

République Algérienne Démocratique et Populaire.

Université A. Mira- Béjaïa

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de fin de Cycle

En vue de l'obtention du diplôme de Master 2 en Informatique

Option : Administration et sécurité des réseaux

Thème :

Conception et Implémentation d'une Application Mobile pour les Services d'Aide aux Etudiants sous Android.

Cas : Service Scolarité et service Bibliothèque de l'université

A.Mira (SSB-UB) .

Présenté par :

M^{elle} BECHAR Anissa et M^r SADELLI Salim.

Soutenu le 22\06\2013 devant le jury composé de :

Présidente	M ^{me} H.KHALED	M.A.B	U. A/Mira Béjaïa.
Promotrice	M ^{me} K.GHIDOUCHE	M.A.B	U. A/Mira Béjaïa.
Co-Promotrice	M ^{me} M.YAICI	M.A.A	U. A/Mira Béjaïa.
Examinatrice	M ^{me} N.BATAT	M.A.A	U. A/Mira Béjaïa.
Examineur	M ^r N.DJEBARI	M.A.B	U. A/Mira Béjaïa.

Béjaïa, juin 2013.

Remerciements

Avant tout, merci DIEU de nous avoir donné le courage, la volonté et la patience de mener ce travail à terme.

Ensuite, il n'aurait pas pu être achevé sans le soutien, les conseils et l'aide de certaines personnes auxquelles on tient ici à exprimer nos sincères remerciements.

Nos sincères remerciements vont à M^{me} K.GHIDOUCHE et M^{me} M.YAICI

pour leurs encadrement avec patience. Leur encouragement et leurs remarques pertinentes nous ont permis de mieux structurer ce travail et de mieux le décrire. Nous les remercions aussi de nous avoir fait profiter de leurs expériences, leurs orientations et leurs conseils nous ont énormément aidé. L'aboutissement de celui-ci doit beaucoup à leur contribution.

Nous remercions M^{me} H.KHALED , pour l'honneur qu'elle nous a fait en présidant notre jury de mémoire.

Nous remercions tout particulièrement M^{me} N.BATAT et M^r N.DJEBARI d'avoir accepté d'examiner notre mémoire.

Nos sincères remerciements s'adressent à nos parents, nos frères , nos sœurs ainsi qu'à toute la famille pour leurs soutien moral, leur encouragement inconditionnel, et surtout pour la confiance qu'ils nous accordent.

Enfin, Nous remercions tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail, en particulier tous nos ami(e)s pour leurs soutien moral et leurs présence à nos cotés.

Dédicaces

Louange A Dieu, le miséricordieux, sans Lui rien de tout cela n'aurait pu être.

Je dédie ce modeste travail :

En hommage à mon regretté père, que dieu l'accueille en son vaste paradis.

A Ma très chère maman : Je ne pourrai jamais assez te remercier ,

Pour ton soutien inconditionnel ,

Pour ta présence dans toutes les étapes de ma vie ,

Pour tout les sacrifices que tu ne cesse de faire pour moi,

J'espère que tu verra en ce travail le fruit des efforts et sacrifices,

Que tu m'as consentis pour mon éducation,

Que Dieu m'accorde la grâce de t'avoir encore à mes côtés pour,

longtemps afin que tu sois témoins de ma reconnaissance,

A Mon très cher frère et sa femme et mes adorés sœurs ainsi qu'à ma tante Nora et mes neveux ,

A Tous mes amis qui m'ont aidé, encouragé, soutenu et on été à mes cotés jusqu'au bout ,

A Tous ceux que j'aime et je respecte.

BECHAR Anissa

Dédicaces

Louange A Dieu, le miséricordieux, sans Lui rien de tout cela n'aurait pu être.

Je dédie ce modeste travail :

A Mes tres chérs parents,

Je vous doit ce que je suis aujourd'hui,

Grace à votre patience et vos innombrables sacrifices,

Que ce medeste travail soit pour vous une petite reconnaissance envers ce que vous avez fait pour moi.

A Toute ma famille,

A Tout mes amis,

SADELLI Salim

Table des matières

Table des Matières	i
Table des Figures	vii
Liste des Tableaux	viii
Liste des Abréviations	ix
1 Les Applications Mobiles	3
1.1 Introduction	3
1.2 l'Informatique mobile	4
1.2.1 Définition de l'informatique mobile	4
1.2.2 Les terminaux mobiles	4
1.2.2.1 Définition des terminaux mobiles	4
1.2.2.2 Les différents types de terminaux mobiles	5
1.2.3 Evolution de l'informatique mobile	7
1.3 Les applications mobiles	10
1.3.1 Présentation des applications mobiles	10
1.3.1.1 Histoire des applications mobiles	10
1.3.1.2 Architecture des applications mobiles	10
1.3.1.3 Fonctionnement des applications mobiles	13
1.3.1.4 Différents types d'applications mobiles	18
1.3.1.5 Les systèmes d'exploitations mobiles	19

1.3.1.6	Avantages et Inconvénients des applications mobiles . . .	22
1.4	Conclusion	24
2	Le système Android	25
2.1	Introduction	25
2.2	Le système d'exploitation Android	25
2.2.1	Présentation d'Android	25
2.2.2	Fonctionnalités d'Android	26
2.2.3	Evolution des versions d'Android	27
2.2.4	Architecture d'Android	29
2.2.5	Cycle de vie d'une application Android	33
2.3	Conclusion	35
3	Présentation du systeme et Conception	36
3.1	Introduction	36
I	Présentation du système	37
3.2	Introduction	38
3.3	Objectifs de l'application mobile SSB-UB	38
3.4	Architecture de l'application mobile SSB-UB	43
3.5	Apport du système mobile par rapport au système classique des deux services scolarité et bibliothèque	44
3.6	Conclusion	46
II	Conception	47
3.7	Introduction	48
3.8	Le Processus Unifié	48
3.8.1	Définition d'un processus de développement logiciel	48
3.8.2	Définition du processus unifié (UP)	48
3.9	Analyse et spécification des besoins	49
3.9.1	Description de l'application SSB-UB	49

3.9.2	Les besoins Fonctionnels	50
3.9.2.1	Le service " Scolarité "	50
3.9.2.2	Le service " Bibliothèque "	51
3.9.3	Les besoins non fonctionnels	52
3.10	Conception	52
3.10.1	Présentation d'UML	52
3.10.2	Présentation des Diagrammes UML	53
3.10.2.1	Définition du diagramme de cas d'utilisation	54
3.10.2.2	Définition du diagramme de séquence	54
3.10.2.3	Définition du diagramme de classe	55
3.10.3	Analyse et conception du service " Scolarité "	56
3.10.3.1	Les cas d'utilisation du service scolarité	56
3.10.3.2	Les diagrammes de séquence du service scolarité	75
3.10.3.3	Le diagramme de classe du service Scolarité	81
3.10.4	Analyse et conception du service " Bibliothèque "	84
3.10.4.1	Les cas d'utilisation du service scolarité	84
3.10.4.2	Les diagrammes de séquence du service bibliothèque	101
3.10.4.3	Le diagramme de classe du service Bibliothèque	108
3.11	Conclusion	112
4	Implémentation	113
4.1	Introduction	113
4.2	Environnement de développement de l'application	113
4.2.1	Environnement logiciel	113
4.2.2	Outils Utilisées	114
4.2.3	Les langages de programmation	117
4.3	Architecture materielle de l'application SSB-UB	118
4.4	Implémentation de la Base de Données	119
4.5	Arborescence de l'application SSB-UB	121
4.6	Interfaces graphiques de l'application SSB-UB	122

4.6.1	Interfaces utilisateur	122
4.6.2	Interfaces Administrateur	134
4.6.2.1	Interface du site web pour l'administrateur de la scolarité	134
4.6.2.2	Interface de l'application java pour l'administrateur du service bibliothèque	135
4.7	Tests	135
4.7.1	Test de téléchargement	136
4.7.2	Test de réservation	137
4.8	Conclusion	138
	Bibliographie	141
	Glossaire	144
	Annexe	149

Table des figures

1.1	Vue générale du Cloud computing.	9
1.2	Architecture d'une application mobile.	12
1.3	Fonctionnement d'une application mobile en mode Connecté.	14
1.4	Fonctionnement d'une application mobile en mode Déconnecté.	16
2.1	Architecture d'Android.	33
2.2	Organigramme du Cycle de vie d'une application Android.	35
3.1	Architecture globale de l'application mobile SSB-UB	44
3.2	La liste des diagrammes UML 2.0.	53
3.3	Diagramme de cas d'utilisation générale du service scolarité.	57
3.4	Diagramme de cas d'utilisation 'S'authentifier ' pour l'administrateur.	60
3.5	Package du cas d'utilisation 'Gestion des Etudiants'.	62
3.6	Package du cas d'utilisation 'Gestion des Publications'.	64
3.7	Diagramme de cas d'utilisation 'S'authentifier ' pour l'étudiant.	66
3.8	Diagramme de cas d'utilisation 'Rechercher une ressource '.	68
3.9	Diagramme de cas d'utilisation 'Envoyer/Recevoir un E-mail '.	72
3.10	Diagramme de séquence ' Authentification '.	75
3.11	Diagramme de séquence ' Rechercher une ressource '.	76
3.12	Diagramme de séquence ' Consulter une ressource '.	77
3.13	Diagramme de séquence ' Envoyer/Consulter E-mail '.	78
3.14	Diagramme de séquence ' Télécharger une ressource '.	79

3.15	Diagramme de séquence 'Consulter une information '	80
3.16	Diagramme de séquence 'Envoyer une Réclamation '	81
3.17	Diagramme de classe du service scolarité.	82
3.18	Diagramme de cas d'utilisation générale du service bibliothèque.	84
3.19	Package du cas d'utilisation 'Gérer emprunt'.	89
3.20	Package du cas d'utilisation 'Gérer Ouvrages '	91
3.21	Diagramme de cas d'utilisation 'Rechercher un ouvrage'	95
3.22	Diagramme de cas d'utilisation 'Renouveler un Emprunt'.	97
3.23	Diagramme de cas d'utilisation 'Noter un Ouvrage '	101
3.24	Diagramme de séquence 'Réservation d'un ouvrage '	102
3.25	Diagramme de séquence 'Renouveau d'un emprunt'.	103
3.26	Diagramme de séquence 'Suggestion d'ouvrages'.	104
3.27	Diagramme de séquence 'Noter un ouvrage '	105
3.28	Diagramme de séquence ' Ajouter Adhèrent '	106
3.29	Diagramme de séquence ' Mettre à jour Adhèrent '	107
3.30	Diagramme de séquence ' Supprimer Adhèrent '	108
3.31	Diagramme de classe du service bibliothèque.	109
4.1	Architecture générale d'un pilote JDBC	116
4.2	Architecture matérielle du système.	119
4.3	Schéma de la base de données général.	120
4.4	la base de données de l'administrateur.	121
4.5	la base de données de l'adhérent.	121
4.6	Arborescence générale de l'applicationSSB-UB.	122
4.7	Interface de l'accueil.	123
4.8	Interface de l'authentification.	124
4.9	Interface des services principaux.	125
4.10	Interface de la liste des services de la scolarité.	126
4.11	Interface des services de recherche de ressources.	127
4.12	Interface du service téléchargement de la ressource	128

4.13	Interface du service réclamation	129
4.14	Interface liste des services de la bibliothèque	130
4.15	Interface de réservation d'ouvrages.	131
4.16	Interface renouvellement d'emprunt.	132
4.17	Interface suggerer ouvrage	133
4.18	Interface de l'ajout d'un étudiant.	134
4.19	Interface de l'ajout d'un adhérent.	135
4.20	Test de téléchargement.	137
4.21	Test de réservation.	138

Liste des tableaux

1.1	Histoire des Smartphones.	6
2.1	Evolution des versions d'Android.	29
2.2	Exemples de Patches	30
3.1	Tableau illustrant l'apport du service scolarité mobile par rapport au service scolarité classique.	45
3.2	Tableau illustrant l'apport du service bibliothèque mobile par rapport au service bibliothèque classique.	46

Liste des Abréviations

AIR	Adobe I ntegrated R untime
API	Application P rogramming I nterface
AVD	Android V irtual D evice
BLOB	Binary L arge O bject
CRM	Customer R elationship M anagement
EDGE	Enhanced D ata rates for G sm E volution
EDI	Environnement de D éveloppement I ntégré
HTC	High T ech C omputer
HTTP	Hyper T ext T ransfer P rotocol
IHM	Interface H omme M achine
JDBC	Java D ata B ase C onnectivity
JDK	Java D eveloppement K it
JSON	Java S cript O bject N otation
JSP	Java S ervlet P rogramming
MMS	Multimedia M essaging S ervice
RSS	Really S imple S yndication
SDK	Software D eveloppement K it
SGBD	Système de G estion de B ases de D onnées
UML	Unified M odeling L anguage
WML	Wireless M ark-up L anguage
XML	e X tensible M arkup L anguage

Introduction Générale

Depuis la fin du XIXe siècle, les hommes ont pris l'habitude de communiquer rapidement et à grande distance, d'abord grâce au télégraphe puis au téléphone et désormais via Internet. De nos jours, l'informatique et les télécommunications occupent une place prépondérante dans notre vie, qu'elle soit professionnelle ou privée.

À une époque où communication et technologie sont les maîtres mots de notre monde, on ne peut douter que l'avenir de l'informatique soit de grandir et de se développer et plus précisément l'informatique mobile qui seul à lui est un moyen de communication important qui occupe une large place dans le domaine d'information et de la télécommunication mobile.

La télécommunication mobile qui jadis se limitait à l'envoi de messages texte ou à de simples appels téléphonique est entrée dans une phase de maturité où applications mobiles et terminaux mobiles sont les principaux acteurs et cela ne cesse d'évoluer vers une nouvelle ère de plus en plus dématérialisé qu'est le Cloud computing.

Ainsi, ayant vue l'importance des applications mobiles au sein de la télécommunication mobile et dans le cadre de notre préparation au diplôme de Master 2 professionnel en Administration et Sécurité des Réseaux, nous avons été amenés à développer une application mobile pour l'université A. Mira ; qui servira à aider les étudiants à avoir accès à quelques services de l'université d'une manière plus rapide et plus pratique.

Notre travail consiste en la conception et l'implémentation d'une application mobile sous Android qui facilitera aux étudiants l'accès aux services les plus importants de l'université de Bejaia à savoir : le service scolarité et le service bibliothèque (SSB-UB). Ses deux services mobiles répondront ainsi à plusieurs demandes des étudiants sans qu'il

ai a se dépalcer .

Pour la réalisation de ce projet nous avons choisi de suivre la démarche de développement logiciel, " Processus Unifié (UP) ", une méthode adaptée à tous types de projets.

Ce mémoire est composé de quatre chapitres qui sont comme suit

Le premier chapitre " Applications Mobiles " consiste à définir l'informatique mobile, les terminaux mobiles puis les applications mobiles ainsi que leur architecture et fonctionnement et nous termineront par leurs avantages et inconvénients.

Dans Le deuxième chapitre " Le système Android ", nous définissons l'OS Android, présenterons ses fonctionnalités, l'évolution de ses versions, son architecture ainsi que le cycle de vie d'une application Android.

Concernant le troisième chapitre " Présentation du Système et Conception ", nous l'avons divisé en deux parties ; la partie " Présentation du Système " ou nous détailleront les objectifs de notre projet, les services offerts par notre application mobile ainsi que son architecture globale. Et la deuxième partie " Conception ", ou nous allons présenter le processus unifié, ensuite nous reviendrons sur la présentation de l'UML dont les diagrammes seront utilisés dans chaque phase et activité du processus unifié.

Le quatrième chapitre " Implémentation " contient l'environnement de développement de notre application,,l'architecture materiel du systeme et les différentes interfaces et pour finir, nous présentons quelques exemples de testes.

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

Les Applications Mobiles

1.1 Introduction

Depuis quelques années, le marché de la téléphonie mobile est en plein essor et une part de plus en plus importante de la population possède un téléphone ou une tablette ayant la capacité de calcul d'un ordinateur. De cela, l'accès à internet a connu un changement majeur, de sorte que les mobiles sont devenus les principaux points d'accès pour l'utilisation d'internet, et grâce à l'évolution de la technologie, les mobiles ne sont pas utilisés juste pour appeler ou jouer à des jeux, mais avec les Smartphones par exemple nous pouvons planifier notre journée complète, consulter nos mails, faire des conférences téléphoniques, se connecter en utilisant un réseau social et effectuer une foule d'autres activités ; tous cela en utilisant ce qu'on appelle ' les applications mobile '. Les applications mobiles ont un impact primordial sur notre quotidien, elles nous permettent par exemple de réduire la charge de travail, substituer le format papier en format électronique, diminuer les coûts téléphoniques et nous offre d'innombrables autres services.

Dans ce chapitre, nous commençons par définir l'informatique mobile suivi des terminaux mobiles et de l'évolution de l'informatique mobile puis nous allons nous concentrer sur les applications mobiles, donner leurs différents types ainsi que leur architecture, leur fonctionnement et rajouter quelques exemples sur les Systèmes d'exploitation mobiles, et nous concluons avec leurs principaux avantages et inconvénients .

1.2 l'Informatique mobile

1.2.1 Définition de l'informatique mobile

L'informatique mobile a fait son apparition à partir de 1992. Elle s'inscrit dans une approche plus globale qui constitue l'informatique ubiquitaire qui est appelée également informatique pervasive. Elle a pour but de rendre accessible toutes sortes de services, n'importe où, tout en masquant l'ordinateur. Ainsi, l'ubiquité permet donc souvent la mobilité.

L'informatique mobile traite de l'ensemble des solutions informatiques sur des plateformes comme les téléphones portables, les Smartphones etc.

En situation de nomadisme, l'informatique mobile permet à un utilisateur de conserver une partie de ces outils numériques tout en lui apportant de nouveaux services. Elle lui permet aussi d'interagir avec son environnement via de nombreux canaux comme les technologies sans-fil.

A l'origine, l'informatique mobile était constitué essentiellement des Pc portables et avait pour but principal de permettre aux commerciaux de présenter des produits, faire des démonstrations, se connecter à une messagerie interne, ou encore de remonter certaines données à une application spécifique au travers d'un équipement composé d'un modem et d'une connexion filaire au réseau téléphonique.

Ainsi, les technologies de l'informatique mobile ont évolué en même temps que sont apparues de nouvelles catégories de collaborateurs nomades (tels que les techniciens de maintenance).

L'informatique mobile est entrée dans une phase de maturité, après plusieurs années, grâce à l'évolution des produits permettant d'accéder à distance à internet : ordinateurs de poche communicants (Pocket pc), téléphone mobiles multifonctions, tablettes tactiles dotés de connexions sans fil (Wifi ou Bluetooth), Assistants personnel (PDA), Smartphones, etc. Tous ces équipements dédiés à la mobilité informatique ont envahi notre quotidien. [29]

1.2.2 Les terminaux mobiles

1.2.2.1 Définition des terminaux mobiles

Les terminaux mobiles se sont grandement améliorés au cours des dernières années ; leurs caractéristiques les distinguent néanmoins des ordinateurs portables classiques par un grand nombre de spécificités uniques malgré leur diversité, on peut citer les plus importantes : la capacité de stockage est limitée, le processeur est faible, les systèmes

d'exploitations sont variés (Android, iOS, Symbian...), les modes de communication sans fil sont multiples (Wifi, GPS, GPRS, etc.).

Bien qu'ils entrent pleinement dans les champs de l'informatique mobile, les ordinateurs portables possèdent un certain nombre de caractéristiques spécifiques qui les rendent fondamentalement différents des autres types terminaux mobiles. On peut citer par exemple : leur poids qui est élevé, leur grande capacité de stockage, la puissance du processeur qui est élevée, etc.[29]

1.2.2.2 Les différents types de terminaux mobiles

Il est possible de regrouper les terminaux mobiles en fonction de leur ergonomie et de leur usage. Nous en avons choisis les plus utilisés qu'on détaille ci-dessous :

1. Les Smartphones



– *Définition des Smartphones*

Smartphone ou " téléphone intelligent " est un téléphone mobile qui possède des fonctions Proches à celles d'un assistant numérique personnel grâce à un système d'exploitation, possédant des fonctionnalités précises comme l'accès à Internet, la bureautique, la gestion de carnet d'adresses, le multimédia (MP3, vidéo, photos,...) , les jeux et sont mêmes équipés de GPS (Système Global de Positionnement).[24]

– *Histoire des Smartphones*

Dans le tableau qui suit (**Tableau 1.1**), nous proposons les dates les plus importantes dans l'histoire des Smartphones [22] .

Année	Événement
1992	Conception du premier smartphone d'IBM.
2001	Lancement du WA3050 de Sagem qui fut l'un des premiers à combiner les fonctions d'un téléphone mobile et d'un PDA tactile. Il était compatible avec la nouvelle norme de l'époque, le GPRS.
2002	Evolution des Réseaux et arrivée du 'EDGE ' et de la 3G/3G+. lancement de NOKIA, LG, Samsung.
2005	En Aout, Google rachète le Startup Android qui développe un Système d'exploitation pour Smartphones, et qui commence à équiper de nombreux constructeur tel que Samsung ou HTC.
2007	Sortie de l'iPhone d'Apple, dont le succès commercial contribua à démocratiser les Smartphones.
2008	Vers la fin de cette année, RIM lance son premier Smartphone tactile, mais BlackBerry OS ne devient réellement un système d'exploitation ergonomique et adapté au tactile qu'en 2011.
2010	Lancement de 'Bada' par samsung, ainsi que le SE 'Windows Phone 7' par Microsoft.
2011-2012	La taille des écrans continue à augmenter avec l'apparition de 1280 px en largeur au lieu de 1136 et amélioration de la définition des écrans.
2013	La surface d'affichage continue d'augmenter avec l'apparition de 1920px en largeur avec une haute définition.

TABLE 1.1 – Histoire des Smartphones.

2. Les Tablettes tactiles [23]



– Définition des tablettes

Tablette (de l'anglais tablet, plaque) est le nom donné à une famille d'ordinateurs portables dépourvus de clavier à touches et munis d'un écran tactile, de la même dimension qu'une feuille A4 ou plus petits. L'écran tactile est toujours multipoints, donc capable de détecter plusieurs touchés simultanés.

– Histoire des tablettes

On considère le 'Newton MessagePad 100', sorti par Apple en 1993, comme l'ancêtre des tablettes. Avant lui, le 'Dynabook' d'Alan Kay avait matérialisé le concept en 1968. L'assistant personnel Newton nécessitait un stylet et reconnaissait l'écriture manuscrite (avec des restrictions). Il n'a pas connu de succès. Microsoft a présenté en 2001 les spécifications (dont une version de Windows XP) pour des " Tablet PC ", utilisables avec un stylet.

Le succès est né avec l'iPad d'Apple, présenté en janvier 2010, et qui repose sur des technologies (matérielles et logicielles) voisines de celles de l'iPhone, téléphone de la marque sorti en 2007. Des centaines de modèles ont ensuite été proposés par différents constructeurs.

– *Utilisation des tablettes*

Ces ordinateurs sont essentiellement tournés vers l'utilisation d'Internet : consultation de pages Web, lecture de journaux en ligne ou de livres électroniques ou messageries. Leurs dimensions excluent l'intégration d'une mémoire de stockage mécanique, comme un lecteur-graveur de DVD ou un disque dur.

3. Les assistants personnels(PDAs) [23]



Assistant personnel ou PDA (Personal Digital Assistant) ou ordinateur de poche en français est un équipement électronique bureautique de poche utilisé principalement pour ses fonctions d'agenda, de répertoire téléphonique et de bloc-notes, mais les avancées technologiques ont permis de lui adjoindre des fonctionnalités multimédia, telles que le dictaphone, le lecteur de mp3, d'images, de vidéos, auxquels s'ajoutent des programmes qui le transforment en outil de navigation associé à un GPS, par exemple. Il s'utilise avec un stylet.

1.2.3 Evolution de l'informatique mobile

L'informatique mobile évolue vers une informatique de plus en plus connectée et répartie, elle se dématérialise et devient 'l'informatique dans les nuages' ou 'Cloud computing'. Ainsi la puissance de l'informatique mobile devient de plus en plus extensible. Chaque fois qu'on utilise une application mobile on devient acteur du Cloud computing.

Cloud Computing (littéralement l'informatique dans les nuages) est un concept majeur dans l'évolution informatique de ces dernières années. Ce concept fait référence à l'utilisation des capacités de calcul et de stockage d'ordinateurs et de serveurs répartis dans le monde entier fournis en tant que service à travers les technologies internet vers une multitude d'utilisateurs externes.

C'est donc un concept qui permet de stocker sur des serveurs à distance des données et traitements informatiques traditionnellement localisés sur des sites locaux.

Mais le Cloud computing ne cesse d'évoluer de jour en jour. Dans un future proche, une grande partie des objets dont nous disposons dans nos maisons seront connectés à internet (système de surveillance, système de fermeture de porte, système d'éclairage ...), collecter et traiter des données de ces appareils sera d'autant plus simple si tout est centralisé et piloté à distance. Cette configuration permettra de plus de simplifier le quotidien, par exemple : le Cloud computing permet à une entreprise ou un utilisateur individuel d'avoir accès à des services, des environnements, des logiciels qu'elle n'aurait pas forcément eu du fait des coûts associés (études, maintenance...). L'avantage est de pouvoir s'affranchir des contraintes informatiques (choix du logiciel, installation, mise à jour des éditeurs...). En gros, accéder à plus de services avec un cout minime.

Et surtout, Avec le Cloud computing, les applications mobiles ne sont plus restreintes que pour certains terminaux mobiles tel que les Smartphones ou systèmes d'exploitations car elles peuvent utiliser les capacités du réseau qui a lui seul pourra offrir les services désirés à la clientèle, ce qui a donné naissance au Cloud computing Mobile. Il existe trois ensembles de Cloud computing [33] :

- **Application** ou **SaaS** (Software as a service) : C'est la couche applicative mettant à disposition des applications complètes fournies à la demande. On y trouvera différents types d'application allant du CRM, à la gestion des contenus en ligne, les collaborations, le monitoring (qui est l'opération qui consiste à suivre méticuleusement le fonctionnement d'un système ou d'un processus en temps réel) et d'autres applications.
- **La plate-forme** ou **PaaS** (Plate-forme as a Service) : est la plate-forme d'exécution, de déploiement et de développement des applications. On peut citer comme exemple : les objets stockés(les fichiers), la base de donnée, l'exécution, etc.

- **L'infrastructure** ou **IaaS** (Infrastructure as a Service) : permet de disposer d'une infrastructure à la demande, pouvant héberger et exécuter des applications, des services ou encore stocker des données. On y trouvera le stockage de block de données, les serveurs distant, le réseau etc.

Le schéma ci-contre **Figure 1.1** illustre un exemple d'une vue générale du Cloud computing.

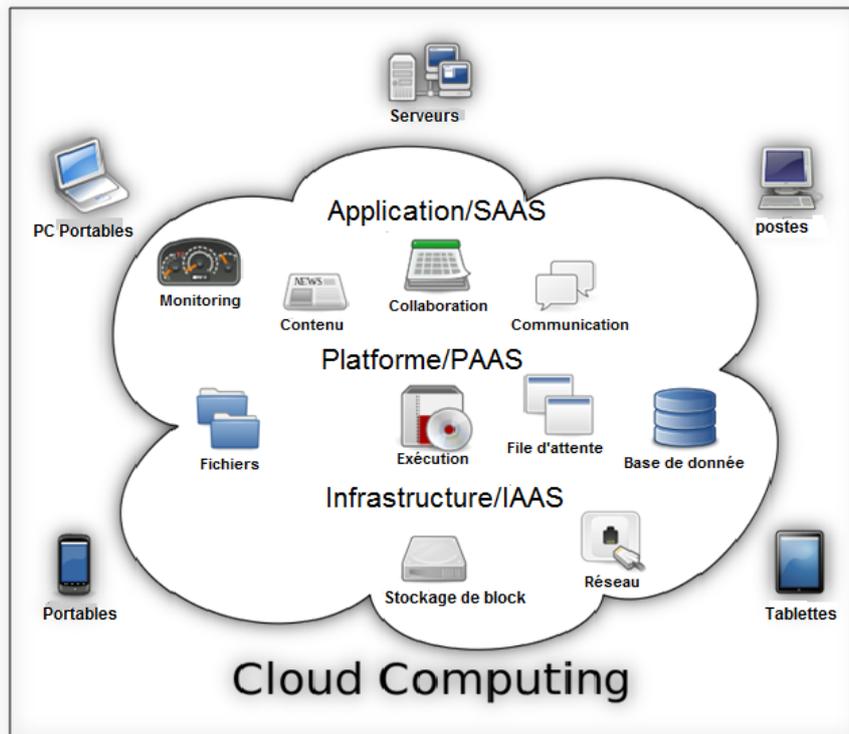


FIGURE 1.1 – Vue générale du Cloud computing.

1.3 Les applications mobiles

1.3.1 Présentation des applications mobiles

Une application mobile est un logiciel applicatif transportable et autonome, développé pour être installé sur un appareil électronique mobile. Elle est identifiée par un ou plusieurs programmes téléchargeable de façon gratuite ou payante depuis un magasin d'applications "Application Store ", permettant d'accéder à un contenu homogène et exécutable à partir du système d'exploitation du Smartphone. Les applications mobiles permettent en général un accès plus pratique, rapide et efficace à des sites en version mobile ou web.[32]

1.3.1.1 Histoire des applications mobiles

La naissance des applications mobiles remonte à la fin du vingtième siècle, Leurs industrie à commencé quand l'iPhone d'Apple et leur magasin iTune ont été introduits en 2007.C'était aussi le moment ou les utilisateurs ont commencé à utiliser leurs appareils mobiles pour surfer sur internet, Seules quelques personnes regardaient leurs courriers électroniques, mais avec l'iPhone le taux de consommateurs a augmenté et les téléphones mobiles ne sont plus utilisés que pour des appels téléphoniques , mais pour beaucoup plus que ça , puisque le nombre de nouvelles applications mobiles s'accroit de jour en jour et ces applications nous offrent plusieurs services tel la consultation du courrier électronique, la téléphonie-IP etc. [3]

1.3.1.2 Architecture des applications mobiles

Une application mobile est normalement structurée comme une application multi-couche. Elle est composée de trois couches principales : [27]

1. **La couche présentation** : correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'interface homme machine. En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en HTML pour être exploitée par un navigateur web ou en WML pour être utilisée par un téléphone portable. On conçoit facilement que cette interface peut prendre de multiples facettes sans changer la finalité de l'application.

La couche présentation relaie les requêtes de l'utilisateur à la couche métier, et en retour lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.

2. **La couche métier ou Business** : correspond à la partie fonctionnelle de l'application, celle qui implémente la " logique ", et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation. Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche.

La couche métier offre des services applicatifs et métier à la couche présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés.

3. **La couche accès aux données** : Elle consiste en la partie gérant l'accès aux données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas car ces derniers sont transparents pour la couche métier qui accède aux données de manière uniforme.

- **Données propres au système** Ces données sont destinées à durer dans le temps, de manière plus ou moins longue, voire définitive. Elles peuvent être stockées indifféremment dans de simples fichiers texte, ou eXtensible Markup Language (XML), ou encore dans une base de données. Quel que soit le support de stockage choisi, l'accès aux données doit être le même.

- **Données gérées par un autre système** Les données peuvent aussi être gérées de manière externe. Elles ne sont pas stockées par le système considéré qui s'appuie sur la capacité d'un autre système à fournir ces informations.

Par exemple, une application mobile de gestion des utilisateurs peut ne pas sauvegarder les informations dont elle a besoin, mais les demander à une application centralisé qui a une capacité de mémoire élevée. Celle-ci est indépendante et pré-existante, et on ne se préoccupe pas de savoir comment elle les obtient ou si elle les sauvegarde, on utilise simplement sa capacité à fournir les données.

Lors de l'élaboration d'une application mobile, deux types d'architectures peuvent être choisis : une application Web cliente basique ou développée.

1. **Une application légère , ou cliente** : les couches métier et données seront situés sur le serveur. Voila un exemple descriptif d'une application basique, illustré dans la figure ci-dessous .
2. **Une application développée** : les services des couches métiers et couches de données sont susceptibles d'être situés sur le périphérique lui-même.

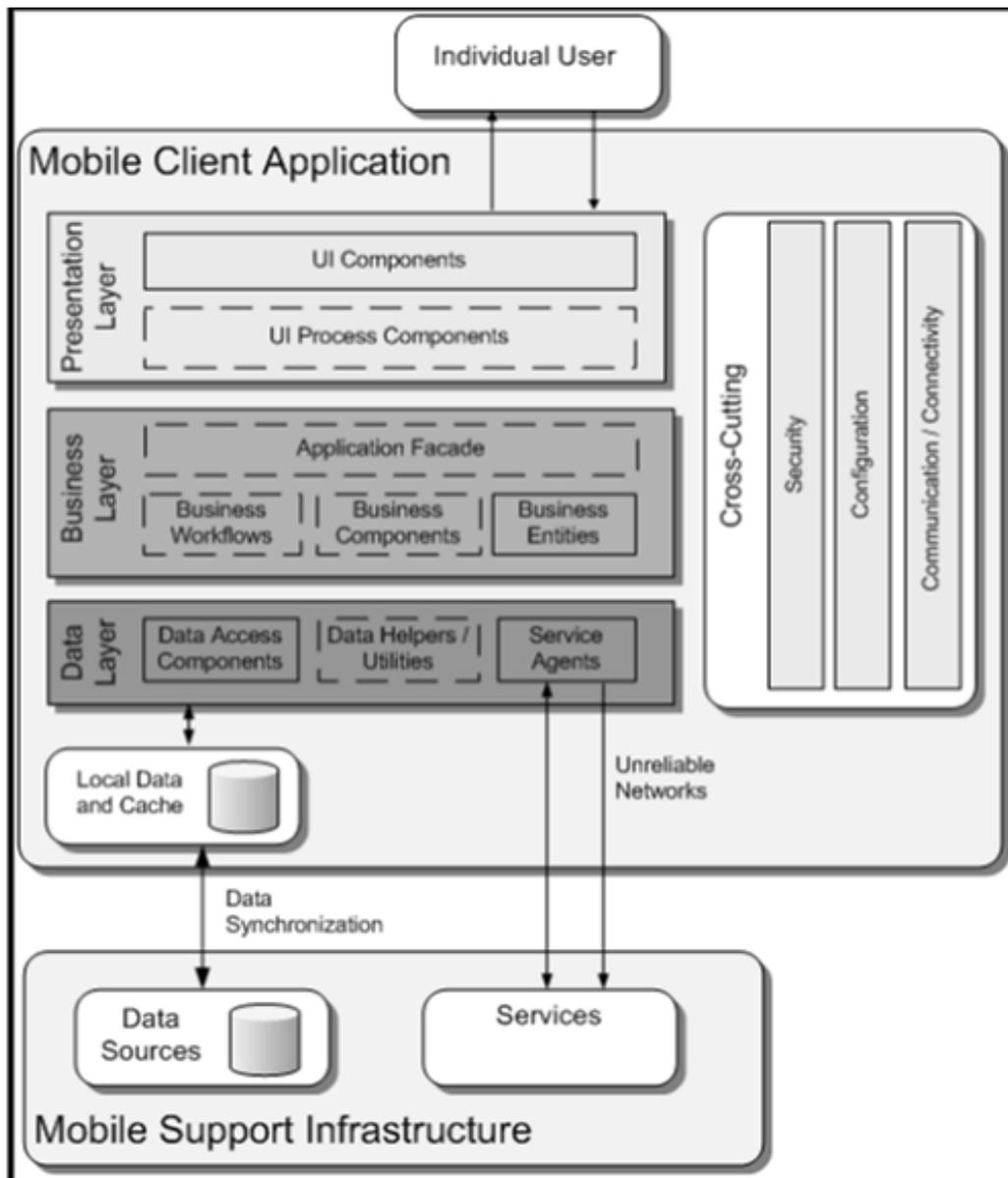


FIGURE 1.2 – Architecture d'une application mobile.

1.3.1.3 Fonctionnement des applications mobiles

Le fonctionnement des applications mobile dépend du mode de l'application, on peut en distinguer trois modes ; le mode connecté, le mode déconnecté et le mode mixte qu'on définit comme suit :

1. Le mode connecté ou On Line

Dans ce mode L'utilisateur est connecté en permanence au serveur. Le terminal mobile ne supporte alors qu'un navigateur et ne nécessite pas la mise en place d'une application cliente. Chaque demande de données est matérialisée par une requête au serveur. Ainsi toutes les opérations de consultation, modification, suppression s'effectuent en ligne.

Le mode connecté suppose que le terminal accède à un réseau (wifi, Edge ou 3G). Dans le cas contraire l'application n'est plus utilisable. [27]

Les applications de ce mode reposent en général sur les points suivant :

- L'application en elle-même, codée en Objective C sur iPhone et JAVA pour Android et Blackberry.
- Un Serveur Web type Apache/PHP.
- Un serveur de base de données (MySQL, SQLite...).

La lecture des données se fait de la manière suivante :

Une fois lancée, l'application se connecte au serveur web et télécharge un fichier, en général au format XML (RSS ou WebServices). Ce flux est généré en temps réel par un script écrit par un langage web qui interprète la demande (et les paramètres GET et POST) et interroge la base de données. Puis l'application interprète ces données et les place dans l'interface. En utilisant ce mode, les différentes versions des applications viennent toutes se connecter à la même source de données, et simplifient la maintenance et les mises à jour.

Le fonctionnement d'une application en mode connecté est illustré dans le schéma de la **Figure 1.3**.

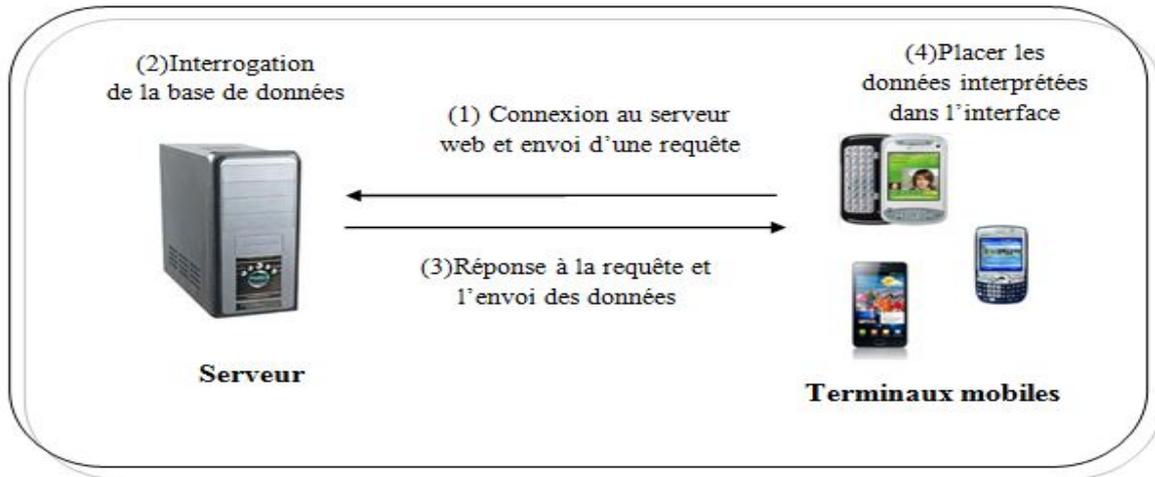


FIGURE 1.3 – Fonctionnement d'une application mobile en mode Connecté.

➤ Avantages et inconvénients du mode connecté [27]

- + la Possibilité de donner des informations en temps réels ;
- + La facilité de mise à jour du contenu ;
- + Les applications sont légères et rapides à télécharger sur les magasins en ligne ;
- + Il garantit la fraîcheur des informations et la sécurité.
- la connexion internet est indispensable ;

➤ Exemples d'applications en mode connecté [4]

Les exemples d'applications mobiles en mode connectées sont nombreux et on pourra citer la gestion d'inventaire (entrées/sorties de stock avec des lecteurs de code barre dans les entrepôts), la navigation web à partir d'un portail, ou les applications machines-to-machines notamment dans la gestion des alertes systèmes en temps réel.

2. Le mode Déconnecté ou Off Line [27]

Le mode déconnecté implique deux choses : la première c'est que Le terminal mobile contient des informations autonomes liées au processus métier¹ . La deuxième c'est qu'il est nécessaire de prévoir non seulement le chargement et le rafraichissement de ses données mais aussi sa fréquence, son mode, et les données métiers utiles à cette population nomade.

En effet, les requêtes utilisateurs ne sont plus adressées directement au serveur mais au terminal mobile d'abord, qui contient de fait une base de données et une application embarquée mais cela engendre quelques problèmes, tels que :

- Le volume et la performance des données.
- les notions d'historique, de purge (suppression), de qualité des données et des objets pertinents ;
- Les logiques métiers embarqués qui désignent toutes les règles ou tests de validation personnalisés appliquées aux données avant de les insérer, de les mettre à jour ou de les supprimer de la base de données , chose qui n'est pas trop pratique .
- Les notions d'habilitation d'accès, de gestion des conflits d'accès au référentiel d'entreprise et bien entendu de temps de réponse acceptable.

Dans les applications en mode déconnecté toutes les fonctions et données média sont embarquées dans le téléphone. Un seul élément dans ce cas de figure reste ; le téléphone avec son logiciel embarqué. Le téléphone lit tout les fichiers qu'il lui faut directement dans la mémoire de stockage du téléphone, ou dans une base de données embarquée. Les informations sont mise à jour lors de la connexion au système d'information et consultable après rafraichissement.

Le fonctionnement d'une application en mode déconnecté se fait comme illustré dans le schéma de la **Figure 1.4**.

1. Un processus métier est un ensemble d'activités, entreprises dans un objectif déterminé .

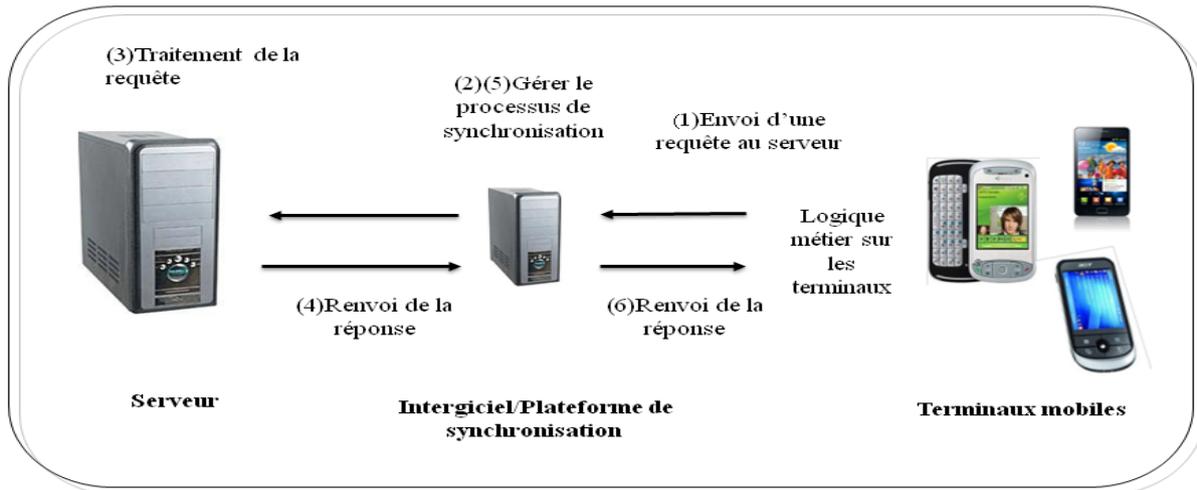


FIGURE 1.4 – Fonctionnement d'une application mobile en mode Déconnecté.

Un Intergiciel (middleware, plateforme de synchronisation, data orchestration engine, ...) sera alors indispensable pour gérer ce processus de synchronisation, réassembler et arranger les données qui peuvent évoluer à chaque instant sur chacun des environnements séparément (terminal mobile et backend²) et parfois de manière incompatible.

➤ Avantages et inconvénients du mode Déconnecté [27]

+ l'application en ce mode est disponible, lors du téléchargement d'une application, tout son contenu est immédiatement disponible.

- La moindre correction orthographique demande une mise à jour de l'application.
- L'application en ce mode peut vite devenir extrêmement lourde.

➤ Exemples d'applications en mode Déconnecté [4]

L'émission/réception des emails sur mobiles, se fait en mode déconnecté synchrone. Le PDA sera chargé à priori automatiquement avec les nouveaux messages dès que le réseau sera accessible. Sur des applications professionnelles telles que la maintenance industrielle ou la gestion d'un CRM, les PDA seront chargés avec des données en début de journée et la synchronisation dépendra du mode d'utilisation, consultatif uniquement, ou interactif.

2. Backend : Endroit où s'effectuent les gros traitements informatiques, typiquement un gros ordinateur équipé d'un système de gestion de base de données et de périphériques de stockage.

3. Le mode Synchronisé ou Mixte [27]

Le mode synchronisé associe les deux modes précédents sur le même terminal léger. Une partie des données est synchronisée sur le terminal, l'autre sera consultée ou transmise en mode connecté.

Les applications mobiles de ce mode reposent sur :

- Un serveur web alimenté par une base de données.
- Le système de stockage interne du téléphone.
- L'application en elle-même.

L'application en utilisation simple lit les données sur son disque de stockage, ou sa base de données interne, mais dispose d'une fonction qui permet de synchroniser son contenu, qui fonctionne comme suit :

- L'application vérifie Sa connectivité.
- Une requête avec la date de dernière mise à jour est envoyée vers le serveur.
- Le serveur répond par une liste de données à récupérer. Toute cette liste est envoyée via un flux XML.
- L'application va alors télécharger toutes les informations. Il est aussi possible, pour des petits fichiers, de les stocker en base de données sous forme de BLOB binaires.
- Une fois toute la séquence terminée, on recharge l'interface.

✎ Avantages et inconvénients du mode synchronisé [27]

- + Application légère à télécharger ; seules les nouvelles informations sont téléchargées (ou celles mises à jours), les autres sont stockées dans le téléphone.
- Les données peuvent ne plus être valables si l'on oublie de faire la mise à jour.
- Lors du premier lancement, cela implique de télécharger tout le contenu. Même en Wifi, cela peut prendre du temps dans certain cas.

1.3.1.4 Différents types d'applications mobiles

On peut classifier les applications mobiles selon quatre différents types ; applications natives, web applications, applications hybride et applications flash ; qu'on définit comme suit [12] :

1. Applications natives

Les applications natives sont des logiciels qui ont été développés spécifiquement pour une plate-forme mobile ou plus exactement pour des systèmes d'exploitation utilisés par les Smartphones et tablettes (iOS, Android, etc.), en utilisant un SDK propre à elle. Les applications ainsi créées sont ensuite téléchargeables depuis une plateforme dédiée au système, généralement un magasin d'application comme 'Android Market' de Android ou 'App store' de Apple .

Le fait de développer une application native permet généralement d'utiliser toutes les fonctionnalités liées au système d'exploitation visé (GPS, accéléromètre, appareil photo, etc.) et permet également de proposer des applications généralement plus riches que les web applications en HTML5. Une fois téléchargées et installées certaines applications peuvent par ailleurs être utilisées sans connexion Internet.

2. Web Applications (Web Apps)

Les web applications sont des sites web optimisés pour mobile, elles sont en général conçues pour ressembler à de vraies applications, on peut y accéder via le navigateur internet du périphérique mobile.

Ces Applications mobiles peuvent être :

-*génériques* ; qui fonctionnent sur toutes les plateformes mobiles.

- *Dédiées à un type de support particulier* ; comme beaucoup d'applications et de frameworks pour mobiles qui ne fonctionnent que sur un navigateur basé sur Webkit (iOS, Android, WebOS etc.).

Le système iOS permet d'utiliser une web application à la manière d'une application, en plaçant un raccourci vers le site web mobile directement sur le bureau, avec une icône. Le navigateur lance alors la web app en plein écran, comme une

vraie application.

3. Applications hybrides

Une application hybride est un mélange de code natif et d'affichage de vues HTML/javascript, elles peuvent être distribuées sur les Stores (boutiques en lignes) des systèmes mobiles, on peut y citer quelques exemples : LinkedIn, Microsoft Bing pour mobiles etc.

Plusieurs stratégies de développement d'applications hybrides sont possibles, selon que nous nous penchons vers le côté natif ou plus vers le côté HTML :

- Ne réaliser que certaines interfaces, voir même que certains composants d'IHM en HTML.

- Réaliser toutes les interfaces en HTML mais garder la logique applicative en code natif, notamment les effets de transitions entre les interfaces et la gestion des transitions (scrolling).

- Réaliser les interfaces en HTML, et les transitions (scrolling) en javascript. Le code natif peut alors se cantonner à quelques composants techniques très ciblés. De la même manière selon les applications la logique métier peut être codée en javascript ou bien en code natif.

4. Applications Flash

Comme pour les applications HTML5/javascript, une application Flash est une solution de développements multiplateformes, qui permet d'accéder à la plupart des ressources des smartphones et tablettes : multitouch, accéléromètre, GPS, bases de données locales SQLite. Ces applications embarquent le runtime AIR dans leur code. On a par exemple Flash Builder / Flex en version 4.5 permettent de packager des applications Android, iOS, et BlackBerry Tablet OS.

1.3.1.5 Les systèmes d'exploitations mobiles

Un système d'exploitation ou OS " Operating System " en anglais est un super logiciel qui permet de gérer toutes les autres applications (mise en marche, arrêt,

allocation des ressources mémoires) et la communication avec le support physique. [17]

Un système d'exploitation mobile est un système d'exploitation conçu pour fonctionner sur un appareil mobile. Ce type de système d'exploitation se concentre entre autres sur la gestion de la connectivité sans fil et celle des différents types d'interface. [17]

Il y'a plusieurs systèmes d'exploitations mobiles, les plus connus sont

✓ **Android**

Elaboré par Google, Android est un système d'exploitation fondé sur un noyau Linux. Disponible via une licence Apache version 2, le système d'exploitation inclut tous les utilitaires requis par un constructeur ou par un opérateur pour mettre en œuvre un téléphone portable. Android a été conçu pour intégrer au mieux des applications existantes de Google comme le service de courrier Gmail, ou celui de cartographie, Google Maps, ou encore Google Agenda, Google Talk, YouTube. [22]

✓ **iOS ou iPhone OS**

iOS, anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone. Reconnu pour sa fluidité, son ergonomie et son intuitivité : c'est le système d'exploitation le plus abouti à ce jour. Il dispose du portail App Store, qui avec un catalogue de 500 000 applications, s'est imposé comme une référence parmi les kiosques d'applications mobiles.[17]

✓ **BlackBerry OS**

BlackBerry est une marque développée par l'entreprise RIM (Research In Motion). Les téléphones possèdent comme système d'exploitation BlackBerry OS. Cet OS se trouve donc uniquement sur les téléphones BlackBerry. Les applications BlackBerry sont développées en Java grâce à une bibliothèque Java et à L'environnement de développement gratuit Eclipse. [17]

✓ **Symbian**

C'est le système d'exploitation développé par Nokia pour ses propres Smartphones. Il est donc présent uniquement sur les téléphones de la marque Nokia. Il y a plusieurs façons de concevoir des applications pour les développeurs. Soit avec un environnement de développement Qt [<http://qt.nokia.com>] (Qt est un outil de développement et une librairie graphique C++) modifié et adapté pour concevoir des applications mobiles, soit avec le langage de programmation Java.[17]

✓ **Windows Mobile** [17]

C'est le système d'exploitation conçu par Microsoft. Il est présent sur les téléphones Windows. Microsoft a récemment signé un accord en 2011 avec Nokia pour l'intégration du système d'exploitation Windows sur les nouveaux téléphones Nokia. Pour le développement d'application sur cet OS, il faut acheter Visual Studio qui est une plateforme de développement créée par Microsoft ou utiliser une version simplifiée de Visual Studio (Visual Studio Express). Le langage utilisé est le C # .

Il y'a d'autres systèmes d'exploitations mobiles, moins connus comme :

✓ **Windows Phone**

Le 15 février 2010 Microsoft a lancé un nouveau système d'exploitation pour mobile, Windows Phone 7. Il inclut des services de Microsoft comme Windows Live, Zune ect. Il intègre aussi des fonctionnalités média sociaux tel Facebook et Twitter. Comme Windows Phone 7 est une nouvelle plate-forme, il n'existe aucune compatibilité avec les applications Windows Mobile. [17]

✓ **HP webOs (Palm webOs)**

C'est le système d'exploitation de Palm qui fut racheté par HP. Il est relativement nouveau et marche aussi bien sur les Smartphones que sur les tablettes. [19]

✓ **Bada**

Ce système d'exploitation a été développé par la société coréenne Samsung qui devrait être installé sur ses téléphones traditionnels pour la fusion de ces derniers avec

les Smartphones. Le langage utilisé est le C++. [17]

✓ **MeeGo (Linux Foundation)**

Est un système d'exploitation open source basé sur Linux, il n'a sorti encore aucun appareil, mais peut avoir un avenir si les fabricants le soutiennent. [15]

✓ **LIMO**

C'est une contraction de " Linux Mobile ", c'est un système ouvert basé, comme son nom l'indique, sur Linux. [16]

1.3.1.6 Avantages et Inconvénients des applications mobiles

Les applications mobiles ont plusieurs avantages et certes ont aussi des inconvénients, en voici les principaux [2]

✍ **Avantages**

+ L'accès direct aux contenus de l'application mobile via l'icône présente sur le dashboard (tableau de bord) du téléphone ou de la tablette (mode d'accès sans URL);

+ Un fonctionnement en mode déconnecté;

+ L'implémentation de fonctionnalités natives comme par exemple les notifications " PUSH ";

+ Toutes les applications sont disponibles sur des plateformes de téléchargement tel que ' l'Android Market', ce qui les rends faciles à trouver.

✍ **Inconvénients**

- Respecter les règles des plateformes mobiles : chaque application doit tenir compte des règles d'Apple, ou de Google, ou encore de Windows ;

- Le développement d'applications mobiles : le coût de création d'une application est relativement élevé surtout si on souhaite que celle-ci soit disponible sur toutes les plateformes ;

- La mise à jour des applications mobiles : lorsqu'on doit mettre à jour la dernière version d'une application, il faut se rendre sur les plateformes de téléchargement, alors que pour les sites mobiles elles se font automatiquement.

1.4 Conclusion

Suite à l'étude faite dans ce chapitre concernant les applications mobiles et leur importance dans le monde d'aujourd'hui, cela nous a aidés à comprendre les notions de base.

Aussi on a pu distinguer que la concurrence entre les différents Systèmes d'exploitation mobile est vraiment rude. En revanche, Android a pu récemment se démarquer de ses concurrents grâce à sa plateforme innovante (qui intègre toutes les nouvelles technologies), son accessibilité et son ouverture (licence open source), qui se sont révélées correspondre aux attentes des consommateurs.

Tous ces critères nous ont amenées à choisir Android comme environnement de développement, qu'on va détailler dans le chapitre suivant.

Le système Android

2.1 Introduction

Le développement d'applications Android nécessite un environnement de travail bien spécifique de par le matériel et les logiciels à utiliser ainsi que le langage de développement.

Nous avons opté pour l'utilisation d'Android pour son ouverture et son accessibilité. Dans ce chapitre nous présenterons le système d'exploitation Android et l'évolution de ces versions, puis nous citerons ces fonctionnalités suivie de son architecture et finiront par le cycle de vie d'une architecture sous Android.

2.2 Le système d'exploitation Android

2.2.1 Présentation d'Android

L'histoire de la naissance d'Android a commencé en août 2005, lorsque Google a acquis Android INC, une startup qui développe des applications pour téléphones mobiles. Depuis lors, 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 annoncée. 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'agissait d'Android, une plateforme pour appareils mobiles.[\[20\]](#)

En effet, Android représente un système d'exploitation open source dédié pour Smartphones, PDA et terminaux mobiles.

Il est basé essentiellement sur la simplicité d'utilisation et surtout sur une capacité de personnalisation importante. Pour promouvoir ce système d'exploitation open source, Google lui a conféré des alliés puissants réunis au sein de l'Open Handset Alliance tel que Samsung, Motorola, Sony Ericsson et LG. Les principaux concurrents d'Android sont Apple avec iPhone OS qui équipe l'iPhone, Research In Motion avec BlackBerry OS, Palm avec Nova ou Web OS, Microsoft et son Windows Mobile, etc. [30]

2.2.2 Fonctionnalités d'Android

Android a été conçu pour intégrer au mieux les applications existantes de Google comme le service de courrier Gmail, l'agenda Google Calendrier ou encore la cartographie Google Maps.

Les fonctionnalités proposées par Android diffèrent d'une version à une autre, on peut citer les plus importantes [37] :

- Augmentation de la performance et de la vitesse ;
- Fonctionnalité de Hot spot Wifi ;
- Partage de contact sur Bluetooth ;
- Ecran d'accueil personnalisable ;
- Disponibilité des Widgets ;
- Installation des applications directement en ligne sur un ordinateur sans avoir à toucher au téléphone pour le faire ;
- La possibilité d'échanger directement des fichiers entre 2 téléphones Android à proximité, simplement en appuyant sur le bouton "Beam" ;
- La possibilité de filmer et de prendre des photos en même temps ;
- Mise à jour automatique des applications.

2.2.3 Evolution des versions d'Android

Le système Android ne cesse d'évoluer, on illustre dans le tableau ci-dessous -Tableau 2.1 - ses différentes versions.[28]

Version	Evolution
Android 1.0 Fin 2007	Version connue uniquement ou presque des développeurs car c'est la version du SDK distribuée avant la sortie du premier téléphone Android.
Android 1.1 dit "petit four" Octobre 2008	<ul style="list-style-type: none"> - Version BETA, incluse dans le premier téléphone ; - Beaucoup de corrections de bug (alarme, Gmail...); - Amélioration des fonctionnalités MMS ; - Introduction des applications payantes sur l'Android Market.
Android 1.5 dit "Cupcake" 30 Avril 2009	<ul style="list-style-type: none"> - Nouveau clavier avec autocomplétion ; - Support Bluetooth ; - Enregistrement de vidéos ; - Widgets et Live Folders ; - Animations améliorées ; - Mise à jour de l'interface graphique.
Android 1.6 dit "Donut" 15 Septembre 2009	<ul style="list-style-type: none"> - Recherche dans les éléments améliorés ; - Recherche vocale améliorée ; - Android market remanié ; - Meilleure intégration entre la galerie multimédia et la prise de clichés ou de vidéos ; - Indicateur d'utilisation de la batterie ; - Apparition de fonctionnalités pour les réseaux privés virtuels (VPN).

<p>Android 2.0 dit "Eclair" 26 Octobre 2009</p>	<ul style="list-style-type: none"> - Interface graphique remaniée ; - Gestion des contacts changée ; - Support d'HTML 5 ; - Support Bluetooth 2.1 ; - Clavier virtuel amélioré. - Refonte de l'installation et des mises à jour du kit de développement.
<p>Android 2.0.1 Décembre 2009</p>	<ul style="list-style-type: none"> - Mise à jour mineure (corrige un ensemble de bugs comme pour l'autofocus, permet une réduction de la consommation de la batterie et quelques améliorations graphiques comme l'écran de verrouillage).
<p>Android 2.1 Janvier 2010</p>	<ul style="list-style-type: none"> - Mise à jour encore mineure, même si elle apporte beaucoup de nouvelles choses ; - le bureau se compose de 5 écrans virtuels ; - fonds d'écran animés (en fonction du toucher ou du son...); - nouveaux effets 3D (notamment sur la gestion des photos) ; - amélioration du client Gmail ; - interface graphique pour l'utilisation du téléphone en voiture.
<p>Android 2.2 Dit " Froyo " 20 Mai 2010</p>	<ul style="list-style-type: none"> - Vitesse améliorée ; - Nouvelles fonctionnalités ; - mises à jour de l'interface graphique ;
<p>Android 2.3 dit "Gingerbread " 6 Décembre 2010</p>	<ul style="list-style-type: none"> - Dernière version dédiée uniquement aux Smartphones ; - Cette version est parfois utilisée sur de petites tablettes.
<p>Android 3.0 dit " Honeycomb" 22 Février 2011</p>	<p>Réservé aux tablettes tactiles et aux téléviseurs connectés, cette mise à jour comprend de nombreux changements dans l'interface.</p>

<p>Android 4.0 dit "Ice Cream Sandwich" 19 Octobre 2011</p>	<p>Cette nouvelle version, fortement inspirée d'Honeycomb, unifiée pour Smartphones, tablettes et Google TV apporte de nombreux changements.</p>
<p>Android 4.1 dit "Jelly Bean " 9 Juillet 2012</p>	<ul style="list-style-type: none"> – Dernière version dédiée principalement aux Smartphones ; – Cette version est parfois utilisée sur de petites tablettes.
<p>Android 4.2 dit "Jelly Bean" 13 Novembre 2012</p>	<ul style="list-style-type: none"> – Nouvelle interface de l'appareil photo et l'introduction de photosphère permettant une prise des photos à 360° ; – ajout d'un système multi-compte uniquement sur tablette et permettant aussi d'écrire avec le clavier rien qu'en glissant le doigt.

TABLE 2.1: Evolution des versions d'Android.

2.2.4 Architecture d'Android

Android bénéficie d'une architecture en couche complète faisant de lui une plateforme riche, dédiée aux appareils mobiles. Il est à base de noyau linux profitant des services système de base tels que la sécurité, la gestion mémoire, gestion de processus, etc. À un niveau supérieur se trouvent un ensemble de bibliothèques écrites en C/C++ jouant le rôle d'un middleware (on en cite le système de bibliothèque C, les médiathèques etc.). C'est sur cette couche que se greffe l'Android Runtime, comprenant la machine virtuelle de Java et ses bibliothèques, vient ensuite la plateforme logicielle, nommée aussi Framework de développement, écrite en java et permettant de mutualiser les ressources entre applications Java. Elle offre aux développeurs la possibilité de produire des applications diverses et innovantes à travers un ensemble d'API. Enfin, un niveau plus supérieur se situe un ensemble d'applications sous forme de paquets APK (Android Package). Exemples d'applications fournies par Android sont : un navigateur web, un client-email, un calendrier, un gestionnaire de contacts, etc. [31]

Les couches constituant cette architecture sont décrites comme suit :

1. *Linux Kernel (Linux le cœur du système)* [30]

La couche la plus basse de l'architecture repose sur un noyau linux 2.6. Cependant, il ne s'agit pas d'un linux proprement dit car X-windows n'est pas implémenté (gestion de fenêtrage) tout comme la librairie glibc qui n'est pas supportée (Android utilise une bibliothèque libc customisée nommée Bionic).

Le noyau linux a été choisis pour sa stabilité et sa performance, pour son modèle de sécurité, pour ses capacités d'abstraction avec le matériel et enfin pour son aspect open source et communautaire fort.

A noter enfin que le noyau linux a été patché (mis à jour ou corrigé) offrant ainsi de nouvelles fonctionnalités. Le tableau ci-dessous -Tableau 2.2-liste quelques exemples de patches :

Nom du Patch	Fonctionnalité
Alarm	Mise en place de timers permettant par exemple de réveiller l'appareil quand il est en veille.
Ashmem	Permet le partage de la mémoire entre application.
Binder	Permet la communication entre processus.
Logger	Offre un système de logs.
Wakelogs	Permet la gestion de l'alimentation.

TABLE 2.2 – Exemples de Patches .

Voici quelques exemples de programmes systèmes du noyau Linux qui servent à piloter les différents périphériques :

- ↳ Contrôleur de la caméra ou camera driver ;
- ↳ Contrôleur de la mémoire flash ou shared memory driver ;
- ↳ Contrôleur du clavier ou keypad driver ;
- ↳ Contrôleur du wifi ou wifi driver ;
- ↳ Contrôleur audio ou audio driver ;
- ↳ Contrôleur d'énergie ou power manager etc.

2. Les bibliothèques [30]

Écrites en C/C++, les bibliothèques sont utilisées par les composants du système Android et sont utilisables par le développeur via le Framework Android, en voici quelques exemples :

- ↳ Surface Manager ou Gestionnaire de surface : gérant l'accès au sous-système d'affichage ;
- ↳ Media Framework ou Armature Environnement : permettant la gestion de fichier multimédia (Audio et vidéo) ;
- ↳ *SQLite* : le moteur de base de données relationnel ;
- ↳ *FreeType* : gérant les bitmap et le rendu des polices ;
- ↳ *WebKit* : permettant la navigation internet ;
- ↳ *SGL* : le moteur graphique 2D ;
- ↳ *SSL* : gérant le protocole SSL ;
- ↳ *Libc* : gérant la bibliothèque C ;

3. Android Runtime (Le moteur d'exécution linux) [31]

Chargé d'exécuter les applications, le moteur Android comporte deux éléments : Des bibliothèques de base et La machine virtuelle Dalvik.

- Les bibliothèques de bases : Elles fournissent les fonctionnalités du langage JAVA et des bibliothèques spécifiques à Android.
- Au lieu d'utiliser une JVM (Java Virtual Machine) classique, Android dispose de sa propre machine virtuelle nommée Dalvik. Basé sur une architecture de registre ; Dalvik exécute les applications Android une fois celles-ci compilées au format requis (.dex) à l'aide de l'outil dx (Un fichier java étant compilé en .class,

ce fichier étant ensuite compilé en .dex).

Il est important de noter que chaque application Android s'exécute dans son propre processus ayant sa propre instance de Dalvik car la VM Android a été écrite à des fins de multiples instanciations sans perte de performances.

Enfin, Dalvik s'appuie sur le noyau linux pour la réalisation des tâches dites de bas niveau telles que le threading, la gestion des processus et de la mémoire.

4. *Le Framework de développement ou applications Framework* [30]

Cette couche intéresse tout particulièrement les développeurs car elle leur permet de créer des applications à l'aide d'une plateforme ouverte.

Utilisant le langage JAVA le Framework met à disposition un ensemble de classes utiles à la création d'applications sans oublier la mise à disposition de classes et méthodes permettant l'accès au matériel, à la gestion de l'interface graphique et aux ressources de l'application.

Enfin, notons que le Framework Android se compose des services applicatifs suivants :

- ↳ *Gestionnaire d'activités ou Activity Manager* : gérant le cycle de vie des Activités;
- ↳ *Les vues du système ou Views* : permettant de créer les interfaces graphiques;
- ↳ *Le gestionnaire de notifications ou Notification Manager* : fournissant un mécanisme d'envois de message aux utilisateurs;
- ↳ *Fournisseur de contenu ou Content Providers* : permettant aux applications de partager des données entre elles;
- ↳ *Gestionnaire de ressources ou Resource Manager* : offrant l'externalisation de ressources telles que les chaînes de caractères pour la notion d'internationalisation, de style, de menus etc;
- ↳ *Gestionnaire de fenêtres ou Window manager* : chargé de l'affichage et du placement des fenêtres d'applications;

5. *La couche applicative*

C'est la Couche la plus haute de l'architecture, celle-ci contient l'ensemble des applications natives, tierces ou développées présentes sur l'appareil.

Toutes les applications présentes dans cette couche sont, comme nous l'avons vu précédemment, exécutées par le moteur d'exécution Android, on peut en citer quelques exemples d'application tel que : Accueil, contacts, messagerie, appels, calendrier,

alarme, appareil, Media Player etc.[31]

L'architecture du système Android est illustrée dans la figure ci-dessous -figure 2.1-. [30]

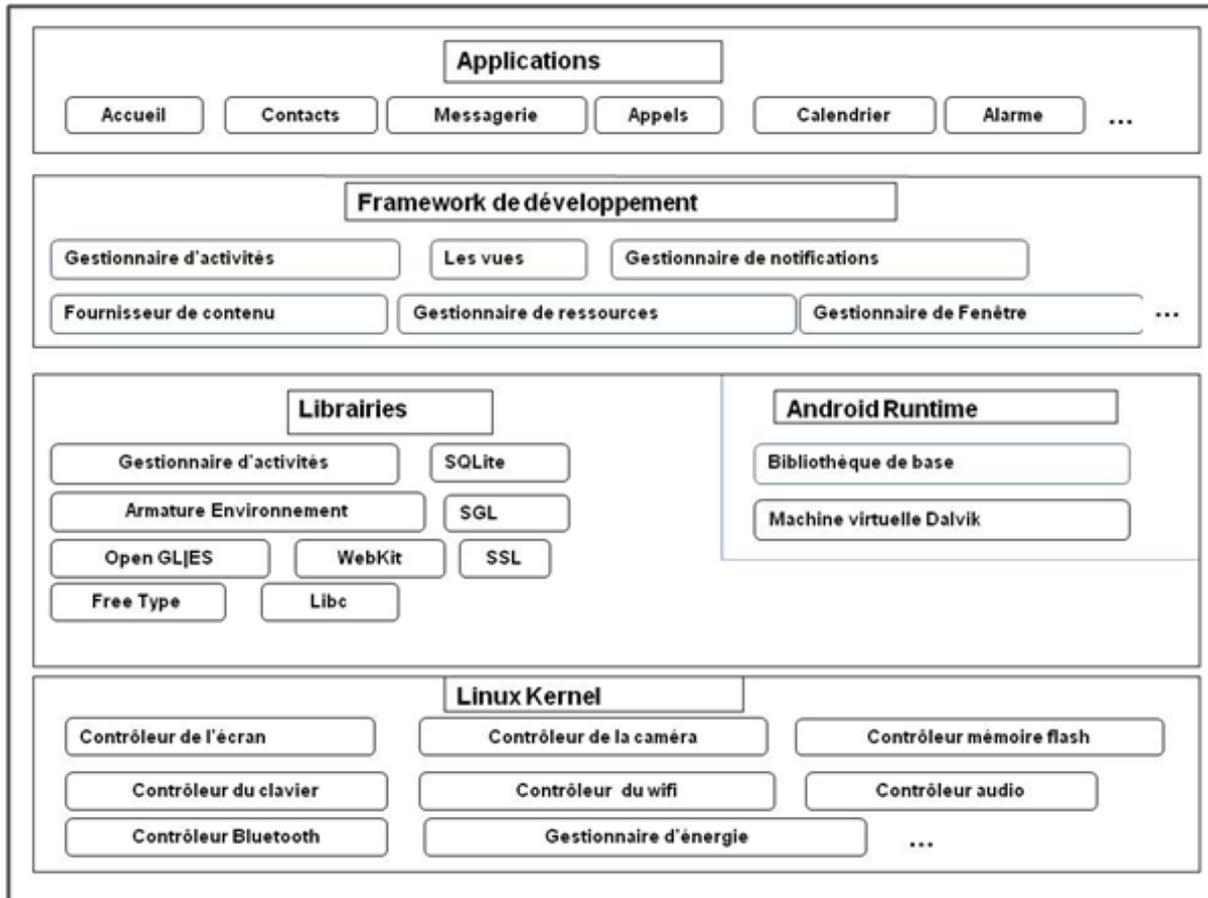


FIGURE 2.1 – Architecture d'Android.

2.2.5 Cycle de vie d'une application Android

Les technologies mobiles prennent de plus en plus de place sur le marché. Les Smartphones sont considérés comme des petits ordinateurs et dotés d'un système d'exploitation s'appuyant sur un noyau Linux. Cependant, ils diffèrent des ordinateurs classiques par le cycle de vie d'une application. Sous Android, une application est composée d'une ou plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faciliter la cinématique (mouvement) de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité. [36]

Le fonctionnement du cycle de vie d'une application mobile qui sera illustré dans la Figure 2.2 se fait comme suit :

-Au Démarrage de l'Activité; la méthode `onCreate` est appelée. Pendant l'utilisation d'une activité, l'utilisateur presse la touche Accueil, ou bien l'application téléphone, qualifiée comme prioritaire et qui interrompt son fonctionnement par un appel téléphonique entrant.

-Une fois l'appel téléphonique terminé, le système réveille l'activité précédemment mise en pause (appel de `onRestart`, `onStart`).

-Si l'activité reste trop longtemps en pause, le système à besoin de mémoire, il détruit l'activité (appel de `onDestroy`).

-`onPause` et `onResume` rajoutent un état à l'activité, puisqu'ils interviennent dans le cas d'activités 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 pour le système. Donc si celui-ci a besoin de mémoire, l'Activité peut être fermée. Ainsi, il est préférable, lorsque l'on utilise cette méthode, de sauvegarder l'état de l'activité dans le cas où l'utilisateur souhaitera y revenir avec la touche Accueil.

La figure 2.2 ci-dessous illustre le cycle de vie d'une application Android.

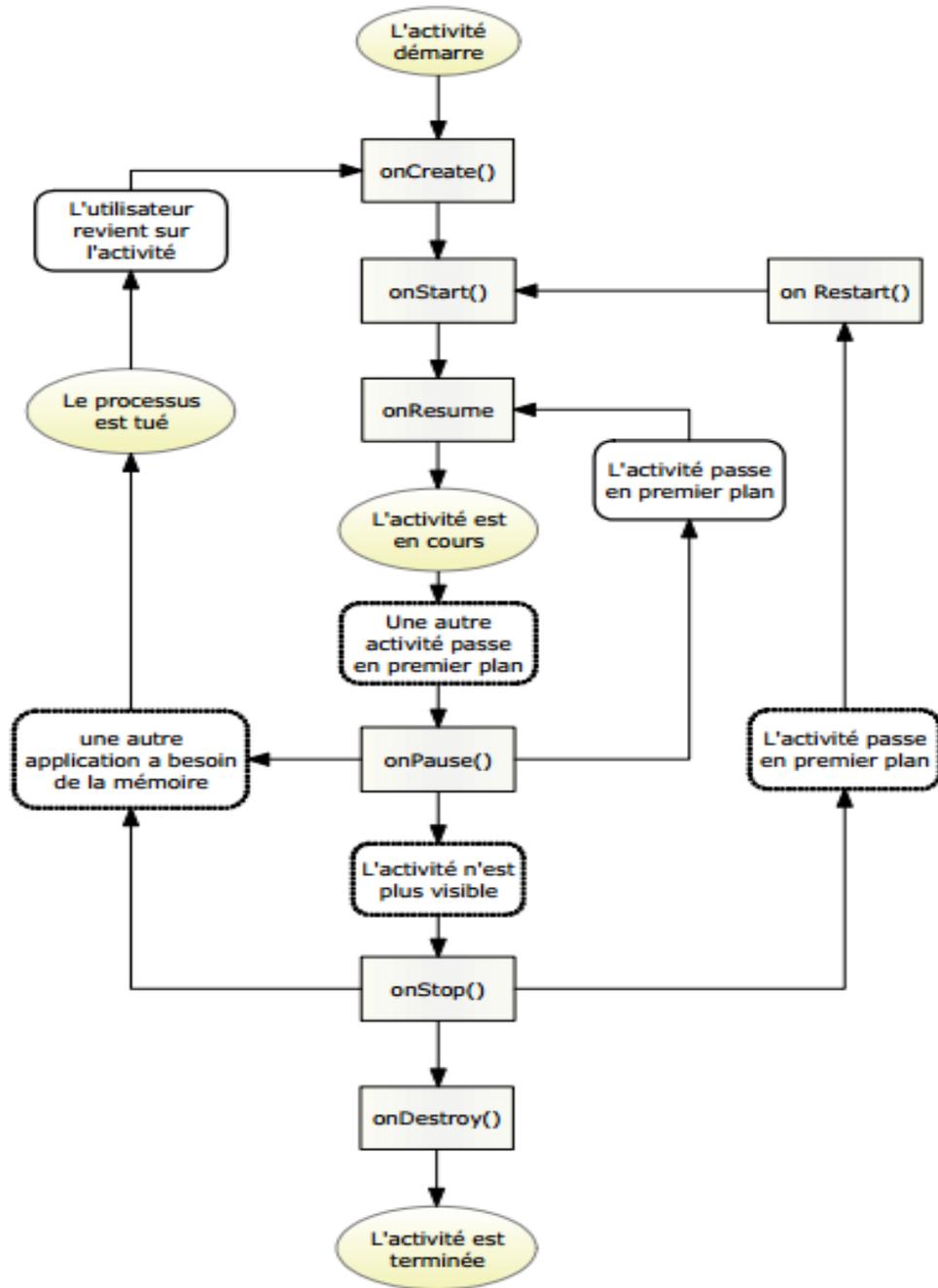


FIGURE 2.2 – Organigramme du Cycle de vie d'une application Android.

2.3 Conclusion

Dans ce chapitre, nous avons fait une étude du Systèmes d'Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et son architecture, à savoir les principaux composants du système. Ceci nous a permis de comprendre les capacités de ce système pour les exploiter par la suite.

Présentation du systeme et Conception

3.1 Introduction

Dans ce chapitre, nous présenterons l'idée principale de l'application mobile SSB-UB (Service Scolarité et Bibliothèque de l'université de Bejaia), pour cela nous avons divisé le travail en deux parties ;

La première partie, intitulée 'Présentation du système' ; explique les objectifs de l'application et les services qu'elle offre aux étudiants qui sera suivie de son architecture général.

Dans la deuxième partie nous nous sommes basés sur la méthode UP (processus unifié) pour présenter l'analyse et la spécification des besoins, suivie de la conception de l'application.

L'étape d'analyse et de spécification des besoins offre une description de l'application mobile SSB-UB suivie des besoins fonctionnels et non fonctionnels.

L'étape de conception consiste à suivre les étapes nécessaires pour concevoir un système de services d'aide aux étudiants sous forme d'application mobile sous Android. Pour cela, nous allons utiliser UML 2 comme langage de modélisation car il offre une souplesse remarquable qui s'exprime par la possibilité d'obtenir des modèles de systèmes reflétant la réalité en utilisant les diagrammes les plus important : digramme de cas d'utilisation diagramme de collaboration, diagramme de séquence et le diagramme de classe .

Enfin nous achèverons le chapitre avec une conclusion.

Première partie

Présentation du système

3.2 Introduction

Vue la diversité des services au sein de l'université A.Mira nous avons opté pour la conception et la réalisation d'un système de services d'aide aux étudiants sous forme d'application mobile afin d'offrir la possibilité d'avoir accès aux différents services et interagir avec ces derniers via leurs terminaux mobiles.

Dans cette partie, nous détailleront les objectifs de notre projet ainsi que les services offerts par l'application mobile SSB-UB , suivie de son architecture globale.

3.3 Objectifs de l'application mobile SSB-UB

L'université A. Mira offre aux étudiants différents services : service bibliothèque, service scolarité, service CNAS, service COUS etc. Afin de rendre les services faciles d'accès aux étudiants nous avons opté pour la conception d'une application mobile pour les deux services les plus importants et les plus utilisés qui sont, le service scolarité et le service bibliothèque ainsi ces derniers seront disponibles sur les terminaux mobiles des étudiants.

Dans l'application mobile SSB-UB le service scolarité offre aux étudiants différents services qui sont :

1. *Le service scolarité*

L'université met à la disposition de ses étudiants un service très important qui est le service Scolarité.

Dans l'application mobile SSB-UB ,nous nous focaliserons sur les services qu'offre le service scolarité au sein du département d'informatique.

Le service scolarité permet aux étudiants de gérer leurs démarches administratives depuis leurs entré à l'université (inscriptions) jusqu'à leurs sortie (délivrance des diplômes), tout en passant par l'organisation de ses emplois du temps, procès verbaux de matières , aussi il permet de gérer les inscriptions, préinscriptions, conventions de stage, édition des certificats de scolarité, l'édition et la réédition des cartes d'étudiants (en cas de vol ou perte) et il met à la disposition des étudiants la

documentation dont il a besoin (programmes d'étude, supports de cours, planning des examens, PV de délibérations de matières, les thèmes PFE, conventions et demandes de stages, avis ou notes importantes).

Dans l'application mobile SSB-UB le service scolarité offre aux étudiants différents services qui sont :

- Service de Recherche de ressources

Ce service permettra de rechercher des ressources qui existent dans la base de données du système en introduisant un mot clef.

Une fois que la première lettre du mot clef est saisit, une liste de ressources portant cette lettre sera affichée, ainsi l'utilisateur aura le choix entre choisir le nom de la ressource parmi la liste affichée ou bien de saisir le mot clef au complet.

Une fois que la ressource trouvée, la fiche technique de la ressource sera affichée, cette fiche technique comportera le nom, la description et la date de publication de cette ressource.

Aussi l'étudiant pourra au lieu de rechercher une ressource à travers ce service ; il pourra accéder à un service optionnel ou la liste des ressource sera répartie comme suit :

- **Planning** :tels que les plannings des examens (session normal et session rattrapage) et planning des emplois du temps.
- **PV** :les PV de délibérations de matière ou de soutenances
- **Avis** : Avis ou Notes importantes de réunion ou CP, avis de formation ou concours.
- **La liste Informations** :Elle regroupe la liste des actualités au sein de toute l'université, tel que l'affichage de conférences, l'affichage de demande à candidatures.
- **Autre** : Calendrier de réception des étudiants, conventions et demandes de stages, les thèmes PFE.

Ces ressources peuvent être sous différents formats : Word, Excel, PDF.

- Service de Consultation de ressources

Ce service a pour rôle de récupérer les documents en les téléchargeant à partir de la base de données du système.

Le téléchargement se fait soit en recherchant la ressource à partir d'une liste exis-

tante et une fois qu'on clique sur la ressource voulue l'URL sera automatiquement chargé dans le champ de l'URL ou bien en introduisant directement l'URL de la ressource (si on l'a déjà).

- Service de Téléchargement des ressources

Ce service a pour rôle de récupérer les documents en les téléchargeant à partir de la base de données du système.

Le téléchargement se fait soit en recherchant le ressource à partir d'une liste existante et une fois qu'on clique sur la ressource voulue l'URL sera automatiquement chargé dans le champ de l'URL ou bien en introduisant directement l'URL de la ressource (si on l'a déjà).

- Service Messagerie

La messagerie va permettre aux étudiants de s'échanger des messages différés à travers la boîte de messagerie.

L'envoi d'un message se fait en choisissant l'identificateur du destinataire à partir d'une liste prédéfinie, cette liste contient seulement les étudiants qui sont du même niveau que l'utilisateur.

Après avoir sélectionné le destinataire, le numéro de téléphone de ce dernier sera importé mais crypté, ainsi, l'utilisateur ne pourra pas voir le numéro de téléphone du destinataire.

- Service Informations

Ce service permettra d'avoir l'actualité sur les événements qui se passent à l'intérieur de l'université : Affichage des dates des examens, affichage du PV général, publications de Notes importantes et des avis aux étudiants etc.

- Service Réclamations

Ce service permettra à l'étudiant de faire leurs réclamations : dans les cas où l'étudiant fait une réclamation à destination de son enseignant (erreur dans l'affichage des notes de l'étudiant) la requête se passera comme suit : l'étudiant va envoyer un message au secrétariat qui se chargera de retransmettre la réclamation

à l'enseignant concerné, erreurs dans les papiers (certificats de scolarité, relevés de notes).

La réclamation se fait en saisissant l'objet et la description du message de la réclamation à envoyer.

2. Le service bibliothèque

L'université A. Mira met à la disposition de ses étudiants un service qui est primordial au bon déroulement de leur cursus scolaire qui est le service Bibliothèque.

Le Service Bibliothèque permet de stimuler les étudiants, faciliter leurs études, les initier à l'emploi des instruments de recherche documentaire, aussi répondre aux besoins de documentation des étudiants ainsi qu'aux exigences et besoins des programmes d'enseignement et de la possibilité de rechercher des ouvrages à travers des espaces réservés pour cela et ainsi d'en emprunter au besoin.

L'application mobile SSB-UB consiste à concevoir un système d'information permettant d'aider à la gestion de la Bibliothèque et plus précisément; aider à la gestion des emprunts, réservations, renouvellement d'emprunt et retours des ouvrages et cela va se faire par l'intermédiaire des terminaux mobiles pour éviter l'encombrement dans la bibliothèque et permettre un gain de temps dans la gestion de cette dernière.

Le service Bibliothèque dans le cas de l'application mobile SSB-UB offre les services suivants :

- Service de Recherche d'ouvrages

Ce service permettra à l'étudiant de rechercher un ouvrage dans la base de données de La bibliothèque et ainsi avoir accès à la fiche technique de ce dernier.

La recherche se fera en saisissant les informations concernant l'ouvrage : le titre de l'ouvrage ou l'auteur de l'ouvrage ou le sujet traité par cet ouvrage.

La recherche pourra aussi se faire d'une manière avancée et cela, en saisissant le titre de l'ouvrage ainsi que le sujet de l'ouvrage pour avoir un résultat de recherche plus précis.

- Service de réservation et d'annulation de réservation d'un ouvrage

Ce service permettra à l'étudiant de réserver un ouvrage

-Dans le cas où l'ouvrage est indisponible, l'étudiant le réserve en l'attente de sa disponibilité, ainsi dès que l'ouvrage sera disponible, une alerte sera envoyée à l'étudiant par le système.

Une fois que l'étudiant aura récupéré l'ouvrage ; le statut de ce dernier passera de " Réserve " à " emprunté ".

La réservation ne sera plus valable après 24h du moment où cet ouvrage est disponible et ainsi il sera attribué au prochain étudiant dans la liste de réservation.

Ce service permettra aussi à l'étudiant d'annuler la réservation d'un ouvrage en accédant au service d'annulation d'ouvrages , une fois qu'il y aura accès il aura à saisir la cote de l'ouvrage et annuler la réservation de l'ouvrage , ainsi cet ouvrage sera attribué à l'étudiant suivant dans la liste de réservations.

L'ouvrage en question peut être : un livre, un article ou un mémoire.

- Service Alerte

Ce service permettra à l'étudiant de voir les alertes qui lui sont envoyées par l'administrateur de la bibliothèque par exemple dans le cas où l'adhérent dépasse la date limite de l'emprunt ou dans le cas où un ouvrage qu'il a déjà réservé est disponible.

- Service Renouvellement d'un Emprunt

Ce service permettra aux étudiants de demander à renouveler l'emprunt d'un ouvrage dans le cas où il n'a pas été réservé, ce renouvellement pourra se faire 24h avant la date limite de l'emprunt de l'ouvrage.

Le renouvellement de l'emprunt pourra se faire si d'autres exemplaires de l'ouvrage existent sinon il ne pourra pas l'emprunter.

- Service Suggestion d'ouvrage

Ce service permettra aux étudiants de suggérer l'acquisition d'un ouvrage aux administrateurs du service de la bibliothèque et cela en remplissant un formulaire ou l'étudiant doit introduire plusieurs champs (nom de l'auteur, titre de l'ouvrage, maison et année d'édition, numéro ISBN) et l'envoyer.

- Service Noter un ouvrage

Ce service permettra aux étudiants de donner leur avis sur un ouvrage.

Cela se fait en saisissant la cote de l'ouvrage qu'on veut noter et choisir dans une liste la note à attribuer à cet ouvrage.

La note de l'ouvrage va varier de 1 à 5.

Cela va servir à aider l'administrateur à faire des statistiques sur la qualité de l'ouvrage et les trier selon la note qui sera donné par l'adhérent, ainsi si l'ouvrage obtient une bonne note alors l'administrateur pourra rajouter d'autres exemplaires de cette ouvrage et dans le cas contraire, l'ouvrage sera résilié de la bibliothèque.

3.4 Architecture de l'application mobile SSB-UB

Le schéma ci-dessous - figure 3.1 - illustre l'architecture de l'application mobile SSB-UB et les services qui y sont disponibles.

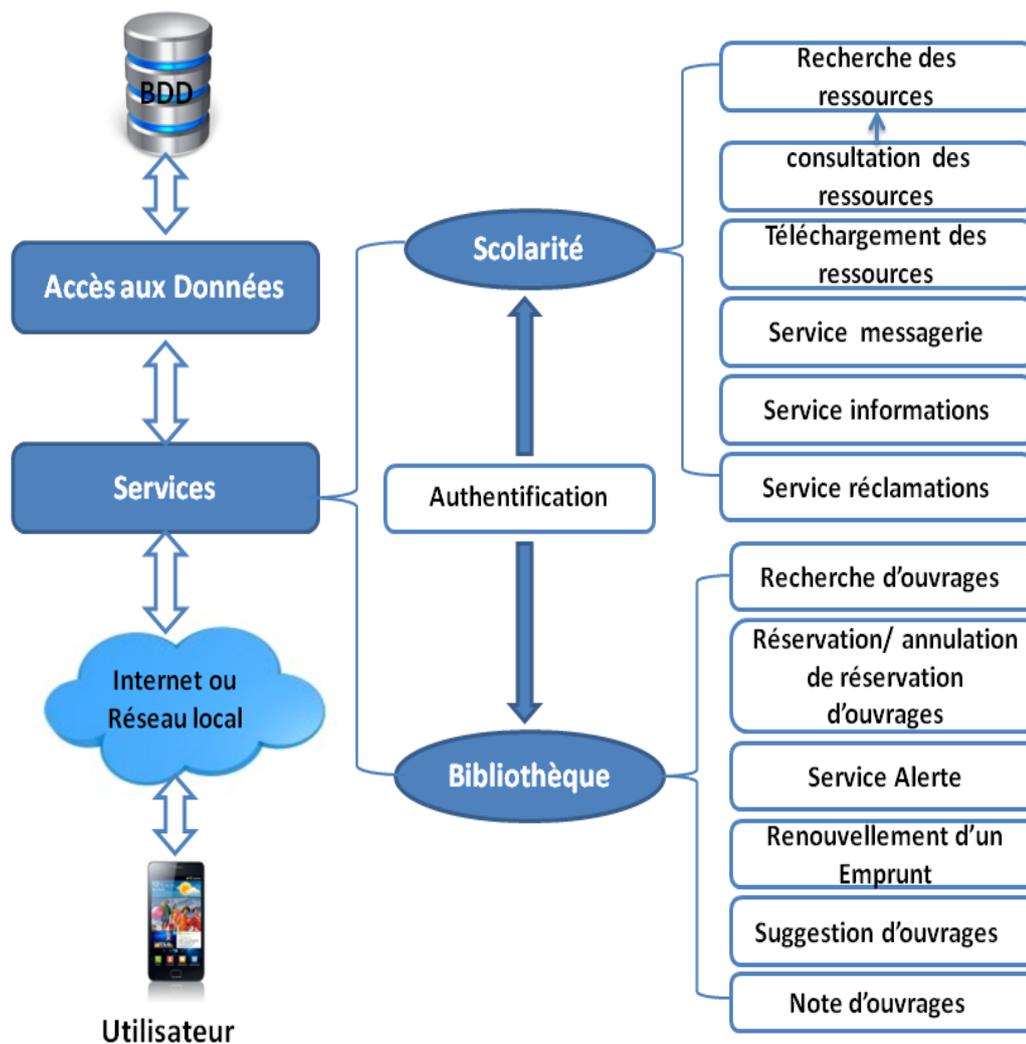


FIGURE 3.1 – Architecture globale de l'application mobile SSB-UB .

3.5 Apport du système mobile par rapport au système classique des deux services scolarité et bibliothèque

A travers une analyse comparative faite entre les services classiques et mobiles du service scolarité et du service bibliothèque, nous avons déduit les tableaux illustrés ci-dessous

Le Tableau 3.1 ci-dessous illustre l'apport du service scolarité mobile par rapport au service scolarité classique.

Service scolarité classique	Service scolarité mobile
Service accessible à travers le site web de l'université dans la rubrique E-Learning.	Service accessible sur un terminal mobile sous un environnement Android .
Pas de service d'alerte.	Disponibilité du service d'alerte.
Indisponibilité du service de réclamations.	Disponibilité du service de réclamations.
Tout utilisateur peut accéder au service	L'accès au service nécessite l'authentification ce qui garantie la confidentialité des ressources.

TABLE 3.1 – Tableau illustrant l'apport du service scolarité mobile par rapport au service scolarité classique.

Le Tableau 3.2 ci-dessous illustre l'apport du service bibliothèque mobile par rapport au service bibliothèque.

Service bibliothèque classique	Service bibliothèque mobile
Le service réservation d'ouvrages n'existe pas dans le site web de l'université	le service réservation d'ouvrages existe dans l'application mobile
Le service ne nécessite pas d'authentification.	Le service nécessite une authentification.
Le service noter ouvrage n'existe pas dans le site de l'université.	Le service noter ouvrage est accessible sur l'application mobile .
Le service renouveler emprunter un ouvrage n'existe pas dans le site de l'université.	Le service renouveler ouvrage est accessible sur l'application mobile .

TABLE 3.2 – Tableau illustrant l'apport du service bibliothèque mobile par rapport au service bibliothèque classique.

3.6 Conclusion

Dans cette partie, nous avons présenté les objectifs de l'application mobile SSB-UB et détaillé les services offerts aux étudiants ainsi que son architecture globale.

Ainsi on a pu conclure les intérêts que pourra offrir l'application mobile SSB-UB aux étudiants, tel que : la mobilité du service, le gain de temps, facilité d'accès aux informations, moins d'encombrement dans les deux services au sein de l'université etc.

Dans la partie suivante, nous allons entamer la conception de l'application mobile SSB-UB .

Deuxième partie

Conception

3.7 Introduction

Pendant plusieurs décennies, le monde informatique a toujours rêvé d'un processus qui puisse garantir le développement efficace de logiciels de qualité, valable quelque soit la grandeur et la complexité du projet, et présentant de bonnes pratiques adaptées à la méthode en question, surtout que, de nos jours, les logiciels demandés sont de plus en plus imposants et exigeants qu'auparavant.

Le processus unifié semble être la solution idéale pour remédier à l'éternel problème des développeurs. En effet, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en un système logiciel quelque soit la classe, la taille et le domaine d'application de ce système.

Le processus unifié utilise et vient compléter le langage UML (Unified Modeling Language). Ce langage de modélisation est une partie intégrante du processus unifié, ils ont été d'ailleurs développés de concret. Nous allons d'abord présenter le processus unifié, ensuite nous reviendrons sur la présentation de l'UML dont les diagrammes seront utilisés dans chaque phase et activité du processus unifié.

3.8 Le Processus Unifié

3.8.1 Définition d'un processus de développement logiciel

Un processus définit une séquence d'étapes, partiellement ordonnées, qui participent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.[\[38\]](#)

L'objectif d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans un court délai et à des coûts moindres.

3.8.2 Définition du processus unifié (UP)

Pour définir le processus unifié, nous allons simplement définir les deux termes qui le composent :

- **Processus** : Suite continue d'opérations constituant la manière de fabriquer. En d'autres termes, c'est une succession de tâches dans le but d'accomplir un travail, un

projet.

- **Unifié** : Participe passé du verbe unifier, être amené à l'unité, se fondre en un tout. En fait, les méthodes d'analyse et de conception orientées objet, étaient variées jusqu'à ce que RAMBAUGH, JACOBSON et BOOCH eut l'idée de les unifier.[40]

Le processus de développement d'une application passe par les étapes suivantes :

- Analyse et spécification des besoins ;
- Conception ;
- Implémentation ;
- Test.

Dans ce chapitre, nous développerons uniquement les deux premières étapes, à savoir l'analyse et la spécification des besoins et la conception, les deux autres étapes seront développées dans le chapitre suivant.

3.9 Analyse et spécification des besoins

L'analyse et la spécification des besoins représentent la première phase du cycle de développement d'un logiciel. Elle sert à identifier les acteurs réactifs du système et leur associer à chacun l'ensemble d'actions avec lesquelles il intervient dans l'objectif de donner un résultat optimal et satisfaisant au client.

3.9.1 Description de l'application SSB-UB

Notre projet consiste à développer une application mobile sous Android qui facilite l'accès aux services de la scolarité (aux étudiants du département d'informatique de l'université A. Mira) et aux services de la bibliothèque (à tous les étudiants de l'université d'A. Mira) ; via un terminal mobile .

Ces services ainsi que la manière d'y accéder ont été définie auparavant dans la partie 'Présentation du système' et notre effort sera orienté vers l'extension de l'accès à ces services sur les terminaux mobiles fonctionnant avec le système Android.

3.9.2 Les besoins Fonctionnels

L'application SSB-UB offre deux services principaux, le service scolarité et le service bibliothèque. Avant d'accéder aux deux services, il faut d'abord que l'étudiant s'authentifie .

L'authentification dans l'application mobile SSB-UB permet de s'assurer de l'identité de l'utilisateur et de garantir la sécurité des données.

Ces deux Services offrent plusieurs sous services qui sont cités ci-dessous :

3.9.2.1 Le service " Scolarité "

Le service Scolarité offre six services qui sont comme suit :

✓ **Service 'Recherche de ressources'** : Ce service permettra aux étudiants de rechercher des ressources ainsi il aura comme options de consulter ou télécharger la ressource.

✓ **Service 'Consultation des ressources'** : Ce service permettra aux l'étudiants de consulter les différentes ressources qui existent dans la base de données.

✓ **Service 'Téléchargement des ressources'** : Ce service permettra aux étudiants de télécharger les différentes ressources qui existent dans la base de données.

✓ **Service 'Messagerie'** : Ce service permettra aux étudiants, l'échange de messages entre eux de manière indifféré.

✓ **Service 'Informations'** : Ce service permettra aux étudiants d'avoir accès à l'actualité sur toutes les actions qui se passent à l'intérieur de la scolarité du département

d'informatique.

✓ **Service 'Réclamations'** : Ce service permettra aux étudiants de faire des réclamations.

3.9.2.2 Le service " Bibliothèque "

Dans le service bibliothèque, l'accès au service ne se fait pas directement comme pour le service scolarité, La condition c'est que l'étudiant soit d'abord adhérent au sein de la bibliothèque donc quand un étudiant essaye d'accéder au service, le serveur vérifie automatiquement si l'étudiant figure dans la base de données de la bibliothèque ou non, si l'étudiant existe ; alors il aura accès au service, sinon l'accès au service lui sera refusé.

Le service bibliothèque offre six services qui sont comme suit :

✓ **Service 'Réservation et Annulation de Réservation d'un Ouvrage'** : Ce service permettra aux adhérents de réserver des ouvrages (au maximum 3 ouvrages) ou d'annuler la réservation d'ouvrages.

✓ **Service 'Recherche d'ouvrages'** : Ce service permettra aux adhérents de rechercher un ouvrage dans la base de données de la bibliothèque.

✓ **Service 'Alerte'** : Ce service permettra d'informer les adhérents à travers une alerte.

✓ **Service 'Renouvellement d'un Emprunt'** : Ce service permettra aux adhérents de renouveler l'emprunt d'un ouvrage.

✓ **Service 'Suggestion d'ouvrages'** : Ce service permettra aux adhérents de suggérer l'acquisition d'un ouvrage aux administrateurs du service de la bibliothèque.

✓ **Service 'Noter Ouvrages'** : Ce service permettra aux adhérents de donner une note aux ouvrages, en lui attribuant un numéro sur une échelle de 1 à 5.

3.9.3 Les besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement.

1. Existence d'un réseau Wifi (Réseau sans fil) ;
2. Existence d'un serveur de Données avec le SGBD MySQL ;
3. Ergonomie et souplesse : 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 ;
4. *Efficacité* : l'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur ;
5. L'application doit optimiser les traitements et l'utilisation de ses ressources (autonomie batterie, mémoire disponible...) afin de ne pas épuiser ses dernières dans des requêtes inutiles.
6. L'accès à la base de données doit être souple et rapide.
7. Rapidité du traitement : l'application doit assurer une rapidité de traitement pour qu'elle s'approche au maximum d'une exécution en temps réel.

3.10 Conception

3.10.1 Présentation d'UML

UML (Unified Modeling Language), se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML modélise l'ensemble des données et des traitements en élaborant des différents diagrammes.[41]

3.10.2 Présentation des Diagrammes UML

UML dans sa version 2 s'articule autour de treize types de diagrammes, chacun d'entre eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier.

Ces diagrammes sont regroupés dans deux grands ensembles : les diagrammes structurels et les diagrammes comportementales.

L'ensemble des quatorze types de diagrammes UML peut ainsi être résumé sur la figure 3.2 Ci-dessous :

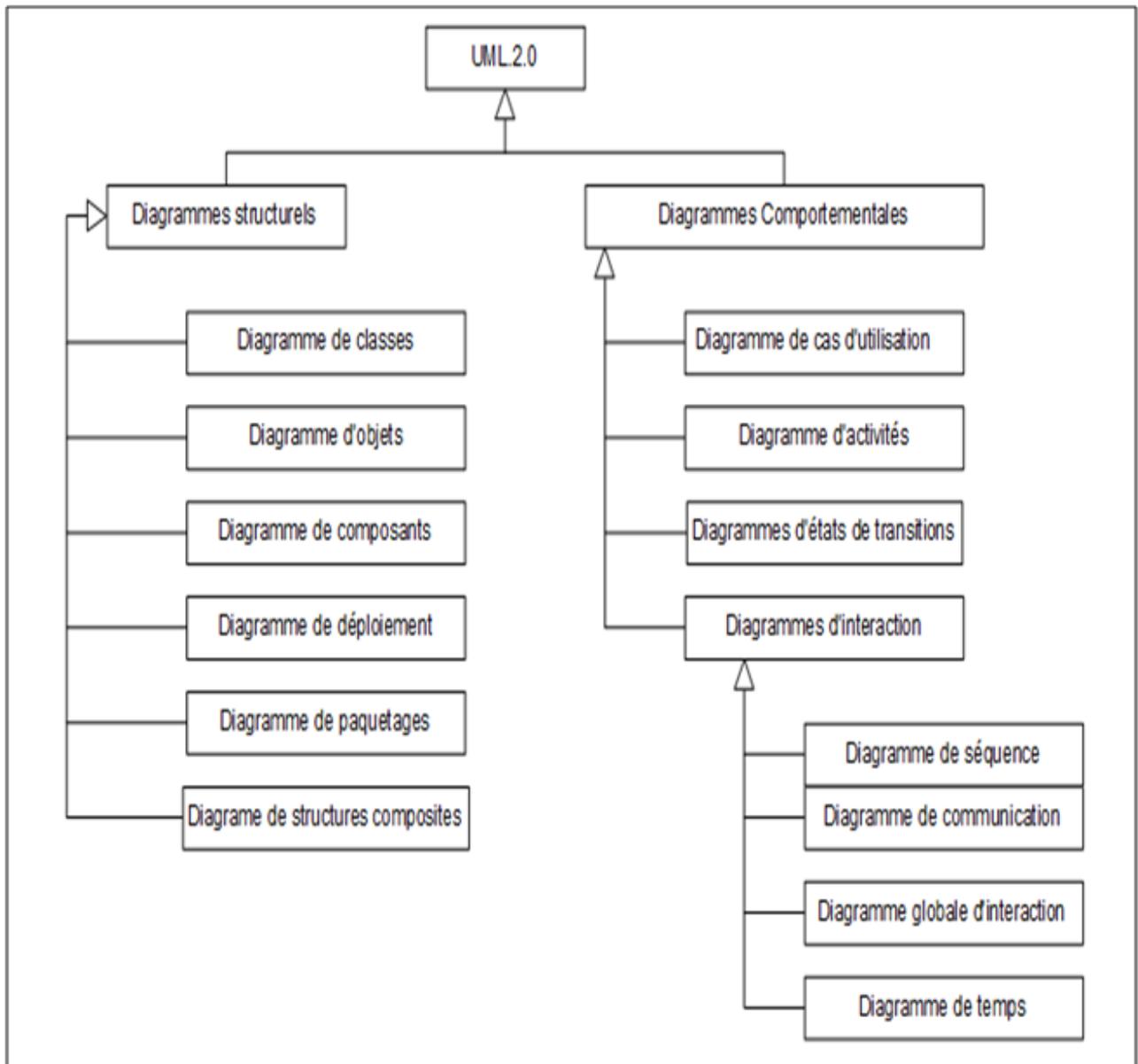


FIGURE 3.2 – La liste des diagrammes UML 2.0.

Nous présentons ci-dessous les diagrammes UML 2.0, que nous avons utilisés dans le cadre de ce projet et quelques notions de base qui leurs sont associées.

3.10.2.1 Définition du diagramme de cas d'utilisation

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système.[35]

Acteur : Représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.[39]

Cas d'utilisation (use case) : Représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.[39]

Les relations entre acteurs : La seule relation entre acteur est la relation de généralisation. Quand un acteur fils hérite d'un acteur père, il hérite en réalité de toutes les associations du père.[39]

Les relations entre cas d'utilisation [39]

- **Relation d'inclusion (Include) :** Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.

- **Relation d'extension (Extend) :** Une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B.

- **Relation de généralisation :** Les cas d'utilisation descendants héritent de la description de leurs parents communs. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires.

3.10.2.2 Définition du diagramme de séquence

Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.[35]

Scénario : Représente une succession particulière d'enchaînements, s'exécutant du début à la fin du cas d'utilisation, un enchaînement étant l'unité de description de séquences d'actions.

Ligne de vie : Représente l'ensemble des opérations exécutées par un objet.[35]

Message : Un message est une transmission d'information unidirectionnelle entre deux objets, l'objet émetteur et l'objet récepteur. Dans un diagramme de séquence, deux types de messages peuvent être distingués [35] :

- **Message synchrone :** Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions.
- **Message asynchrone :** Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations.

3.10.2.3 Définition du diagramme de classe

Le diagramme de classe est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet.

Une classe : Représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On peut parler également de type.

Un objet : Est une entité aux frontières bien définies, possédant une identité et encapsulant un état et un comportement. Un objet est une instance (ou occurrence) d'une classe.

Un attribut : Représente un type d'information contenu dans une classe.

Une opération : Représente un élément de comportement (un service) contenu dans une classe.

Une association : Représente une relation sémantique durable entre deux classes.

Une superclasse : Est une classe plus générale reliée à une ou plusieurs autres classes plus spécialisées (sous-classes) par une relation de généralisation. Les sous-classes " Héritent " des propriétés de leur superclasse et peuvent comporter des propriétés spécifiques supplémentaires.

Après avoir expliqué le fonctionnement des différents diagrammes nous allons détailler leur utilisation dans l'analyse et la conception .

3.10.3 Analyse et conception du service " Scolarité "

3.10.3.1 Les cas d'utilisation du service scolarité

Nous illustrons ci-dessous Figure 3.3 , le diagramme du cas d'utilisation générale du service scolarité.

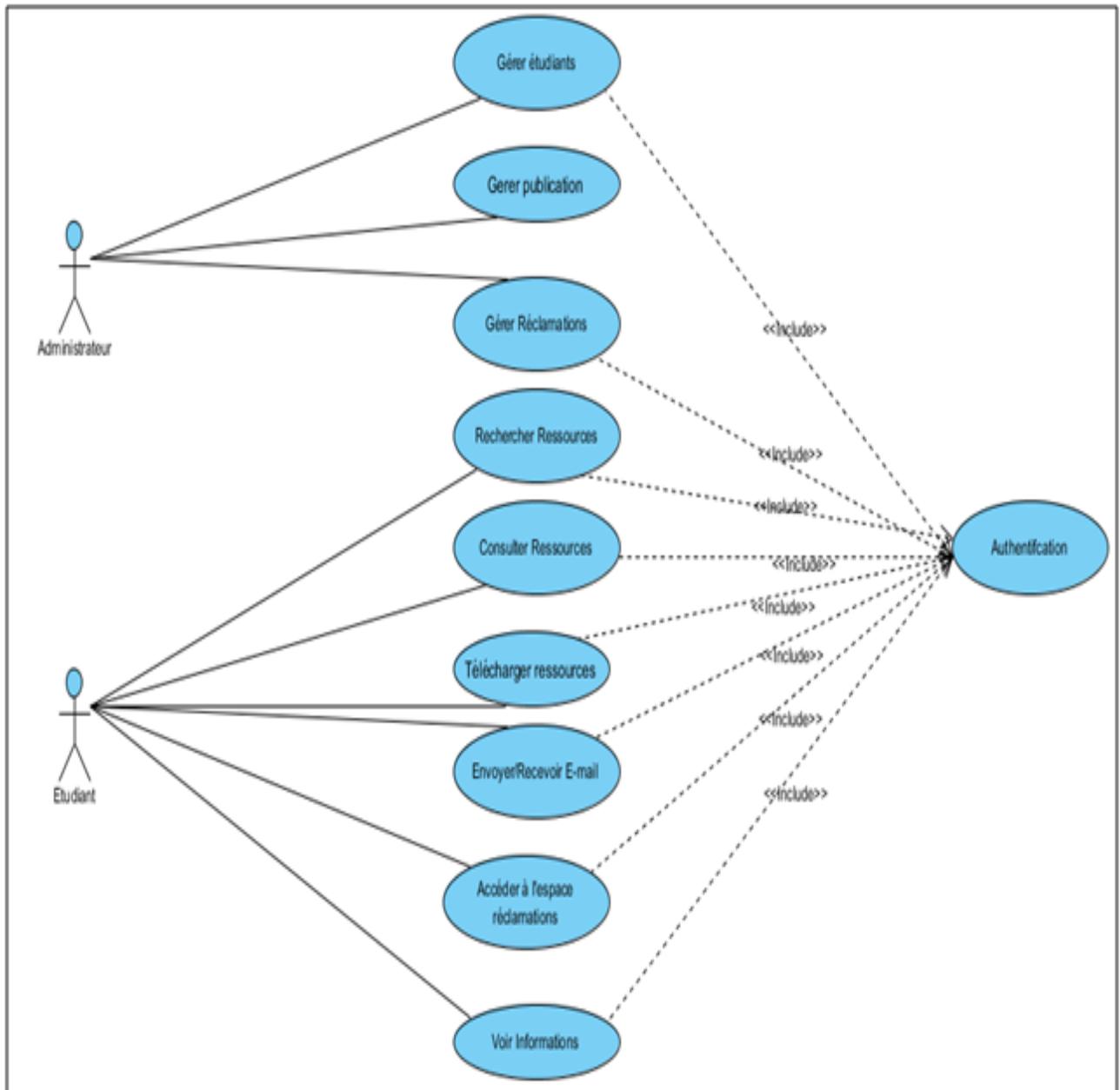


FIGURE 3.3 – Diagramme de cas d'utilisation générale du service scolarité.

1. Identification des acteurs

Dans le cas du service scolarité, on a deux acteurs l'administrateur et l'étudiant .

✓ **L'administrateur** : l'administrateur va gérer les deux services scolarité et bibliothèque et chacun des deux services disposent d'un ou plusieurs administrateurs. Les administrateurs de la bibliothèque vont gérer ce service en utilisant une application java et les administrateurs de la scolarité vont gérer ce service en utilisant un site web conçu pour cela.

✓ **L'étudiant** : l'étudiant peut accéder à tous les services du service scolarité mais ne peut accéder aux services de la bibliothèque que s'il devient adhérent c'est-à-dire que son compte sera confirmé dans la base de données de la bibliothèque.

Afin de mieux éclaircir les diagrammes de cas d'utilisation nous allons faire une description textuelle des cas d'utilisation.

2. Analyse des cas d'utilisations pour l'administrateur

✓ *Cas d'utilisation " s'authentifier "*

- *La Description*

- Le diagramme de cas d'utilisation

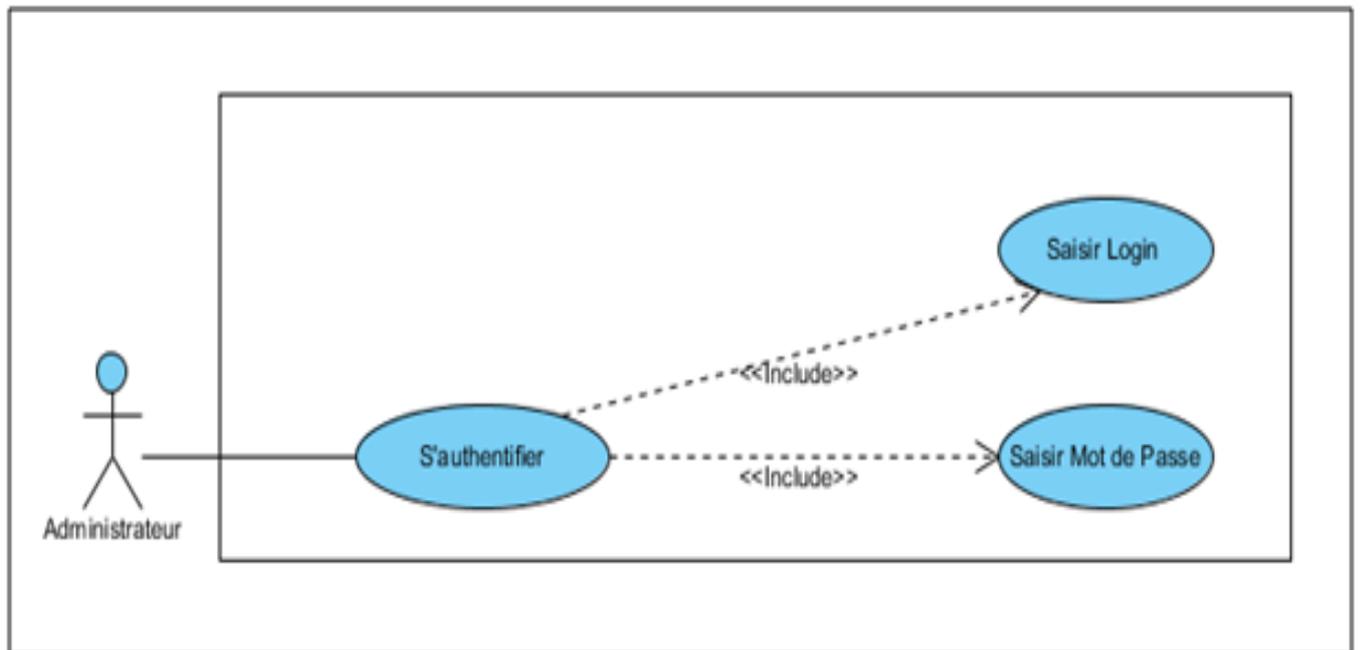


FIGURE 3.4 – Diagramme de cas d'utilisation 'S'authentifier ' pour l'administrateur.

✓ Cas d'utilisation " Gestion des étudiants "

- La Description

- Le diagramme de cas d'utilisation

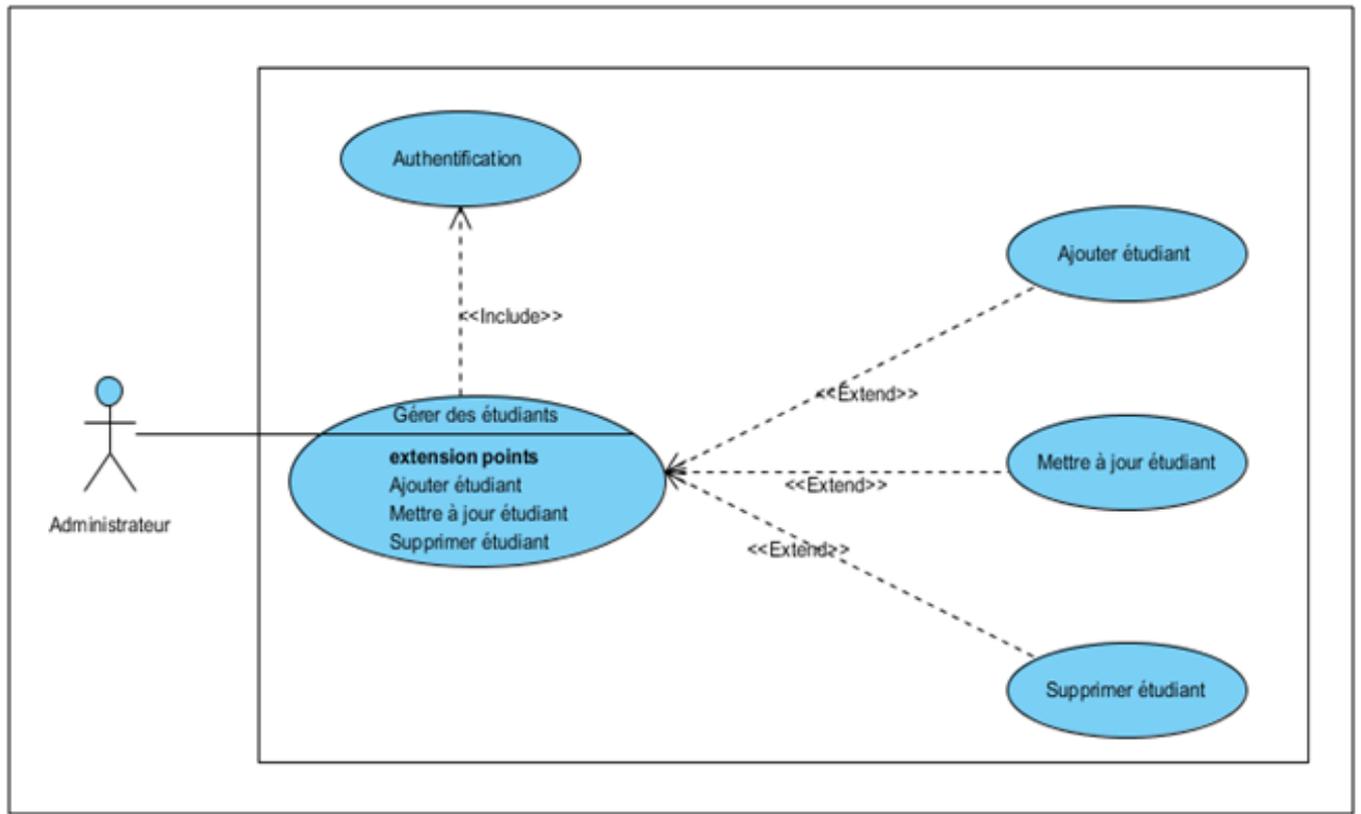


FIGURE 3.5 – Package du cas d'utilisation 'Gestion des Etudiants'.

✓ Cas d'utilisation " Gestion des Publications "

- La Description

- Le diagramme de cas d'utilisation

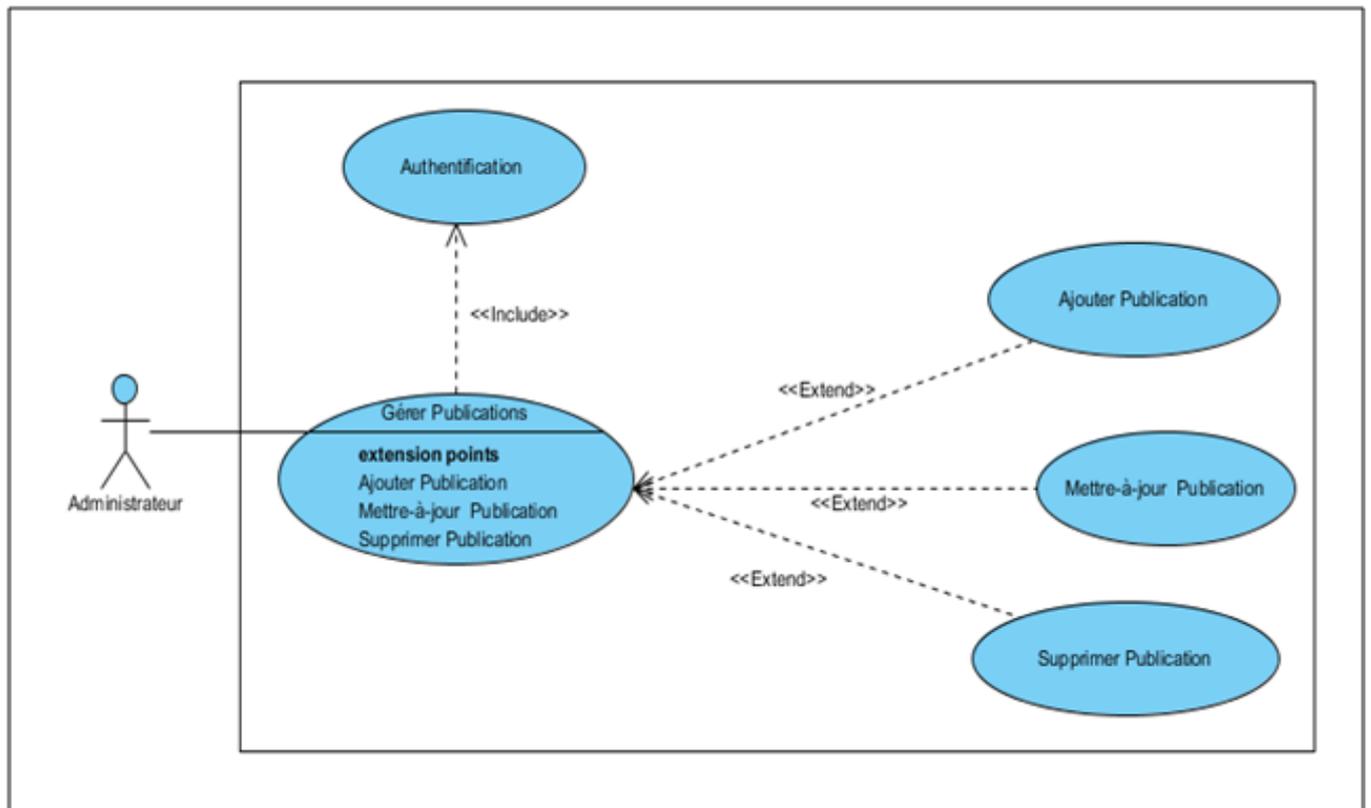


FIGURE 3.6 – Package du cas d'utilisation 'Gestion des Publications'.

3. Analyse des cas d'utilisations pour l'étudiant

✓ Cas d'utilisation " s'authentifier "

- La Description

- Le diagramme de cas d'utilisation

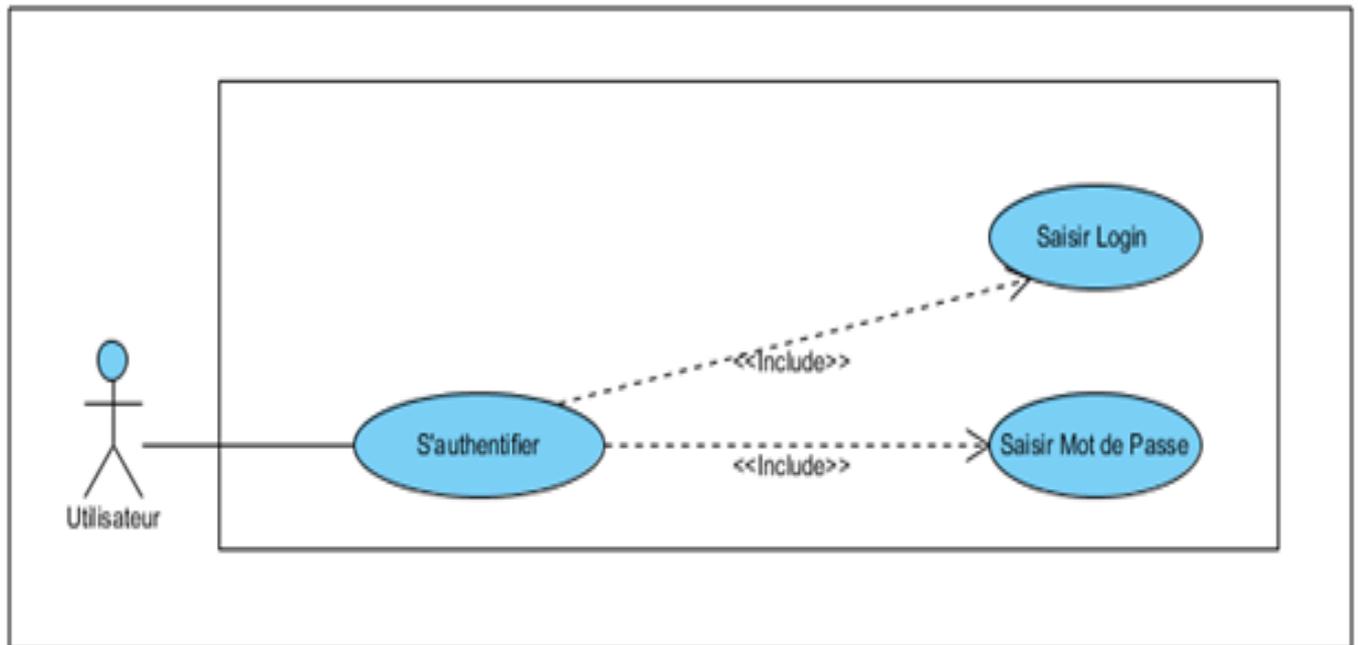


FIGURE 3.7 – Diagramme de cas d'utilisation 'S'authentifier ' pour l'étudiant.

✓ Cas d'utilisation " Rechercher une ressource "

- La Description

- Le diagramme de cas d'utilisation

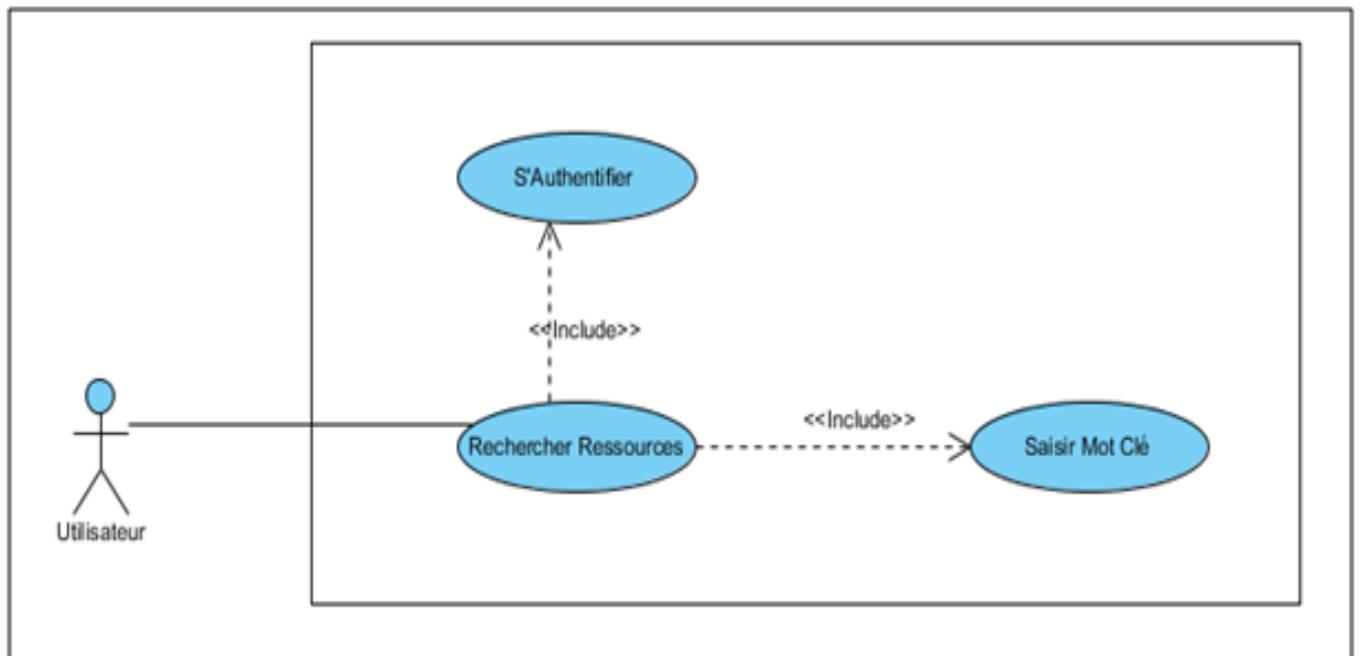


FIGURE 3.8 – Diagramme de cas d'utilisation 'Rechercher une ressource'.

✓ Cas d'utilisation " Consulter une ressource "

- La Description

✓ *Cas d'utilisation " Envoyer/Recevoir un E-mail "*

- *La Description*

- Le diagramme de cas d'utilisation

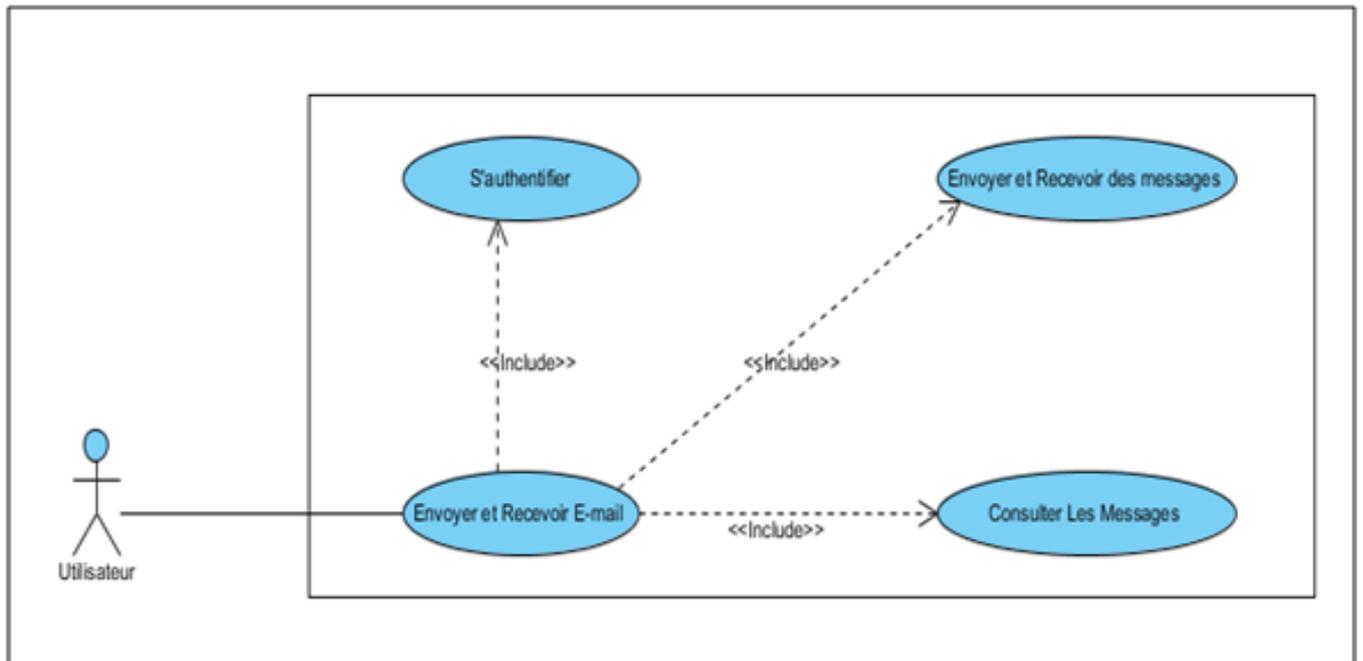


FIGURE 3.9 – Diagramme de cas d'utilisation 'Envoyer/Recevoir un E-mail '.

✓ Cas d'utilisation " Consulter une Information "

- La Description

✓ *Cas d'utilisation " Envoyer une réclamation "*

- *La Description*

3.10.3.2 Les diagrammes de séquence du service scolarité

Nous allons détailler quelques exemples de diagrammes de séquence des cas d'utilisations que nous avons déjà cité au-dessus.

☞ *Diagramme de séquence du cas d'utilisation 'Authentification' de l'utilisateur*

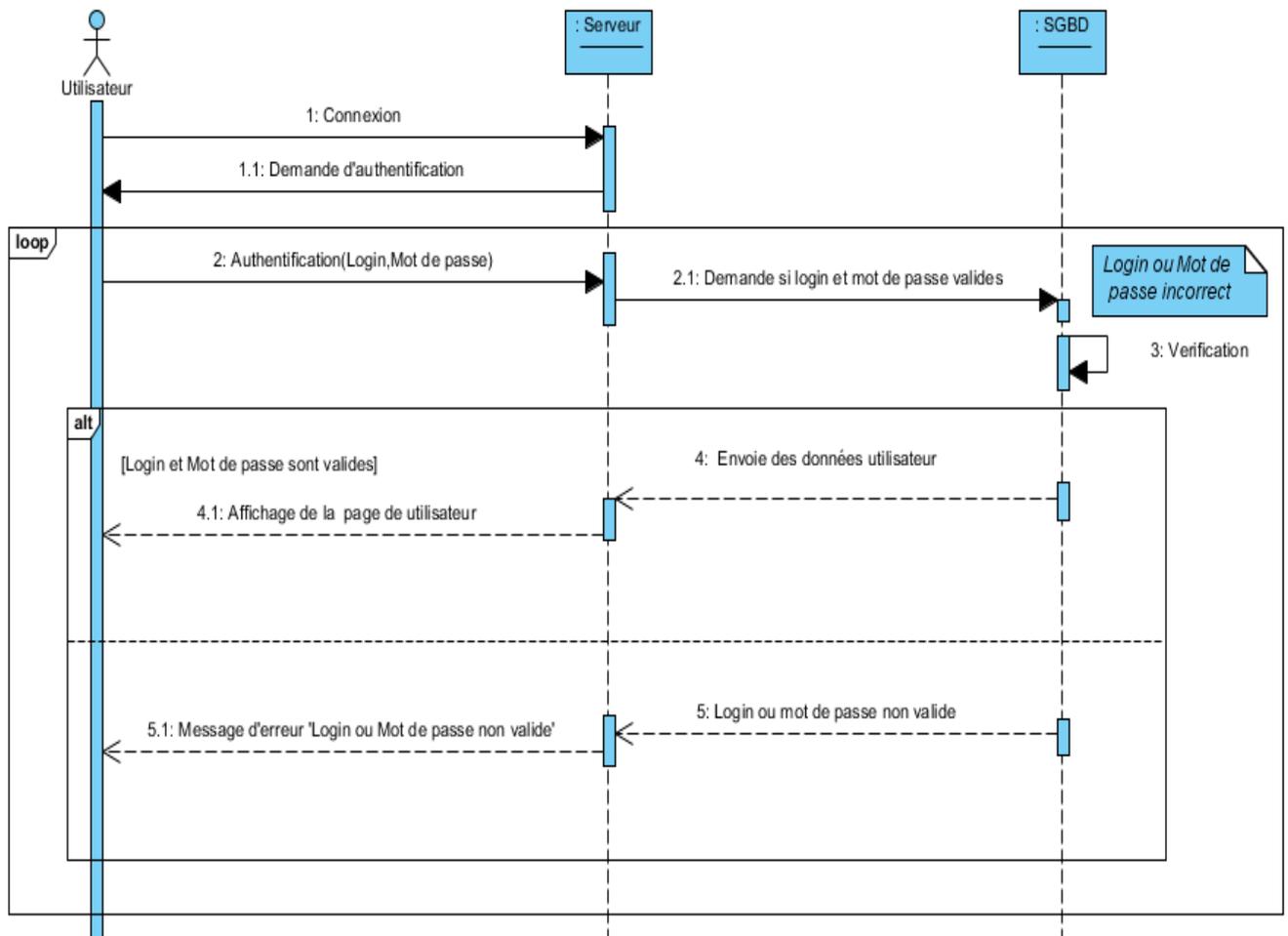


FIGURE 3.10 – Diagramme de séquence 'Authentification'.

☞ Diagramme de séquence du cas d'utilisation 'Rechercher une ressource'

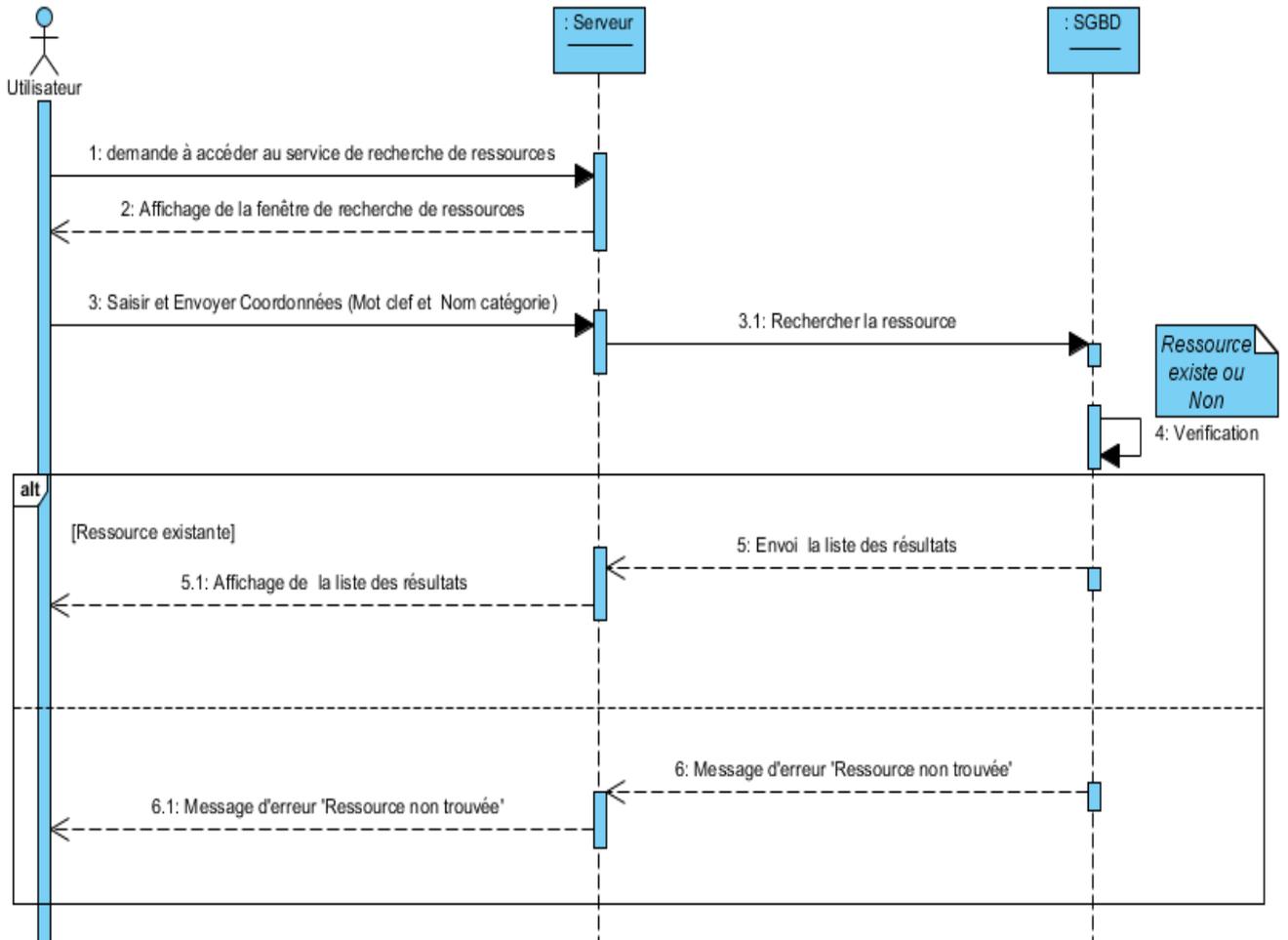


FIGURE 3.11 – Diagramme de séquence 'Rechercher une ressource'.

☞ Diagramme de séquence du cas d'utilisation 'Consulter une ressource'

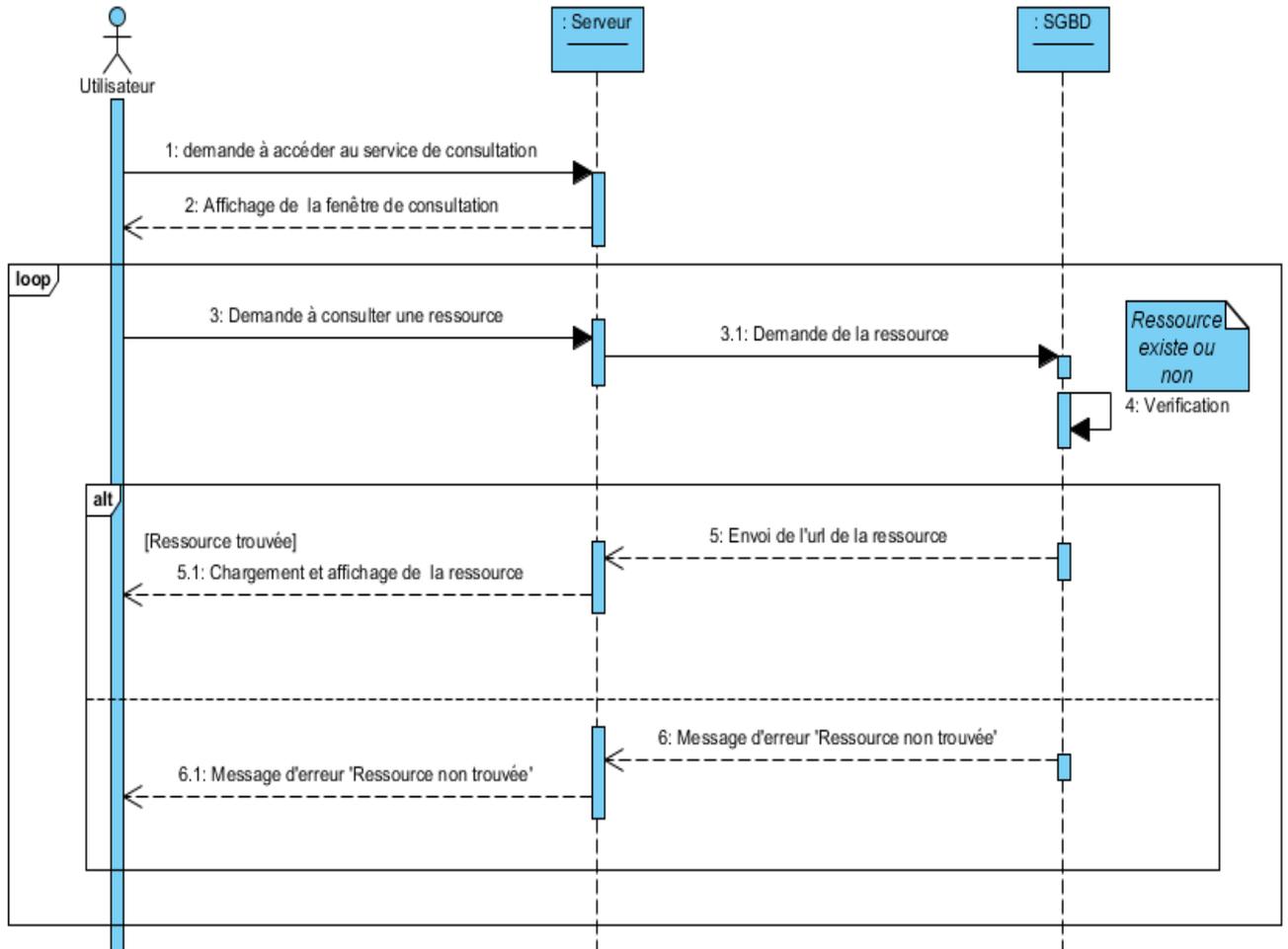


FIGURE 3.12 – Diagramme de séquence ' Consulter une ressource '.

☞ Diagramme de séquence du cas d'utilisation 'Envoyer/Consulter E-mail '

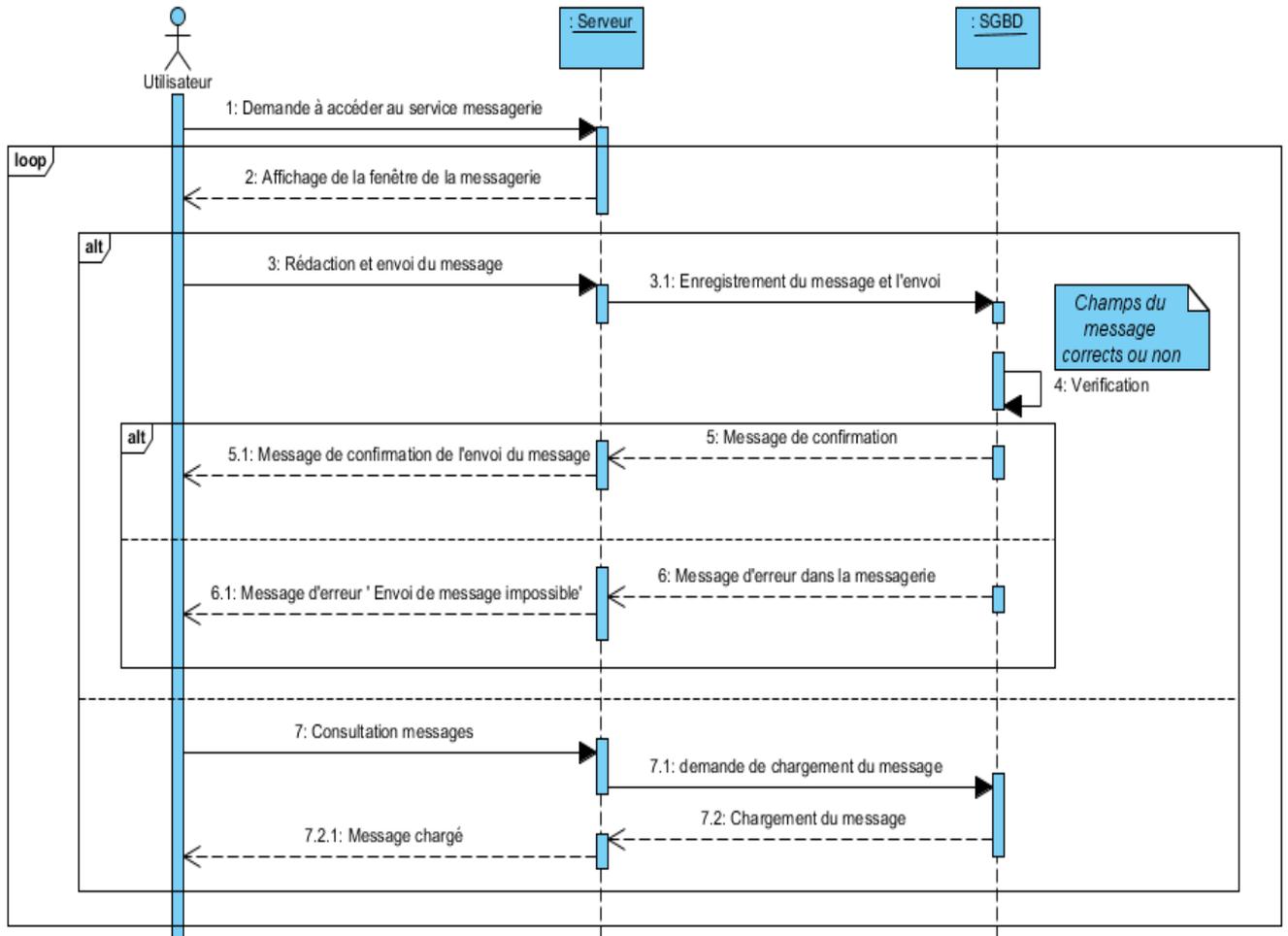


FIGURE 3.13 – Diagramme de séquence ' Envoyer/Consulter E-mail '.

☞ Diagramme de séquence du cas d'utilisation 'Télécharger une ressource '

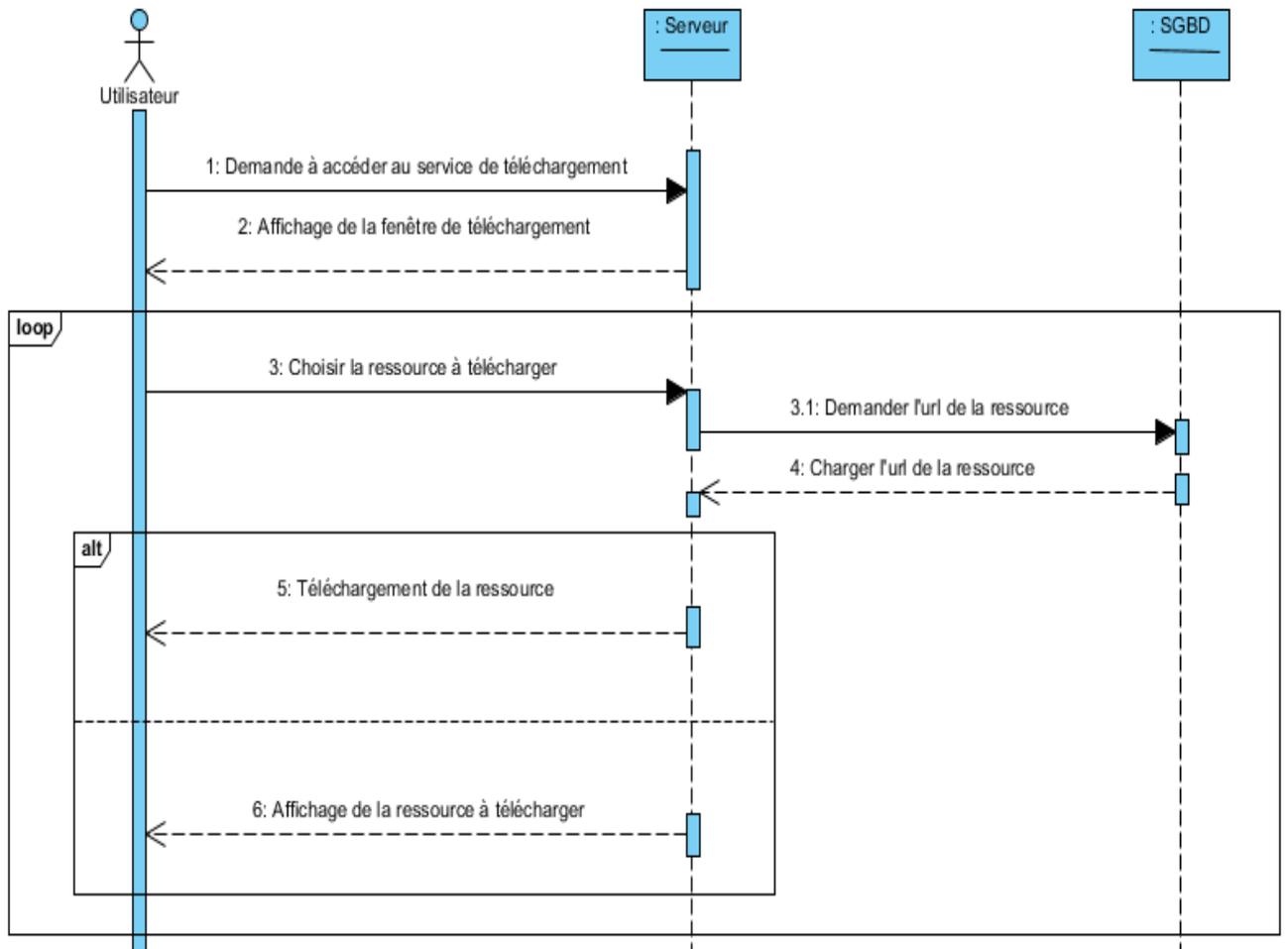


FIGURE 3.14 – Diagramme de séquence 'Télécharger une ressource'.

☞ Diagramme de séquence du cas d'utilisation 'Consulter une information'

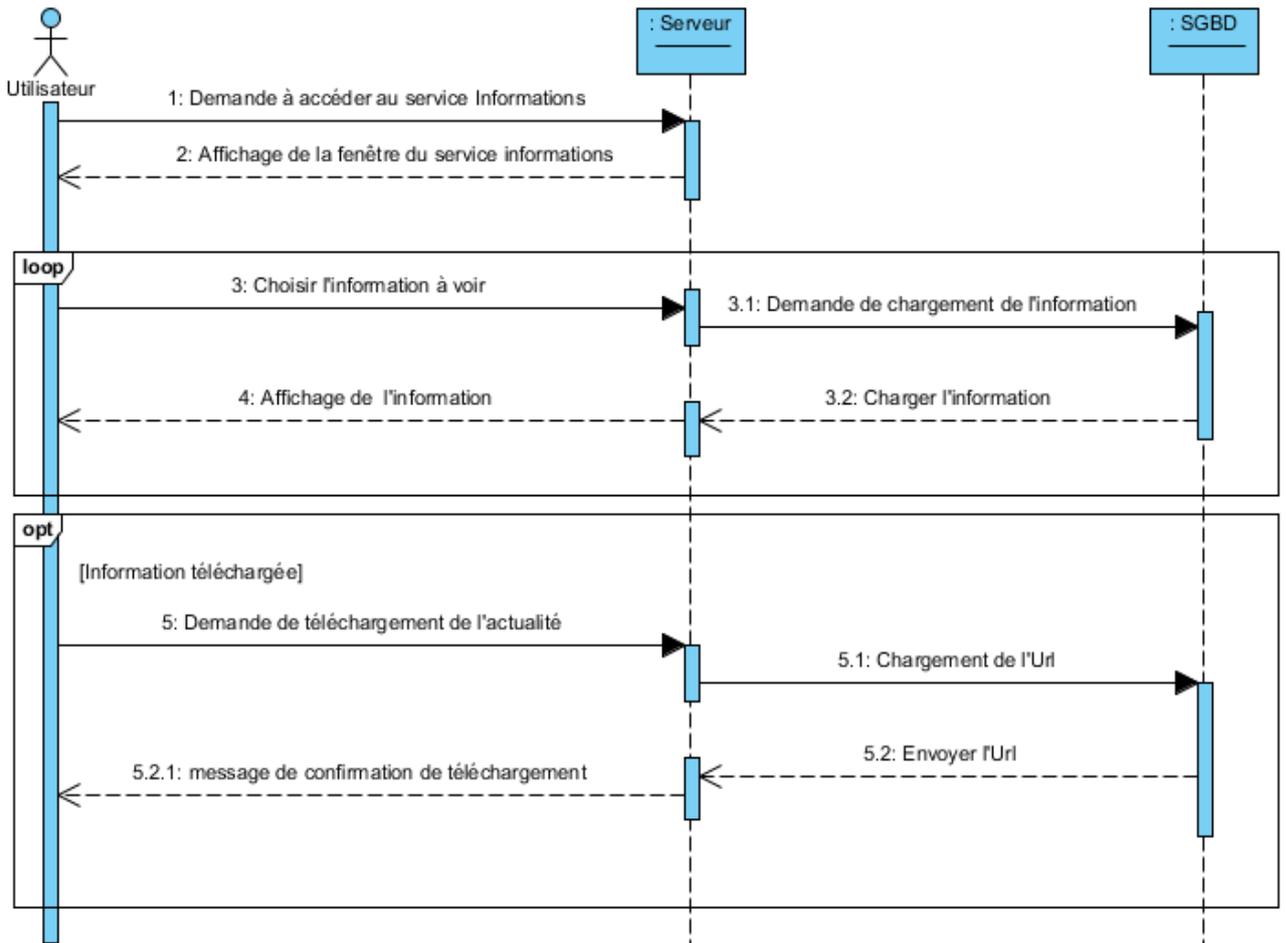


FIGURE 3.15 – Diagramme de séquence 'Consulter une information'.

⚡ Diagramme de séquence du cas d'utilisation 'Envoyer une Réclamation'

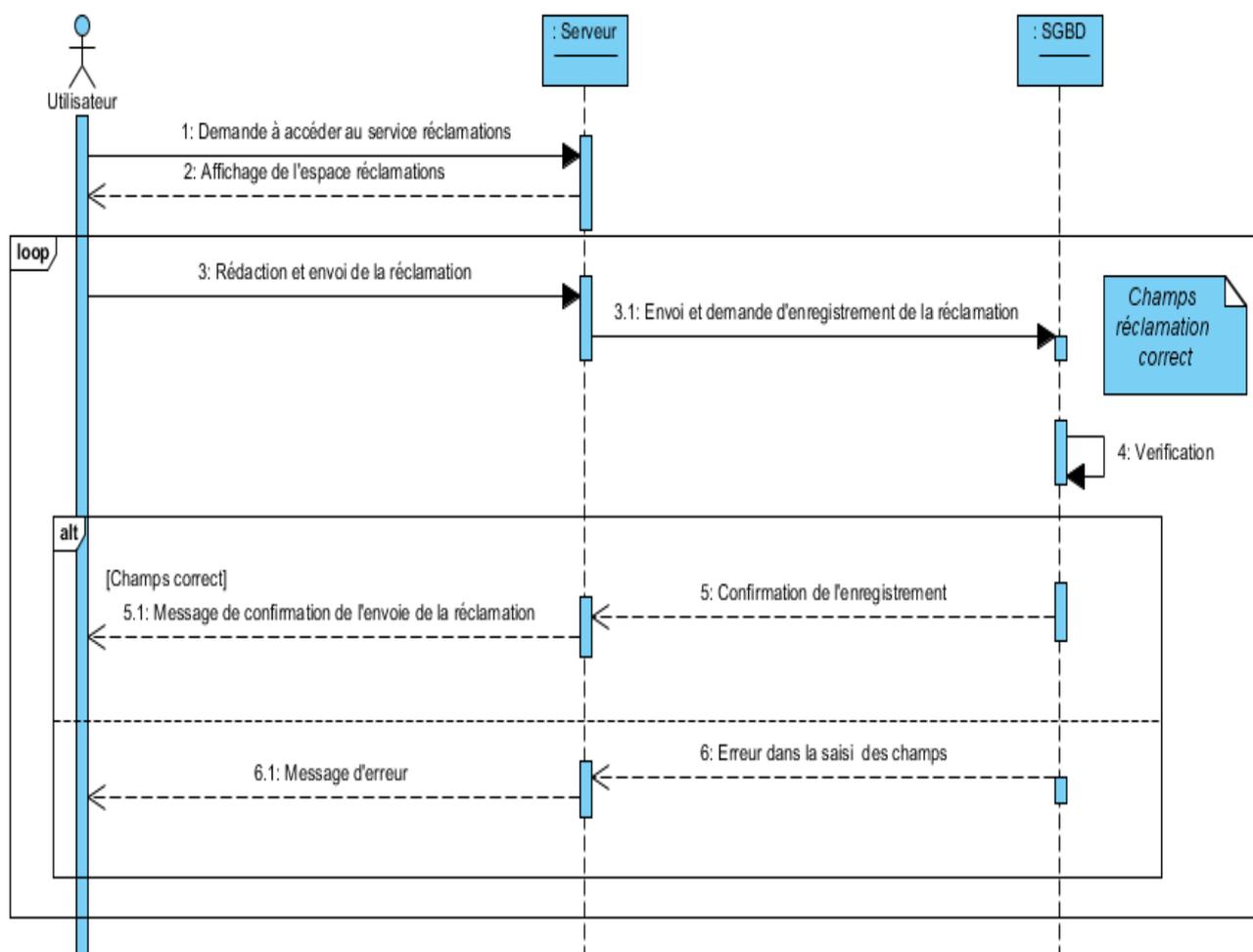


FIGURE 3.16 – Diagramme de séquence 'Envoyer une Réclamation' .

3.10.3.3 Le diagramme de classe du service Scolarité

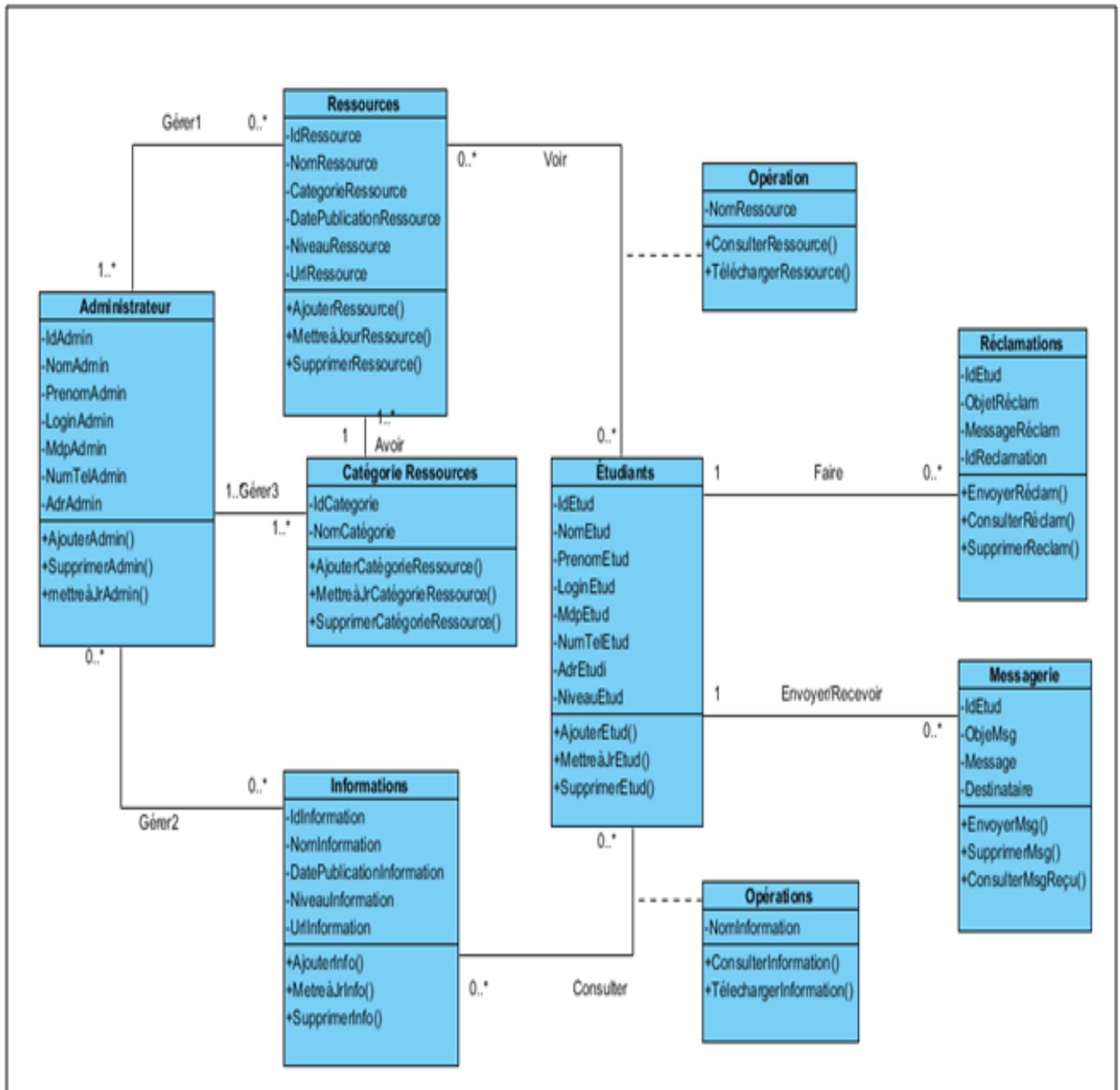


FIGURE 3.17 – Diagramme de classe du service scolarité.

↳ *Table de données du diagramme de classe du service scolarité*

3.10.4 Analyse et conception du service " Bibliothèque "

3.10.4.1 Les cas d'utilisation du service scolarité

Nous illustrons dans la figure ci-dessous , le diagramme de cas d'utilisation générale du service bibliothèque.

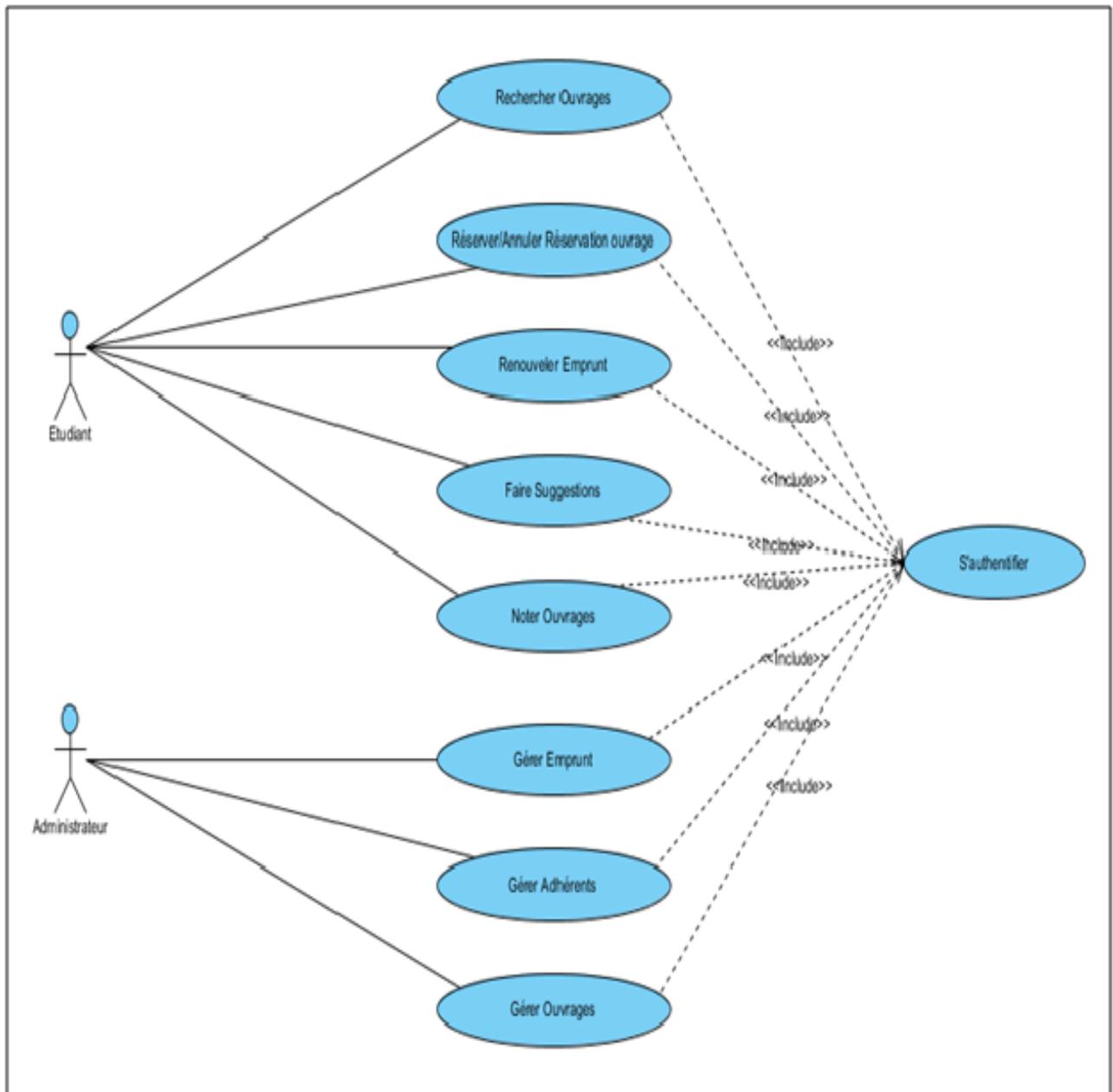


FIGURE 3.18 – Diagramme de cas d'utilisation générale du service bibliothèque.

1. Identification des acteurs

Dans le cas de notre projet on a deux acteurs l'administrateur et l'adhérent

- **l'administrateur** : La bibliothèque peut disposer d'un ou de plusieurs administrateurs. Les administrateurs de la bibliothèque vont gérer ce service à travers une application java conçue pour cela .

- **L'adhérent** : une fois l'étudiant authentifié ,il ne pourra accéder aux services de la bibliothèque que s'il est adhérent c'est-à-dire que l'administrateur vérifiera d'abord si son compte a été confirmé dans la base de données de la bibliothèque si c'est pas le cas , il va l'ajouter.

Afin de mieux éclaircir les diagrammes de cas d'utilisation nous allons faire une description textuelle des cas d'utilisation.

2. Analyse des cas d'utilisations pour l'administrateur

✓ *Cas d'utilisation 'Gérer adhérents'*

- *La Description*

✓ *Cas d'utilisation 'Gérer emprunt'*

- *La Description*

- Le diagramme de cas d'utilisation

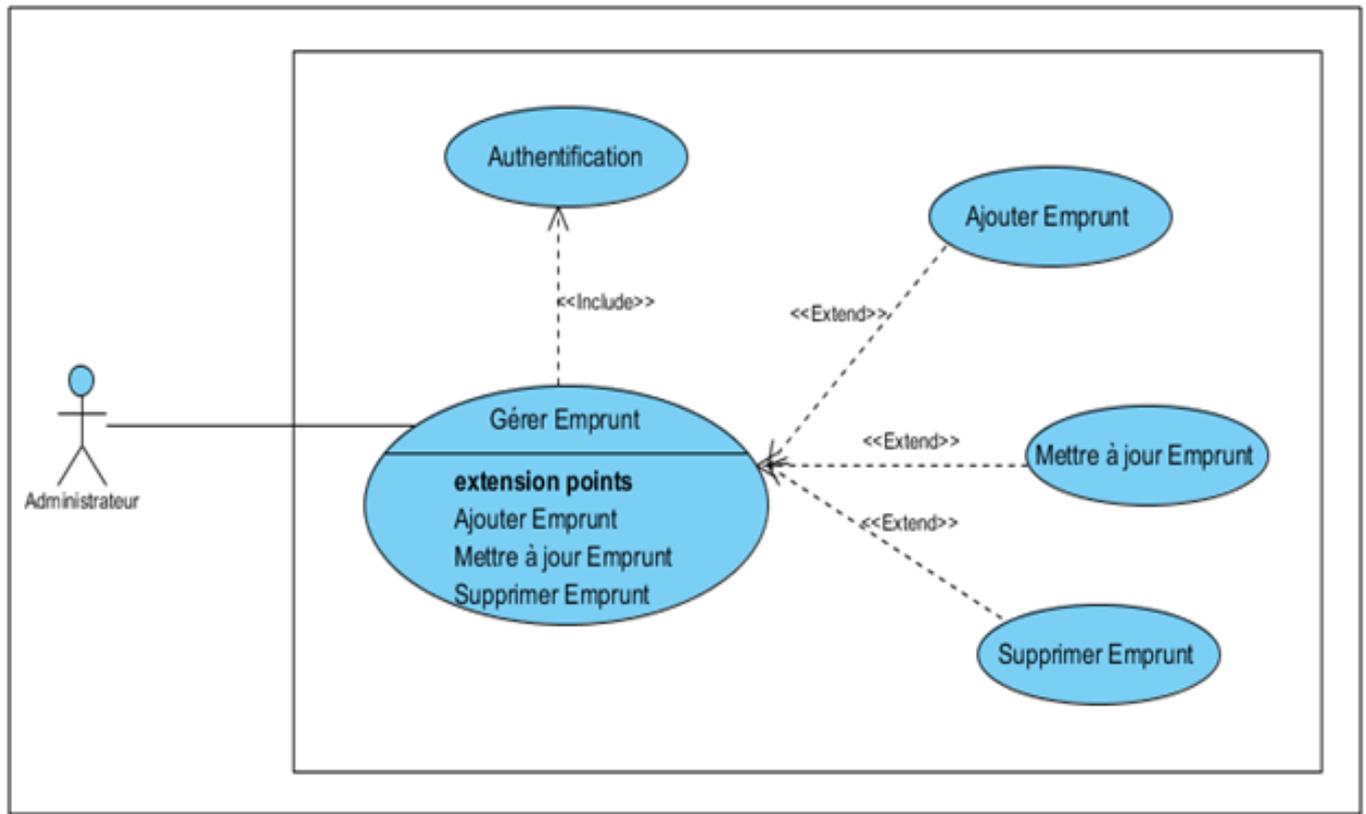


FIGURE 3.19 – Package du cas d'utilisation 'Gérer emprunt'.

✓ Cas d'utilisation 'Gérer Ouvrages '

- La Description

- Le diagramme de cas d'utilisation

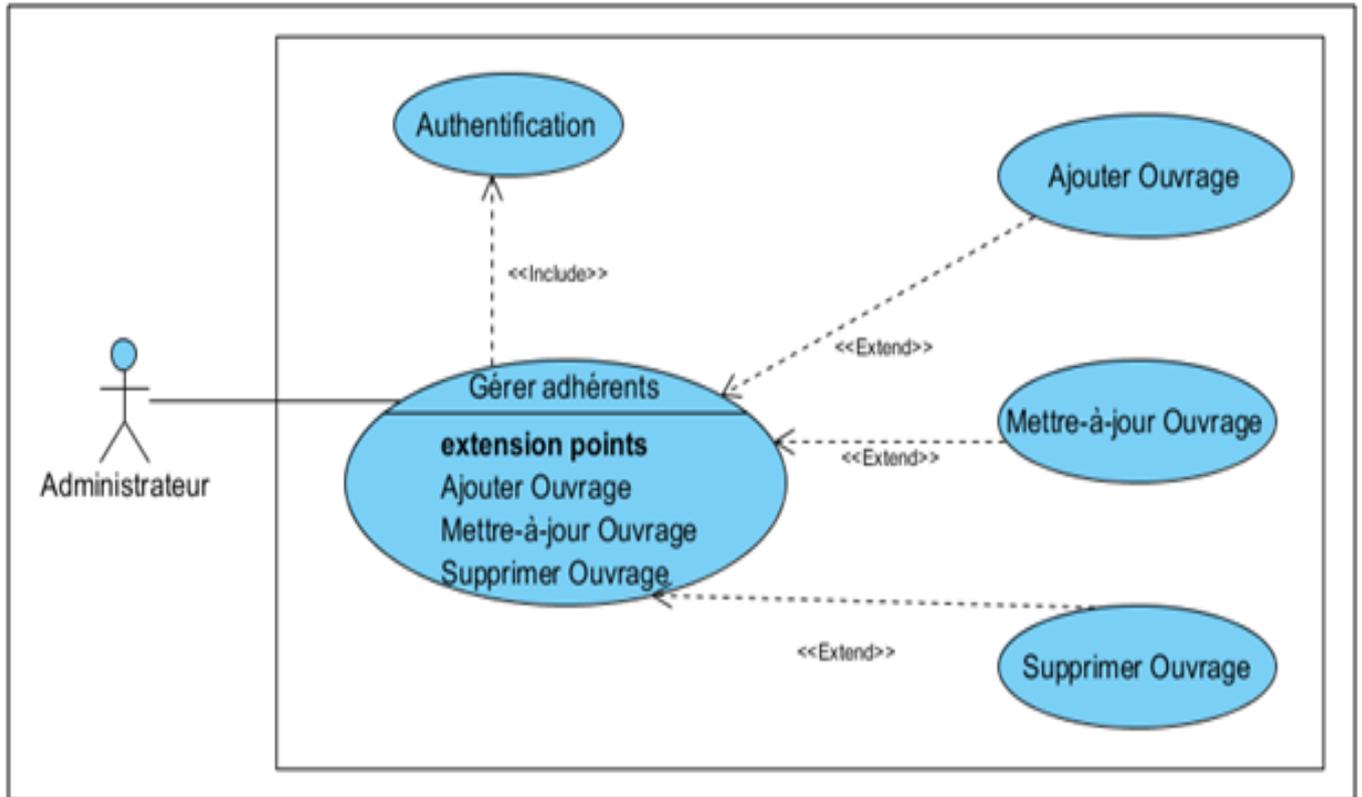


FIGURE 3.20 – Package du cas d'utilisation 'Gérer Ouvrages'.

3. Analyse des cas d'utilisation pour l'adhérent

✓ cas d'utilisation ' Réserver/Annuler Réserve d'un ouvrage'

- La Description

✓ *Cas d'utilisation 'Rechercher un ouvrage'*

- *La Description*

- Le diagramme de cas d'utilisation

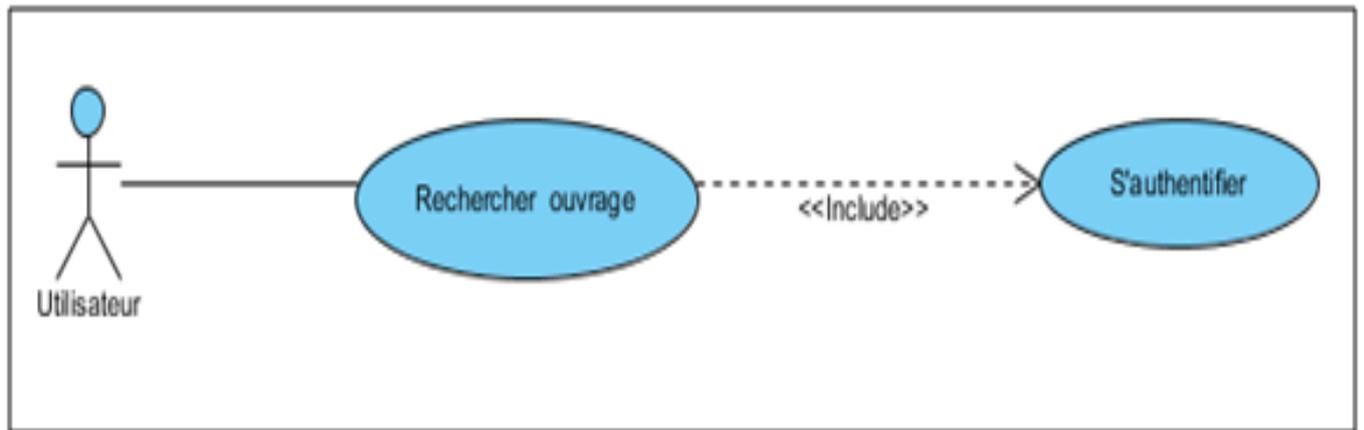


FIGURE 3.21 – Diagramme de cas d'utilisation 'Rechercher un ouvrage' .

✓ Cas d'utilisation 'Renouveler un Emprunt'

- La Description

- Le diagramme de cas d'utilisation

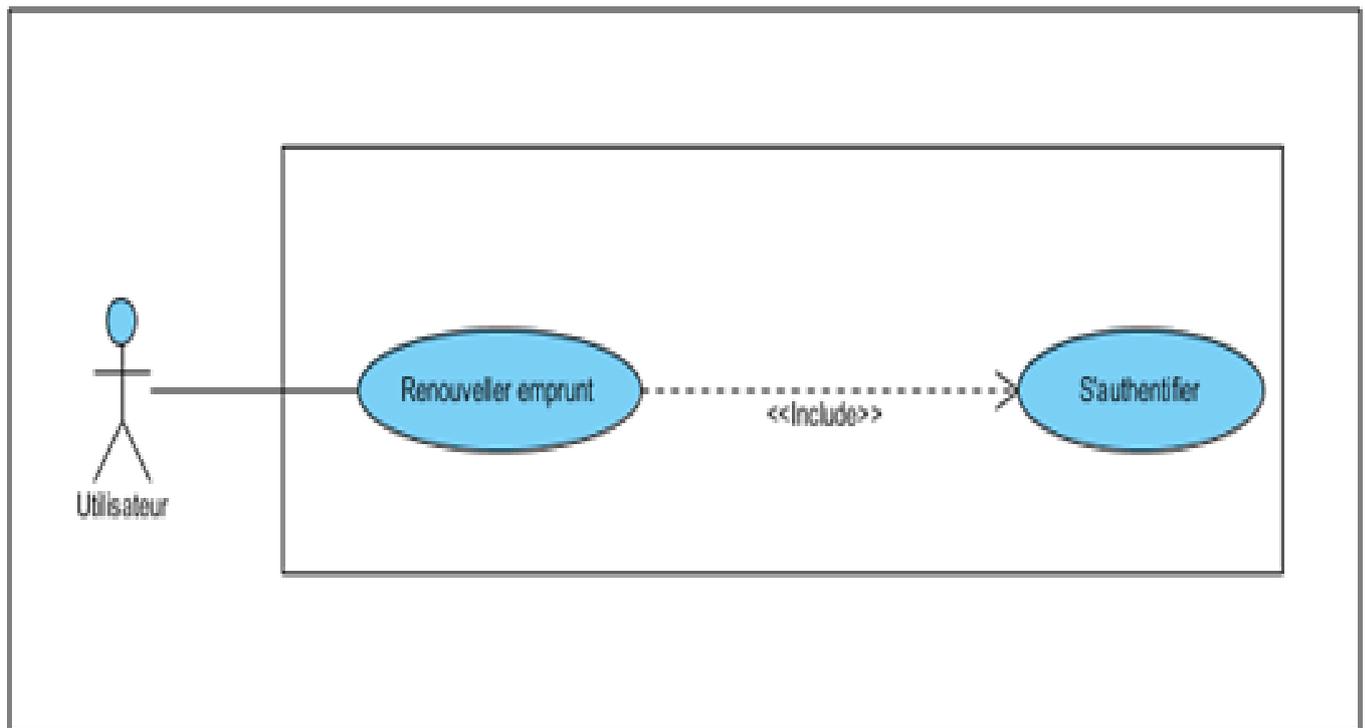


FIGURE 3.22 – Diagramme de cas d'utilisation 'Renouveler un Emprunt'.

✓ Cas d'utilisation 'Voir les Alertes'

- La Description

✓ *Cas d'utilisation 'Noter un Ouvrage '*

- *La Description*

- Le diagramme de cas d'utilisation

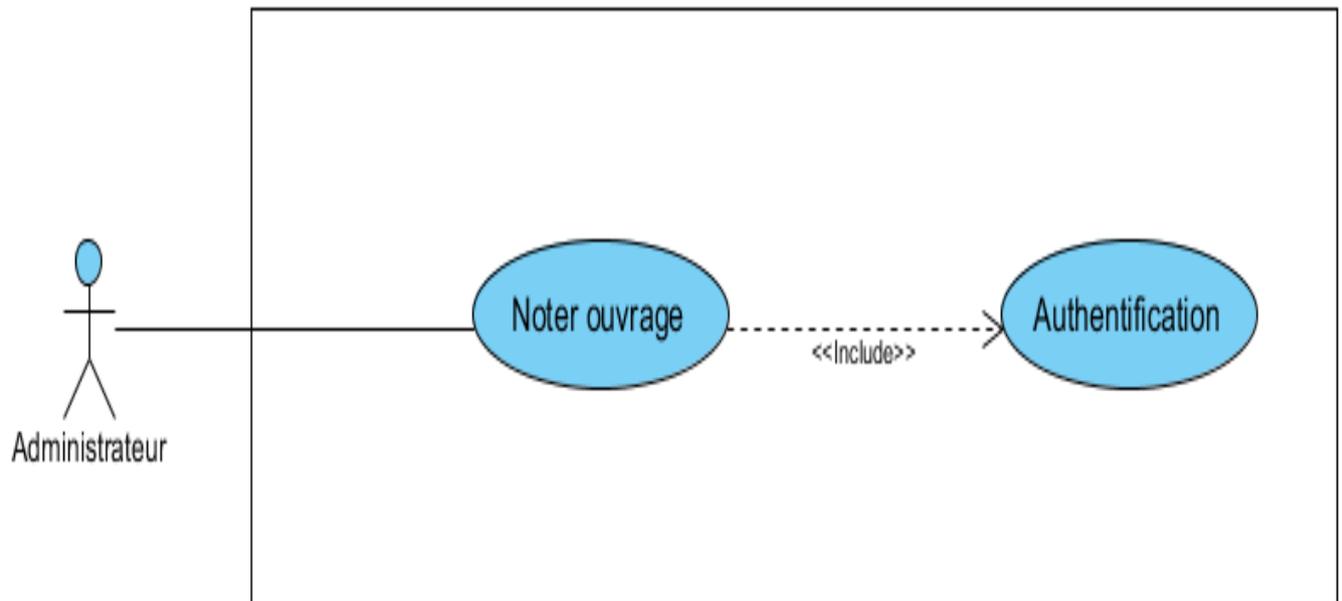


FIGURE 3.23 – Diagramme de cas d'utilisation 'Noter un Ouvrage'.

3.10.4.2 Les diagrammes de séquence du service bibliothèque

Nous allons détailler quelques exemples de diagrammes de séquence des cas d'utilisations que nous avons déjà cité au-dessus.

☞ *Diagramme de séquence du cas d'utilisation 'Réservation d'un ouvrage'*

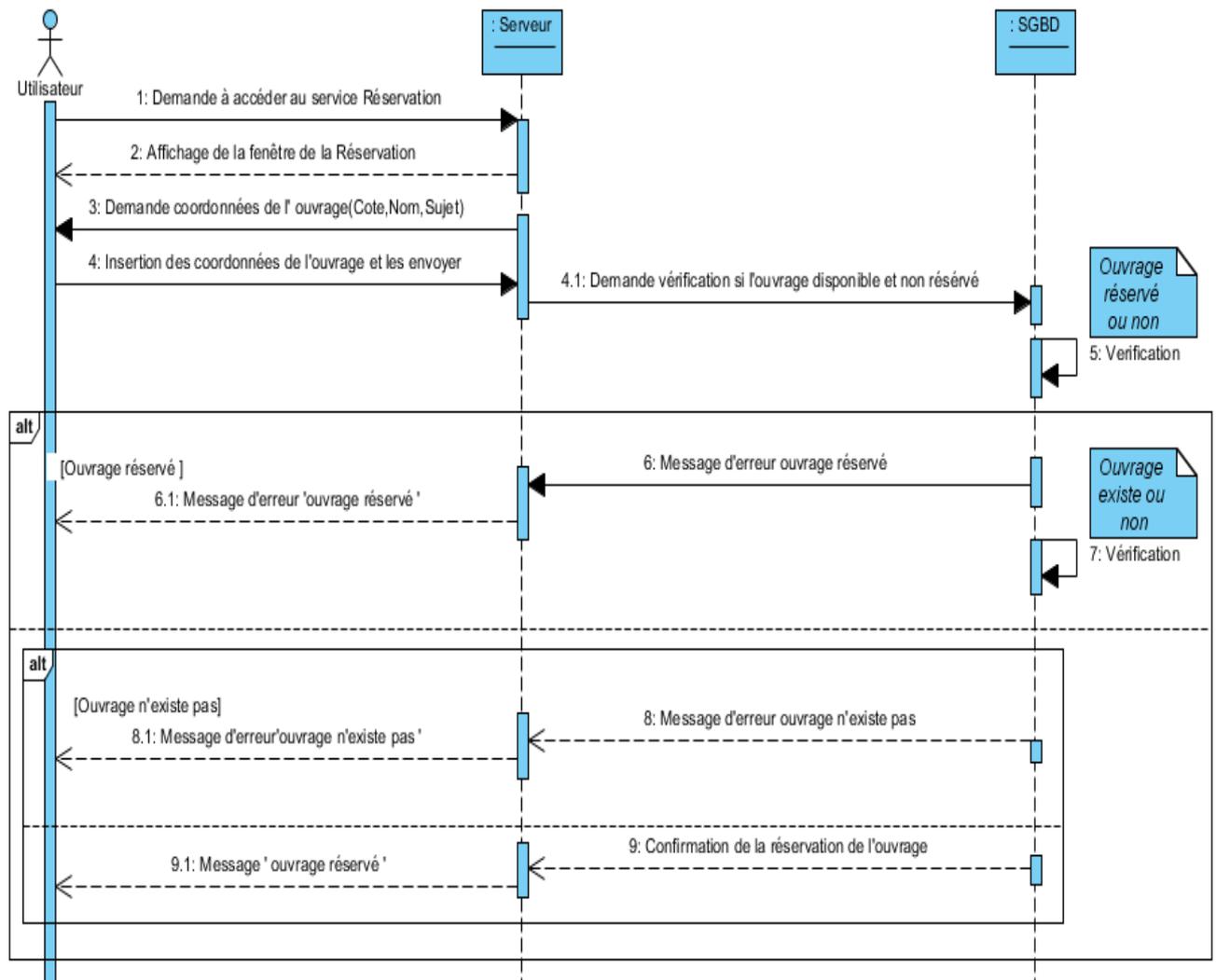


FIGURE 3.24 – Diagramme de séquence 'Réservation d'un ouvrage '.

↳ Diagramme de séquence du cas d'utilisation 'Renouvellement d'un emprunt'

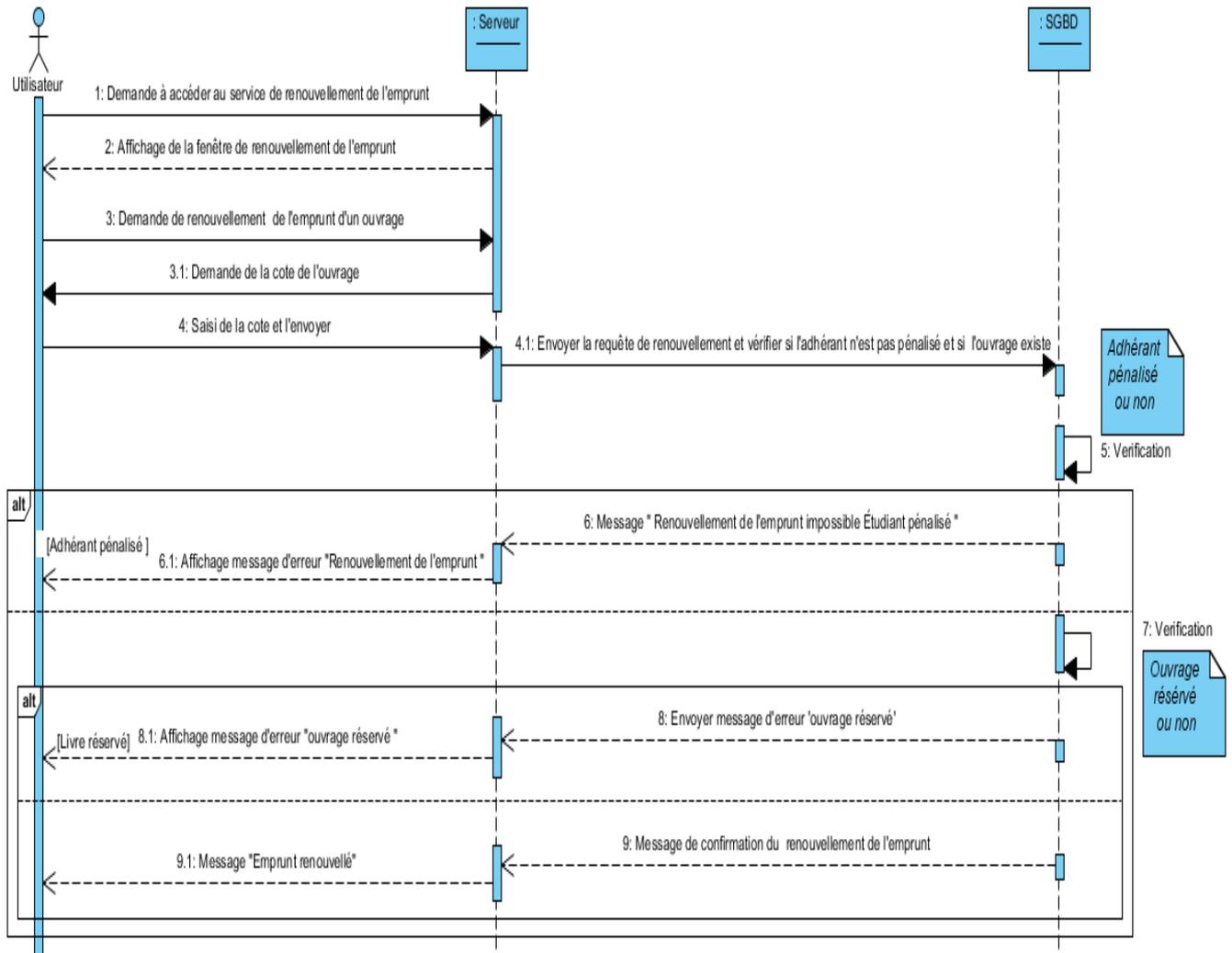


FIGURE 3.25 – Diagramme de séquence 'Renouvellement d'un emprunt'.

➤ Diagramme de séquence du cas d'utilisation 'Suggestion d'ouvrages'

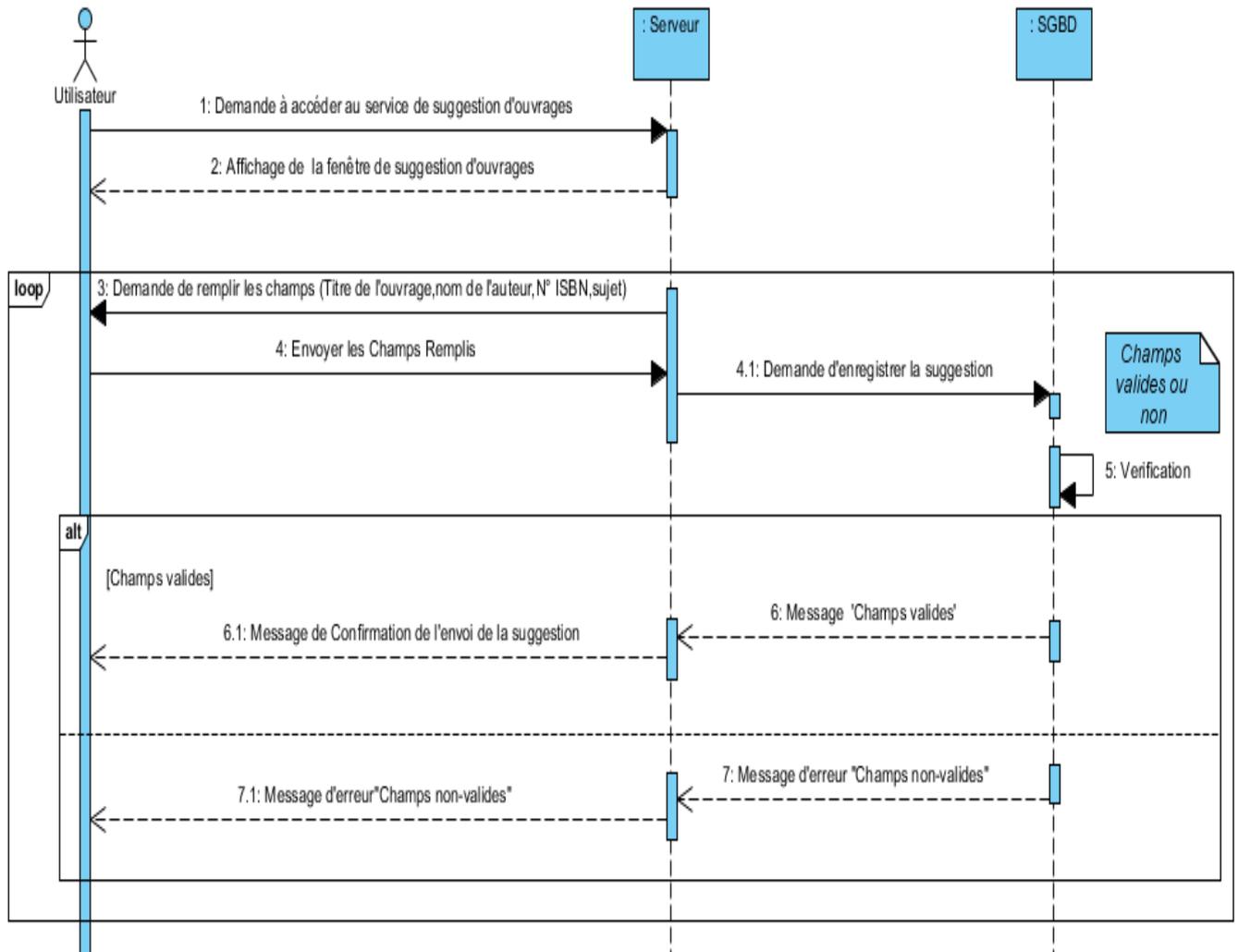


FIGURE 3.26 – Diagramme de séquence 'Suggestion d'ouvrages'.

↳ Diagramme de séquence du cas d'utilisation 'Noter un ouvrage '

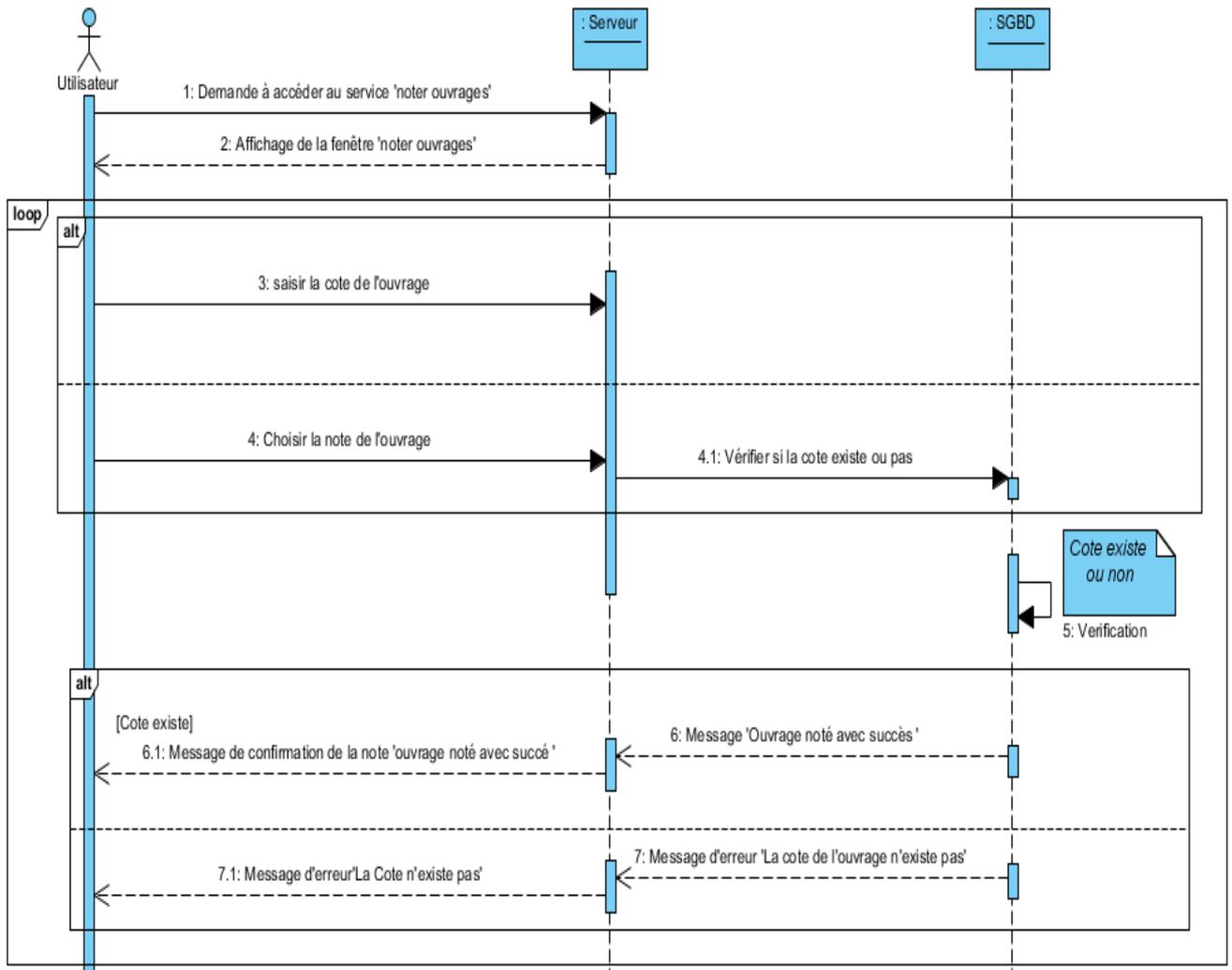


FIGURE 3.27 – Diagramme de séquence 'Noter un ouvrage'.

↳ Diagramme de séquence du package des cas d'utilisation 'Gérer les Adhérents'

- Diagramme de séquence 'Ajouter Adhèrent'

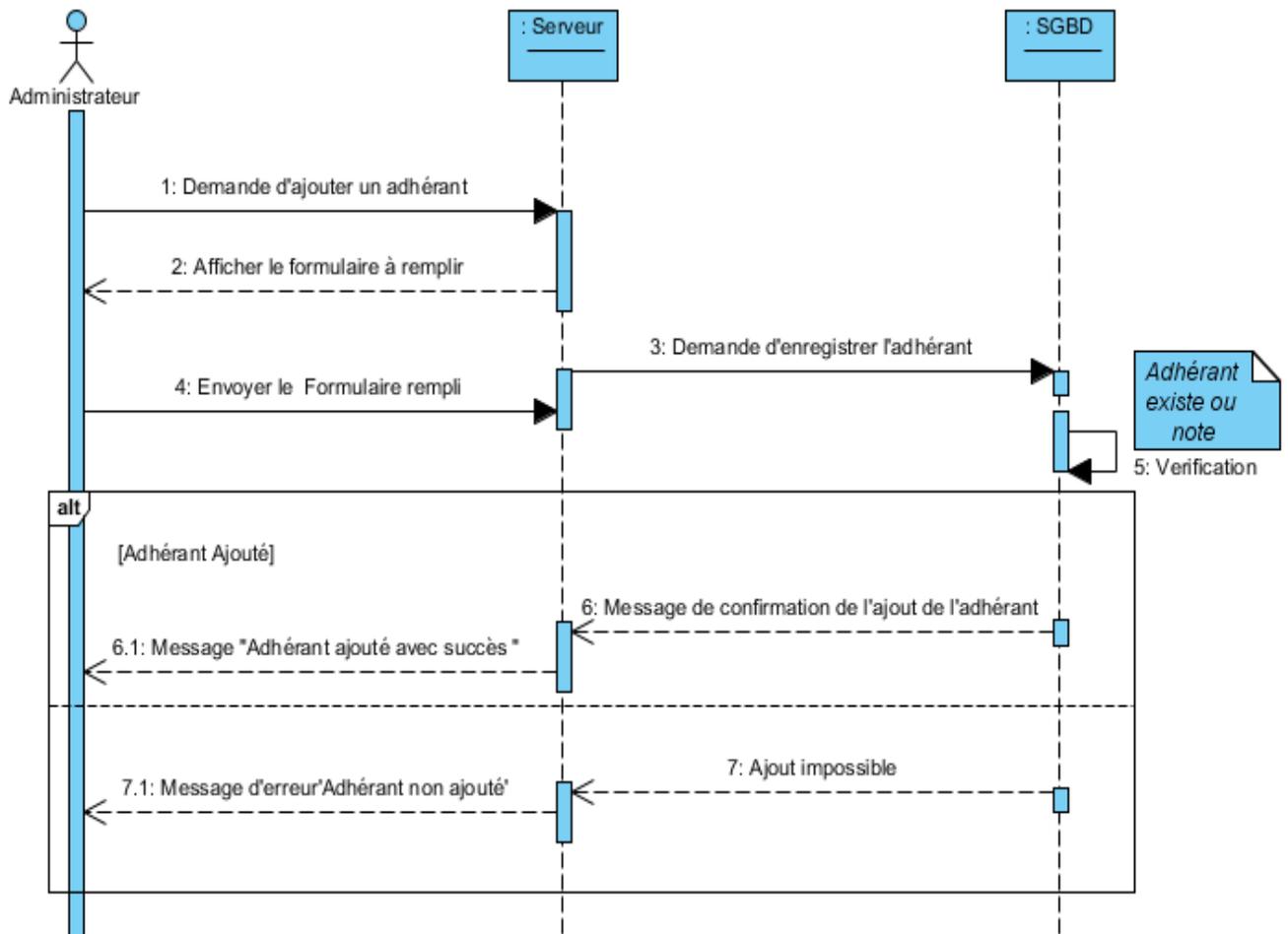


FIGURE 3.28 – Diagramme de séquence 'Ajouter Adhèrent'.

- Diagramme de séquence 'Mettre à jour Adhèrant'

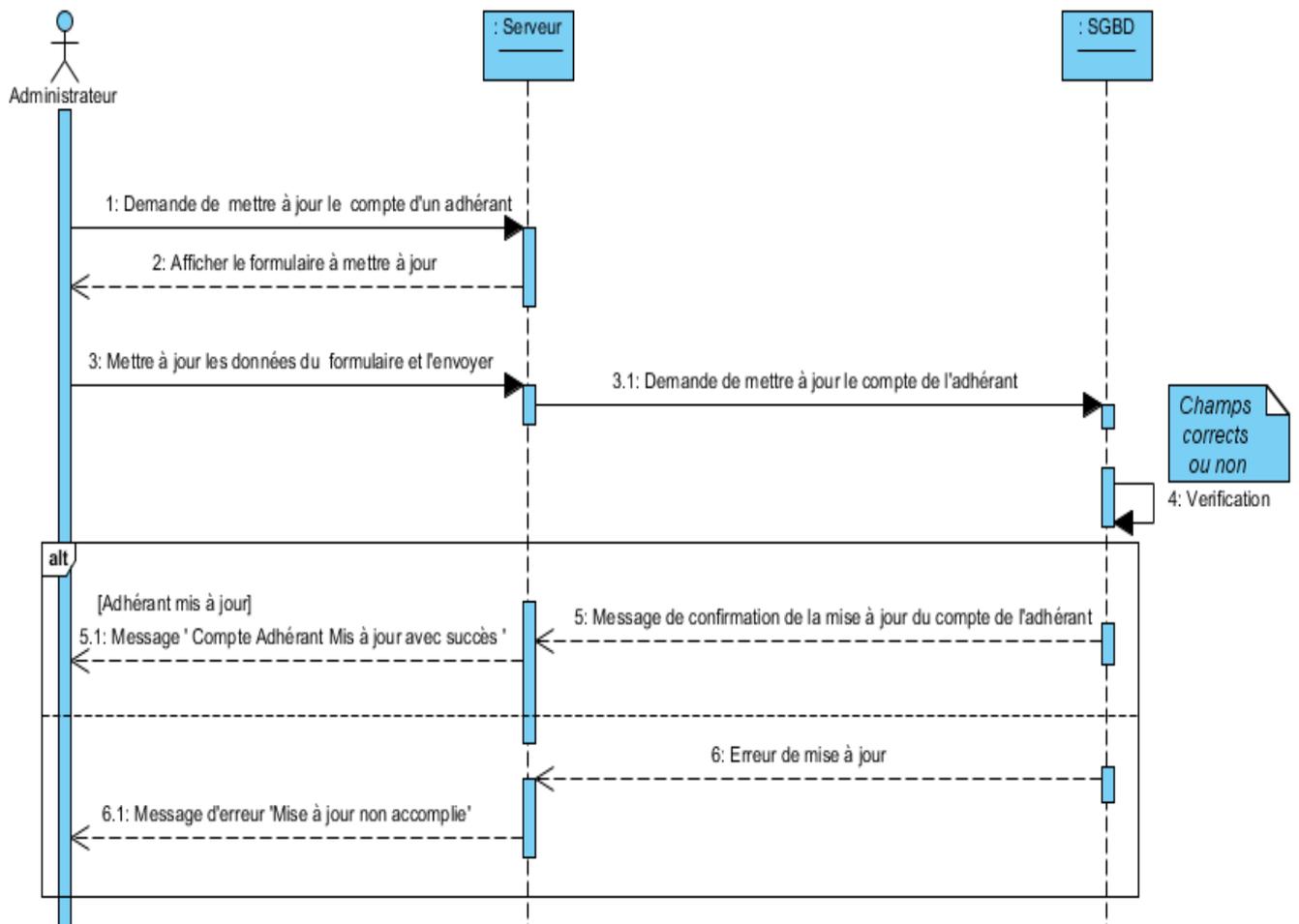


FIGURE 3.29 – Diagramme de séquence 'Mettre à jour Adhèrant'.

- Diagramme de séquence ' Supprimer Adhèrent '

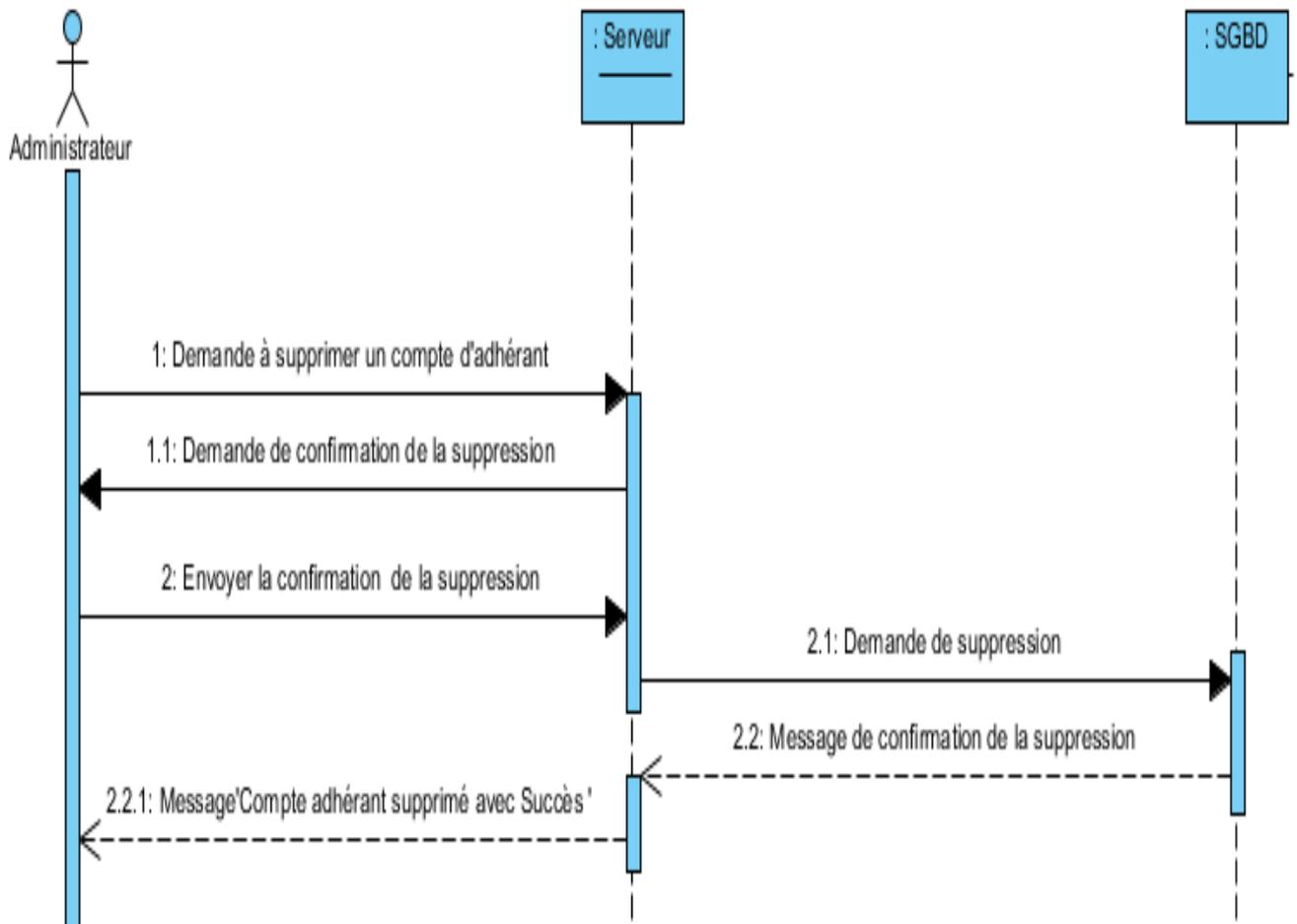


FIGURE 3.30 – Diagramme de séquence ' Supprimer Adhèrent '.

3.10.4.3 Le diagramme de classe du service Bibliothèque

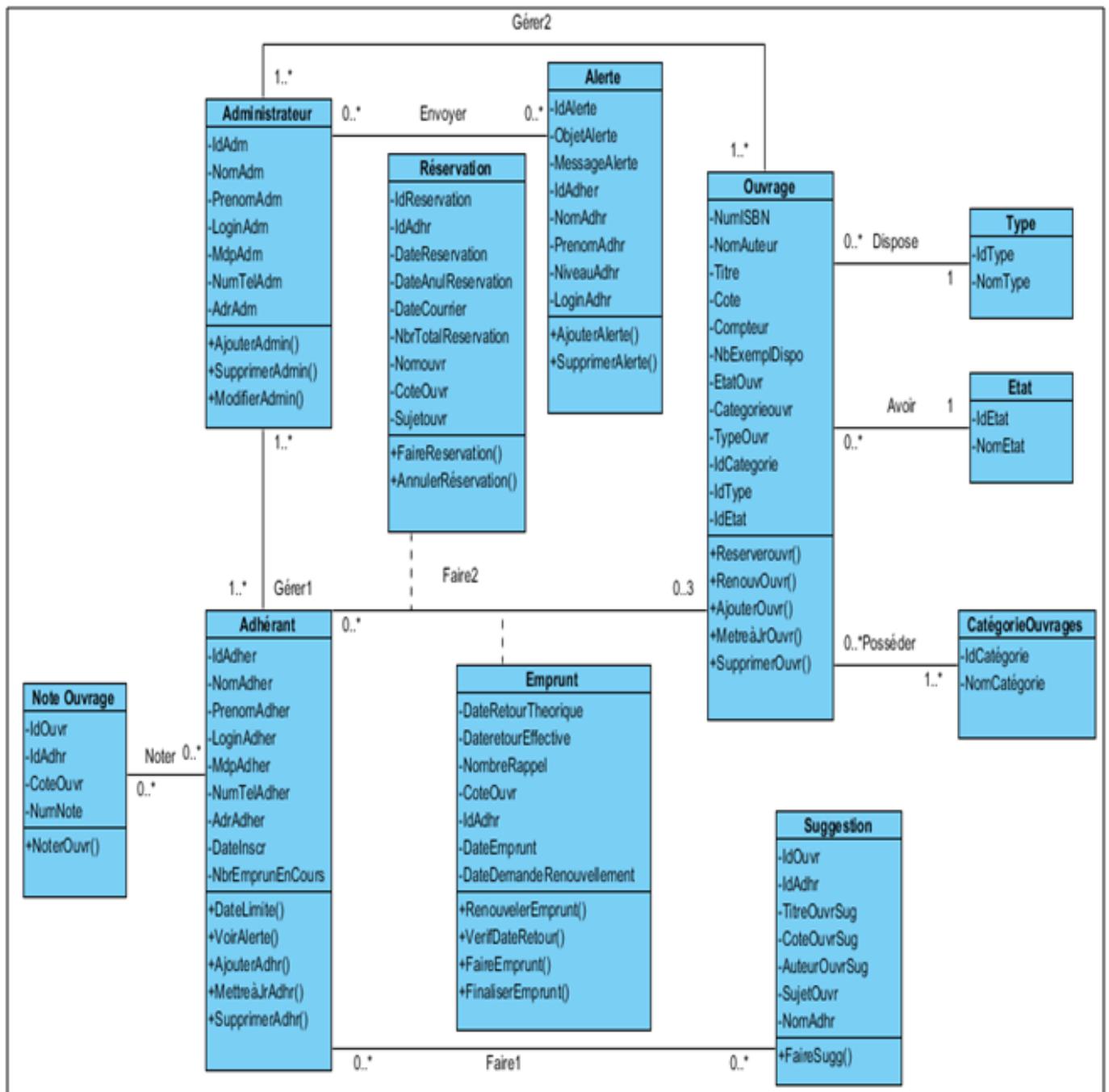


FIGURE 3.31 – Diagramme de classe du service bibliothèque.

✎ *Table de données du diagramme de classe du service Bibliothèque*

3.11 Conclusion

A l'issus de cette étape, nous avons pu exprimer les objectifs attendus du futur système à concevoir, ainsi que les différents diagrammes UML afin de bien entamer l'implémentation et la réalisation de l'application mobile SSB-UB.

Implémentation

4.1 Introduction

A ce stade du processus, les cas d'utilisation sont terminés, le problème a été analysé en profondeur ; nous avons défini une conception que nous espérons la mieux appropriée. Nous pouvons alors entreprendre la dernière activité du Processus Unifié composée de deux parties (implémentation et test), avec comme objectif d'aboutir à un produit final, exploitable par les utilisateurs.

Dans ce chapitre nous allons présenter l'environnement logiciel, les technologies et les langages de programmation que nous avons utilisés ainsi que l'architecture matérielle mise en place du système de l'application mobile, suivie de quelques interfaces de l'implémentation de la base de données, ensuite illustrer l'arborescence ainsi que quelques exemples d'interfaces graphiques de l'application mobile ainsi que celles de la partie administration, à savoir une interface de l'application Java pour la partie administration du service bibliothèque, une interface du site web de l'application pour la partie administration du service scolarité, et nous terminerons le chapitre par la phase de test.

4.2 Environnement de développement de l'application

4.2.1 Environnement logiciel

- NetBeans 7.0.1

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000. En plus de Java, NetBeans permet également de supporter

différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, éditeur graphique d'interfaces et de pages Web ...).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Mac OS X etc ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit (JDK) est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)) et s'enrichit à l'aide de plugins.

4.2.2 Outils Utilisés

- IDE Eclipse

Eclipse IDE est un environnement de développement intégré libre, extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

- Le Plugin ADT (Android Développement Tools) [1]

Android Development tools est un plugin pour l'IDE Eclipse, il est conçu pour donner un environnement puissant et intégré afin de développer des applications Android. ADT étend les capacités d'Eclipse afin de permettre de configurer rapidement de nouveaux projets Android, créer des interfaces utilisateurs d'une application et même exporter le fichier '.APK' qui permet de distribuer l'application.

- Software Development Kit (SDK) [21]

C'est un kit de développement basé sur le langage Java. Il s'agit des outils que Google a fournis pour interagir avec Android pour la réalisation des applications de type défini (Par exemple iOS, Android...). Le SDK propose, en plus, un émulateur Android. Ce dernier 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. A chaque version d'Android est associée

une version de l'émulateur, permettant au développeur de tester et voir exactement à quoi ressemblera son application sur un matériel réel.

- Java DataBase Connectivity (JDBC) [8]

JDBC est l'acronyme de Java DataBase Connectivity et désigne une API Java (ensemble de classes et d'interfaces défini par SUN et les acteurs du domaine des Bases de Données) permettant d'accéder aux bases de données à l'aide du langage Java via des requêtes SQL. Cette API permet d'atteindre de manière quasi-transparente des bases : My Sql, Sybase, Oracle, Informix, ... avec le même programme Java JDBC.

Cette API permet de faire tout types de requêtes tel que :

- La sélection de données dans des tables ;
- La création de tables et l'insertion d'éléments dans les tables ;
- La gestion des transactions etc.

Dans un système Client/Serveur, l'accès aux bases de données avec JDBC peut s'effectuer selon une architecture à deux couches ou bien une architecture à trois couches. La figure suivante -figure 4.1 - spécifie l'architecture générale d'un pilote JDBC :

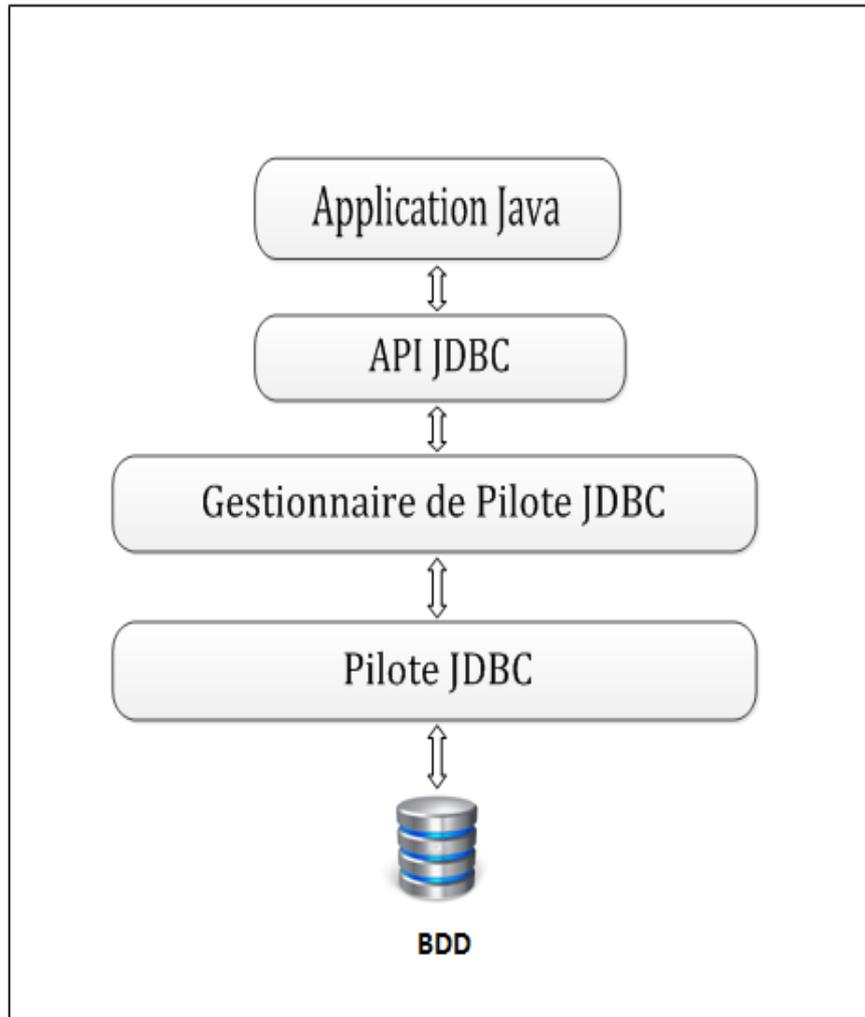


FIGURE 4.1 – Architecture générale d'un pilote JDBC .

- Tomcat [5]

Apache Tomcat est un conteneur web libre (et pas un serveur à part entière) permettant d'exécuter des servlets¹ et des pages serveur Java (JSP). Il est développé sous licence open-source par la fondation Apache. Il peut être utilisé ou couplé avec un serveur Web (dont Apache), et porté sur n'importe quel système sur lequel une machine virtuelle Java est installée.

1. Une servlet est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également l'être au format XML ou tout autre format destiné aux navigateurs web. Les servlets utilisent l'API Java Servlet (package javax.servlet).

- **MySQL** [42]

MySQL est un système de gestion de base de données relationnel, un langage de requêtes vers les bases de données exploitant le modèle relationnel et utilise le langage SQL comme langage de requête.

SQL est un langage de manipulation de bases de données mis au point dans la années 70 par IBM, il permet d'effectuer trois types de manipulations :

1. La manipulation des tables : Création, suppression, modification de la structure.
2. Les manipulations des données de la base : Sélection, modification, suppression d'enregistrement.
3. La gestion des droits d'accès aux tables : Contrôle des données, droit d'accès, validation des modifications.

- **JSON** [10]

JSON (JavaScript Object Notation) est un format de données textuelles, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter de l'information structurée. Créé par Douglas Crockford, il est décrit par la RFC 4627 de l'IETF .

4.2.3 Les langages de programmation

- **Java** [7]

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).

Enfin, nous rappelons que le Java, étant un langage de programmation orienté objet utilisable sur divers systèmes d'exploitation, est un langage assez robuste, portable et à hautes performances.

- XML [25]

Extensible Markup Language est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode structurées en champs arborescents.

Ce langage a été standardisé par le W3C² en février 1998 et est maintenant très populaire. Il est similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet.

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.

- JavaScript [9]

JavaScript est un langage de script orienté objet principalement utilisé dans les pages HTML. A l'opposé des langages serveurs (qui s'exécutent sur le site), JavaScript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au dessus d'un élément, du redimensionnement de la page etc). La version standardisée de JavaScript est l'ECMAScript.

- PHP [18]

HyperText Preprocessor, plus connu sous son sigle PHP est un langage de programmation interprété libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes.

4.3 Architecture matérielle de l'application SSB-UB

L'application mobile SSB-UB est une application qui se connecte à un serveur de bases de données afin de récupérer et d'envoyer les données en utilisant internet et cela se

2. W3C ou World Wide Web Consortium est une organisation non lucrative permettant de définir des standards pour les technologies liées aux web.

fait via les requêtes HTTP (GET/POST) qu'on trouve dans les scripts PHP , en voila un schéma qui illustre l'architecture de ce systeme .

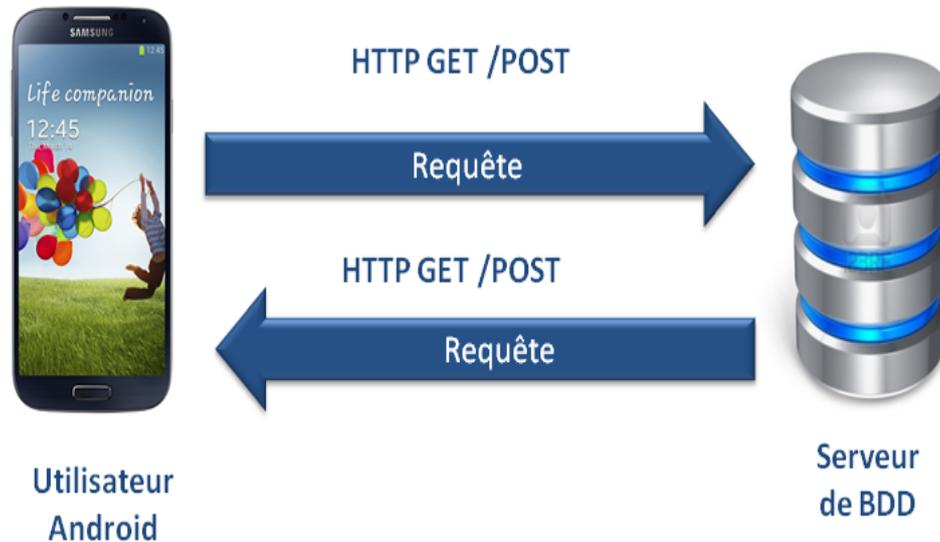


FIGURE 4.2 – Architecture materielle du systeme.

4.4 Implémentation de la Base de Données

Pour implémenter notre base des données, nous avons utilisé l'environnement de création de base de données PHPMysqlAdmin et le système de gestion de base de données MySQL. la table ci-dessous présente le schéma de la base de données générale :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
adherent		2	InnoDB	utf8_general_ci	32,0 Kio	-
administrateur		1	InnoDB	utf8_general_ci	32,0 Kio	-
alerte		4	InnoDB	utf8_general_ci	16,0 Kio	-
categorieouvrage		4	InnoDB	utf8_general_ci	32,0 Kio	-
catégorieressource		4	InnoDB	utf8_general_ci	32,0 Kio	-
etudiant		2	InnoDB	utf8_general_ci	32,0 Kio	-
information		0	InnoDB	latin1_swedish_ci	16,0 Kio	-
noteouvrage		4	InnoDB	utf8_general_ci	16,0 Kio	-
ouvrage		48	InnoDB	utf8_general_ci	48,0 Kio	-
ouvrageemprunt		0	InnoDB	utf8_general_ci	32,0 Kio	-
reclamations		0	InnoDB	utf8_general_ci	32,0 Kio	-
reservation		2	InnoDB	utf8_general_ci	16,0 Kio	-
ressource		48	InnoDB	utf8_general_ci	96,0 Kio	-
suggestionouvrage		1	InnoDB	utf8_general_ci	16,0 Kio	-
typeouvrage		0	InnoDB	latin1_swedish_ci	16,0 Kio	-
15 table(s)	Somme	120	InnoDB	latin1_swedish_ci	464,0 Kio	0 o

FIGURE 4.3 – Schéma de la base de données général.

- La table suivante illustre la base de donnée de l’administrateur :

←T→	id_administrateur	nomadministrateur	prenomadministrateur	mailadministrateur	identifiantadministrateur	motdepasseadministrateur
 	1	Sadelli	salim	sadellisalim@gmail.com	admin	admin
 	2	Bechar	Anissa	Bechar.anissa@gmail.com	admin	admin

FIGURE 4.4 – la base de donnée de l’administrateur.

– La table suivante illustre la base de donnée de l’adhérant :

←T→	id_etudiant	identifiantetudiant	motdepasseetudiant	nometudiant	prenometudiant	emailetudiant	telephonetudiant	adresetudiant	Niveau
 	3	sadellisalim	salimo	sadelli	salim	sadellisalim@gmail.com	0558564712	4 rue du Bejaia	M2
 	4	Becharmissa	Anissa	Bechar	Nissa	bechar.anissa@gmail.com	0558451208	05.rue maurice audin Bejaia	M2

FIGURE 4.5 – la base de donnée de l’adhérant.

4.5 Arborescence de l’application SSB-UB

La figure ci - dessous -figure 4.5 - illustre l’arboréscence de notre application mobile .

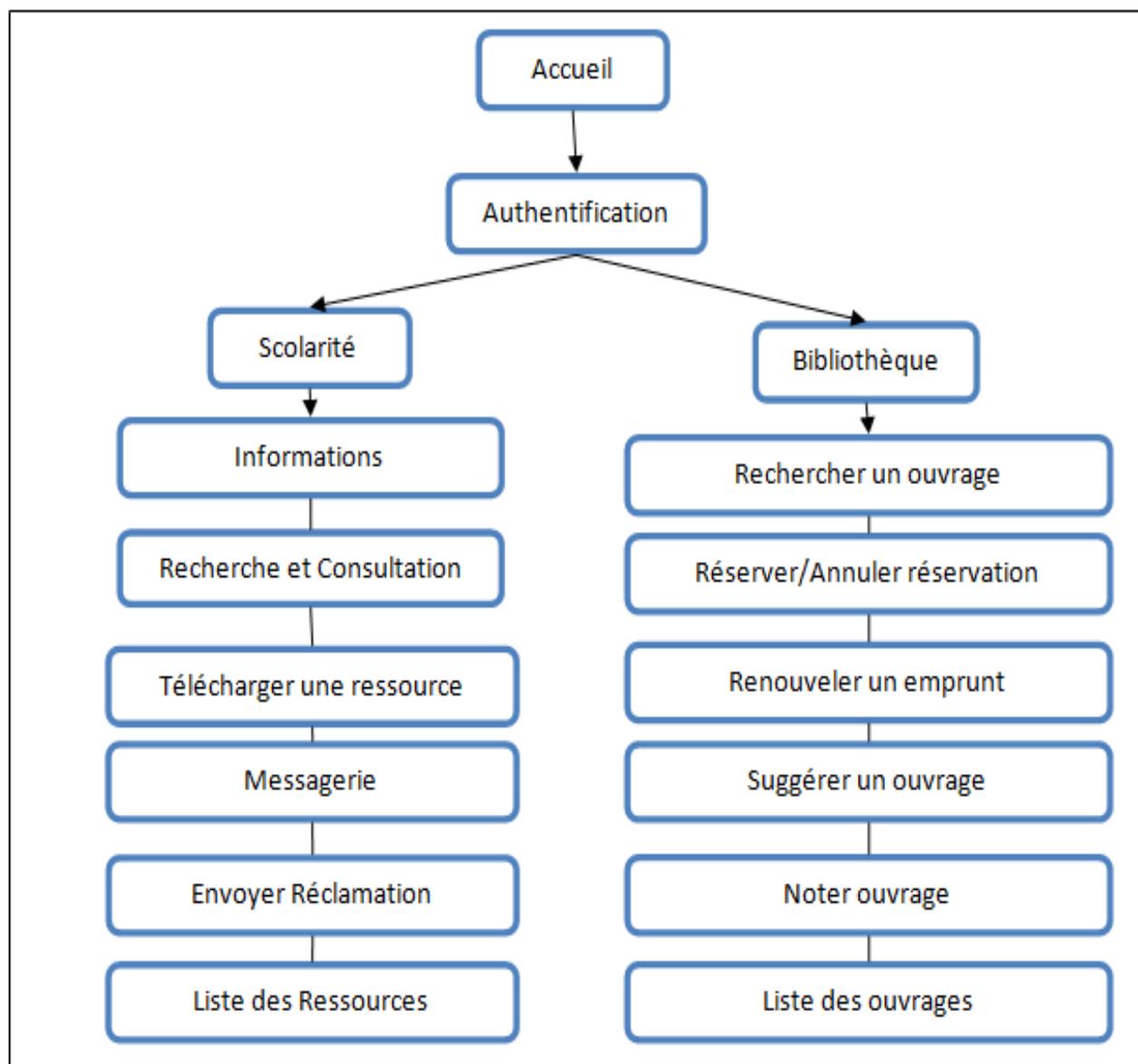


FIGURE 4.6 – Arborescence générale de l'applicationSSB-UB.

4.6 Interfaces graphiques de l'application SSB-UB

Dans ce qui suit, nous allons présenter quelques interfaces de l'application mobile SSB-UB

4.6.1 Interfaces utilisateur

✓ **Interface d'accueil**

Cette interface offre un aperçu sur la page d'accueil de l'application SSB-UB .



FIGURE 4.7 – Interface de l'accueil.

✓ **Interface de l'authentification**

Cette interface offre un aperçu sur la page d'accueil de l'application SSB-UB.

Dans cette interface , nous pouvons voir les deux champs 'username' et 'Password' que l'étudiant doit saisir puis il aura à cliquer sur le bouton 'Login' afin de s'authentifier.

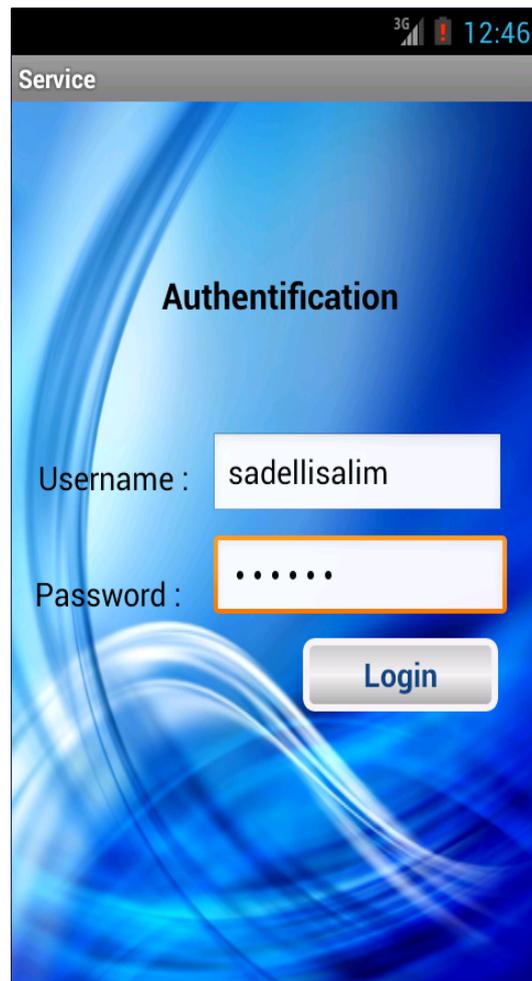


FIGURE 4.8 – Interface de l'authentification.

✓ Interface des services principaux

Cette interface offre un aperçu les deux services principaux de notre application mobile, Nous pouvons voir un bouton 'Scolarité' pour accéder au service scolarité et un autre bouton 'Bibliothèque' pour accéder au service bibliothèque.



FIGURE 4.9 – Interface des services principaux.

1. Interfaces des services de la scolarité

✓ Interface de la liste des services scolarité

Cette interface offre un aperçu de toute la liste des services de la scolarité de l'application SSB-UB. Dans cette interface, nous pouvons voir six boutons, chaque bouton nous permet d'accéder à un service en cliquant dessus.



FIGURE 4.10 – Interface de la liste des services de la scolarité.

✓ Interface de la recherche de Ressources

Cette interface offre un aperçu de l'interface de la recherche de ressources de l'application SSB-UB.

Dans cette interface, nous pouvons voir un champ vide ou nous devons saisir un mot clef sur la ressource à rechercher et cliquer sur l'icône rechercher et la réponse de la recherche va s'afficher dans le champ de la description.

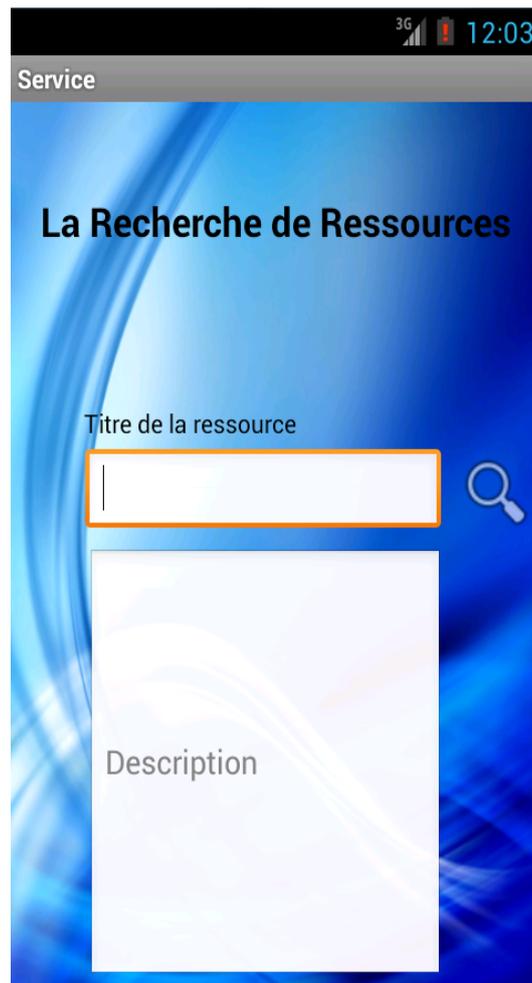


FIGURE 4.11 – Interface du services de recherche de ressources.

✓ Interface du téléchargement de Ressources

Cette interface offre un aperçu de l'interface du téléchargement de ressources de l'application SSB-UB.

Dans cette interface, nous trouverons un champs URL ou on doit soit saisir l'url de la ressource à télécharger (si on l'a déjà) ou bien choisir une ressource parmi la liste affichée et cliquer dessus , ainsi l'url va automatiquement s'afficher dans le champs de l'URL et puis cliquer sur le bouton 'télécharger' , ainsi la ressource va etre téléchargée.

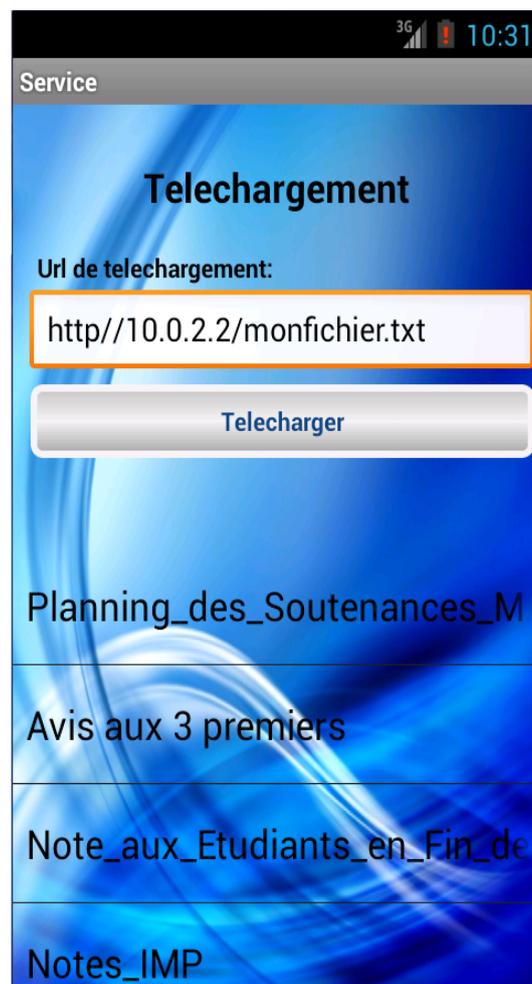
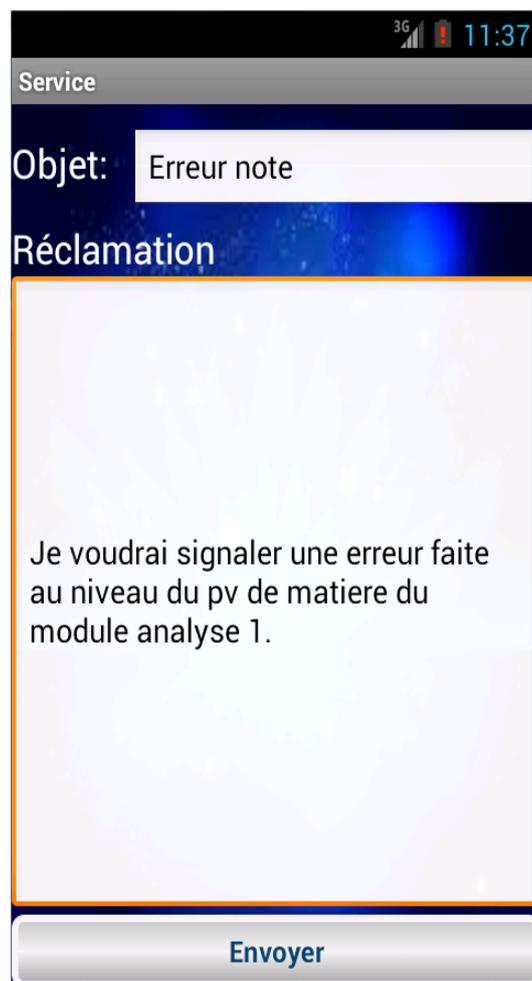


FIGURE 4.12 – Interface du service téléchargement de la ressource .

✓ Interface du service de réclamations

Cette interface offre un aperçu de l'interface du service d'information de l'application SSB-UB.

Dans cette interface, il y'a un champ objet ou nous devons saisir l'objet de la réclamation , et en dessous il y'a le champs 'Réclamation' ou nous aurons a saisir le texte de la réclamation , puis cliquer sur le bouton 'envoyer', ainsi la réclamation sera envoyé à l'administrateur.



The image shows a mobile application interface for submitting a complaint. At the top, there is a status bar with '3G' signal, a battery icon, and the time '11:37'. Below that is a grey header with the word 'Service'. The main content area has a dark blue background. It features a label 'Objet:' followed by a white text input field containing 'Erreur note'. Below this is another label 'Réclamation' followed by a large white text area with a thin orange border. The text in this area reads: 'Je voudrai signaler une erreur faite au niveau du pv de matiere du module analyse 1.'. At the bottom of the screen is a wide, rounded button with a gradient and the text 'Envoyer'.

FIGURE 4.13 – Interface du service réclamation .

2. Interfaces de la bibliothèque

✓ Interface de la liste des services de la bibliothèque

Cette interface offre un aperçu de toute la liste des services de la bibliothèque de l'application SSB-UB.

Dans cette interface nous pouvons voir six boutons dont chaque bouton reprise un service du service scolarité.

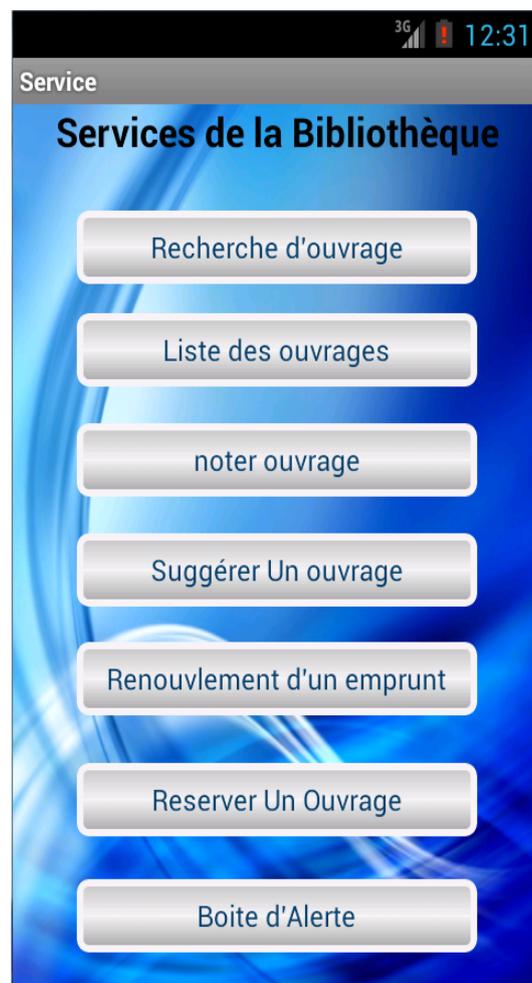


FIGURE 4.14 – Interface liste des services de la bibliothèque .

✓ **Interface de la réservation d'ouvrages**

Cette interface offre un aperçu de l'interface de réservation d'ouvrages de l'application SSB-UB.

Dans cette interface nous remarquons trois champs : cote de l'ouvrage , Nom de l'ouvrage et sujet de l'ouvrage ; l'étudiant devra saisir les champs et cliquer sur 'reserver l'ouvrage'.

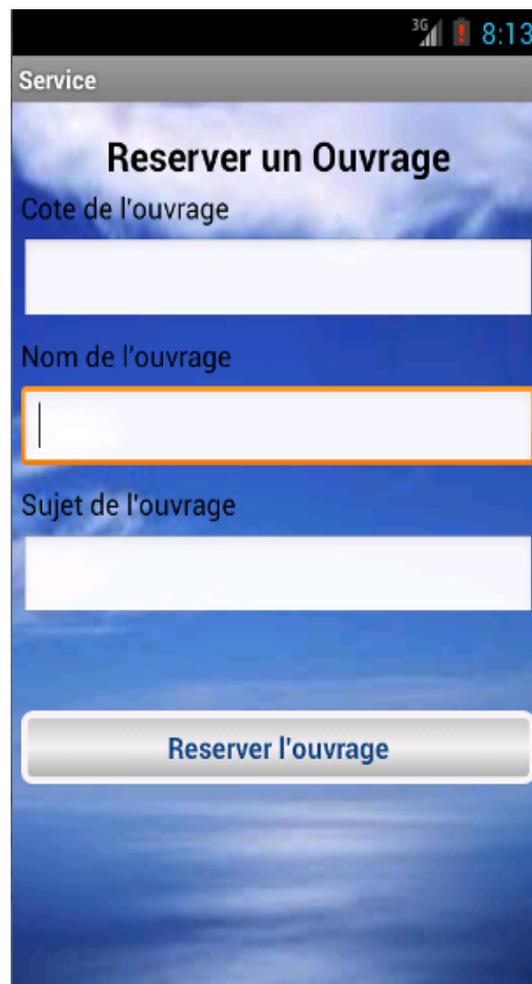


FIGURE 4.15 – Interface de réservation d'ouvrages.

✓ Interface du renouvellement d'emprunt

Cette interface offre un aperçu de l'interface de renouvellement d'emprunt de l'application SSB-UB.

Dans cette interface, nous remarquons le champs cote , que l'étudiant devra saisir et puis cliquer sur 'renouveler emprunt'; puis la réponse sera affichée dans le champs de description.

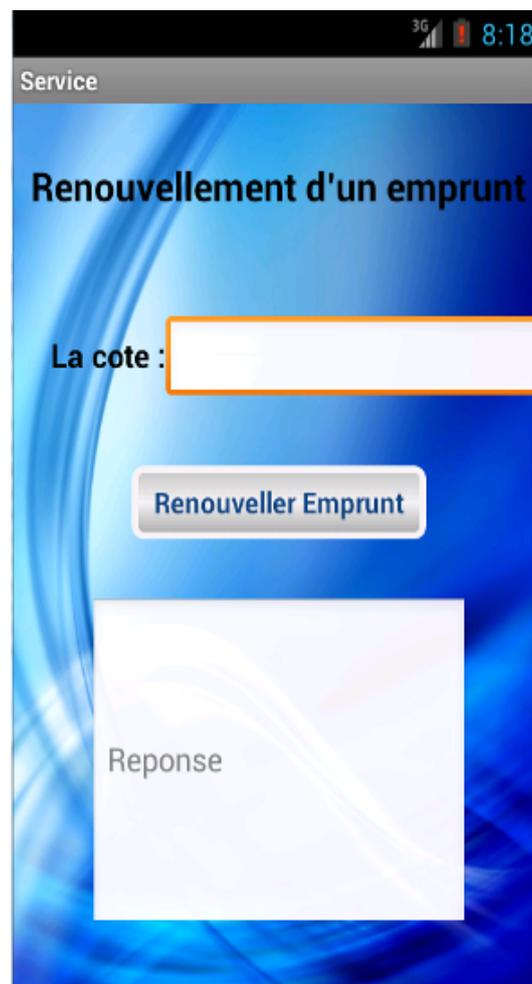
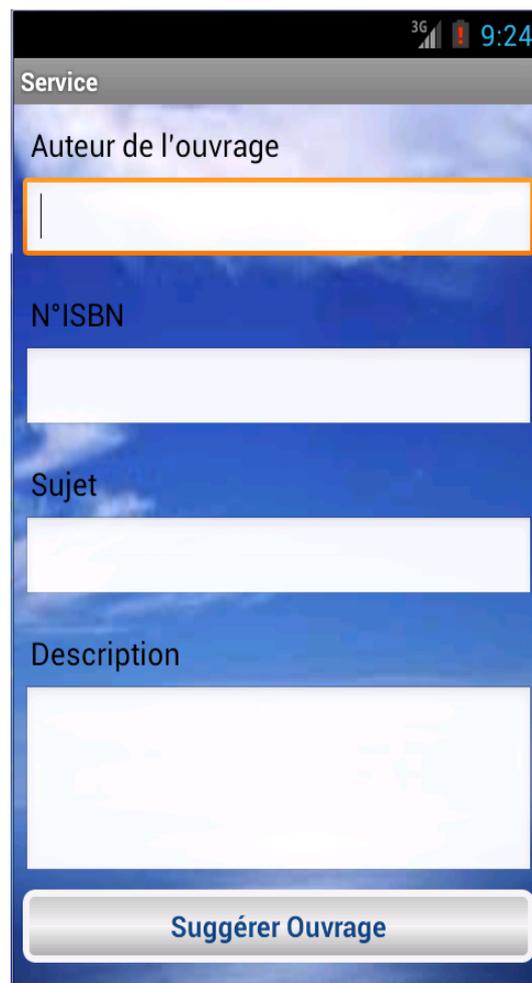


FIGURE 4.16 – Interface renouvellement d'emprunt.

✓ Interface de Suggestion d'ouvrages

Cette interface offre un aperçu de l'interface de suggestion d'ouvrages de l'application SSB-UB.

Dans cette interface, nous pouvons voir quatre champs vides (Auteur de l'ouvrage, Numéro ISBN, Sujet et description), l'étudiant devra saisir tout ces champs puis cliquer sur le bouton 'suggerer ouvrage'.



The screenshot shows a mobile application interface titled 'Service'. It contains four text input fields stacked vertically. The first field is labeled 'Auteur de l'ouvrage' and is highlighted with an orange border. The second field is labeled 'N°ISBN', the third 'Sujet', and the fourth 'Description'. At the bottom of the form is a button labeled 'Suggérer Ouvrage'. The background of the form is a blue sky with clouds. The top status bar shows '3G' signal, a battery icon, and the time '9:24'.

FIGURE 4.17 – Interface suggerer ouvrage .

4.6.2 Interfaces Administrateur

4.6.2.1 Interface du site web pour l'administrateur de la scolarité

Voilà une interface de du site web que nous avons implémenté afin de gérer le service scolarité (gérer les étudiants, gérer les publications et gérer les réclamations).

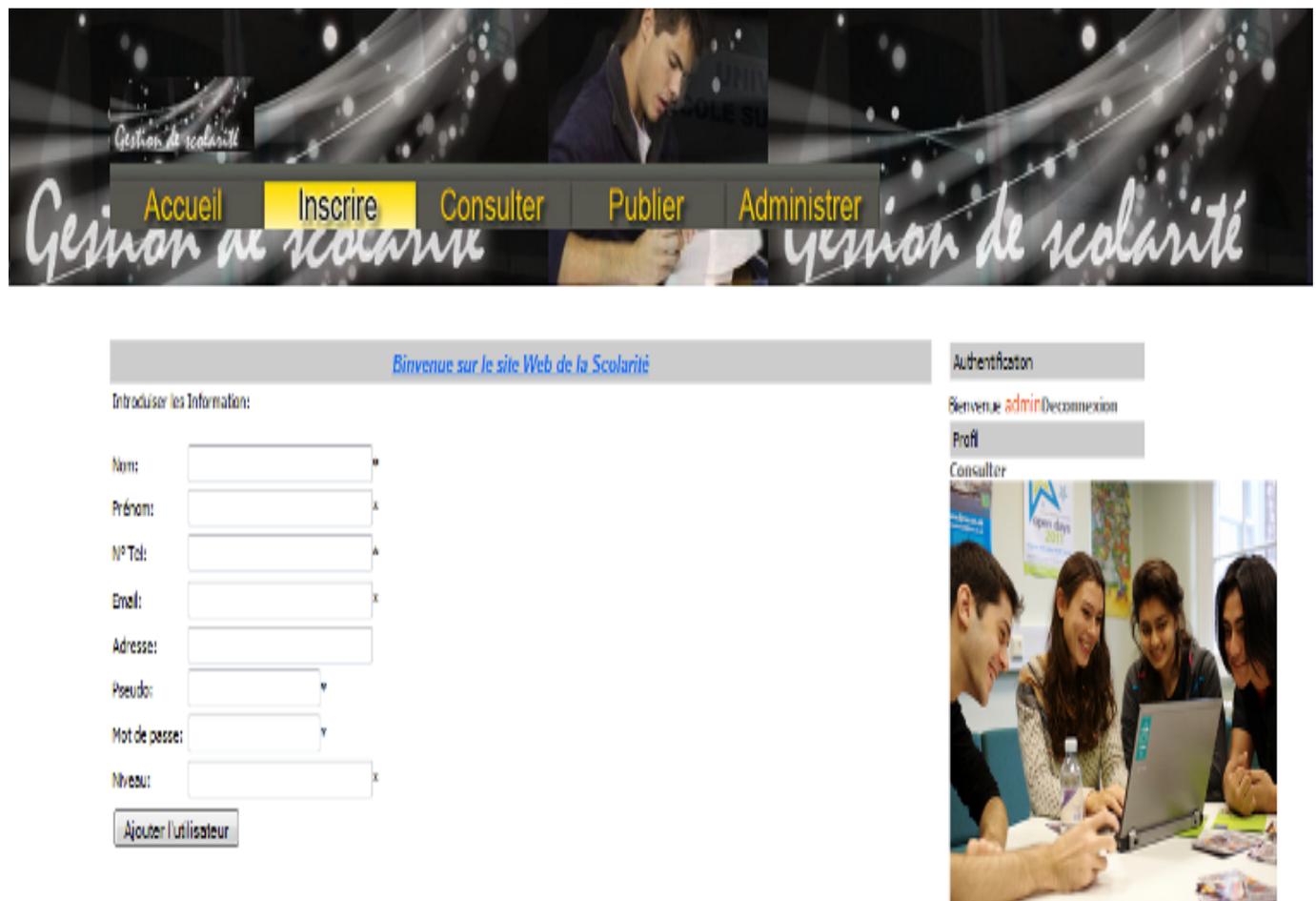
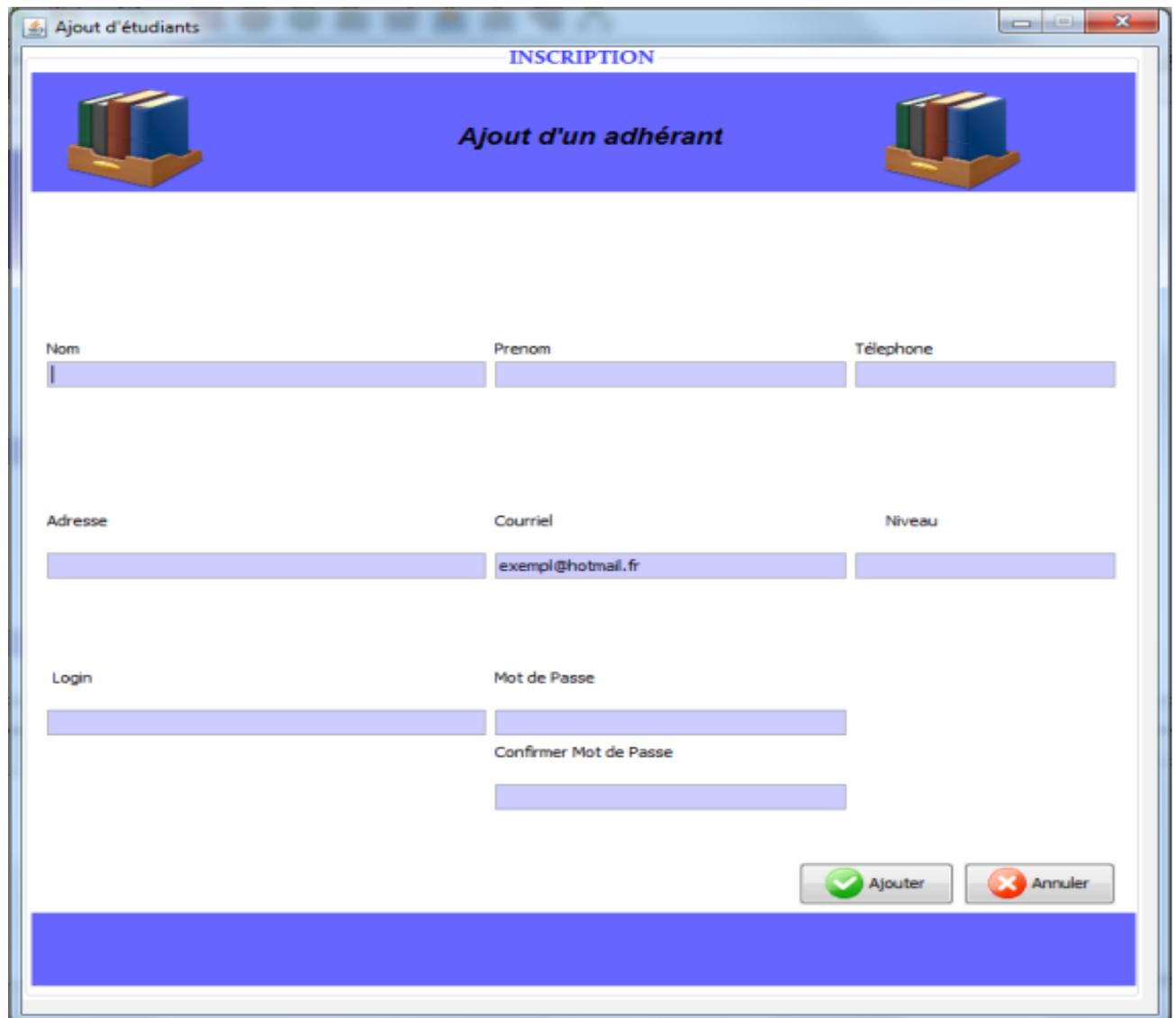


FIGURE 4.18 – Interface de l'ajout d'un étudiant.

4.6.2.2 Interface de l'application java pour l'administrateur du service bibliothèque

Voilà une interface de l'application java que nous avons implémenté afin de gérer le service bibliothèque(gérer les publications,gérer les étudiants et gérer les réclamations).



The screenshot shows a Java application window titled "Ajout d'étudiants". The window has a blue header bar with the text "INSCRIPTION" and "Ajout d'un adhérent" in the center. There are two book icons on either side of the header. Below the header, there are several input fields for user registration:

- Nom: []
- Prenom: []
- Téléphone: []
- Adresse: []
- Courriel: [exempl@hotmail.fr]
- Niveau: []
- Login: []
- Mot de Passe: []
- Confirmer Mot de Passe: []

At the bottom right, there are two buttons: "Ajouter" (with a green checkmark icon) and "Annuler" (with a red X icon).

FIGURE 4.19 – Interface de l'ajout d'un adhérent.

4.7 Tests

Cette phase contient les différents tests qu'a subit notre l'application afin de détecter les anomalies et les défauts : tester la rapidité et le déroulement des fonctionnalités.

Pour cela nous avons choisis un exemple de chacun des services scolarité et bibliothèque qui sont : le service de téléchargement d'une ressource pour le service scolarité et le service de réservation d'ouvrages pour le service bibliothèque.

4.7.1 Test de téléchargement

A travers ces trois interfaces nous pouvons voir le déroulement du service de téléchargement de ressources, cela commence en saisissant l'URL de la ressource ou en choisissant une ressource dans la liste prédéfinis des ressources (comme affichée dans la première interface) puis cliquer sur le bouton 'télécharger' , ainsi la ressource va commencer a ce télécharger et cela en affichant une barre de progression du téléchargement (comme illustré sur la deuxième figure) et une fois que le téléchargement sera terminé, un message de confirmation que le fichier à bien été téléchargé va s'afficher à l'écran(comme illustré dans la troisième figure).

Ainsi , apres avoir testé ce service nous constatons que l'opération du téléchargement s'est faite de manière rapide à part si la ressource à télécharger est tres lourde cela risque de prendre un peu plus de temps.

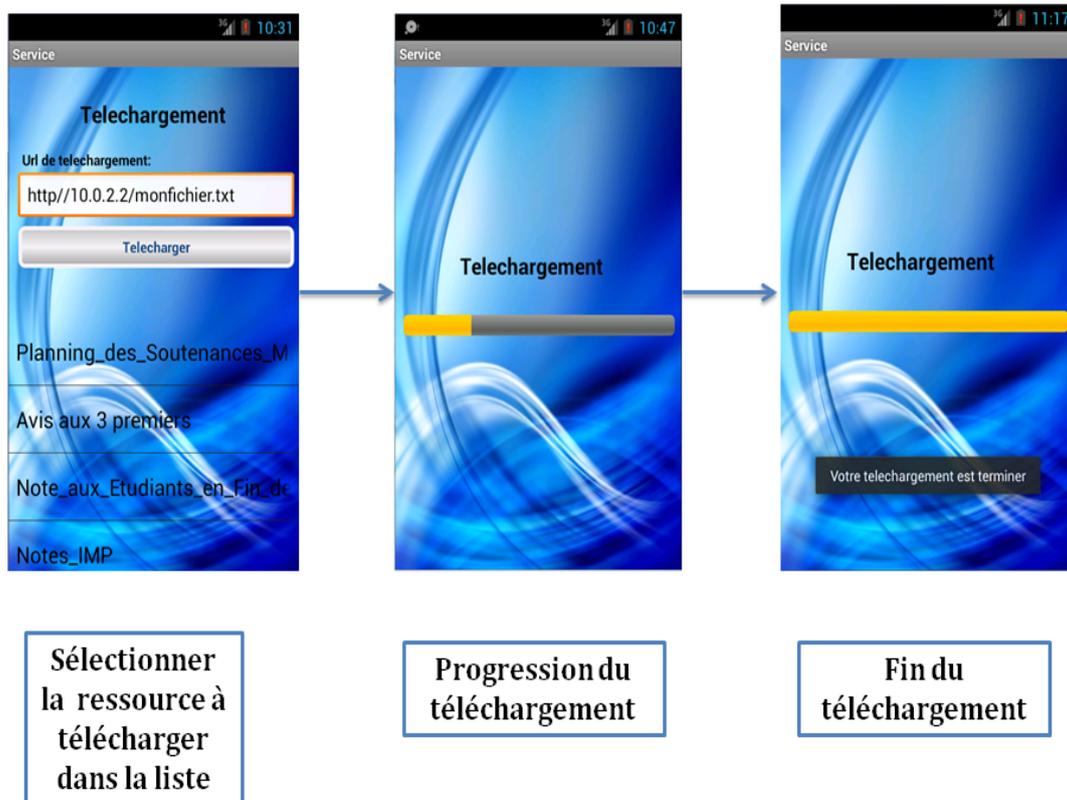


FIGURE 4.20 – Test de téléchargement.

4.7.2 Test de réservation

A travers ces trois interfaces nous pouvons voir le déroulement du service de réservation d'ouvrages , cela commence en saisissant le champs de la cote et les autres champs(Nom et sujet) vont automatiquement etre importé, puis cliquer sur le bouton 'reserver l'ouvrage' (comme illustré dans le première figure) puis une verification va se faire afin de valider la réservation (si l'ouvrage n'est pas deja réservé et qu'il n'est pas disponible comme illustré dans la deuxième figure) , et apres cela un message de validation de la réservation va s'afficher à l'écran (comme illustré dans la troisième figure).

Ainsi,apres avoir testé ce service,nous concluons que l'opération de réservation d'ouvrages s'est faite facilement et rapidement .

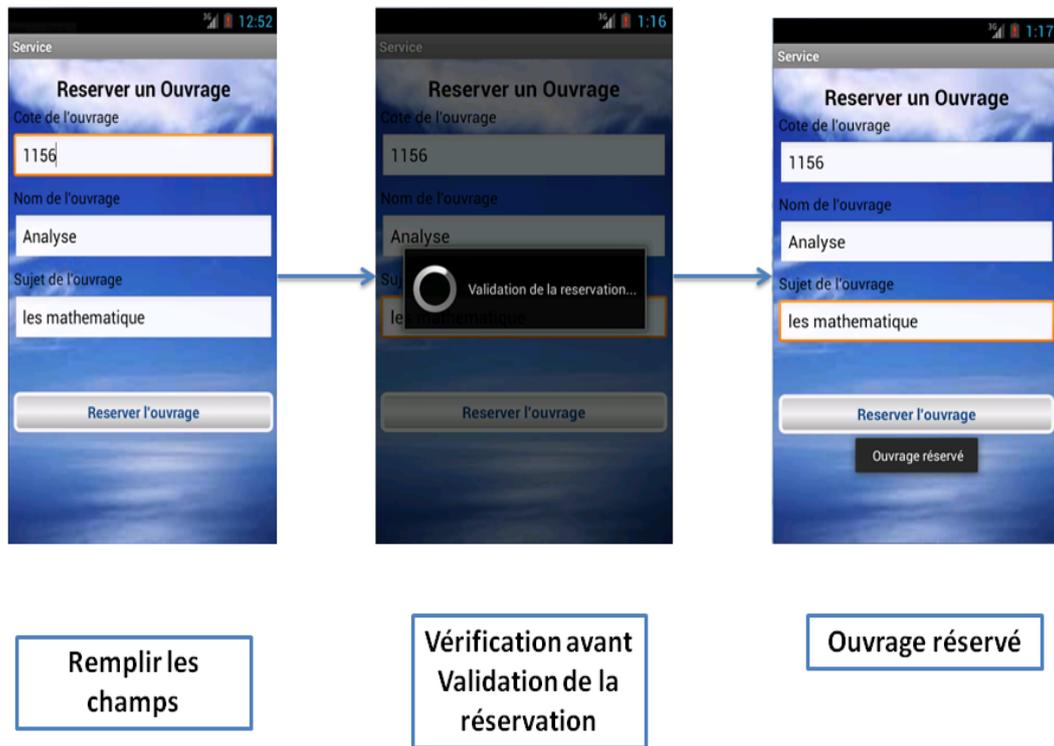


FIGURE 4.21 – Test de réservation.

4.8 Conclusion

Dans ce chapitre, nous avons décrit brièvement l'environnement logiciel en spécifiant l'environnement de développement, l'architecture matérielle du système, l'implémentation de quelques exemples de la base de données, puis donné des exemples des interfaces graphiques de l'application mobile SSSB-UB ainsi que celles de la partie administration (l'application Java qui administre le service bibliothèque et le site web qui administre le service scolarité) et nous avons achevé le chapitre par deux exemples de tests importants.

Conclusion Générale

Au bout de notre cursus en Master professionnel " Administration et Sécurité des Réseaux ", nous avons été chargés de réaliser un projet de fin de cycle. Notre travail s'est basé sur la conception et l'implémentation d'une application pour les services d'aide aux étudiants de l'université A.Mira et plus spécialement les étudiants du département d'informatique ; sous Android.

Cette application mobile permet en premier lieu de répondre aux besoins des étudiants par la résolution de certaines contraintes qu'ils ont quotidiennement dans le service scolarité et le service bibliothèque .

En deuxième lieu, elle facilitera la tâche non seulement aux étudiants par le fait de la mobilité mais aussi aux administrateurs des deux services qui n'auront plus de problème d'encombrement au sein du lieu du service.

En plus de cela ,notre application mobile offrira de multiples services supplémentaires par rapport au système classique tel que le service d'alerte(par exemple,évitera à un étudiant d'être exclu de la bibliothèque en l'informant du délai de l'emprunt),le service de réservation d'ouvrages(qui permettra à l'étudiant de réserver un ouvrage sans avoir à se déplacer à la bibliothèque),le service noter un ouvrage (qui permettra à l'étudiant de donner son avis sur la qualité d'un ouvrage).

Ainsi nous avons décentralisé l'endroit où l'étudiant peut rechercher une ressource ou un ouvrage.

En effet,ce projet nous a été bénéfique, de par l'utilisation de diverses technologies (android, J2EE, JAVA, PHP) et notre analyse des besoins ainsi qu'une conception détaillée de notre base de données et de nos interfaces a fait que nous avons pu capturer

certains besoins de l'université que nous avons essayé d'implémenté du mieux que l'on a pue.

Dans notre travail,l'ambition est de fournir un maximum de services aux étudiants,ce que nous avons essayé de faire,pour ce qui reste, nous avons dressé quelques perspectives :Ajouter les fonctionnalités manquantes aux deux services : scolarité et bibliothèque ; comme la gestion des documents ,en utilisant les codes barres et la gestion des prêt.Elargir le champ de cette application pour gérer les autres services de l'université, comme le service COUS ou CNAS.Faire une extension de notre application mobile pour les deux universités de Bejaia afin que ça soit une application qui va regrouper toutes les facultés existantes et aussi optimiser les communications entre l'application et le serveur.

Bibliographie

- [1] *ADT Plugging*. Date de consultation 06 2013. <http://developer.android.com/tools/sdk/eclipse-adt.html>.
- [2] *Avantages et inconvénients d'une application mobile*. Date de consultation 11 2012. <http://www.contenus-en-ligne.com/avantages-inconvenients-dune-application-mobile>.
- [3] *The Birth of Mobile Applications*. Date de consultation 11 2012. <http://skylineapps.wordpress.com/2012/01/16/the-birth-of-the-mobile-applications/>.
- [4] *Connecté ou déconnecté, synchrone ou asynchrone ?* Date de consultation 01 2013. <http://194.51.85.66/nomad-consulting/alors-mode-connecte-ou-deconnecte>.
- [5] *Définition de Tomcat*. Date de consultation 05 2013. <http://www.journaldunet.com/encyclopedie/definition/972/34/20/tomcat.shtml>.
- [6] *Définition de PDA*. Date de consultation 12 2012. <http://www.6ma.fr/lexique/informatique/pda/>.
- [7] *Java*. Date de consultation 06 2013. http://www.futura-sciences.com/fr/definition/t/internet-2/d/java_485/.
- [8] *Java SE Technologies Database*. Date de consultation 06 2013. <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>.
- [9] *Javascript*. Date de consultation 06 2013. http://www.futura-sciences.com/fr/definition/t/internet-2/d/javascript_509/.
- [10] *Json*. Date de consultation 06 2013. <http://www.json.org/>.
- [11] *Le Cloud Computing*. <http://www.wygam.com/documents/cloud-computing.pdf>.
- [12] *Les différents types d'applications mobiles : natives, web apps, hybrides, flash*. Date de consultation 11 2012. <http://olivierguillet.com/category/applications-mobiles/>.

- [13] *L'IDE Eclipse*. Date de consultation 05 2013. <http://www.techno-science.net/?onglet=glossaire&definition=517>.
- [14] *L'informatique mobile*. Date de consultation 11 2012. <http://deptinfo.unice.fr/~renevier/corrections/theseHTML/informatiquemobile.html>.
- [15] *Linux mobile : MeeGo*. Date de consultation 01 2013. <http://www.pcinpact.com/news/66060-tizen-mobile-linux-meego-evolution-html5-lancement.htm>.
- [16] *Linux Mobile Foundation (LiMo Foundation)*. Date de consultation 06 2013. <http://www.techopedia.com/definition/3383/linux-mobile-limo-foundation>.
- [17] *L'évolution des systèmes pour smartphone*. Date de consultation 11 2012. <http://www.techno-science.net/?onglet=news&news=11005>.
- [18] *Oral PHP MySQL ASP*. Date de consultation 06 2013. <http://www.ed-productions.com/FTP/expose/Php-MySql-ASP.pdf>.
- [19] *Palm OS (Palm Operating System)*. Date de consultation 01 2013. <http://searchmobilecomputing.techtarget.com/definition/Palm-OS>.
- [20] *Présentation d'Android*. Date de consultation 02 2013. <http://www-igm.univ-mlv.fr/~dr/XPOSE2011/SDKAndroid/android.html>.
- [21] *SDK*. Date de consultation 06 2013. <http://www.techterms.com/definition/sdk>.
- [22] *Smartphones*. Date de consultation 11 2012. http://www.futura-sciences.com/fr/doc/t/telecoms/d/smartphones-mobile_1487/c3/221/p2/.
- [23] *Tablette*. Date de consultation 12 2012. http://www.futura-sciences.com/fr/definition/t/technologie-2/d/tablette_12437/.
- [24] *What is a smartphone ?* Date de consultation 11 2012. http://cellphones.about.com/od/smartphonebasics/a/what_is_smart.htm.
- [25] *XML*. Date de consultation 06 2013. http://www.futura-sciences.com/fr/definition/t/internet-2/d/xml_3997/.
- [26] C. Battard, S. Brivet, F. Gacogne, and M.Nouchy. *Informatique nomade*. Mai 2004. <http://frantzgacogne.free.fr/ftp/informatiquenomade.pdf>.
- [27] M. Benslimane. *Conception d'une Application Mobile sous IOS*. PhD thesis, Université A.Mira, Bejaia, Juin 2012.
- [28] A. Bounceur. *Programmation Android et Géolocalisation*. Bretagne, 2012.
- [29] G. Carpentier and T. C. B. Crombe. *Solutions Mobiles avec les logiciels IBM Lotus, DB2, WebSphere, trivoli et Rational*. paris, edition dunod edition, 2003.
- [30] M. Dahdeh. *Conception développement et intégration d'une application embarquée de téléchargement des applications Android 'FTAB Store'*. Tunis, 2011.

- [31] M. Farinone. *Présentation de la Platform Android*. 2011. <http://cedric.cnam.fr/~farinone/SETM0/seance1PourSETM0.pdf>.
- [32] F.Clark. *history of mobile applications*. 2012. <http://www.uky.edu/~jclark/mas490apps/History%20of%20Mobile%20Apps.pdf>.
- [33] A. Huth and J. Cebula. *The Basics of Cloud Computing*. Etats Unis, 2011.
- [34] M. R. Islam, M. R. Islam, and T. Mazumder. Mobile application and its global impact. *International Journal of Engineering and Technology (IJET)*, 10 :06, 2010.
- [35] J.Gabay and D.Gabay. *Mise en œuvre guidée avec études de cas*. 2011.
- [36] M. Meier. *Android Application Développent*. Etats Unis (Indiana), wiley publishing edition, 2009.
- [37] A. Messaoudi and Y. Djebara. *Conception et Réalisation d'une Application Réseau pour le Système ANDROID*. PhD thesis, Université A.Mira, Bejaia, Juin 2012.
- [38] P.Gerard. *Processus de Développement Logiciel*. Paris, Éditions eyrolles edition, 2008. <http://lipn.univ-paris13.fr/~gerard/docs/cours/methodo-support.pdf>.
- [39] P.Roques. *UML 2 par la pratique : étude de cas et exercices corrigés*. Paris, Éditions eyrolles edition, Septembre 2011.
- [40] P.Roques and F.Vallée. *uml en action de l'analyse des besoins à la conception en Java*. Paris, Éditions eyrolles edition, 2003.
- [41] P. Roques. *Le cahier du programmeur : UML 2 Modéliser une application web*. Paris, 4 Éditions eyrolles edition, Juin 2008.
- [42] C. Soutou. *Apprendre SQL avec MySQL*. Paris, 2006.

Glossaire

-A-

Accéléromètre : calcul l'accélération linéaire d'un mobile ou autre c'est-à-dire calculer la vitesse du processus.

Application store : est une plateforme de téléchargement d'applications, similaire au Google Play distribué par les appareils mobiles fonctionnant sous Android ou bien l'App Store distribuée par Apple sur les appareils mobiles fonctionnant sous iOS (iPod Touch,iPhone et iPad) .

AVD : **A**ndroid **V**irtual **D**evice qui est un émulateur ou en d'autres termes, c'est un terminal mobile virtuel .

-B-

BLOB ou **BLOB Binaire** : pour **B**inary **L**arge **O**bject, est un type de donnée permettant le stockage de données binaires (le plus souvent des fichiers de type image, son ou vidéo) dans le champ d'une table d'une base de données.

-C-

CRM : signifie **C**ustomer **R**elationship **M**anagement en Anglais soit Gestion de la Relation Client (GRC). La gestion de la relation est une stratégie par laquelle l'entreprise vise à comprendre, à anticiper et à gérer les besoins de ses clients actuels et potentiels.

-D-

Dalvik : est la machine virtuelle pour le langage Java des applications du système d'exploitation Android de Google. Le nom provient d'une ville d'Islande, d'où le créateur est originaire. Dalvik est produit par l'Open Handset Alliance.

-E-

ECMAScript : est un langage de programmation de type script standardisé par Ecma International . Il s'agit donc d'un standard, dont les spécifications sont mises en œuvre dans différents langages script, comme JavaScript ou ActionScript, ainsi qu'en C++ (norme 2011). C'est un langage de programmation orienté objet.

EDGE : **E**nhanced **D**ata Rates for **G**SM **E**volution (acronyme anglais signifiant tranchant (d'une lame)) est une norme de téléphonie mobile, une évolution du GPRS qui est une extension de GSM avec une compatibilité ascendante(la compatibilité d'un produit récent vis-à-vis de ses anciennes versions).

EDI ou **E**nvironnements de **D**éveloppement **I**ntégré : Un EDI est un ensemble d'outils destinés à programmer dans un langage donné, qui sont distribués ensemble. Il permet de manipuler les outils de programmation depuis une interface graphique simplifiée.

-G-

3G /3G+ : La troisième génération (3G) désigne une génération de normes de téléphonie mobile permettant des débits bien plus rapides (de 2 à 42 Mb/s prévus à maturité du réseau) qu'avec la génération précédente, par exemple le GSM. Et la 3G+ est l'évolution de la 3G en 3.5G.

-H-

HTC (**H**igh **T**ech **C**omputer Corporation) : est un fabricant taiwanais qui fabrique des smartphones destinés au départ à la plate-forme Windows Mobile de Microsoft et depuis

2008, à Android, le système d'exploitation open-source de Google mais aussi à Windows Phone.

-I-

IHM : Les **interactions homme-machine** (les IHM) définissent les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine.

iTunes : est un logiciel de lecture et de gestion de bibliothèque multimédia numérique distribué gratuitement par Apple. Il est disponible officiellement sur Mac OS X, Windows, et peut fonctionner sur GNU Linux par émulation.

-J-

JSP : Extension de la technologie Java Servlet de Sun qui permet de programmer simplement l'affichage de contenus dynamiques sur le Web. JSP consiste en une page HTML incluant du code Java qui s'exécutera soit sur le serveur Web, soit sur le serveur d'application.

-M-

Android Market : Google Play, anciennement dénommé Android Market, est le magasin en ligne de Google. Celui-ci permet de télécharger des logiciels, des livres, des films ou de la musique, payants ou non.

Métier (processus) : Un processus métier est un ensemble d'activités, entreprises dans un objectif déterminé. La responsabilité d'exécution de tout ou partie des activités par un acteur correspond à un rôle. Le déroulement du processus utilise des ressources et peut être conditionné par des événements, d'origine interne ou externe. L'agencement des activités correspond à la structure du processus.

MMS (**M**ultimedia **M**essaging **S**ervice) : " service de messagerie multimédia ", est un système d'émission et de réception de messages multimédias pour la téléphonie mobile.

Multitouch : Un dispositif multi-touch, multi-tactile ou tactile multipoint est à

la fois une technique d'interaction humain-ordinateur et le matériel qui la met en application. Ce dispositif sert à interagir avec le matériel informatique par le biais de plusieurs points de contact (souvent avec plusieurs doigts).

-P-

Plugin : c'est un petit logiciel qui se greffe à un programme principal **PUSH** (notification) : c'est un message d'alerte envoyé à l'utilisateur d'un Smartphone et qui est lié à l'installation d'une application mobile.

-R-

Runtime AIR : **A**dobe **I**ntegrated **R**untime; anciennement nommé Apollo, est un logiciel moteur pour exécuter des logiciels créés avec Adobe Flash et ActionScript sans utiliser de navigateur Web.

RSS (**R**eally **S**imple **S**yndication) : Est une famille de formats de données utilisés pour la syndication de contenu Web.

-S-

SAP : Est un système dans lequel les différentes fonctions de l'entreprise (comptabilité, finances, production, approvisionnement, marketing, ressources humaines, qualité, maintenance, etc.) sont reliées entre elles par l'utilisation d'un système d'information centralisé sur la base d'une configuration client/serveur.

SDK (**S**oftware **D**éveloppement **K**it) ou kit de développement ou trousse de développement logiciel : Est un ensemble d'outils permettant aux développeurs de créer des applications de type défini (par exemple pour iOS, Android, Symbian, Bada ou Windows Phone 8 etc).

Système multi-compte : Est un système qui permet d'avoir plusieurs profils

utilisateurs sur une tablette, comme c'est déjà le cas sur un ordinateur, et donc une tablette personnalisée pour chaque utilisateur de la tablette.

-T-

Threading : C'est la gestion d'un fil d'instructions à l'intérieur d'un processus.

-X-

XML : De l'anglais 'e**X**ensible Markup **L**anguage' (En français 'langage de balisage extensible') est un langage informatique de balisage générique.

-W-

Webkit : Est une bibliothèque logicielle permettant aux développeurs d'intégrer facilement un moteur de rendu de pages Web dans leurs logiciels.

Web Services : Un web service est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et de manière synchrone.

Windows Live : Est un service mis en place par Microsoft destiné à faciliter l'identification sur les sites Internet. Il permet, grâce à une unique adresse de messagerie et à un mot de passe, de se connecter sur tous les sites web autorisés.

WML (Wireless Mark-up Language) : Il correspond au langage XML réalisé dans le but d'afficher des données sur l'écran d'un mobile.

-Z-

Zune : Le Zune est le lecteur multimédia de la société Microsoft et principal concurrent de l'iTunes.

Annexe

Etapes d'insatation des outils nécessaires à la création d'applications mobiles sous Android

1.Installation du JDK

- Téléchargement

Il faut commencer par télécharger le JDK , voila un lien ou nous pouvons le télécharger : <http://www.oracle.com/technetwork/java/javase/downloads/index.html> . Ensuite , cliquer sur le bouton "Download JDK".Puis cliquer sur le bouton : "Download" dans la page qui s'affiche.

Sur la page suivante, choisir le système d'exploitation (Platform)puis Cliquer sur le bouton "Continue".

Enfin,il faut cliquer sur l'exécutable à télécharger : jdk-6u21-windows-i586.exe dans mon cas.

- Installation

Pour lancer l'installation,il faut double-cliquer sur l'exécutable précédemment téléchargé et se laisser guider .

2.Le SDK android

Maintenant que nous avons installé le JDK, nous avons donc les outils nécessaires pour développer un programme java mais une application Android est un programme particulier, nous avons les outils de bases mais il nous manque les outils spécifiques à Android.

L'outil qui nous manque est le SDK Android. Il contient ce dont nous avons besoin, non

pas pour développer un programme Java classique, mais pour développer une application Android. On peut dire qu'il s'ajoute au JDK.

- Téléchargement du SDK Android et décompression

Avant d'installer le SDK, il faut d'abord le télécharger, voilà un lien où c'est possible de le télécharger : <http://developer.android.com/sdk/index.html>, il faut choisir un fichier qui convient au système d'exploitation de l'ordinateur.

Le fichier téléchargé sera un fichier compressé (zip pour Windows et Mac ou tgz pour GNU/Linux).

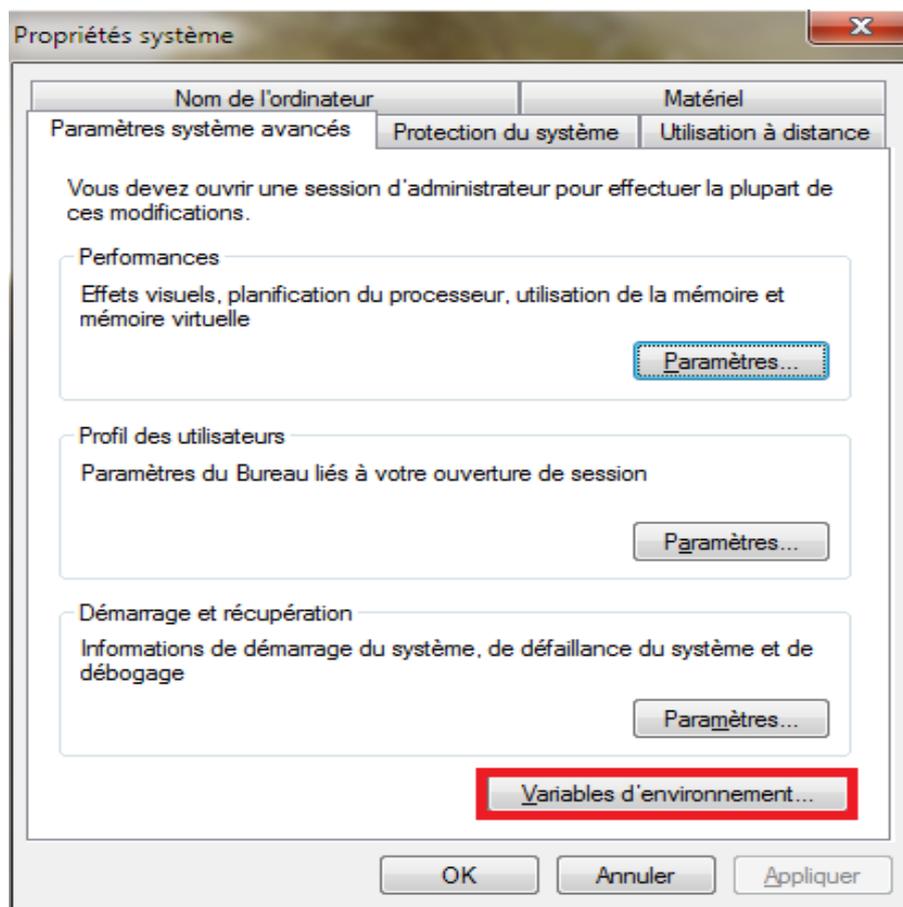
Après téléchargement du fichier il faut le décompresser et le sauvegarder dans le disque dur de l'ordinateur.

La prochaine étape consiste à ajouter un PATH, c'est à dire ; définir le répertoire où on a mis le SDK au sein d'une variable d'environnement.

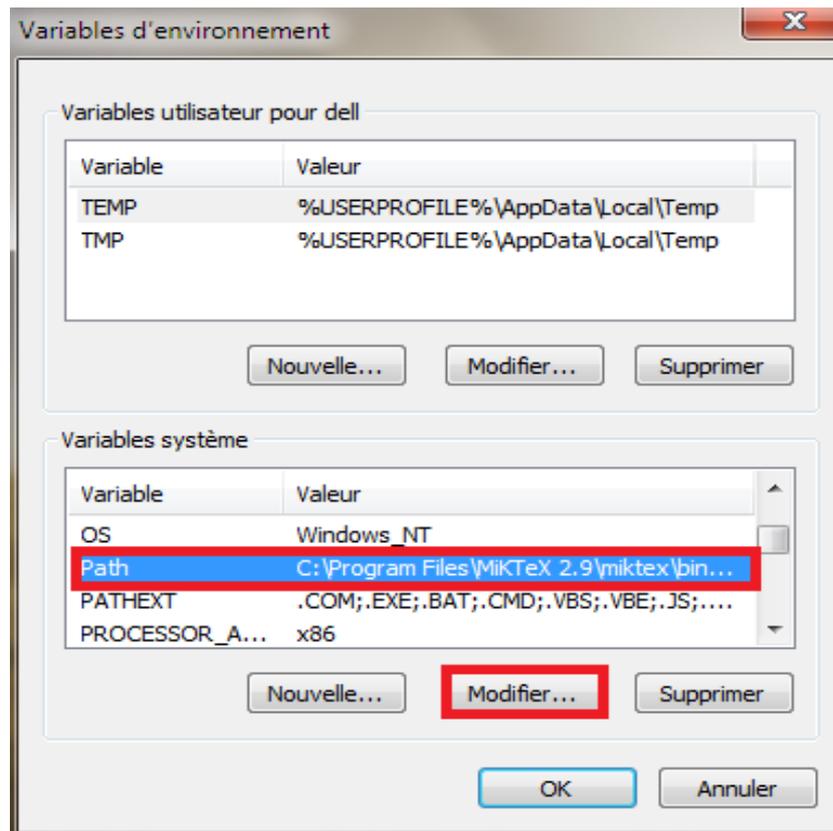
- Ajout du SDK au PATH

Dans Windows, faire un clic-droit sur le poste de travail (ou sur "Ordinateur" depuis Vista) et choisir "Propriétés".

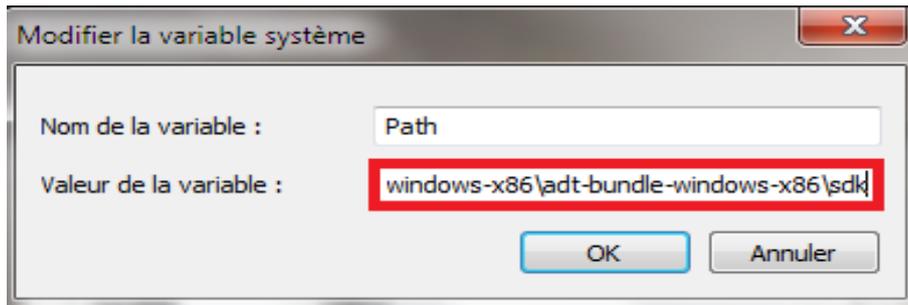
Se placer sur l'onglet "Avancé" (ou bien cliquer sur "Paramètres système avancés" dans le menu de gauche depuis Vista) et cliquer sur "Variables d'environnement" :



Dans les "Variables système", chercher "Path" et cliquer sur "Modifier" :

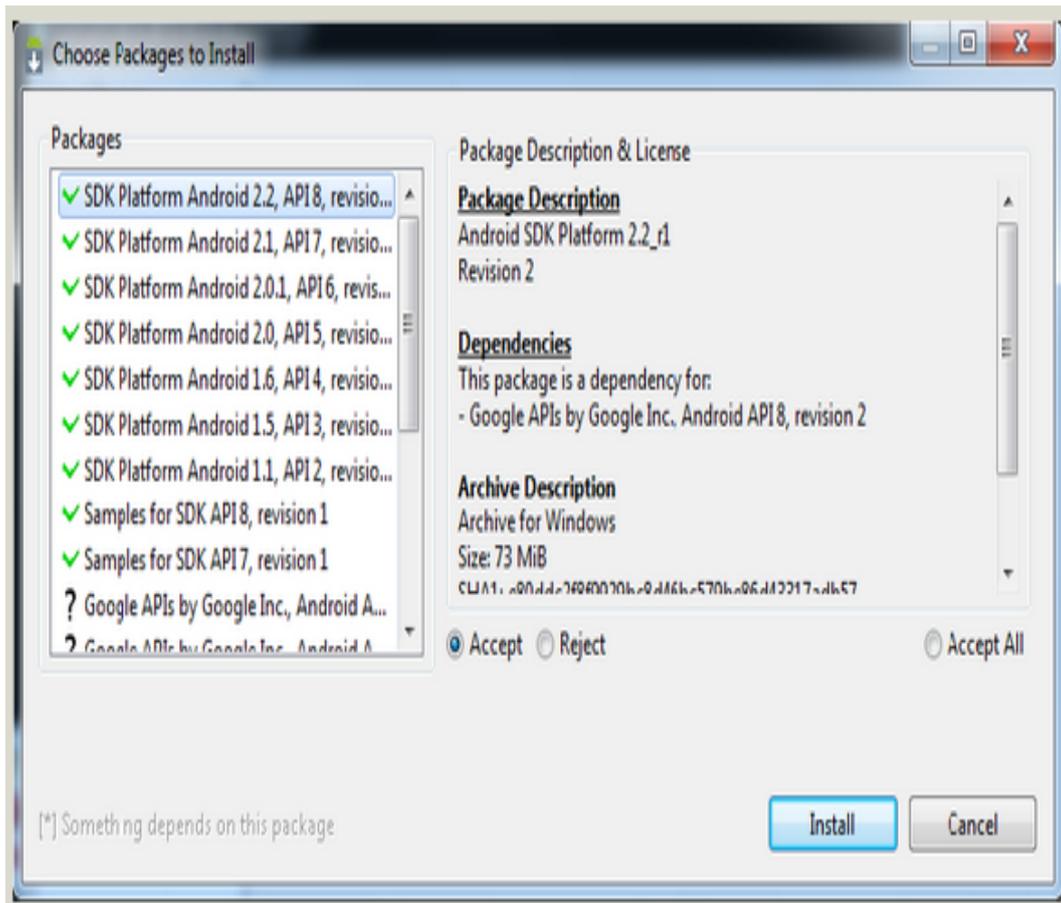


Ajouter à la fin de la ligne "Valeur de la variable" le chemin du SDK Android, précéder d'un point-virgule (afin de le séparer du répertoire précédent) :



3.Installation des plates-formes Android

- La dernière étape consiste à installer au moins une version de plate-forme Android.
- Dans le répertoire du SDK, un exécutable est présent : SDK Manager.exe ,il s'agit d'une petite interface permettant de -gérer les versions d'Android.
 - Lancez-le pour démarrer l'installation.
 - Parmi les packages disponibles, en choisir un :



Sur Mac et GNU/Linux, il faut lancer depuis un terminal la commande "android" se trouvant dans le répertoire /tools/.

Une fois l'installation terminée, nous pouvons passer à la suite : l'installation d'Eclipse.

4. Installation d'Eclipse

- Téléchargement

Tout d'abord, nous devons télécharger eclipse, voila un lien ou nous pouvons le faire : <http://www.eclipse.org/downloads/> .

Nous aurons le choix entre beaucoup de "versions" d'Eclipse différentes mais c'est mieux de choisir "Eclipse IDE for Java Developers" :



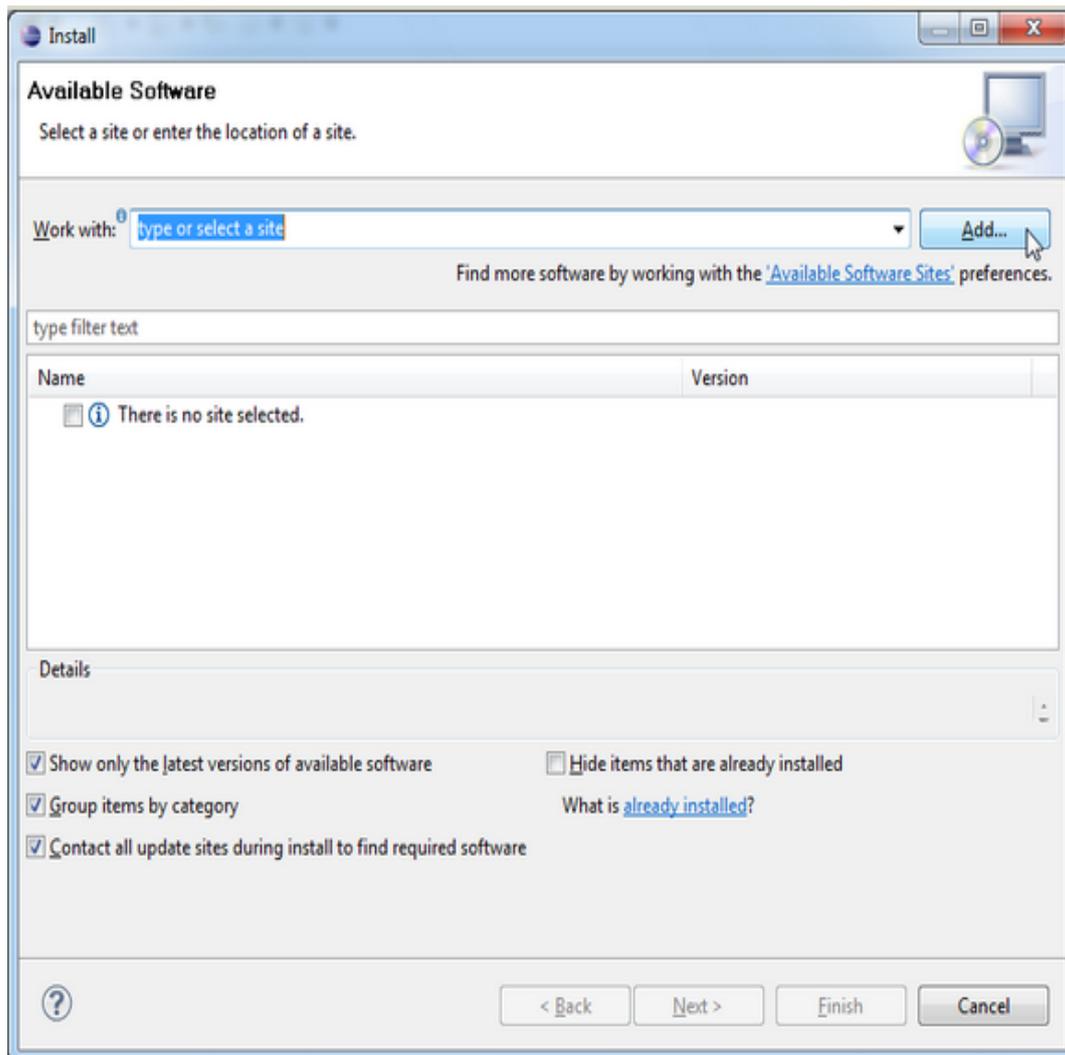
Choisir entre 32bits et 64bits.

- *Installation*

L'installation d'Eclipse se résume en réalité à décompresser le fichier téléchargé. Nous aurons après à le placer n'importe où sur le disque dur. Pour démarrer Eclipse, nous devons lancer l'exécutable présent dans le répertoire "eclipse". Une fois eclipse installé, nous aurons besoin d'un plugin afin de développer nos applications Android. Le plugin pour les développements d'applications Android s'appelle ADT (qui signifie en français "Outils de développement Android").

5. Installation d'ADT

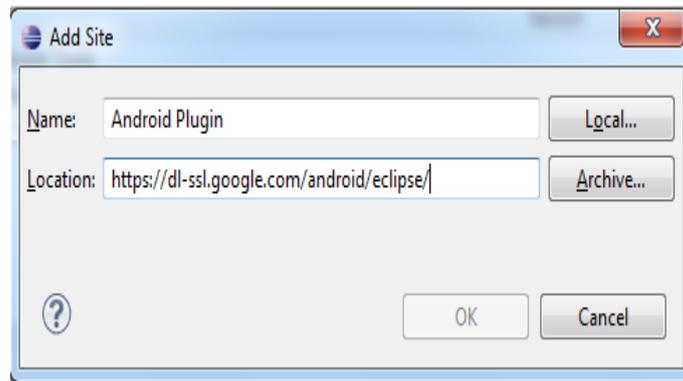
Après le démarrage d'Eclipse. Dans le menu "Help", choisir "Install New Software...". Dans la fenêtre qui s'ouvre, nous allons cliquer sur le bouton "Add..." :



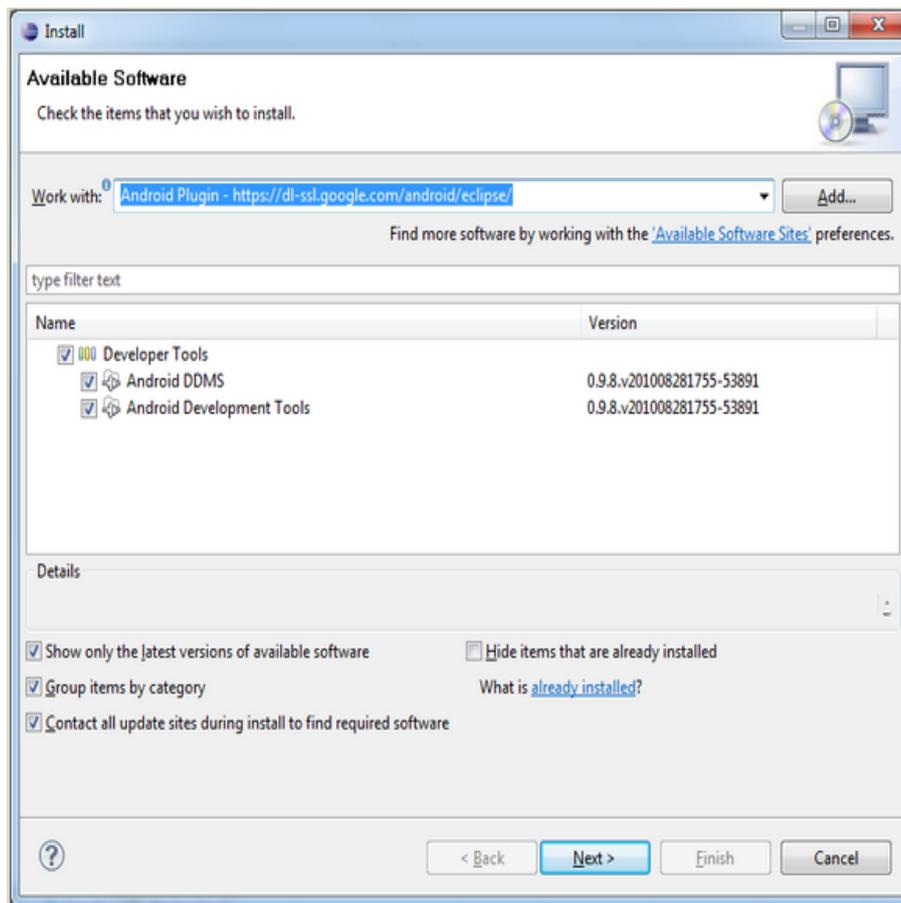
Puis apres , introduire les valeurs suivantes :

Name : Android Plugin (par exemple).

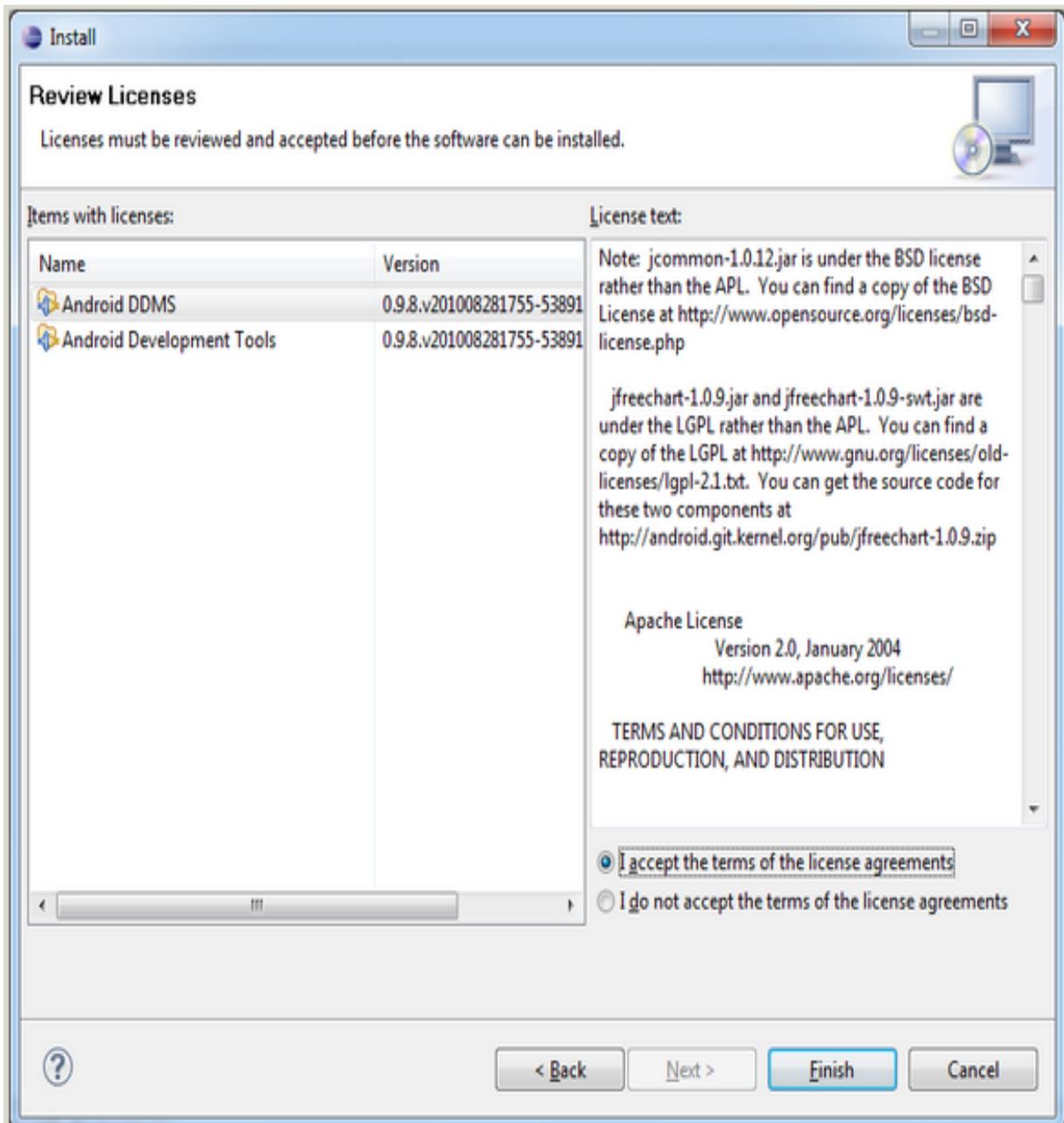
Location : [https ://dl-ssl.google.com/android/eclipse/](https://dl-ssl.google.com/android/eclipse/) .



Après, il faut cliquer sur OK puis patienter quelques secondes et la fenetre suivante apparaîtra :

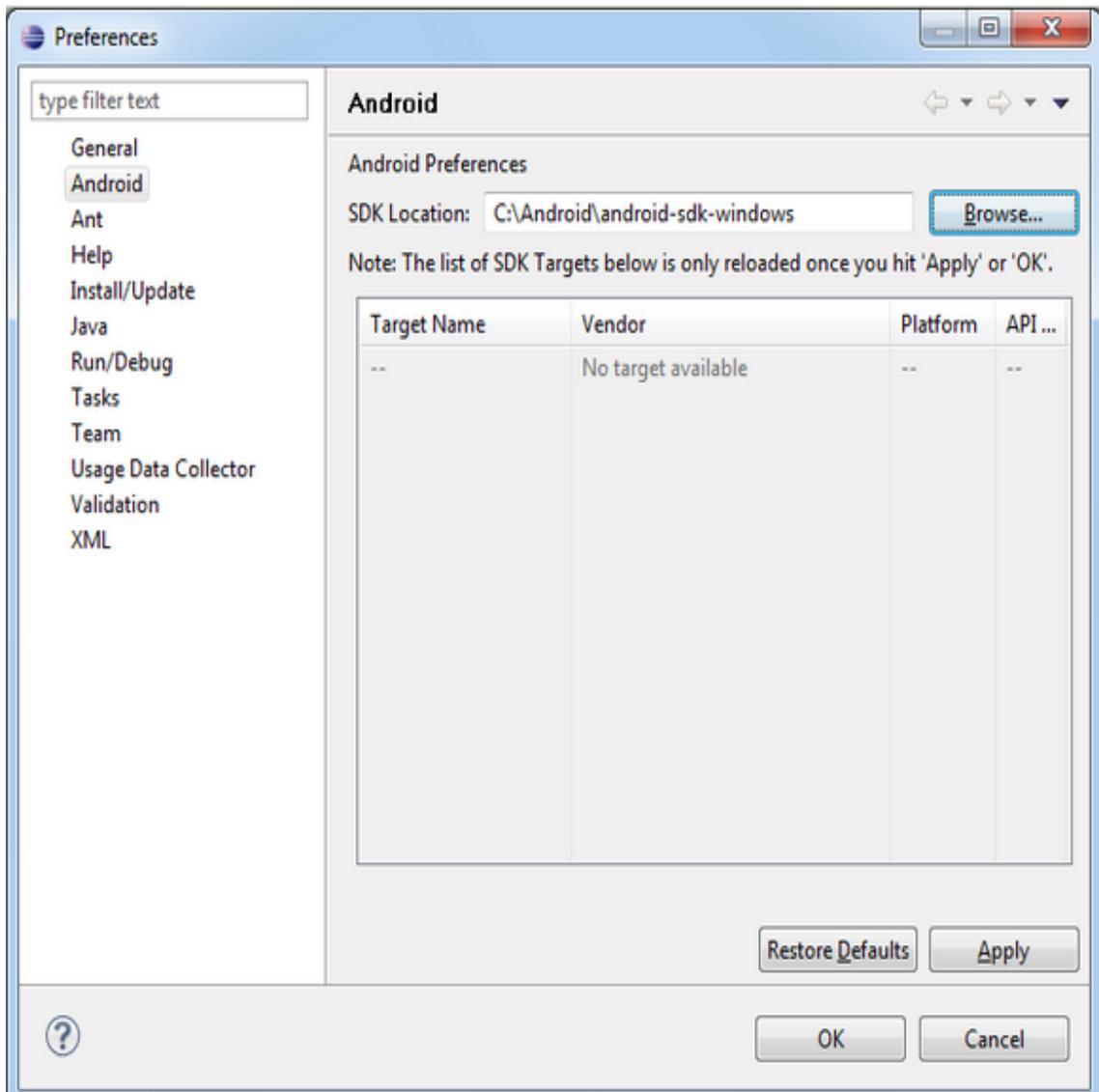


Il faut Cocher sur "Developer Tools" puis cliquer sur "Next". Sur la fenêtre suivante et cliquer à nouveau sur "Next". Vous arrivez alors à ceci :



Nous devons accepter les termes du contrat de licence et cliquer sur 'Finish', une fois que l'installation se termine, nous devront redémarrer Eclipse .

- Après cela , il faudra indiquer au plugin ADT où se trouve le SDK ,comme suit :
- Dans le menu "Window", sélectionner "Preferences".
 - Dans le panneau de gauche, choisir "Android" .

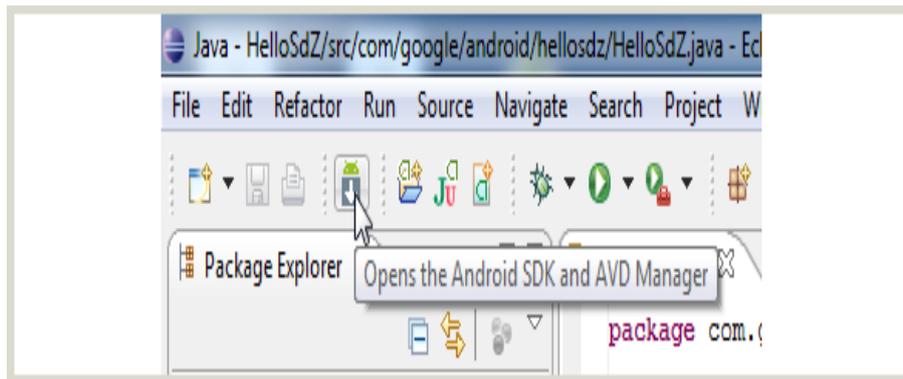


-Ainsi , nous devons cliquer sur 'Browse ' et indiquer l'emplacement du SDK puis cliquer sur ' Apply' .

-Pour tester notre application, nous allons utiliser un téléphone virtuel.

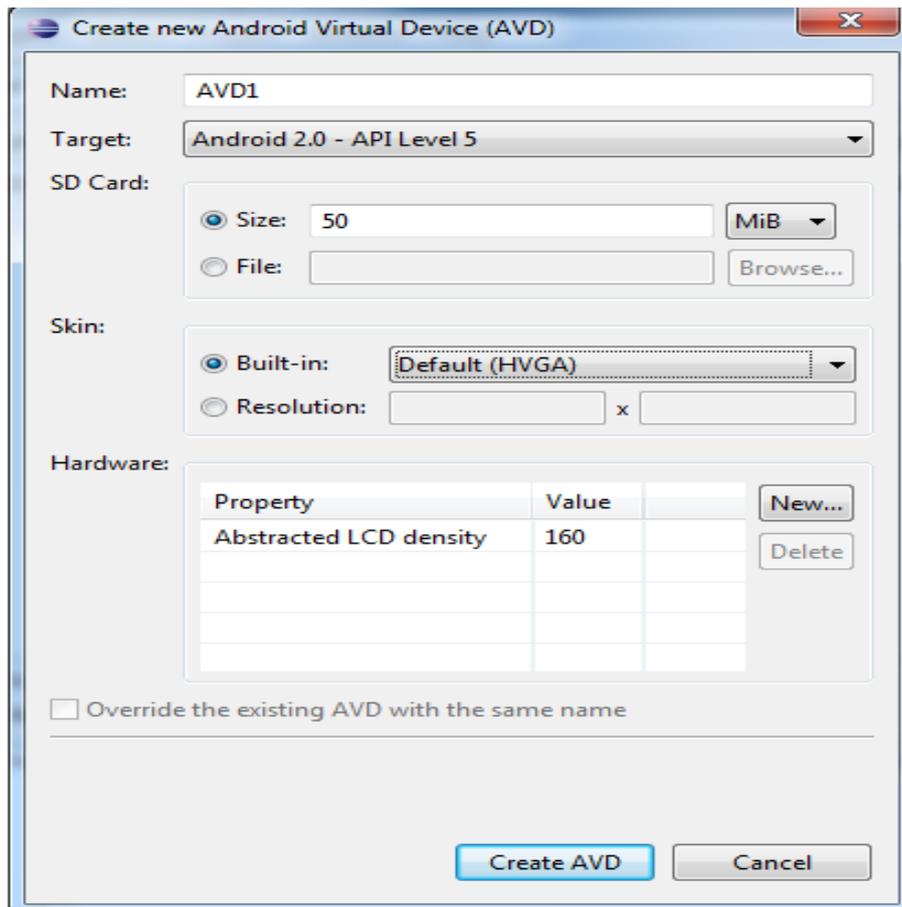
- Nos téléphones virtuels s'appellent des AVD, que l'on peut écrire en français "Périphérique Android Virtuel".

-Pour commencer, Il nous faut cliquer sur le petit bouton au long titre "Opens the Android SDK and AVD Manager



Cela ouvre le gestionnaire de SDK et d'AVD. Il nous faut Cliquer sur "Virtual Devices" dans le panneau de gauche.

Mais puis-ce-qu'on n'a aucun AVD pour le moment. On doit en créer un . Pour en créer un, On doit cliquer sur "New..." puis remplir les champ ;comme illustré dans la figure ci-dessous :

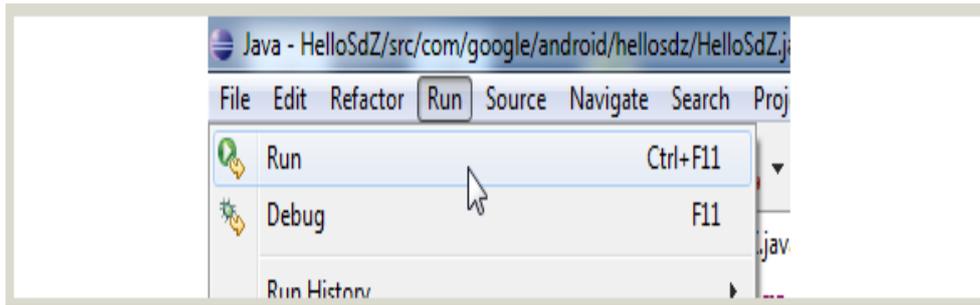


Nous devons cliquer sur le bouton "Create AVD" pour finaliser la création de notre mobile virtuel.

6. Créer un nouveau projet Pour commencer, il faut choisir dans le menu : "File", -> "New" -> "Project" puis cliquer sur "Android Project".
Après cela, cliquer sur "Next", remplir les champs puis cliquer sur 'finish'.
Ainsi un code sera généré par défaut.

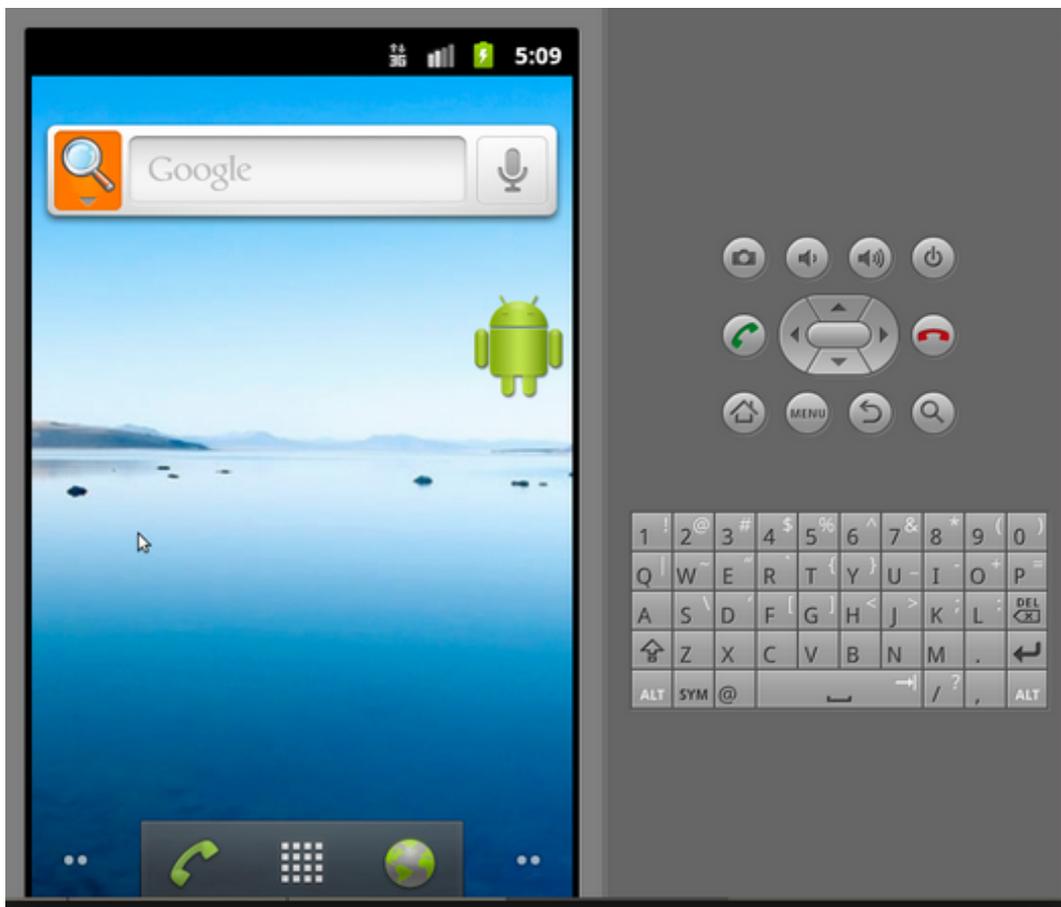
- Lancer l'application dans l'AVD

Pour lancer l'application il faut cliquer sur "Run" dans le menu. "Run" :



L'AVD va alors mettre quelques instants à se lancer : Nous verront apparaître les lettres "ANDROID", en gris sur fond noir.

Une fois que l'AVD sera entièrement démarré, l'application se lancera automatiquement :



RÉSUMÉ

Le travail présenté dans ce mémoire consiste en la conception et L'implémentation d'une application mobile pour les services d'aide aux étudiants de l'université A. Mira sous Android ; Il offre plus précisément deux services très importants au sein de l'université qui sont les services scolarité et bibliothèque, ainsi, il permet aux étudiants du département d'informatique d'avoir accès aux principaux services offerts par le département et en plus de cela il offre à l'ensemble des étudiants l'accès aux services de la bibliothèque ; tout cela à travers un terminal mobile.

Dans le cadre de ce travail nous avons défini de façon général les applications mobiles puis présenté d'une façon globale le système d'exploitation mobile utilisé, qu'est le système Android.

Pour concevoir ce travail, nous avons opté pour UP comme processus de développement logiciel et UML comme langage de modélisation .

Nous avons également présenté les langages de programmation Android 4.2,XML,PHP et javaScript avec lesquels notre application mobile à été implémenté ainsi que d'autres langages et sans oublier les SGBD MySQL pour l'implémentation de la base de données et notre application a été développée en utilisant un AVD qui est un émulateur d'applications android.

Mots clés : Informatique mobile, Applications mobiles, terminaux mobiles, Android, Java.

ABSTRACT

The work presented in this document is the design and implementation of a mobile application for support services for students of the University A.Mira in Android .It offers two very important services specifically in the university,are département and library services and it allows students from Department of Computer Science to have access to all services offered by the department and on top of that it offers to all students access to library services, all through a mobile device.

In this work,we defined in general,mobile applications then presented in a comprehensive way the mobile operating system, which is the Android system.

In designing this study, we opted for UP as développement software process and UML as modeling language.

Concerning the implementation, we have also presented the programming language JAVA Android 4.2, XML, PHP and javaScript with which our mobile application was implemented and others languages and not forgetting the MySQL DBMS for the implementation of the database and our application was defined using an AVD which is an emulator of Android.

Key words : mobile computing ,Mobile applications, mobile devices, Android, Java .