/actualites/windows-phone-iphone-android-blackberry/
- [2] <http://www.dicofr.com/cgi-bin/n.pl/dicofr/definition/20010101005515>
- [3] www.xpose.avenir.asso.fr/viewxpose.php?site=17subpage=/index.php?page=conclusion
- [4] <http://www.techno-science.net/?onglet=glossairedefinition=11122>
- [5] <http://www.webopedia.com/TERMA/A/apple-ios.html>

- [6] <http://www.Blog.boulangier.fr/high-tech-multimedia/windoxs-phone-7-ios-android-quelles-diafferneces/>
- [7] <http://www.handimobility.org/blog/handicap-et-apple-ios-5-integre-des-vibrations-personnalisees/>
- [8] <http://www.pro.bouyguetelecom.fr/selection-pro/les-mobiles/univers-du-smartphone/blackberry-os>
- [9] www.ifi.auf.org/site-data/rapports/tpe../tpe-nguyen-tien-thinh.pdf
- [10] <http://www.franceyou.blogspot.com/2013/01/avantages-et-inconvenients-de.html>
- [11] <http://www.memoireonline.com>
- [12] <http://www-igm.univ-mlv.fr/dr/XPOSE2008/android/>
- [13] <http://www.openhandsetalliance.com/>
- [14] <http://www.ibm.com/developerworks/opensource/library/x-android>
- [15] <http://socialcompare.com/fr/comparison/android-versions-comparison>
- [16] <http://www.android-help.fr/article/les-versions-Android>
- [17] <http://www.assistance.orange.fr/A-faq/3510.pdf>
- [18] <http://www.frandroid.com/actualites-generales/15901-android-2-2-froyo-cest-pour-bientot/>
- [19] <http://developer.android.com/about/dashboards/index.html>
- [20] <http://www.androideur.com/architecture-et-caracteristiques-du-os-android>
- [21] <http://www-igm.univ-mlv.fr/dr/XPOSE2008/android/>
- [22] <http://manu.kegtux.org/Java/Tutoriels/Android/android.html>
- [23] <http://www.slideshare.net/MohamedAchrafKarmous/cours-1-android>
- [24] <http://fr.slideshare.net/Android2EE/android-a-quick-course-fr-part-i-by-android2ee-mathias-seguy-la-devoux-france-2012>
- [25] <http://manu.kegtux.org/Java/Tutoriels/Android/android.html>
- [26] <http://book.cakephp.org/1.3/fr/The-Manual/Beginning-With-CakePHP/Understanding-Model-View-Controller.html>
- [27] <http://julien-pauli.developpez.com/tutoriels/php/mvc>
- [28] <http://www.MySQL.com>
- [29] <http://www.apache.org>
- [30] <http://manu.kegtux.org/Java/Tutoriels/Android/android.html>
- [31] <http://ebookbrowse.com/tutoriel-wampserver-pdf-d86101764>
- [32] <http://www.ibm.com/developerworks/opensource/library/x-android>
- [33] <http://manu.kegtux.org/Java/Tutoriels/Android/android.html>
- [34] <http://www.lenumeriquefacile.com/definition/gsm>
- [35] <http://www.lirmm.fr/ajm/Cours/01-02/DESS-TNI/TER13/dochttp.html>
- [36] <http://www.w3.org/Protocols/>

[37] <https://www.vpnblog.net/definition-explication-vpn/>

.....A.....

Apache

Apache est le serveur libre le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais qui a désormais été porté sur d'autres systèmes, dont Microsoft Windows. Grâce à sa convivialité et sa facilité d'utilisation, il est devenu l'un des piliers du succès du trio Apache/MySQL/PHP de développement Web. [29]

.....C.....

CSS

Les feuilles de styles CSS(Cascading Style Sheets) sont l'un des gages de qualité de toute conception web moderne. On peut grâce au CSS créer des sites élégants et légers, conformes aux standards du Web et accessibles à tous. Il a été mis au point en complément au langage XHTML concernant la mise en page et la présentation [L15]

.....D.....

Dalvik

est le nom de la machine virtuelle open-source utilisée sur les systèmes Android. Cette machine virtuelle exécute des fichiers .dex, plus ramassés que les .class classiques. Ce format évite par exemple la duplication des String constantes. La machine virtuelle utilise elle-même moins d'espace mémoire et l'adressage des constantes se fait par un pointeur de 32 bits. [33]

.....G.....

GSM

Global System for Mobile Communications. Il présente un fonctionnement cellulaire : au cur

de chaque cellule, qui désigne une zone géographique plus ou moins étendue, il y a une antenne relais qui permet de faire le lien entre les téléphones mobiles et le réseau. Le réseau permet de passer d'une cellule à une autre sans interrompre la communication et sans que l'utilisateur ne s'en rende compte. [34]

GPS

global positioning system cest un Système de positionnement par satellite. Lié à des données cartographiques, cette technologie permet notamment des calculs ditinéraires [L17]

.....H.....

HTTPS

C'est le protocole HTTP lié à une connexion sécurisée par SSL (Secure Socket Layer). Sa principale utilisation est le commerce électronique. Son rôle est de crypter la communication entre le client et le serveur. Pour cela, il est nécessaire d'établir une session SSL avant la connexion HTTP [35]

HTTP

(HyperText Transfer Protocol) ou protocole de transfert d'hyper-texte est un protocole qui permet de transmettre nimporte quel type de fichier ou de donnée. Il a été pensé afin de pouvoir être utilisé dans de futures applications orientées objet. [37]

.....M.....

MySQL

(My Structured Query Language) est un système de gestion de base de données. Distribué sous une double licence GPL et propriétaire .Il fait partie des logiciels de gestion de base de données les plus utilisé dans le monde. [28]

.....N.....

Notepad++

C'est un éditeur de texte libre générique qui intègre la coloration syntaxique. Il édite les codes sources et supporte plusieurs langages (PHP, ASP, JavaScript, HTML, XML).Développé en C++, il demeure un éditeur de code source petit par sa taille (taille binaire) et efficace par sa performance d'exécution. Il constitue une alternative au traditionnel « Bloc-notes » qui ne comporte pas la coloration syntaxique. Il offre contrairement au Bloc-notes la possibilité de définir un langage, l'auto complétion, le multi document (possibilité d'éditer plusieurs documents à la fois grâce à des onglets). [11]

.....P.....

PhpMyAdmin

PhpMyAdmin est développé en PHP et offre une interface intuitive pour l'administration des bases de données de MySQL sur un serveur web, il permet la création de base, l'affichage, l'ajout, la suppression des tuples dans des tables, gérer les privilèges des utilisateurs, l'exécution de requêtes...Etc. [L12][30]

Photoshop

C'est un programme d'édition graphique, il permet d'effectuer un nombre impressionnant de retouches sur images. La mise à l'échelle intelligente par exemple, permet d'étendre ou de rétrécir une photo dans de multiples sens sans qu'il n'y ait pour autant de déformation de celle-ci sur le résultat final [L13].

PHP

C'est un langage appartient à la grande famille des descendants du C, dont la syntaxe est très proche du c, Dans une utilisation Web, l'exécution du code PHP se déroule ainsi lorsqu'un visiteur demande la consultation d'une page Web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension .PHP), le serveur appelle l'interpréteur PHP qui va traiter et générer le code final de la page. Ce contenu est renvoyé au serveur HTTP, qui l'envoie ensuite au client. [L14]

Push mail

Technologie : née dans les années 90, initiée et popularisée par BlackBerry, la technologie du push mail est adoptée par un maximum de smartphones. Le principe est que la messagerie électronique s'actualise à chaque nouveau message reçu.[m6]

.....V.....

VPN

Crée une connexion sécurisée entre votre ordinateur et le serveur VPN. Les données sortant de votre ordinateur étant cryptées, elles deviennent quasi impossibles à pirater. Cette sécurité, ce cryptage est valable pour une connexion filaire ou Wifi et empêche toute malveillance ou surveillance. [38]

.....W.....

WampServer

Wampserver est une plate-forme de développement web sous Windows. Il nous permet de développer des applications web dynamiques grâce à son environnement qui comprend deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi qu'une administration pour les deux bases SQL PhpMyAdmin et SQLiteManager. [31]

XML

Extensible Markup Langage est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode en champs arborescents. [32] En Android, grâce à ce langage, nous décrivons les interfaces dans un format spécial, et Android le converti automatiquement en objets Java qui seront par la suite disponibles comme tout autre objet dans notre code. Il offre ainsi plus de souplesse de développement, facilite les modifications du code et assure la séparation entre la présentation et le comportement des objets.

Résumé

Aujourd'hui, l'informatique mobile a atteint une prodigieuse évolution technologique dans différents domaines (réseaux informatiques, bases de données, le Web, etc.). Cette évolution est nécessaire pour remédier aux problèmes rencontrés dans la vie actuelle. Actuellement Smartphones, PDA et autres terminaux mobiles connaissent une véritable explosion. Dans ce contexte, Android, le système d'exploitation par Google, présente le double avantage d'être gratuit et open-source. Libre donc à tout un chacun d'en exploiter l'énorme potentiel !

Notre travail consiste à concevoir et de réaliser une application pour le système Android permettant la gestion du stock au sien de Cevital Le but de ce projet était donc dans un premier temps de découvrir la plateforme ANDROID et les outils qui gravitent autour (émulateur, machine virtuelle, etc.). Une fois familiarisé avec l'environnement, nous devons prendre en main les concepts du développement pour appareils mobiles et de manière plus précise comprendre le fonctionnement des applications destinés à Android. Cette solution facilitera et augmentera les possibilités de travail pour le personnel de Cevital.

Mots-clés : MySQL, SGBD, UML, UP, Java, Web, Android.

Abstract

Today, mobile computing has achieved tremendous technological developments in various fields (computer networks, databases, web, etc.). This is necessary to address the problems in the present life. Currently Smartphones, PDAs and other mobile devices experiencing an explosion. In this context, Android operating system by Google, has the double advantage of being free and open-source. Therefore free to everyone to exploit the huge potential !

Our job is to design and build an application for the Android system for the management of the stock in Cevital. The purpose of this project was to initially discover the ANDROID platform and tools that revolve around (emulator, virtual machine, etc..). Once familiar with the environment, we had to take over the development concepts for mobile devices and a more precise understanding of how applications for Android. This solution will facilitate and increase work opportunities for Cevital staff.

Keywords : MySQL, SGBD, UML, UP, Java, Web, Android.