

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Bejaïa



Faculté de Technologie
Département : Génie électrique

Projet de Fin d'étude

En vue de l'obtention du diplôme de Master en
Télécommunication

Option : Réseaux et Télécommunication

Thème

*Application Android pour la Recherche Documentaire
Dans la Bibliothèque de Bejaia*

Réaliser par :

BENAABLA Lynda

Dirigé par :

Mme Meriem GAGAOUA.

Devant les jurys composés de :

Mme BELAHCEN Noura Rezika

Mr NAIT MOHAND Nassim

Année universitaire 2019/2020.

Dédicaces

À MES CHÈRES PARENTS

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consentie pour mon Instruction et mon bien être.

Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en n'acquitterai jamais assez.

À mon très cher frère Lyazid et mes adorables sœurs Asma et Imane

Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

À ma chère cousine Kahina et son mari Nacer pour leur aide depuis le début de mon parcours, à tous mes cousins Hakima, Sabrina, Sissa, Maya et Zouina, Mahdi et Ramzi.

À tous mes oncles Nacer, Aziz, Bakli... et tantes Sadrina, Hayet, Tonsia et Fahima...

À mes grand mères Nouara et Djamilia que dieu leur donne une longue et joyeuse vie.

À mes copines et amis avec qui j'ai passé ce formidable parcours universitaire Sonia, Sarah, Tinhinane, Dalil, Yacine, Adem. Merci pour leur amour et encouragements.

À mon très cher ami Lakhdar, qui n'a pas cessée de me conseiller, encourager et soutenu tout le long de mon travail. Merci pour m'avoir donnée autant d'attention.

Sans oublier mon binôme Syphax pour son soutien morale, sa patience et compréhension, tout au long de notre projet.

À toute ma famille,

À tous mes autres ami(e)s,

À tous ceux que j'aime et tous qui m'aiment.

Remerciements

Tout d'abord au terme de ce travail je tiens à remercier :

Mon DIEU qui m'a donnée le courage, et le pouvoir d'achever ce modeste travail.

Mon encadrante GAGAOUA Marieme pour son encadrement, et ces remarques constructives tout le long de ce travail.

Et ces qualités professionnelles et ces conseils judicieux,

Enfin je tiens à remercier également tous les membres du jury qui ont accepté d'évaluer mon travail.

Table des Matières

Table des Matières	
Liste des Figures.....	8
Liste des Tableaux.....	10
Liste des Abréviations	11
INTRODUCTION GENERALE.....	13
Chapitre 1 : Présentation de la Bibliothèque de BEJAIA.....	15
Introduction.....	16
1 Présentation de l'université de Bejaia	16
1.1 Historique.....	16
1.2 Présentation de la bibliothèque universitaire	17
1.2.1 Services de la bibliothèque	19
1.2.1.1 Service gestion des fonds (acquisition et traitement)	19
1.2.1.2 Service de l'orientation et de la recherche bibliographique.....	20
1.2.2 Le rôle et enjeux de la bibliothèque	20
1.2.3 Organigramme de la bibliothèque	21
1.2.4 Les types de bibliothèques.....	22
1.2.5 Présentation du site de recherche documentaire de la bibliothèque (6)	24
Conclusion	29
Chapitre 2 : Présentation des Applications Mobiles	30
Introduction.....	31
2 Présentation des Applications Mobiles	31
2.1 Définition d'une application mobile.....	31
2.2 Domaine d'application mobile.....	31
2.3 Types d'application mobile	32
2.3.1 Les Applications Natives ou « Natives Apps »	32
2.3.2 Les Applications Web ou « WEB Apps »	32
2.3.3 Les Applications Hybride ou « Hybrid Apps »	33
2.4 Système d'exploitation mobile :.....	34
2.5 Types de systèmes d'exploitation mobiles (9):.....	34
2.6 Système d'exploitation Android	35
2.6.1 Présentation.....	35
2.6.2 Historique	35
2.6.3 Architecture	36

Table des Matières

2.7	Application Android.....	40
2.7.1	Présentation.....	40
2.7.2	Cycle de vie d'une application.....	40
2.7.2.1	Etat d'une activité (Application)	40
	Conclusion	43
	Chapitre 3 : Conception et Analyse et Spécification des besoins	44
	Partie 1 : Analyse et Spécification des Besoins	45
	Introduction.....	45
3	Analyse et Spécification des besoins.....	45
3.1	Problématique	45
3.2	Le processus de modélisation UML.....	46
3.2.1	Définition d'UML :	46
3.2.2	Les différents types de diagrammes :	46
3.3	Spécification des besoins	48
3.3.1	Besoins fonctionnels :.....	48
3.3.2	Besoins non fonctionnels :	49
3.4	Analyse des besoins.....	49
3.4.1	Identification des acteurs	49
3.4.2	Identifications des cas d'utilisations	51
3.4.3	Diagrammes des cas d'utilisations	52
	Partie 2 : Conception	55
3.5	Diagrammes de séquences des cas d'utilisation :	56
3.5.1	Diagramme de séquence de cas d'utilisation « s'authentifier » :.....	57
3.5.2	Diagramme de séquence de cas d'utilisation « s'inscrire » :.....	58
3.5.3	Diagramme de séquence de cas d'utilisation « Gérer les emprunts » :.....	59
3.5.4	Diagramme de séquence de cas d'utilisation « Gérer les ouvrages » :	60
3.5.5	Diagramme de séquence de cas d'utilisation « Gérer les pénalités » :.....	61
3.5.6	Diagramme de séquence de cas d'utilisation « Gérer les recherches » :.....	62
3.5.7	Diagramme de séquence de cas d'utilisation « Gérer les Adhérent » :.....	63
3.5.8	Diagramme de séquence de cas d'utilisation « Gérer les restitutions » :.....	64
3.6	Diagramme de classe de l'application :	65
3.7	Modélisation	65
3.7.1	Présentation de MCD	65
3.7.2	Diagramme entité association :.....	68

Table des Matières

3.7.3	Passage au modèle relationnel :	68
	Conclusion	69
	Chapitre 4 : La mise en œuvre	70
	Introduction.....	71
4	Mise en œuvre	71
4.1	Environnement de développement	71
4.1.1	Android Studio :.....	71
4.1.2	WampServer :	72
4.1.3	PhpMyAdmin :.....	73
4.1.4	Apache :.....	74
4.2	Langage de développement	75
4.2.1	JAVA.....	75
4.2.2	PHP.....	76
4.2.3	SQL	78
4.2.4	XML	78
4.3	Les services Web.....	79
4.3.1	Définition	79
4.4	Description des Application.....	81
4.4.1	Interfaces de l'application Android.....	81
4.4.1.1	Interface "authentification d'utilisateur"	82
4.4.1.2	Interface d'identification.....	83
4.4.1.3	Interface d'Inscription « Etudiant ».....	83
4.4.1.4	Interface recherche de document « PFE ».....	84
4.4.1.5	Interface recherche de document « PFC »	84
4.4.1.6	Interface Catalogue de bibliothèque	85
4.4.1.7	Interface de recherche de documents « Livre ».....	85
4.4.1.8	Interface réserver un livre	86
4.4.1.9	Interface emprunter un livre	86
4.4.1.10	Interface d'inscription « Enseignant ».....	87
A.1	Table utilisateur (Agent).....	88
A.2	Table de données pour Emprunt	88
	Conclusion	89
	CONCLUSION GENERALE.....	90
	Bibliographie.....	91
	ANNENES.....	94

Table des Matières

A.I	Base de données	94
A.1	Table utilisateur (Etudiant)	95
A.2	Table utilisateur (Enseignant)	95
<input type="checkbox"/>	Le code pour l'Interface de l'application Android	96
<input type="checkbox"/>	Le code pour l'Interface authentification d'utilisateur(login)	98
<input type="checkbox"/>	Programme de l'interface en java (authentification) :	100
<input type="checkbox"/>	Programmation de l'interface en PHP (authentification)	101
<input type="checkbox"/>	Programmation de l'interface en java (Etudiant)	102
<input type="checkbox"/>	Programmation en PHP (Etudiant).....	104
Résumé	105

Liste des Figures

Figure 1.1:Organigramme de la bibliothèque centrale.	22
Figure 1.2:Catalogue bibliothèque, recherche simple	25
Figure 1.3:Catalogue bibliothèque, recherche avancée.	25
Figure 1.4:Démarche d'une recherche simple.	26
Figure 1.5:Démarche de la recherche avancée.....	26
Figure 1.6:Une recherche simple sur le catalogue bibliothèque.	27
Figure 1.7:Résultats de la recherche.....	28
Figure 1.8:Une recherche avancée sur le catalogue bibliothèque.	29
Figure 1.9:Résultat de la recherche avancée.....	30
Figure 2.1:Architecture Android.	40
Figure 2.2:Cycle de vie d'une activité.....	45
Figure 3.1:La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme.	49
Figure 3.2 :Relation de généralisation entre acteurs.	54
Figure 3.3:Diagramme de cas d'utilisation (Adhérent).	57
Figure 3.4:Diagramme de cas d'utilisation (Administrateur).	58
Figure 3.5:Diagramme de cas d'utilisation (Bibliothécaire).....	59
Figure 3.6:Diagramme de séquence du cas d'utilisation « s'authentifier ».	62
Figure 3.7:Diagramme de séquence de cas d'utilisation « s'inscrire ».	63
Figure 3.8:Diagramme de séquence de cas d'utilisation « Gérer des emprunts ».	64
Figure 3.9:Diagramme de séquence de cas d'utilisation « Gérer des ouvrages ».	65
Figure 3.10:Diagramme de séquence de cas d'utilisation « Gérer les pénalités ».	66
Figure 3.11:Diagramme de séquence de cas d'utilisation « Gérer les recherches ».	67
Figure 3.12:Diagramme de séquence de cas d'utilisation « Gérer les Adhérent ».	68
Figure 3.13:Diagramme de séquence de cas d'utilisation « Gérer les restitutions ».	69
Figure 3.14:Diagramme de classe de l'application.	71
Figure 3.15:Modèle conceptuel de données.....	72
Figure 3.16:Diagramme entité association.	73
Figure 4.1 création d'une base de données sur PhpMyAdmin.....	74
Figure 4.2:Fonctionnement du PHP.	77
Figure 4.3:Interface d'accueil de l'application.	80
Figure 4.4: Interface authentification d'utilisateur <i>Xml</i>	81
Figure 4.5: Interface identification	82
Figure 4.6: Interface d'inscription pour étudiant.....	83
Figure 4.7: Interface recherche de document « PFE ».....	84
Figure 4.8: Interface recherche de document « PFC ».....	85
Figure 4.9: Interface Catalogue de bibliothèque.....	86
Figure 4.10:Interface de recherche de documents « Livre ».....	87
Figure 4.11:Interface réserver un livre.....	88
Figure 4.12:Interface emprunter un livre.....	89
Figure 4.13:Interface d'inscription « Enseignant ».....	90
Figure 4.14 table de données Agent.....	91
Figure 4.15 : table de données pour Emprunt.....	92
Figure 5.1 La Base de données.....	93
Figure 5.2 table de données Etudiants.....	94
Figure 5.3 table de données Enseignant.....	95

Liste des Figures

Figure 5.4 code de l'interface de l'application en Xml.....	96
Figure 5.5 Programmation en java de l'application.....	97
Figure 5.6 code de l'interface authentification en Xml.....	98
Figure 5.7 Programme de l'interface authentification en java.....	99
Figure 5.8 Programme de l'interface authentification en PHP.....	100
Figure 5.9 Programmation en java de l'interface inscription étudiant.....	101
Figure 5.10 Programme en PHP d'interface inscription étudiant.....	102

Liste des Tableaux

Tableau 1.1:Base de l'offre de service en bibliothèque académique selon C. Ollendorff.	19
Tableau 2.1: Avantage et inconvénients des types d'applications.	33
Tableau 3.1:identifications des acteurs.....	50
Tableau 3.2:Cas d'utilisation associé au système.	51

Liste des Abréviations

AAPT	Android Asset Packaging Tool
ADB	Android Debug Bridge
ANSI	American National Standards Institute
API	Application programming interface
BU	Bibliothèque universitaire
CERIST	Centre de recherche sur l'information scientifique et technique
CSS3	Cascading Style Sheets
DDMS	Dalvik Debug Monitor Service
DVM	Machine virtuelle Dalvik
HTML5	HyperText Markup Language 5
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines
IDE	Integrated development environment
ISBN	International Standard Book Number
ISNN	International Standard Serial Number
JVM	Java Virtual machine
LAMP	Linux, Apache, MySQL, PHP
MCD	Modèle conceptuel des données
MLD	Modèle Conceptuel de Données
MPD	Modèle Physique des Données
OD	Diffraction Optics
OS	Operating System
PHP	Hypertext Preprocessor
RCP	Rich Client Platform
SDK	Software développement kit de la plateforme mobile
SGBD	Système de gestion de base de données
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SWT	Standard Widget Toolkit
TCP / IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
WAMP	Windows, Apache, MySQL, PHP
WAP	Wireless Application Protocol
XML	Extensible Markup Language

INTRODUCTION GENERALE

L'informatique a donné une nouvelle vie à certaines activités humaines. Avec le développement de l'informatique de nouveaux logiciels sont apparus, ce qui a facilité la recherche et l'accès à l'information et minimisent le temps de son traitement. Ainsi que la réalisation moderne des projets.

Les applications mobiles sont apparues dans les années 1990, elles sont liées aux développements d'Internet et des télécommunications, des réseaux sans fils et des technologies agents.

Le domaine des applications mobiles reconnaît de plus en plus un essor phénoménal et devient une source financière importante tant pour le marché public que privés. Dans cette démarche notre établissement a voulu interagir pour exploiter les avancées technologiques qui ne cessent de progresser afin d'offrir des services innovants et rapides à leurs adhérents.

L'objectif de notre étude, est la conception d'un système d'information permettant d'aider à la gestion de la Bibliothèque et plus précisément, aider à la gestion des emprunts, réservations et retours des livres... :

L'application devra permettre donc de :

- ✓ Gérer les personnes inscrites
- ✓ Gérer l'emprunt, le retour et le rappel sur les dates de retour des livres.
- ✓ Ainsi que les réservations par des personnes inscrites.

Introduction Générale

Dans ce mémoire nous présentons l'ensemble des étapes pour le développement de notre solution organisée sous forme de chapitre.

Le premier chapitre est consacré à la présentation de l'université de Bejaia en général, puis la présentation de la bibliothèque universitaire, ses différents services, rôles et enjeux ainsi que les types de bibliothèque. On achève notre présentation par l'exposition du site de recherche documentaire de la bibliothèque de Bejaia.

Dans le second chapitre nous décrivons Les applications mobiles, leurs domaines et différents types

Le troisième chapitre représente l'analyse et spécification des besoins.

Le quatrième chapitre « Conception et réalisation » décrit les différents aspects conceptuels de données de l'application, pour entamer enfin la partie réalisation qui présente ainsi les différents outils et langage de programmations utilisés au cours de cette étude.

En concluant avec une conclusion générale et des atouts ainsi que des perspectives pour des développements au futur.

***Chapitre 1 : Présentation de la
Bibliothèque de BEJAIA***

Introduction

Dans ce chapitre nous allons présenter l'université de Bejaia, ensuite nous présentons la bibliothèque de Bejaia, ses missions principales et ses différents services. Puis nous décrivons le site de recherche documentaire de la bibliothèque (le site Opac Bejaia).

1 Présentation de l'université de Bejaia

1.1 Historique

L'Université de Bejaïa, créée en octobre 1983, est un établissement public pluridisciplinaire. Elle compte aujourd'hui plus de 45 700 étudiants, 1714 enseignants et 1227 personnels techniques et administratifs, répartis sur huit facultés : Technologie - Sciences Exactes - Droit et Sciences Juridiques et Administratives - Sciences de Nature et de la Vie - Lettres et Langues - Sciences Humaines et Sociales - Sciences Economiques, Sciences de Gestion et Sciences Commerciales - Sciences Médicales.

L'université de Bejaia a réussi à mettre sur pied des formations de plus en plus en phase avec le monde du travail. Cette démarche lui a permis d'être mieux à l'écoute des besoins de ses partenaires économiques en matière de ressources humaines et de compétences.

Elle dispose actuellement d'une trentaine de laboratoires et unité de recherche. Ces derniers sont agréés par le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique portant sur plusieurs domaines : Modélisation et Optimisation des Systèmes - Technologie des Matériaux et du Génie des Procédés-Matériaux organiques - Génie et de l'environnement – Hydraulique -Technologie Industrielle et de l'Information-Génie Electrique - Biomathématique, Biophysique Biochimie - Mathématiques Appliquées -Physique Théorique - Ecologie et Environnement - Economie et Développement - Microbiologie Appliquée - Biochimie Appliquée - Formation en langues Appliquées et Ingénierie des Langues en milieu Multilingue - Ecosystèmes Marin et l'Aquacole) (1).

1.2 Présentation de la bibliothèque universitaire

Les bibliothèques universitaires se développèrent à partir du XIII^e siècle dans les célèbres universités européennes « *en particulier en Angleterre (Cambridge et Oxford), en Allemagne et en France (La Sorbonne)* » Pour définir ce que c'est qu'une bibliothèque universitaire, nous avons retenu deux définitions qui expliquent clairement cette notion [2] :

1. Le dictionnaire encyclopédique des sciences de l'information et de la communication :

Il définit les BU à travers leur vocation et les missions qui leurs sont assignées :

- Les bibliothèques universitaires sont chargées de fournir aux étudiants, aux enseignants et chercheurs la documentation nécessaire à l'étude et à la recherche dans toutes les spécialités.
- Disciplines enseignées dans les universités et d'assurer une mission d'orientation et d'enseignement bibliographique et documentaire
- Les prestations d'une BU sont donc basées, à la lumière de cette définition, sur un nombre d'éléments déterminants à savoir :
 - La nature du public potentiellement utilisateur (les étudiants et les enseignants),
 - L'activité intellectuelle et professionnelle de ce public (préparation des examens, des exposés, des travaux de recherche tels que les mémoires, thèses et communications scientifiques, ainsi que la mise au point des contenus des cours dispensés au sein de l'université),
 - Les disciplines académiques assurées par l'université de rattachement (les spécialités admises par cette université déterminent la composition du fonds documentaire),
 - Un rôle pédagogique (qui se traduit par la mise en place de dispositifs de formation des usagers à l'exploitation optimale des ressources mises à leur disposition via des répertoires, catalogues et divers autres outils bibliographiques) [2].

2. Christine Ollendorff

Dans le cadre d'une thèse qui s'est donnée pour objectif : « *proposer aux managers de bibliothèques académiques un outil d'aide à la décision en milieu complexe* »

C. Ollendorff ¹s'est attelée à mettre sur pied un modèle d'offre de service adapté à cette catégorie de bibliothèques. Cette offre est constituée, selon elle, de trois services de base :

- La mise à disposition de documents,
- La recherche documentaire,
- La formation.

Ces trois types de services et leurs subdivisions, peuvent être enrichis de compléments selon les moyens mis en œuvre pour leur réalisation. Cette variété des prestations de service constitue "la largeur de l'offre de service".

Cette largeur a comme ambition : répondre à la globalité des besoins des usagers. Elle doit être, appuyée, pour qu'elle soit plus efficace, par une politique de ciblage qui donnerait naissance à des services spécialisés, personnalisés par segments d'utilisateurs. Pour chaque service, on détermine également un niveau de finesse, une profondeur en termes de rapidité, de coûts, de délais, de moyens mis en œuvre.

Les trois services cités plus haut, dans leur largeur comme dans leur finesse ont été mis en évidence dans un tableau que nous reproduisons ci-dessous (2) (Tableau 1.1) :

Services	Mise à disposition	Recherche documentaire	Formation
Service de base	Consultation sur place Catalogue accès local Prêt	Outils en local Banques de données commerciales Orientation vers d'autres sources	Formation aux méthodes de recherche d'information

¹ Directrice de la Documentation et de la Prospective chez Arts et Métiers ParisTech.

Service complémentaire	Information, nouveautés Catalogues accès à distance Photocopieur Photocopie sommaire Espaces de consultation	Fourniture des documents (prêt entre bibliothèques)	Mise en place d'un module d'autoformation
------------------------	--	---	---

Tableau 1.1 : Base de l'offre de service en bibliothèque académique selon C. Ollendorff.

1.2.1 Services de la bibliothèque

L'organisation et la gestion administrative de la bibliothèque est conforme au décret exécutif n°03-279 fixant les missions et les règles particulières d'organisation et de fonctionnement de l'université et l'arrêté interministériel correspondant au 24 août 2004 fixant l'organisation administrative du Rectorat, de la Faculté, de l'Institut, de l'annexe de l'Université et ses services communs, dont fait partie la bibliothèque et comprend (3):

1.2.1.1 Service gestion des fonds (acquisition et traitement)

Le service assure les opérations d'acquisition et de traitement suivant les taches de la chaîne documentaire qui se finalise par la communication de l'information aux usagers.

Pour cela la bibliothèque utilise des normes de gestions universelles normalisées et de l'outil informatique pour faciliter et simplifier l'accès à l'information, la recherche et la consultation des documents comme l'utilisation du logiciel Syngeb ²élaboré par le Centre de recherche sur l'information scientifique et technique CERIST³.

Le service communique et affiche la liste de ses nouvelles acquisitions chaque année.

² Un logiciel complet et évolutif permettant de gérer tous types de documents (Livres, périodiques, thèses, articles et les non livres). Le produit permet de gérer rapidement et efficacement vos documents.

³ un établissement public algérien à caractère scientifique et technologique sous la tutelle du ministère de l'enseignement supérieur et de la recherche scientifique.

1.2.1.2 Service de l'orientation et de la recherche bibliographique

Le service mis à jour l'inventaire du fonds documentaire de la bibliothèque chaque fin d'année. Ce service veille à orienter et informer les usagers dans leurs quêtes du savoir et de recherche documentaire en mettant à leur disposition des outils de recherche adéquats et les conditions appropriées d'utilisation du fonds documentaire (catalogue en ligne) de la bibliothèque et de ses espaces. Ce service organise chaque nouvelle année universitaire des portes ouvertes pour initier les nouveaux étudiants à la recherche bibliographique.

1.2.2 Le rôle et enjeux de la bibliothèque

Dans un contexte marqué par l'explosion du numérique, la bibliothèque est le vecteur de l'accès à l'information scientifique, critère premier de la compétitivité internationale des universités. Face à la concurrence internationale accrue, elle est un élément constitutif de l'identité et de l'image de l'établissement, notamment auprès des étudiants et enseignants-chercheurs étrangers. Les rôles principaux des bibliothèques sont :

- D'accompagner et de soutenir les activités d'enseignement et de recherche.
- Plus importantes que jamais dans ce monde de l'information numérique, elles identifient, acquièrent et rendent accessibles les ressources documentaires indispensables aux étudiants et aux chercheurs.
- Elles ont également pour mission de former à l'utilisation des ressources documentaires.
- Gardiennes de la mémoire et du patrimoine documentaire de l'établissement, elles occupent une place centrale dans la vie sociale et culturelle d'une université
- La B.U. est non seulement un lieu de travail, d'étude et de recherche, mais aussi de convivialité (4).

Le développement et la modernisation des bibliothèques d'enseignement supérieur et de recherche représente donc un enjeu stratégique. Dans cette perspective, le ministère soutient particulièrement les axes suivants (4):

- Faciliter l'accès à l'information, par l'extension des horaires d'ouverture, le développement du libre accès et de l'accès à distance aux collections ;
- Développer les ressources numériques (manuels pédagogiques ; revues scientifiques ; bases de données ; documents numérisés) ;
- Diffuser la production pédagogique et scientifique de l'université (thèses, publications scientifiques, ressources pédagogiques, etc.) ;
- Former les étudiants et enseignants-chercheurs à la recherche et à l'exploitation de l'information.

1.2.3 Organigramme de la bibliothèque

La bibliothèque centrale de l'université placée sous la responsabilité d'un directeur chargé du fonctionnement et de la gestion des structures (Figure1.1).

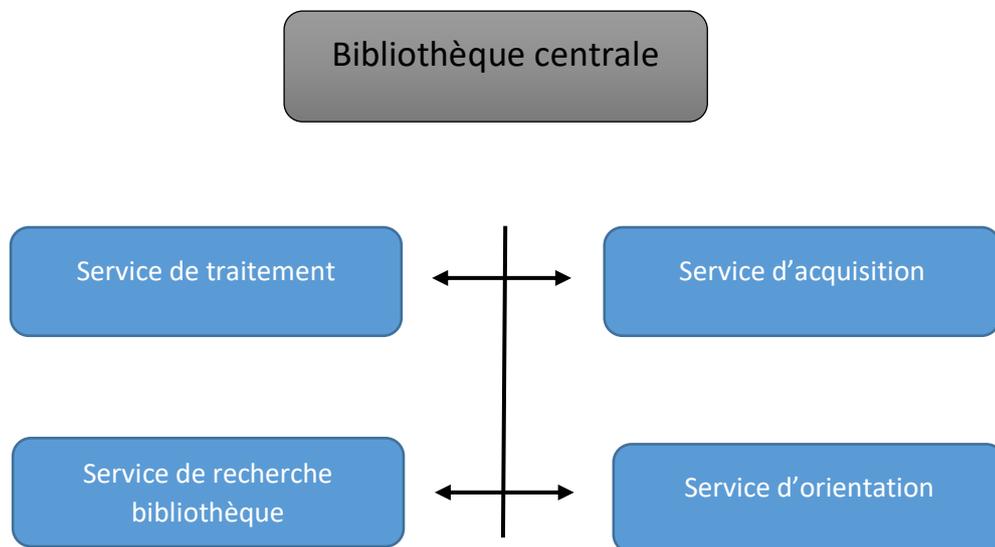


Figure 1.1 : Organigramme de la bibliothèque centrale.

1.2.4 Les types de bibliothèques

Il existe une variante à la bibliothèque universitaire, la bibliothèque de recherche qui peut être réservée exclusivement à la communauté des chercheurs, que constituent les étudiants du troisième cycle universitaire, et les enseignants-chercheurs, ou rattachée organiquement à la bibliothèque universitaire (5)

- **La bibliothèque de recherche**

Au niveau de la recherche, les ouvrages deviennent plus spécialisés, plus coûteux et les chercheurs scientifiques ont besoin de plus en plus de périodiques. Les revues académiques sont, en effet, des publications de prédilection des chercheurs, leur principale matière de travail. L'expression "bibliothèque de recherche" renferme trois acceptions selon Maurice Garden⁴: celle du chercheur, celle ayant une orientation patrimoniale, et enfin la bibliothèque faisant office de réservoir d'ouvrages de références.

- **La bibliothèque du chercheur**

Le chercheur, pour réaliser son "rêve" qui consistait à rassembler toute la documentation d'une manière exhaustive entrant dans ses domaines d'intérêt, et d'assurer ainsi un accès immédiat aux documents, essayait au temps où l'imprimerie faisait ses premiers pas, de constituer "son cabinet de lecture pour reprendre l'expression de M. Garden. Cependant, l'essor prodigieux des nouvelles technologies de l'information et de la communication et les possibilités d'échange et d'accès "instantané" à la communication scientifique peut remettre au jour cette tendance, de bibliothèques individuelles.

⁴ Professeur d'histoire (1970) et vice-président (1972) de l'Université Lumière-Lyon 2, puis professeur à l'ENS Cachan. Recteur vice-chancelier de l'Académie de Paris (1993), il a également été responsable de la recherche universitaire en sciences humaines et sociales au ministère de la Recherche et de l'Enseignement supérieur. Auteur de "Lyon et les Lyonnais au XVIIIe siècle" (Flammarion 1975).

- La bibliothèque patrimoniale

Selon M. Garden, ce type de bibliothèques savantes est limité à quelques disciplines de sciences humaines et sociales. Elle se caractérise, par son caractère « unique » de la recherche. C'est une instrumentation, un laboratoire, un peu à l'image de ce que sont les dépôts d'archives pour les historiens. Or, Elles sont souvent comme des trésors patrimoniaux, pour lesquels les soins de protection et de conservation l'emportent sur le désir de communication.

- La bibliothèque de référence : elle existe à travers deux modèles différents :

- Une bibliothèque de référence généraliste

L'exemple donné est celle créée au *Wissenschaftskolleg*⁵ de Berlin, en Allemagne, constituée de quelques grandes séries de répertoires bibliographiques, des séries de catalogues sur CD-Rom ou sur microfiches, quelques écrans, à peine quelques outils de référence. Le personnel spécialisé du service cherche à satisfaire les demandes de chaque chercheur à partir des ressources de toute la ville, de la bibliothèque de l'Etat jusqu'aux bibliothèques des multiples instituts.

L'inconvénient est que ce système ne peut fonctionner que pour un nombre limité d'utilisateurs, et dans un contexte économique favorable. La balkanisation⁶ très poussée des spécialités universitaires fait que leur couverture en matière d'instruments bibliographiques devient une tâche hasardeuse.

- Une bibliothèque de référence spécialisée :

Cette catégorie de bibliothèque doit être un instrument de pointe, ouvert sur l'ensemble de la production internationale sur le thème qu'est le sien. Cependant, la miniaturisation des champs de recherche associée au caractère "généraliste" des plus grandes revues souligne M. Garden remettent en question cette exhaustivité souhaitée. La dispersion des travaux scientifiques sur

⁵ un institut interdisciplinaire fondé en 1981 à Grunewald, Berlin, Allemagne, dédié aux projets de recherche dans les sciences naturelles et sociales. Il est calqué sur l'IAS original à Princeton, New Jersey et est membre de Some Institutes for Advanced Study.

⁶ Notion politique dérivée du toponyme Balkans, elle est utilisée pour la première fois par un Allemand, Walther Rathenau, en septembre 1918 utilisé pour décrire le processus de fragmentation et de division d'une région ou d'un État en des États et régions plus petits et souvent hostiles les uns envers les autres.

plusieurs publications, et la variété des thématiques qui y figurent imposent la sélectivité dans l'acquisition. La gestion a priori complexe d'un tel fonds qui ne peut être que gigantesque et le public très restreint attendu rendent l'obtention d'une surface financière suffisante aléatoire.

1.2.5 Présentation du site de recherche documentaire de la bibliothèque (6)

Pour leurs recherches, les usagers peuvent utiliser l'outil informatique mis à leur disposition pour accéder aux bases de données de la bibliothèque via :

- Intranet à l'adresse : Bejaia/**Opac** comme illustrer dans les figures 1.2, 1.3.
- Internet à l'adresse : <http://recherche.univ-bejaia.dz/opac>.

Catalogue bibliothèque :

Catalogue

Catalogue Bibliothèque

Recherche simple

Entrez le(s) critère(s) de recherche

Mot(s) du titre

Mot(s) auteur

Mot(s) sujet

AR Recherche avancée

Tous droits réservés © 2007, CERIST

Figure 1.2:Catalogue bibliothèque, recherche simple

Catalogue

Catalogue Bibliothèque

Recherche avancée

Entrez le(s) critère(s) de recherche

Mot(s) du titre

Mot(s) auteur

Mot(s) sujet

Mot(s) editeur

ISBN, ISSN...

Année

Type de document

Langue

AR Recherche simple

Tous droits réservés © 2007, CERIST

Figure 1.3:Catalogue bibliothèque, recherche avancée.

1. Recherche simple :

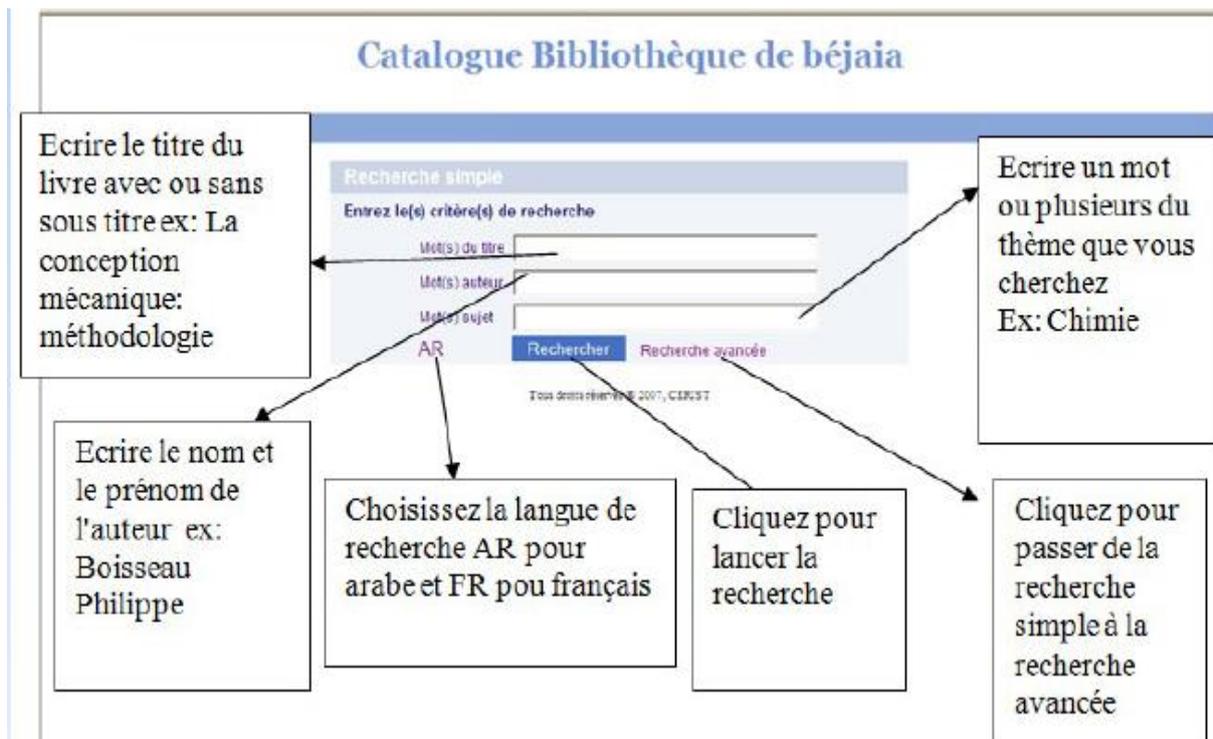


Figure 1.4:Démarche d'une recherche simple.

2. Recherche avancée

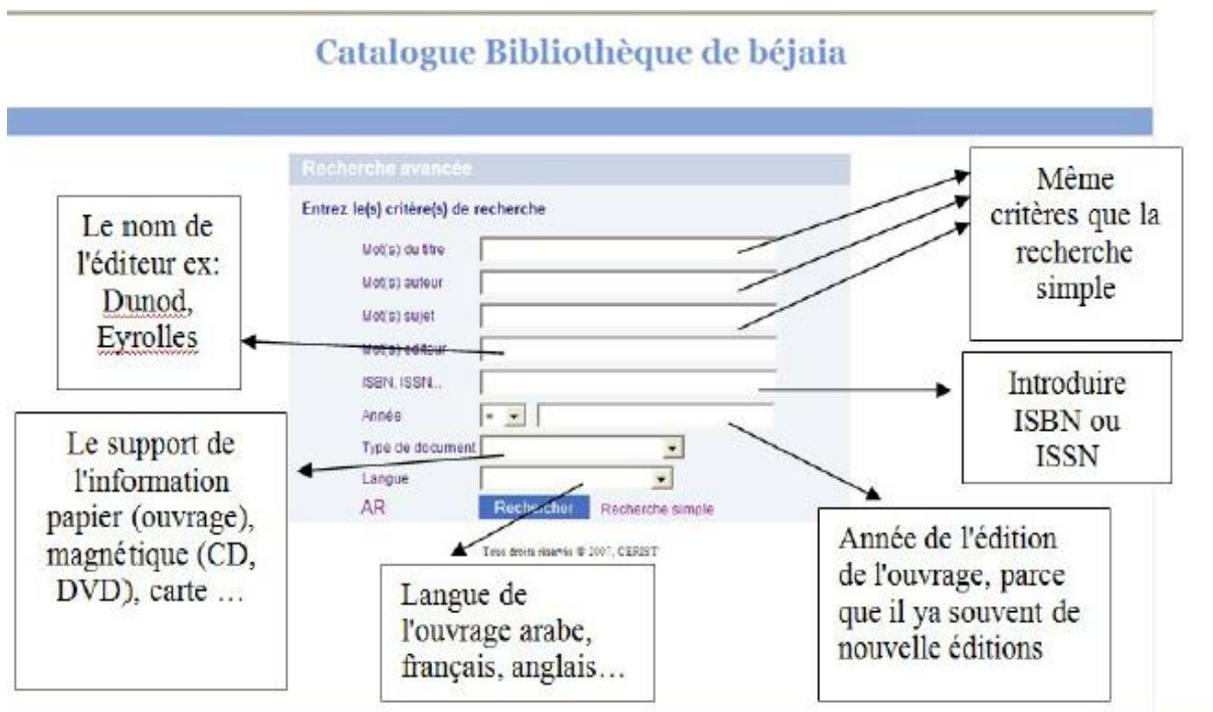


Figure 1.5:Démarche de la recherche avancée.

3. Explication pour une recherche simple :

Catalogue Bibliothèque

Recherche simple

Entrez le(s) critère(s) de recherche

Mot(s) du titre

Mot(s) auteur

Mot(s) sujet

AR Recherche avancée

N°	Titre	Auteur	Type	Année
1.	Conception et Développement d'une application mobile de Vente Flash sous ANDROID	Hacene , Messaoudi	Thèse	2015
2.	Conception et implémentation d'une application mobile de gestion d'un cabinet médical sous Android. : Cas d'étude : "Un cabinet de pneumo-phtisiologie"	Nassim , Yaici	Thèse	2014
3.	Conception et implémentation d'une application mobile pour les services d'aide aux étudiants sous Android	Salim, Sadelli	Thèse	2013
4.	Conception et mise en oeuvre d'une application mobile sous Android dédié à la gestion du département d'Informatique de l'Université de Béjaïa	Rabah, Brahmi	Thèse	2015
5.	Conception et réalisation d'une application de messagerie instantanée sous Android	Lahcene, Kedjar	Thèse	2015
6.	Conception et réalisation d'une application mobile sous Android « Cas d'étude : gestion de stocks dans un domicile »	Yacine, Bouadjnek	Thèse	2014
7.	Conception et réalisation d'une application mobile sous Android « Cas d'étude : Le suivi des diabétiques »	Djamel, Benmahiddine	Thèse	2016
8.	Conception et réalisation d'une application mobile sous Android pour l'affichage des emplois du temps et planning des examens	Anis , Barka	Thèse	2014
9.	Conception et Réalisation d'une application mobile sous Android pour le domaine médical.	Assia , Ait Braham	Thèse	2014
10.	Conception et réalisation d'une application mobile sous Android. Assistance d'un corps de sécurité	Safi eddine, Belhaouchet	Thèse	2015
11.	Conception et réalisation d'un application android dédiée aux étudiants de l'université de béjaïa	Boubekeur, Cherid	Thèse	2015.
12.	Conception et réalisation d'une application android : localisation des boutiques, hotes,	Sara, Messouaf	Thèse	2015

Figure 1.6: Une recherche simple sur le catalogue bibliothèque.

La figure ci-dessus présente la démarche à suivre pour faire une recherche simple, il est recommandé d'introduire sur le premier champ le titre du livre recherché avec ou sans sous-titre par exemple « Application Android », sur le deuxième champ on écrit le nom et prénom de l'auteur du livre recherché « Hacene MESAOUDI », finalement sur le dernier champ de saisie on écrit un mot clé ou plusieurs du thème recherché. Après le choix de la langue effectuer soit en Ar (Arab) ou Fr (français) il reste que à lancer la recherche en cliquant sur l'icône « Recherche » et pour l'impression sur l'icône « Imprimer ». Tous les titres similaires seront affichés comme le montre la figure ci-dessus (Figure 1.6, 1.7).

recherche.univ-bejaia.dz/opac/notice.php?id=96067 - Google Chrome

Non sécurisé | recherche.univ-bejaia.dz/opac/notice.php?id=96067

Public **ISBD** **UNIMARC**

Type de document	Thèse
Langue	Français
Titre	Conception et Développement d'une application mobile de Vente Flash sous ANDROID [texte imprimé]
Auteur(s)	Hacene , Messaoudi Abdelouahid, Ouakkouche
Adresse bib.	Béjaia : Université A/ Mira Bejaia, 2015
Collation	72f. ; 30 + CD
Notes	Bibliogr.
Notes de thèse	Master : Informatique : Béjaia : 2015
Sujet(s)	Application mobile : ANDROID : Vente Flash
Résumé	La Vente Flash de m-Commerce n'a pas de vocation pour remplacer les canaux traditionnels, y compris le e-Commerce sur ordinateur, il convient particulièrement à certains types d'achat basés sur la simplicité des produits et l'urgence de la demande. D'après une étude Gartner, les consommateurs ont dépassé en 2011 6,2 milliards de téléchargement d'applications mobiles ANDROID, soit une progression de près de 60 pourcent par rapport à 2010. La progression ne devrait pas s'arrêter en si bon chemin puisqu'en 2013 le nombre a dépassé le chi-re de 21,6 milliards de téléchargements. Le projet que nous présentons aborde précisément une applica-tion conçu pour fonctionner sur la plate-forme ANDROID. Le projet permet la Vente Flash et la fédélisation des clients en restant au courant en permanence des nouveautés commerciaux des produits. L'application permet l'achat des produits selon la catégorie et les produits en promotions. L'application proposera l'achat sur site du NecheriFenet. - Augmenter le trafic du site NecheriFenet.com. - Fidéliser les abonnés du NecheriFenet. L'application nécessite un espace administrateur qui permet de faire des mises à jour du ca-talogue de produit et des promotions. L'espace administrateur nécessite également la mise en place des interfaces de gestion des commandes. Par ailleurs, nous proposons dans notre appli-cation une nouvelle forme d'achat et sélection des produits en vue d'améliorer les méthodes de navigation dans l'application. En e-et, il est nécessaire de disposer d'outils pour gérer les achats, les commandes, les stocks, la logistique et tous les autres aspects liés à la gestion com-merciale, gestion du catalogue, gestion des achats et gestion du contenu.
Indices	004

Exemplaire(s)

Cote	Localisation	Support	Statut
004MAS/409	NB	livre	Disponible
004MAS/409.2	NB	livre	Disponible

Figure 1.7:Résultats de la recherche.

4. Explication pour une recherche avancé :

Catalogue Bibliothèque



The screenshot shows a web form titled "Recherche avancée" (Advanced Search). Below the title is the instruction "Entrez le(s) critère(s) de recherche" (Enter the search criterion(s)). The form contains several input fields and dropdown menus:

- Mot(s) du titre: application android
- Mot(s) auteur: hitachi, ghilas
- Mot(s) sujet: application mobile
- Mot(s) editeur: chercheurs
- ISBN, ISSN...: 71f
- Année: = 2017
- Type de document: Thèse
- Langue: Français

At the bottom left, there is a label "AR" and two buttons: "Rechercher" (highlighted in blue) and "Recherche simple" (Simple Search).

Figure 1.8: Une recherche avancée sur le catalogue bibliothèque.

La figure ci-dessus présente la démarche à suivre pour effectuer une recherche avancée, dans le cas où le résultat n'est pas retrouvé après une recherche simple il est recommandé de faire une recherche avancée pour donner plus de précision. Parmi les champs ajoutés a cette recherche on trouve le champ « mot(s) éditeur » pour écrire le nom de l'éditeur ou un mot relatif, le champ « ISBN ou ISSN » représente le numéro international qui permet d'identifier de manière unique chaque livre publié, ensuite le champs « Année » pour préciser l'année de l'édition parce que y'a toujours de nouvelles éditions, le champs « Type de document » le support de l'information qui peut être un livre (ouvrage), CD, DVD..., le champs « Langue » pour définir la langue de l'ouvrage (français, anglais, arabe...), quand tous les champs sont remplis il reste que à effectuer la recherche (Figure1.8).

recherche.univ-bejaia.dz/opac/notice.php?id=105421 - Google Chrome

Non sécurisé | recherche.univ-bejaia.dz/opac/notice.php?id=105421

Public **ISBD** **UNIMARC**

Type de document	Thèse
Langue	Français
Titre	Conception et réalisation d'une application sous Android pour la consultation des états des navires au port de Bejaia [texte imprimé]
Auteur(s)	Hitachi , Ghiles Kaiba , Yanis Brahmi Houda , Née Elbouhissi
Adresse bib.	Béjaia : Université A/ Mira Bejaia, 2017
Collation	71f. : ill. : 30 + CD
Notes	Bibliogr., Annexe
Notes de thèse	Master : Informatique : Béjaia : 2017
Sujet(s)	Application Mobile : Android, UP : UML : Service Web : Java :XML : MVC : Http : JSON : PHP : SQL.
Résumé	L'évolution de la technologie du mobile a remarquablement étendu les moyens d'accès à l'information : elle est devenue exploitable en tout lieu couvert par un réseau. En tirant profit de cette évolution, les entreprises ont souvent tendance à étendre leur gestion sur des terminaux mobiles. Dans ce but, l'entreprise portuaire de Bejaia voulait se doter de cette pratique qui se révèle très efficace quant à l'accès aux informations indépendamment du temps et de lieu. Notre travail consiste donc à concevoir et implémenter une application Android dédiée à la consultation des états des navires au port de Bejaia, cette application devra reproduire quelques fonctionnalités disponibles sur le site web de la firme.

cation une nouvelle forme d'achat et sélection des produits en vue d'améliorer les méthodes de navigation dans l'application. En e-et, il est nécessaire de disposer d'outils pour gérer les achats, les commandes, les stocks, la logistique et tous les autres aspects liés à la gestion com-merciale, gestion du catalogue, gestion des achats et gestion du contenu.

Exemplaire(s)

Cote	Localisation	Support	Statut
004MAS/409	NB	livre	Disponible
004MAS/409.2	NB	livre	Disponible

Figure 1.9:Résultat de la recherche avancée.

Les figure 8 et 9 représente les résultats de la recherche simple ainsi que la recherche avancée, indiquant ainsi toutes les informations relatives au livres (type de document, langue, nom de l'auteur, l'adresse, le sujet, le résumé ainsi que la disponibilité du livre dans la bibliothèque (Figure 1.9).

Conclusion

Dans ce chapitre nous avons présenté l'université de Bejaia, par la suite nous avons présenté la bibliothèque centrale de Bejaia, ainsi que les différentes missions et services de la bibliothèque et son organigramme, et on a fini par une explication de l'utilisation du site Opac de recherche documentaire de la bibliothèque. Dans le chapitre suivant nous allons présenter les applications et les systèmes d'exploitation mobiles.

Chapitre 2 : Présentation des Applications Mobiles

Introduction

Un système d'exploitation mobile est 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. Dans ce chapitre nous allons définir c'est quoi une application mobile, et quelques notions de base, puis nous allons présenter c'est quoi un système d'exploitation mobile tout en définissant c'est quoi un système Android, son architecture et le cycle de vie d'une application Android.

2 Présentation des Applications Mobiles

2.1 Définition d'une application mobile

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation d'un smartphone ou d'une tablette. Selon la Médiamétrie⁷, début 2016, les utilisateurs de smartphones avaient en moyenne 28 applications mobiles installées sur leur téléphone référence.

Les applications mobiles sont adaptées aux différents environnements techniques des Smartphones et à leurs contraintes et possibilités ergonomiques (écran tactile notamment). Elles permettent généralement un accès plus confortable et plus efficace à des sites ou services accessibles par ailleurs en versions mobile ou web. L'essentiel du temps passé sur les Smartphones est consacré à des applications mobiles (7).

2.2 Domaine d'application mobile

Le domaine d'utilisation des applications mobiles est très vaste, on peut citer par exemple la géolocalisation, itinéraire, paiement mobile, traitement du texte, les jeux, les réseaux sociaux et éducation, etc.

⁷ une société anonyme spécialisée dans la mesure d'audience et l'étude des usages des médias audiovisuels et numériques en France. Elle est notamment connue pour sa marque Audimat, dont le nom est devenu, par antonomase, un mot du langage courant ; la marque se nomme aujourd'hui Médiamat.

2.3 Types d'application mobile

Actuellement il existe trois types d'applications mobiles :

2.3.1 Les Applications Natives ou « Natives Apps »

Celles-ci correspondent à des logiciels créés uniquement pour une plateforme mobile spécifique. Le développement de ces logiciels se fait au travers du SDK ou software development kit de la plateforme mobile en question. Le nom de ces applications vient du fait qu'elles sont développées exclusivement avec le langage « natifs », par exemple le langage JAVA ou le langage Objective-C. Les natives App sont téléchargées à partir d'une plateforme de téléchargement qui est souvent un Store applicatif. C'est par exemple le cas pour l'Apple store ou encore Google Play (8).

2.3.2 Les Applications Web ou « WEB Apps »

Correspondent à des sites Web qui sont conçus spécialement pour un affichage optimisé pour mobile. Pour accéder à ces sites Web, on utilise le navigateur Internet qui est sur le mobile. Ces applications mobiles sont développées principalement à partir de technologies Web comme le HTML5 ou encore CSS3. Grâce au support HTML5, il est dès lors, possible d'accéder à environ 80 % des fonctions présentes sur le mobile. Par exemple, cela permet d'accéder à différentes fonctions, comme la géolocalisation, à l'accéléromètre, gérer la fonction multitouche ou encore permettre la synchronisation offline lorsque le mobile perd et retrouve ensuite sa connexion. Ces applications Web peuvent être de deux sortes : les génériques, qui sont compatibles avec toutes les plateformes mobiles et donc utilisables sur n'importe laquelle, ou alors celles conçues spécifiquement pour un genre de support en particulier (8).

2.3.3 Les Applications Hybride ou « Hybrid Apps »

Ces dernières sont considérées comme un mix, entre les Web Apps et les Natives Apps. En effet, elles sont compatibles avec toutes les plateformes mobiles. Mais ces applications sont principalement développées à l'aide d'HTML5, aujourd'hui, qui est très performant mais qui utilisent aussi d'autres langages Web comme le CSS et le JavaScript.

Ainsi, une application dite hybride, contrairement à une application native, n'est pas dépendante d'une plateforme mobile en particulier. De la même manière, et contrairement aux applications Web, les applications hybrides peuvent accéder à toutes les fonctions présentes sur le mobile. Cela est rendu possible par des liens qui sont faits entre le langage natif et la technologie Web présente dans l'application hybride.

En combinant les points forts des applications natives et Web, celles-ci suscitent, de plus en plus, l'intérêt des développeurs d'applications mobiles (8) (Tableau 2.2).

	Avantages	Inconvénients
Application Native	<ul style="list-style-type: none"> - Utilise l'accès à des fonctionnalités du mobile sans besoin d'internet. - Adaptative à des « business model ». - Installation gratuite. 	<ul style="list-style-type: none"> - Prennent du temps à être développées. - Relativement couteuses. - Problèmes de rétrocompatibilité. - Possibilité de refus d'être mis sur certains stores d'appli
Application Web	<ul style="list-style-type: none"> - Comptabilité avec tous les navigateurs. - Simple à développer. 	<ul style="list-style-type: none"> - Pas accès à toutes les fonctions présentes sur le mobile
Application hybride	<ul style="list-style-type: none"> - Il est plus facile à développer et le temps de développement est rapide. - Son coût est abordable 	<ul style="list-style-type: none"> - Pas adaptative à tous les systèmes d'exploitation. - C'est aussi le cas pour la performance. - Possibilité de refus d'être mis sur certains stores d'application.

Tableau 2.2: Avantage et inconvénients des types d'applications.

2.4 Système d'exploitation mobile :

Le système d'exploitation est le garant des fonctions et des caractéristiques disponibles sur votre appareil mobile, telles que la molette, le clavier, le WAP, la synchronisation avec les applications mobiles, le courriel, les SMS... Le système d'exploitation du mobile choisira également quelles applications tierces peuvent fonctionner sur votre appareil.

Choisir son appareil en fonction de son système d'exploitation peut être un critère très important (9).

2.5 Types de systèmes d'exploitation mobiles (9):

- Système d'exploitation Android par Google
- Système d'exploitation Bada par Samsung Electronics
- Système d'exploitation de BlackBerry
- Le iOS par Apple
- Le système d'exploitation MeeGo OS par Nokia et Intel
- Le Palm OS (Garnet OS)
- Le Symbian OS par Nokia
- Le webOS par Palm et HP
- Le Windows Mobile pour appareils Windows

Dans notre travail on a adopté le système d'exploitation Android pour les raisons suivantes :

- Un grand nombre de constructeurs proposent Android sur leurs appareils (Samsung, HTC, LG...).
- La possibilité de customiser son androphone à volonté (live wallpaper, launcher, widget...), vous pouvez personnaliser votre appareil Android.
- La communauté Android est très active qui crée des ROMs non officiels et des kernels.
- La compatibilité matérielle facilité par Google.

- Les services Google parfaitement intégrés au système et aux terminaux Android (YouTube, Google new, Gmail, Google Search, etc....).
- Android évolue plus rapidement que l'iOS et dépasse aujourd'hui l'iOS en termes de fonctionnalités.
- Possibilité de mettre plusieurs comptes d'utilisateurs sur les terminaux Android.
- Mise à jour gérée directement par wifi ou Connexion mobile.

2.6 Système d'exploitation Android

2.6.1 Présentation

Android est un système d'exploitation de Google destiné à une multitude de smartphones, tablettes et autres objets connectés. Aujourd'hui dans sa version 7 dite "Nougat"⁹, Android est l'OS mobile le plus utilisé au monde avec deux milliards d'utilisateurs mensuels actifs en mai 2017, dominant les parts de marché devant l'iOS d'Apple sur iPhone ou iPad. Il est créé par Andy Rubin¹⁰ autour d'un noyau Linux, il permet l'installation d'une infinité d'applications de messagerie et téléphonie, de Facebook à Twitter en passant par Snapchat ou Instagram, ou encore d'actualité, comme L'Express et des jeux vidéo mobiles. Sa conception open-source et sa gratuité l'ont rendu populaire chez de nombreux fabricants de smartphones, qui le personnalisent la plupart du temps avec leurs propres interfaces ou fonctionnalités (propriétaires). La protection de ce système et ses données personnelles, comme tous les autres OS mobiles, est devenu un enjeu crucial de cyber sécurité (10).

2.6.2 Historique

Android a été créé par une société nommée Android Inc. à Palo Alto de Californie vers l'an 2003. L'idée originale de la compagnie était de créer et de développer les OD¹¹ avancées pour appareils photo numériques, mais après avoir étudié le marché, ils ont réalisé que le marché pour ce produit n'était pas assez grand et profond afin qu'ils puissent faire une énorme quantité de bénéfices et d'attirer suffisamment de clients. Donc, ils ont décidé d'aller pour la création d'un système d'exploitation pour téléphones

⁹ la version 7.x du système d'exploitation mobile Android développé par Google. Elle est disponible depuis le mois d'août 2016. Elle est décrite par les niveaux d'API 24 et 25.

¹⁰ le cofondateur et ancien CEO de Danger Inc. et du système d'exploitation Android Inc. Jusqu'en mars 2013, il fut vice-président de l'ingénierie chez Google, où il supervisa le développement d'Android, un système d'exploitation open-source pour smartphones.

¹¹ Objectifs avec lentilles Diffringentes qui permettent de réduire les aberrations chromatiques ainsi que le poids.

mobiles. Google ayant acquis Android en 2005 et plus tard a apporté le système d'exploitation en production et distribution en 2007. Ça a été aussi suivi par la création de l'Open Handset Alliance¹², qui a été mené par Google.

Android est basé sur le noyau de Linux qui est très populaire et qui grandit sans cesse. Avec le pouvoir du noyau Linux, le système d'exploitation mobile Android ne peut que grandir de plus en plus. Tout comme Java qui utilise une machine virtuelle afin de compiler et d'exécuter son code, Android aussi possède sa propre machine virtuelle parfaite appelé la Machine virtuelle Dalvik¹³ qui est utilisé afin d'exécuter les différentes applications sur le noyau Linux. Android a eu beaucoup de succès au cours des années et domine aujourd'hui l'essentiel du marché en ce qui concerne le marché du téléphone mobile. Il est maintenant en concurrence directe avec l'IOS d'Apple. La plupart des fabricants de téléphone mobile ont maintenant adopté Android comme leurs systèmes d'exploitation. Une des raisons de la popularité croissante d'Android est qu'ils travaillent avec les différents fabricants afin de mieux répondre à leurs besoins et de savoir comment faire pour s'améliorer (11).

2.6.3 Architecture

Android est basé sur un kernel linux 2.6.xx, au-dessus du kernel il y a "le hardware abstraction layer" qui permet de séparer la plateforme logique du matériel. Au-dessus de cette couche d'abstraction on retrouve les bibliothèques C/C++ utilisées par un certain nombre de composants du système Android. Au-dessus des bibliothèques on retrouve l'Android Runtime, cette couche contient les bibliothèques cœur du Framework ainsi que la machine virtuelle exécutant les applications. Au-dessus la couche "Android Runtime" et des bibliothèques cœur on retrouve le Framework permettant au développeur de créer des applications. Enfin au-dessus du Framework il y a les applications comme illustrée dans la figure suivante (Figure 2.1) :

¹² Open Handset Alliance (abrégé OHA) est un consortium de plusieurs entreprises dont le but est de développer des normes ouvertes pour les appareils de téléphonie mobile.

¹³ Une machine virtuelle destinée aux téléphones mobiles et tablettes tactiles, qui est incorporée dans le système d'exploitation Android.

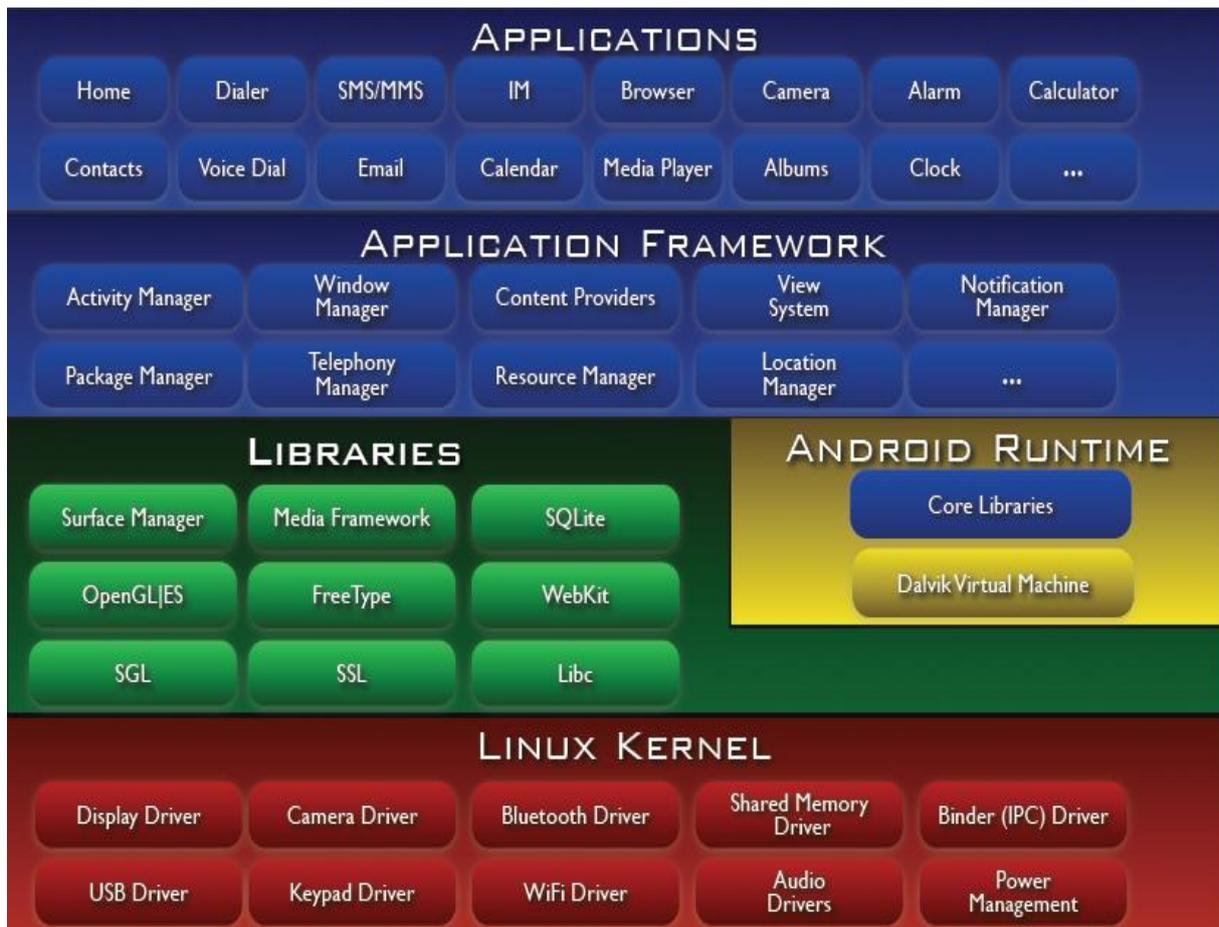


Figure 2.1 : Architecture Android.

- La couche Application

Les applications sont sur la couche supérieure de la pile logicielle Android. L'utilisateur de l'appareil Android a plus d'interaction avec la couche d'application. Android offre un ensemble d'applications, y compris un client de messagerie, navigateur, cartes, calendrier, contacts, d'autres à l'appareil.

Avec le noyau Linux fonctionnant sur le sous-jacent, Android offre plusieurs couches pour sa pile logicielle. La couche application se trouve au-dessus de la couche d'Android. Android propose un ensemble d'applications différentes : Email client, Navigateur, Calendrier, Contacts etc.... Ces applications s'avèrent pour être les plus familières applications utilisées par les utilisateurs d'Android et ils se trouvent sur le haut niveau de la couche application. Tout comme ces applications natives, Android offre également un accès aux troisièmes parties des applications (11).

- Application Framework

Le cadre de l'application a un rôle important afin de fournir des services de haut-niveau qui peuvent être utilisés par de nombreuses applications. Ceci est rendu disponible sous la forme de classes JAVA.

Les outils d'aide sont disponibles pour les développeurs Android dans le monde, ce qui leurs donnent accès à ces services dans leurs applications. Cela permet d'économiser du temps et ne pas écrire les mêmes fonctions plusieurs fois (11).

Le Framework Android inclut les services principaux suivants :

- ✓ Activity Manager : en charge du commence et l'arrêt des applications en respect de leur activité.
- ✓ Content Providers : cela permet aux applications de partager des données entre eux.
- ✓ Notifications Manager : fournit des alertes et des notifications aux utilisateurs en ce qui concerne les différentes applications.
- ✓ View System : utilisée pour créer des interfaces utilisateur pour les applications afin que l'utilisateur puisse interagir avec eux à travers le toucher et des gestes.
- ✓ Package Manager : demandes de permis de communiquer et de recueillir des informations d'autre application actuellement installée sur le système.

- Android Runtime: Dalvik Virtual Machine:

JVM aka Java Virtual machine est une machine virtuelle fictive conçue comme la machine parfaite, cela sert à compiler le code Java. Pour Android, qui utilise Java, la JVM est remplacée par la Machine virtuelle Dalvik (DVM). Il est conçu et optimisé afin de compiler du code pour Android et est principalement utilisé pour les téléphones mobiles.

En comparaison avec la machine virtuelle Java qui utilise une instruction de pile de 8 bits, la Machine virtuelle Dalvik utilise une instruction de pile personnalisée de 16 bits. Alors DVM est plus optimisé que la JVM. C'est aussi le car le code doit être compilé plus rapidement que le code Java en raison de la limitation initiale des ressources matérielles sur les téléphones mobiles par rapport nombre de processeurs. Avec une pile

d'instruction de 16 bits cela permet la réduction du temps requis par rapport au nombre d'instructions donc à son tour augmente la performance et le nombre de code étant interprété par l'interpréteur. Cela permet une meilleure performance car il permet aux systèmes d'exploitation à compiler le code plus rapidement pour les applications différentes.

Le runtime Android est composé de deux éléments principaux : la machine virtuelle Dalvik et les bibliothèques de Java de base. La machine virtuelle Dalvik est l'implémentation de Google de Java qui est optimisé pour être utilisé sur des appareils mobiles (11).

- Noyau Linux

La dernière couche du système d'exploitation Android est le noyau Linux qui n'interagit pas vraiment avec les utilisateurs et les développeurs. Au cours de ses premiers jours, Android a été créé et basé sur le noyau Linux 2.6 optimisés pour qu'il soit rendu et utilisé sur le matériel mobile. Avec l'évolution et le développement continu du noyau Linux au cours des années, de nouvelles versions d'Android sont basées sur le noyau Linux 3.1. Ceci permet un accès plus proche au matériel de sorte que leurs applications interagissent directement avec le matériel plus vite. À son tour en fait, cela signifie que le noyau est chargé de gérer toutes les bases et les éléments primitifs de la téléphonie mobile comme la gestion de l'alimentation et la recharge de la batterie.

Tout comme Android, le noyau Linux est un projet open source, maintenu par la Linux Fondation et utilisé partout dans le monde en solutions d'entreprise (11).

2.7 Application Android

2.7.1 Présentation

Une application Android est une application mobile spécifiquement développée pour les smartphones et les tablettes utilisant le système Android. Elles sont de nature très variable tel que les applications jeux, mobile commerce, utilitaire, service d'information (12).

2.7.2 Cycle de vie d'une application

Les activités du système sont gérées comme des piles d'activités . Lorsqu'une nouvelle activité est lancée, elle est généralement placée en haut de la pile actuelle et devient l'activité en cours d'exécution, l'activité qui la précède reste toujours en dessous de celle-ci dans la pile et ne reviendra pas au premier plan jusqu'à la fin de la nouvelle activité. Il peut y avoir une ou plusieurs piles d'activités visibles à l'écran (13).

2.7.2.1 Etat d'une activité (Application)

Une activité a essentiellement quatre états (13)

- Si une activité est au premier plan de l'écran (à la position la plus élevée de la pile), elle est *active* ou en *cours d'exécution*. Il s'agit généralement de l'activité avec laquelle l'utilisateur interagit actuellement.
- Si une activité a perdu le focus mais est toujours présentée à l'utilisateur, elle est *visible*. Cela est possible si une nouvelle activité transparente se concentre sur votre activité, une autre activité a une position plus élevée en mode multi-fenêtres ou l'activité elle-même n'est pas focalisable en mode de fenêtrage actuel. Une telle activité est complètement vivante (elle conserve toutes les informations d'état et de membre et reste attachée au gestionnaire de fenêtres).
- Si une activité est complètement masquée par une autre activité, elle est *arrêtée* ou *masquée*. Le système conserve toujours toutes les informations d'état et de membre, cependant, il n'est plus visible pour l'utilisateur, donc sa fenêtre est masquée et il sera souvent tué par le système lorsque de la mémoire est nécessaire ailleurs.
- Le système peut supprimer l'activité de la mémoire soit en lui demandant de terminer en appelant la commande **finishAffinity()**, soit en tuant simplement son processus, ce qui

le *détruit*. Lorsqu'il est à nouveau affiché à l'utilisateur, il doit être complètement redémarré et restauré à son état précédent.

Le diagramme suivant montre les chemins d'état importants d'une activité. Les rectangles carrés représentent les méthodes de rappel qui peuvent être implémenté pour effectuer des opérations lorsque l'activité se déplace entre les états. Les ovales colorés sont des états majeurs dans lesquels l'activité peut être.

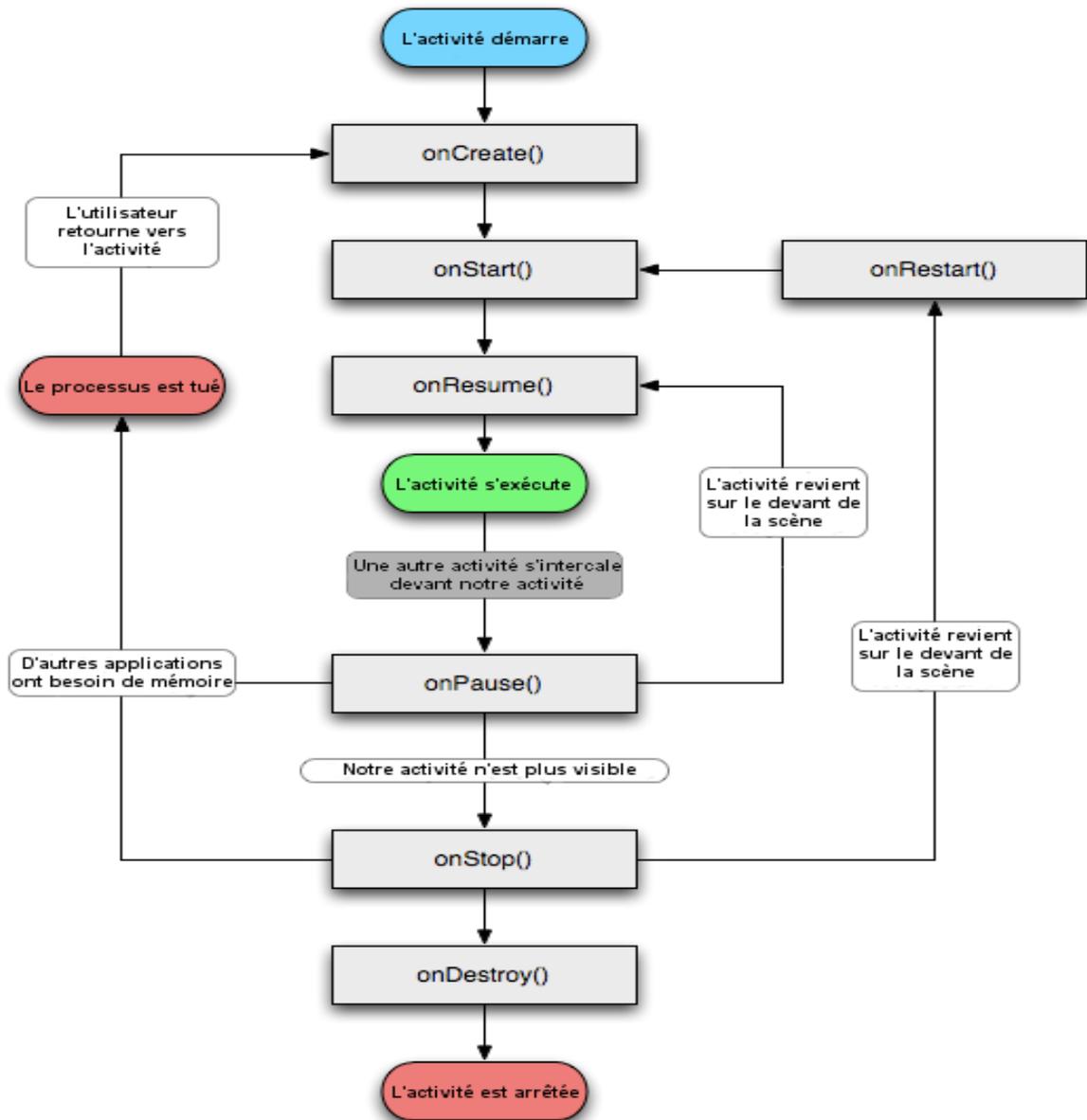


Figure2.2 : Cycle de vie d'une activité.

Il existe trois boucles clés composant une activité :

- La durée de vie d'une activité se déroule entre le premier appel et le **onCreate(Bundle)** dernier appel final **onDestroy()**. Une activité fera toute la configuration de l'état « global » dans **onCreate ()** et libérera toutes les ressources restantes dans **onDestroy ()**. Par exemple, si un thread s'exécute en arrière-plan pour télécharger des données à partir du réseau, il peut créer ce thread dans **onCreate ()**, puis arrêter le thread dans **onDestroy ()**.
- La durée de vie visible d'une activité se produit entre un appel à **onStart()** et un appel correspondant à **onStop()**. Pendant ce temps, l'utilisateur peut voir l'activité à l'écran, bien qu'elle ne soit pas au premier plan et n'interagisse pas avec l'utilisateur. Entre ces deux méthodes, vous pouvez gérer les ressources nécessaires pour montrer l'activité à l'utilisateur. Par exemple, vous pouvez enregistrer un **BroadcastReceiver** dans **onStart ()** pour surveiller les modifications qui ont un impact sur votre interface utilisateur et le désinscrire dans **onStop ()** lorsque l'utilisateur ne voit plus ce que vous affichez. Les méthodes **onStart ()** et **onStop ()** peuvent être appelées plusieurs fois, car l'activité devient visible et masquée pour l'utilisateur.
- La durée de vie au premier plan d'une activité se produit entre un appel à **onResume()** et un appel correspondant à **onPause()**. Pendant ce temps, l'activité est visible, active et en interaction avec l'utilisateur. Une activité peut fréquemment passer de l'état de reprise à celui de pause – par exemple lorsque l'appareil se met en veille, lorsqu'un résultat d'activité est fourni, lorsqu'une nouvelle intention est délivrée – le code de ces méthodes doit donc être assez léger.

Le cycle de vie complet d'une activité est défini par les méthodes d'activité suivantes. Tous ces éléments sont des crochets que vous pouvez remplacer pour effectuer le travail approprié lorsque l'activité change d'état. Toutes les activités seront mises **onCreate(Bundle)** en œuvre pour effectuer leur configuration initiale ; beaucoup mettront également en œuvre **onPause()** pour valider les modifications des données et se préparer à interrompre l'interaction avec l'utilisateur et **onStop()** à ne plus être visible à l'écran. Votre superclasse doit être appelé lors de l'implémentation de ces méthodes (13) (Figure 2.2).

Conclusion

Dans ce chapitre nous avons présenté brièvement l'application mobile ainsi que le système d'exploitation Android, son architecture et son historique. Cela nous a aidés à bien comprendre le fonctionnement de ce système, tout en décrivant les applications Android

Le chapitre suivant sera consacré à la conception de l'application.

Chapitre 3 : Conception et Analyse et Spécification des besoins

Partie 1 : Analyse et Spécification des Besoins

Introduction

Dans ce chapitre nous présentons les objectifs de notre application. Ce qui nous amène à identifier les acteurs du système et les besoins des utilisateurs que nous essayons de projeter dans les diagrammes de cas d'utilisations globales et détaillés. Et on finira par présenter les différents organigrammes de notre méthode de conception du projet (UML¹⁵). L'étude de l'existant est une étape essentielle de tout projet informatique. Cette étude permet de dégager les défaillances et les insuffisances auxquelles nous devons y remédier.

3 Analyse et Spécification des besoins

3.1 Problématique

De nombreux étudiants et enseignants-chercheurs qui rédigent un mémoire ou une thèse s'intéressent à la recherche documentaire pour préparer leur étude empirique. Pour répondre à des questions de recherche, il est toujours nécessaire de collecter soi-même des données par le biais de recherches qualitatives ou quantitatives. Par ailleurs nous attestons des difficultés auxquelles sont confrontés ces étudiants, dans le cadre de leurs recherches documentaires, liées à la rédaction de leur mémoire et autres. Ils rencontrent des difficultés majeures dans deux domaines.

- Ils perdent les repères spatiaux. Le système d'organisation des connaissances des bases de données génère une déroute.
- Par ailleurs, ils peinent à comprendre et utiliser les langages documentaires spécifiques à chacun des outils consultés.

Le but de notre travail est de réaliser une application sous Android pour la recherche documentaire dans la bibliothèque de Bejaia, pour permettre aux utilisateurs de réaliser leurs recherches documentaires dans des bonnes conditions tout en gagnant du temps et d'énergies.

¹⁵ Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système.

3.2 Le processus de modélisation UML

3.2.1 Définition 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. En clair, il ne faut pas désigner UML en tant que méthode mais plutôt comme une boîte d'outils qui sert à améliorer les méthodes de travail (14).

3.2.2 Les différents types de diagrammes :

UML dans sa version 2.0 s'articule autour de treize 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 de comportement (15).

La figure 3.1 présente les différents types de diagramme de l'UML :

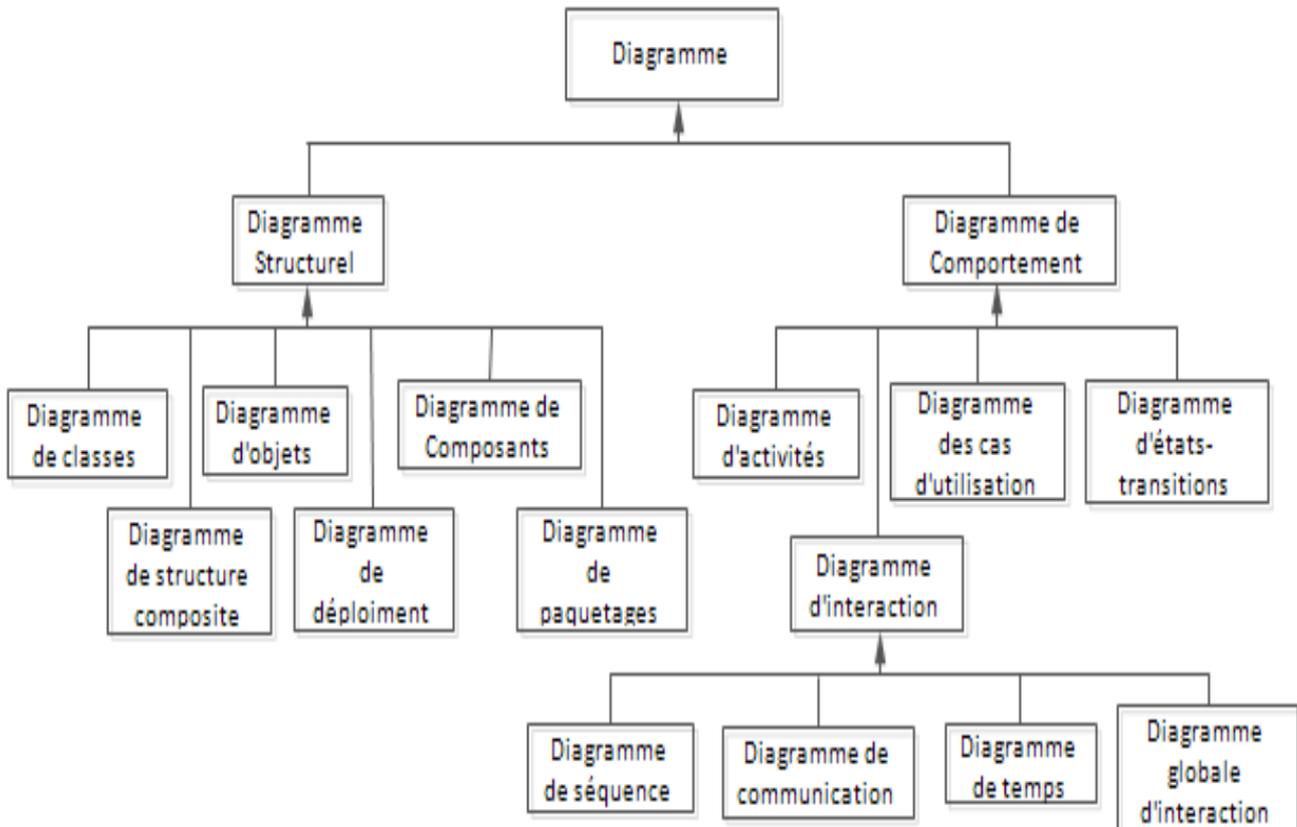


Figure 3.1: La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme.

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont :

- **Diagramme de cas d'utilisation** : il représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.
- **Diagramme de classes** : est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation).

➤ **Diagramme de séquence** : représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. On peut représenter les mêmes opérations par un diagramme de communication, graphe dont les nœuds sont des objets et les arcs (numérotés selon la chronologie) les échanges entre objets. En fait, diagramme de séquence et diagramme de communication sont deux vues différentes, mais logiquement équivalentes (on peut construire l'une à partir de l'autre) d'une même chronologie.

Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique (16).

3.3 Spécification des besoins

La spécification des besoins représente la première phase du cycle de développement d'une application. Elle doit décrire sans ambiguïté l'application à développer.

Elle représente l'ensemble des services que doit fournir le logiciel à son utilisateur. Selon le processus unifié chaque service est modélisé sous un cas d'utilisation, pour élaborer à la fin un diagramme UML de cas d'utilisation. Chaque cas d'utilisation est décrit sous une forme textuelle représentant un scénario nominal (ensemble des actions à réaliser pour atteindre l'objectif).

Dans cette étape nous allons spécifier l'ensemble des besoins fonctionnels et non fonctionnels liées à notre application. Ensuite, nous allons modéliser les spécifications semi-formelles des besoins à l'aide des diagrammes de cas d'utilisation et les diagrammes de séquences.

3.3.1 Besoins fonctionnels :

Il s'agit des fonctionnalités que doit fournir le système aux délégués acteurs. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système. Le système doit permettre :

- A la bibliothécaire :
- De s'identifier
- D'ajouter, consulter, supprimer, modifier des ouvrages et vérifier la disponibilité des exemplaires.
- D'ajouter, consulter, supprimer, modifier, rappeler un emprunt.

- D'ajouter, supprimer une pénalité.
 - De consulter la liste des pénalités.
 - Définir la durée de la pénalité.
 - Définir la durée des emprunts.
 - Définir le nombre maximal d'ouvrage à emprunter.
 - Consulter les ouvrages populaires (les plus sollicités).
 - Consulter l'ouvrage le moins emprunté.
 - Consulter les nouveautés.
- A adhérent :
- D'effectuer une recherche selon un critère défini.
 - De réserver un ouvrage.

3.3.2 Besoins non fonctionnels :

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD, de système d'Exploitation...), dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités. L'application devra être capable de :

- Tourner en réseau.
- Être compatible avec n'importe quel système d'exploitation.
- IL faudra aussi noter que l'application devra être hautement sécurisée car les informations ne devront pas être accessibles à tout le monde.

3.4 Analyse des besoins

3.4.1 Identification des acteurs

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec le système. Tout d'abord, nous commençons par définir ce qui est un acteur (Figure 3.2) :

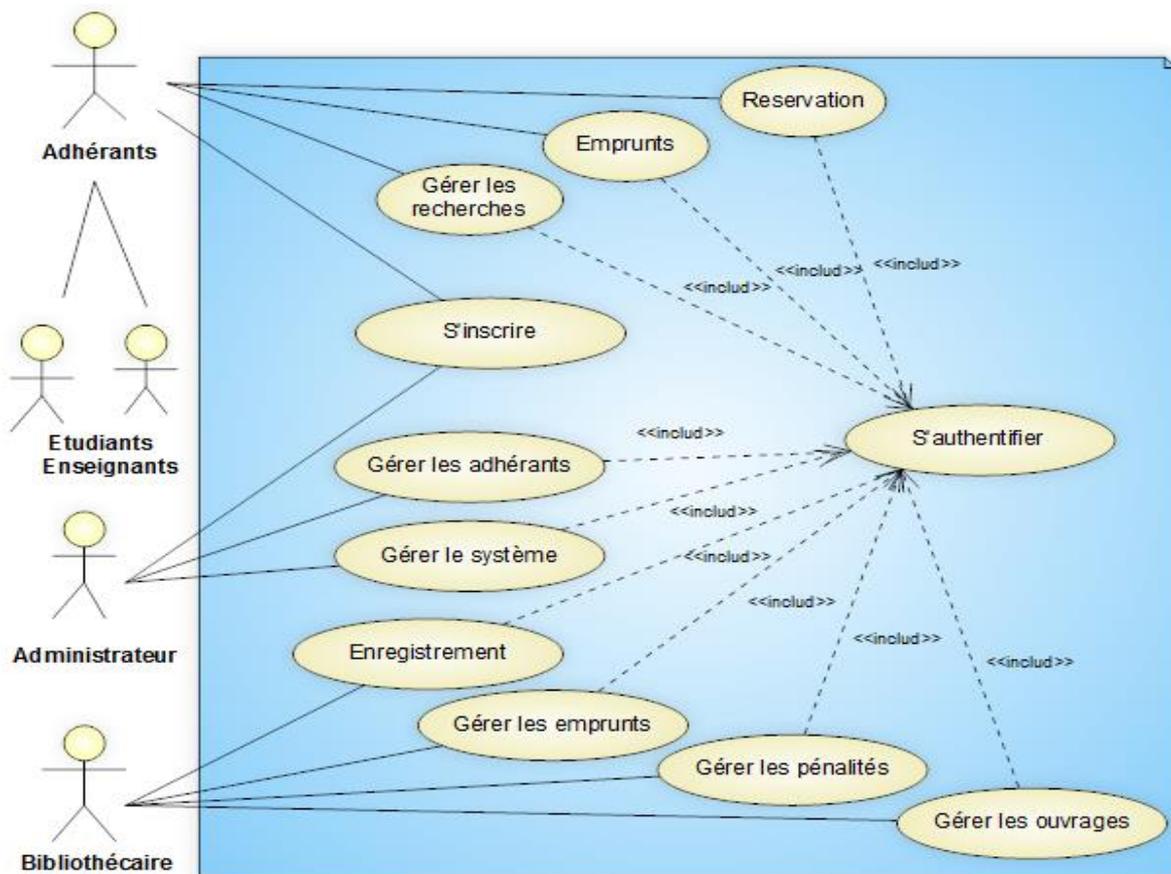


Figure 3.2 :Relation de généralisation entre acteurs.

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Un acteur est une personne, un matériel ou un logiciel qui interagit directement avec le système pour réaliser une tâche. Ainsi, un acteur peut consulter et/ou modifier directement. Les acteurs de notre application sont (Tableau3.1) :

N°	Acteurs	Description
1	Administrateur	Le rôle de l'administrateur est la gestion des utilisateurs qui peuvent accéder au système. Comme il se charge de l'administration et garde le contrôle sur la plate-forme web et mobile de l'établissement.
2	Bibliothécaire	Personne à utiliser le système pour gérer les emprunts, les réservations, les restitutions, les emprunteurs et le fond de bibliothèque
3	Les adhérents (étudiants,	L'utilisateur consulte le site de recherche, choisir l'ouvrage dont il a besoin, leurs détails, ainsi que leur localisation.

	enseignants, chef de département...)	
--	--------------------------------------	--

Tableau 3.1 : identifications des acteurs.

3.4.2 Identifications des cas d'utilisations

Un cas d'utilisation est une fonctionnalité de système qui produit un résultat observable pour un utilisateur potentiel du système. Le cas d'utilisation regroupe une famille de scénario ou chaque scénario est un traitement particulier du système. C'est un schéma qui montre les cas d'utilisation en forme (ovales) liés par des associations (lignes) à leurs acteurs « stick man », ou représentation graphique équivalente. Chaque association signifie simplement « participe à » et un cas d'utilisation doit obligatoirement relier à au moins un acteur.

Ce diagramme permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs aux systèmes).

Lors de notre analyse des besoins nous avons pu identifier les actions importantes que nous présenterons ci-dessous et nous les modélisons par la suite avec les diagrammes cas utilisation d'UML (Tableau 3.2).

N°	Cas d'utilisation	Acteurs
1	S'inscrire	Les adhérents Administrateur Bibliothécaire
2	Gérer des adhérents	Administrateur
3	Gérer des emprunts	Bibliothécaire
4	Gérer des ouvrages	Bibliothécaire
5	Gérer des pénalités	Bibliothécaire
6	Gérer les recherches	Les adhérents
7	Gérer les restitutions	Agent de la bibliothèque
8	L'identification de la bibliothécaire	Administrateur
9	Enregistrement	Bibliothécaire
10	Gérer les compte ou authentification	Administrateur

Tableau 3.2 : Cas d'utilisation associé au système.

- **L'Enregistrement** : La bibliothécaire doit enregistrer les coordonnées des ouvrages empruntés, la date de retour de l'ouvrage et le matricule de l'emprunteur.
- **La gestion des adhérents** : Le système doit permettre à la bibliothécaire d'ajouter, de modifier, de supprimer ou encore de consulter un adhérent pour vérifier le nombre d'emprunts et/ou les dates d'échéances de ses emprunts.
- **La gestion des ouvrages** : L'application doit aussi permettre à la bibliothécaire de gérer la réservation des ouvrages. Elle doit aussi mettre à jour la base de données lorsqu'il y a des nouveaux ouvrages.
- **La recherche d'un ouvrage** : Les membres peuvent accéder à la bibliothèque à partir de n'importe quel terminal de l'université afin de consulter la liste de livres existants, et ensuite se présenter chez la bibliothécaire afin d'effectuer l'emprunt.
- **La gestion des emprunts** : La bibliothécaire doit, avec cette application, pouvoir effectuer les différentes opérations de gestions des emprunts suivants : ajouter, retourner, consulter.
- **La gestion des pénalités** : Le système doit aussi permettre à la bibliothécaire d'enregistrer des pénalités pour un membre, de la supprimer ou de vérifier les listes des adhérents pénalisés ainsi que la durée.
- **Gestion des comptes ou authentification** : Chaque compte est authentifié par un Login et mot de passe, chaque utilisateur doit introduire son login et mot de passe après inscription et remplissage de la fiche d'inscription.
- **Inscription** : chaque nouveau adhérent (enseignants, étudiants, agents administratifs) doit s'inscrire sur la plateforme, si ce n'est pas déjà fait pour bénéficier de ces services. Chaque adhérent doit remplir tous les champs du formulaire pour être inscrit et enregistré dans la base.

3.4.3 Diagrammes des cas d'utilisations

✓ Adhérent :

L'adhérent (étudiants, enseignant, etc...) peut effectuer un emprunt ou bien une réservation. Cette fonctionnalité permet au client de s'inscrire et de choisir, emprunter, réserver un livre (Figure 3.3).

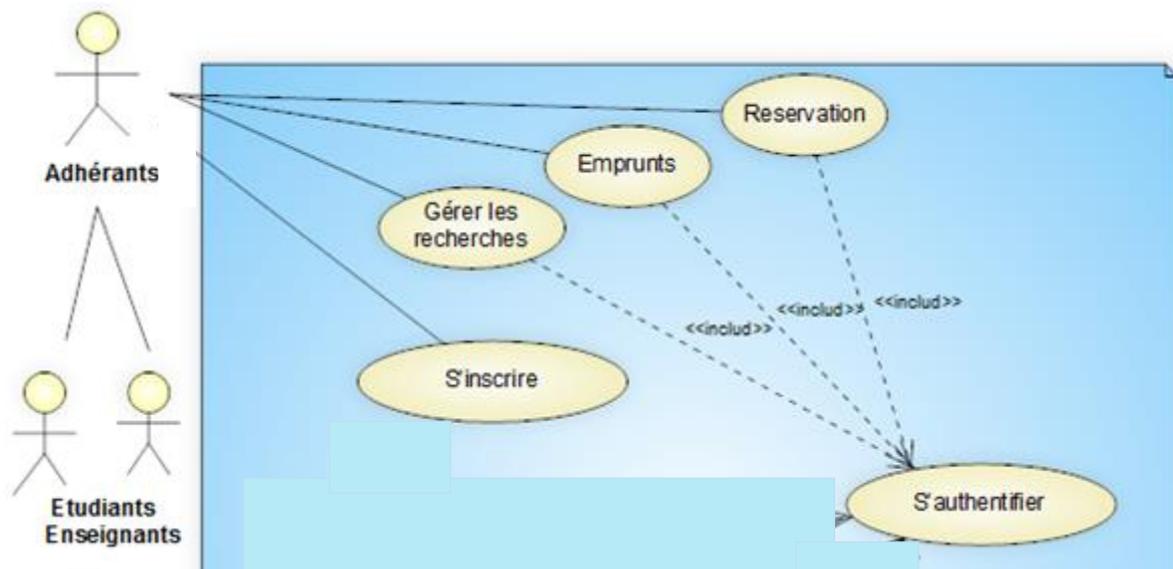


Figure 3.3 : Diagramme de cas d'utilisation (Adhérent).

✓ Administrateur :

L'administrateur doit gérer la plate-forme web et mobile de l'établissement. Cette fonctionnalité permet à l'administrateur de gérer les utilisateurs de la bibliothèque et leurs rôles comme il doit être authentifié (Figure 3.4).

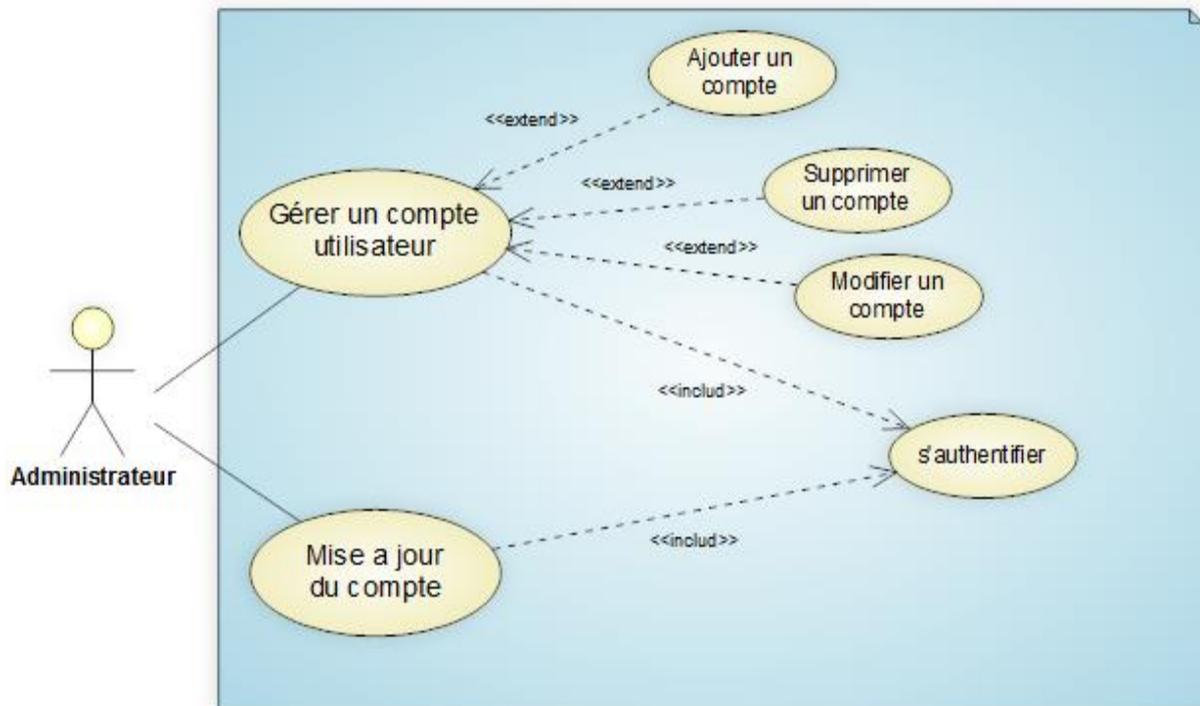


Figure 3.4 : Diagramme de cas d'utilisation (Administrateur).

✓ Bibliothécaire :

Le bibliothécaire doit gérer la bibliothèque. Cette fonctionnalité lui permet de gérer les emprunts, les réservations, les ouvrages, les pénalités, et l'enregistrement. Il doit être authentifié (Figure3.5).

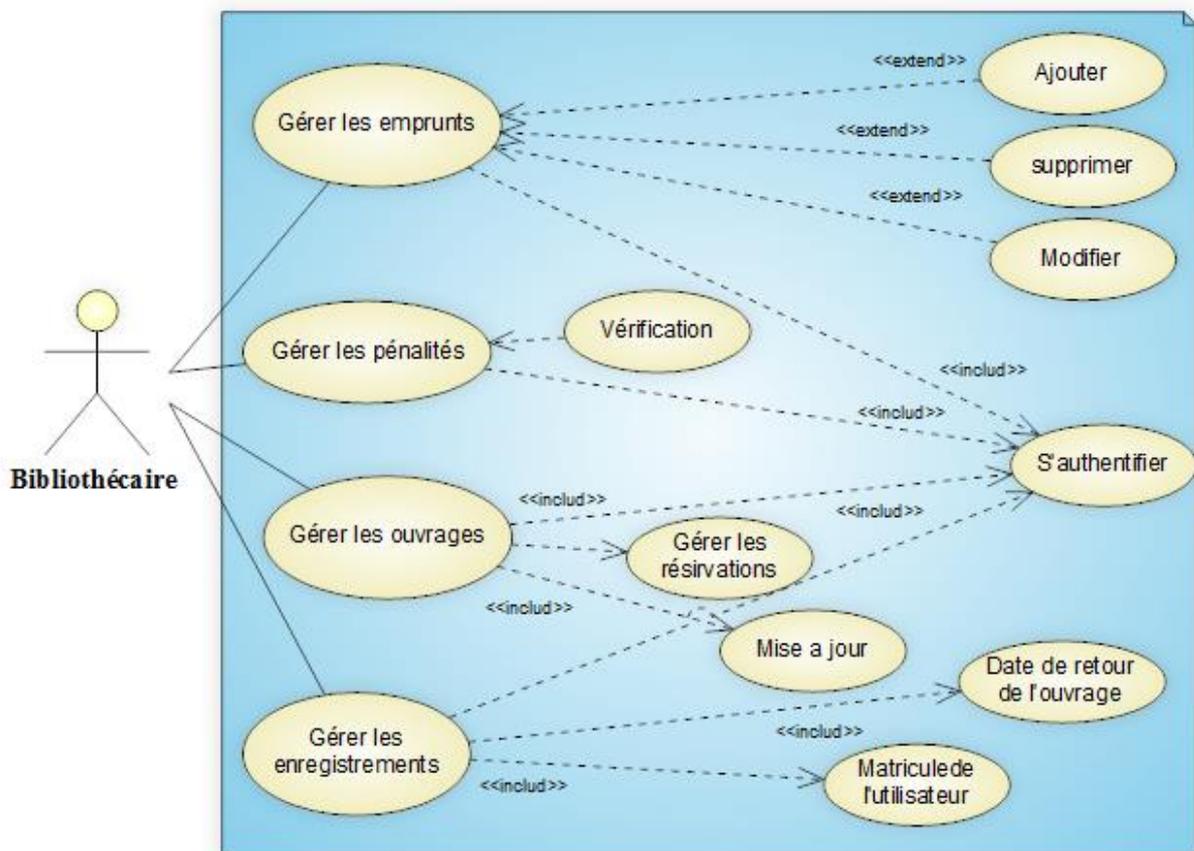


Figure 3.5 : Diagramme de cas d'utilisation (Bibliothécaire).

Partie 2 : Conception

Après avoir tracé les grandes lignes de phase de spécification de besoins, mettons l'accent maintenant sur une phase fondamentale dans le cycle de vie d'un logiciel, la phase de conception. Cette phase a pour objectif de déduire la spécification de l'architecture de système.

En premier lieu, la méthodologie de conception sera présentée, l'organisation des sections de ce chapitre suivra alors la logique de cette technologie. Cette phase aboutira à la conception et la représentation des diagrammes de séquences et d'activités en se basant sur le langage de modélisation UML.

3.5 Diagrammes de séquences des cas d'utilisation :

Les diagrammes de séquence sont une solution populaire de modélisation dynamique en langage UML, car ils se concentrent plus précisément sur les *lignes de vie*, les processus et les objets qui vivent simultanément, et les messages qu'ils échangent entre eux pour exercer une fonction avant la fin de la ligne de vie.

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Ces diagrammes sont utilisés à la fois par les développeurs logiciels et les managers d'entreprises pour analyser les besoins d'un nouveau système ou documenter un processus existant. Les diagrammes de séquence sont parfois appelés diagrammes d'événements ou scénarios d'événements.

Le diagramme de séquence permet de représenter les échanges entre les composants et les objets du système, dans le cadre d'exécution des cas d'utilisation, de point de vue temporel.

Dans ce qui suit, nous présentons toutes les fonctions de l'utilisateur à travers des différents diagrammes de séquences (17).

3.5.1 Diagramme de séquence de cas d'utilisation « s'authentifier » :

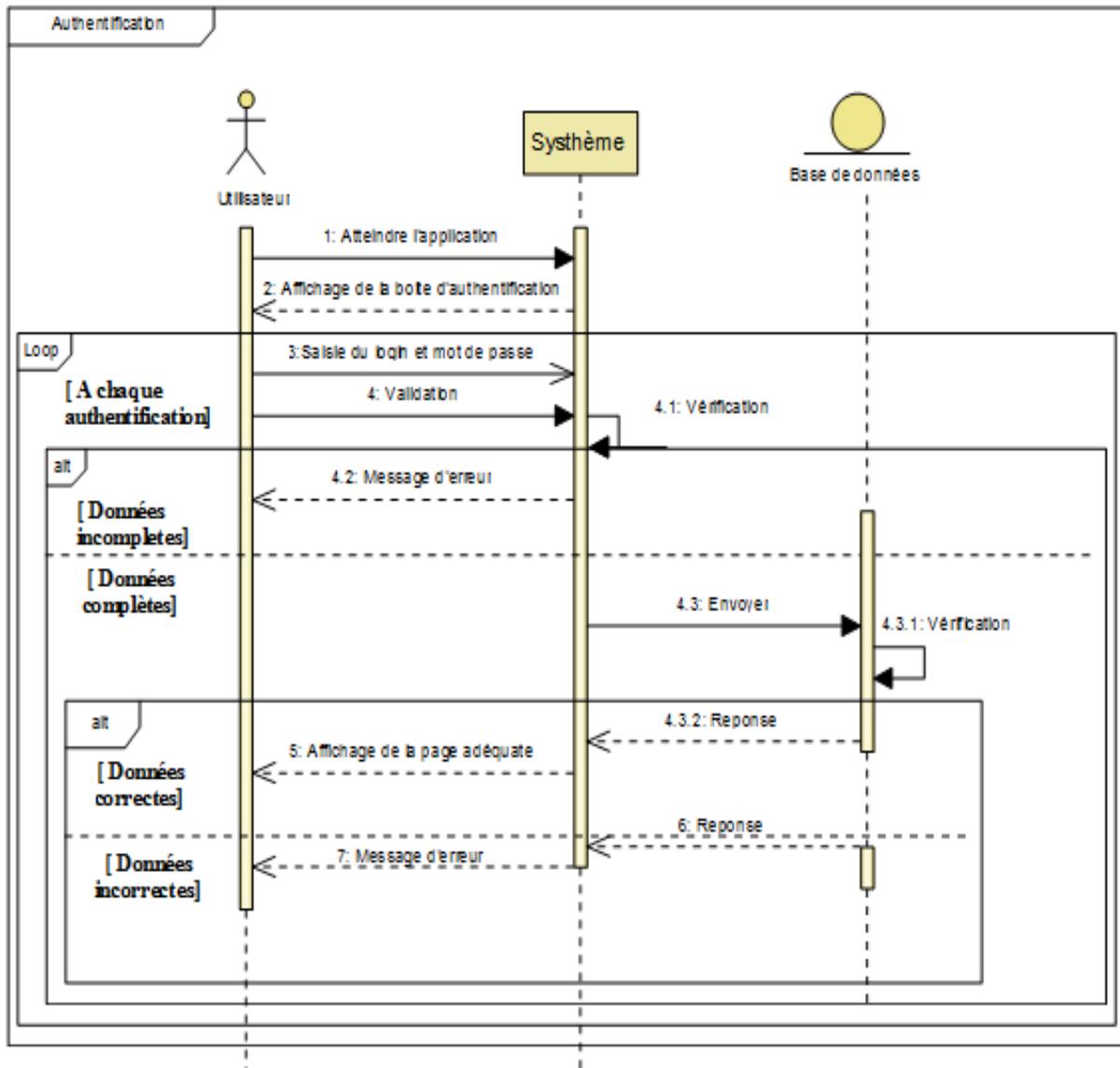


Figure 3.6 : Diagramme de séquence du cas d'utilisation « s'authentifier ».

3.5.2 Diagramme de séquence de cas d'utilisation « s'inscrire » :

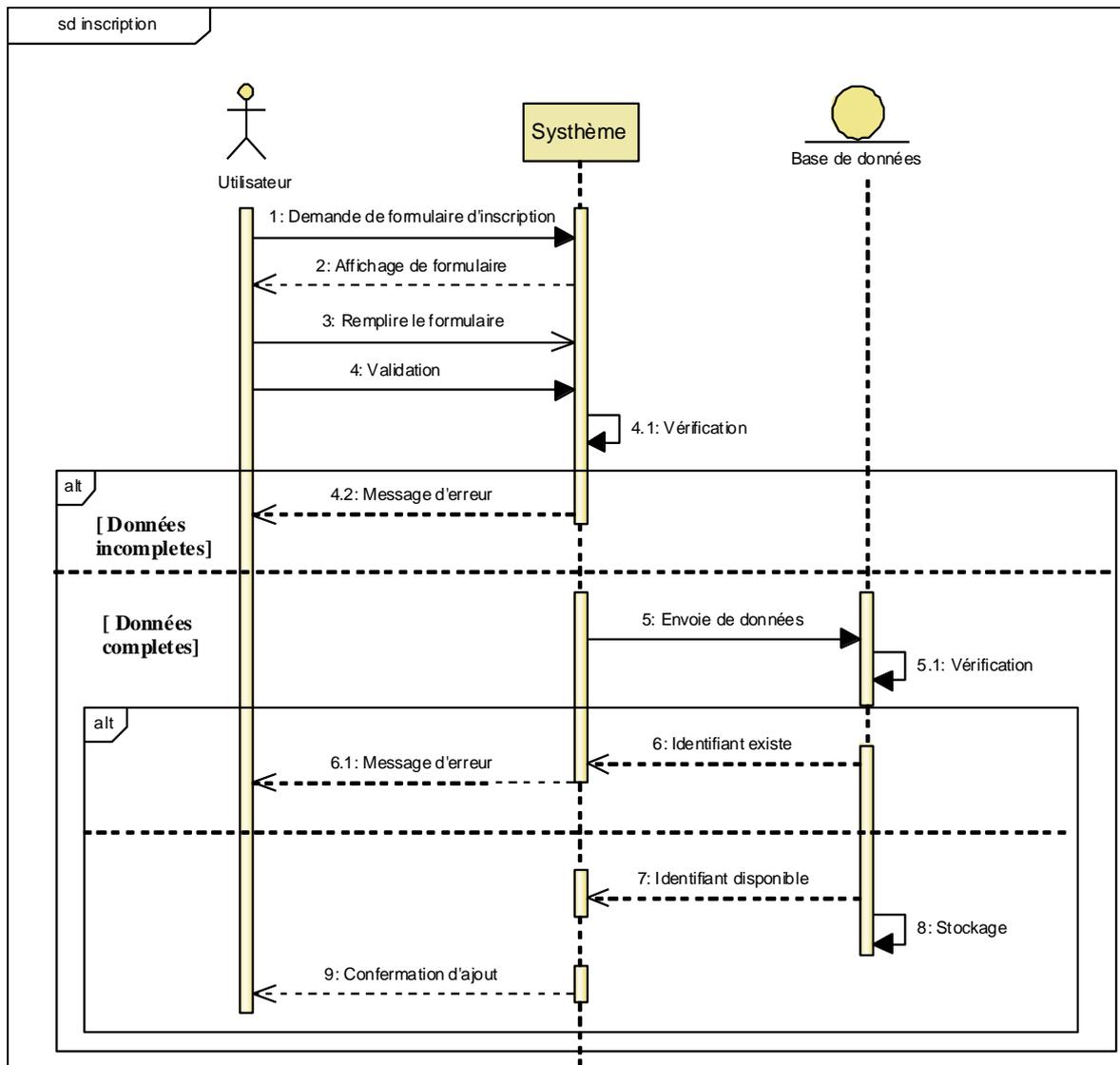


Figure 3.7 : Diagramme de séquence de cas d'utilisation « s'inscrire ».

3.5.3 Diagramme de séquence de cas d'utilisation « Gérer les emprunts » :

La figure suivante présente l'emprunt de documents par un adhérent dans la bibliothèque, telle que l'existence de document et la réservation éventuelle (Figure 3.8).

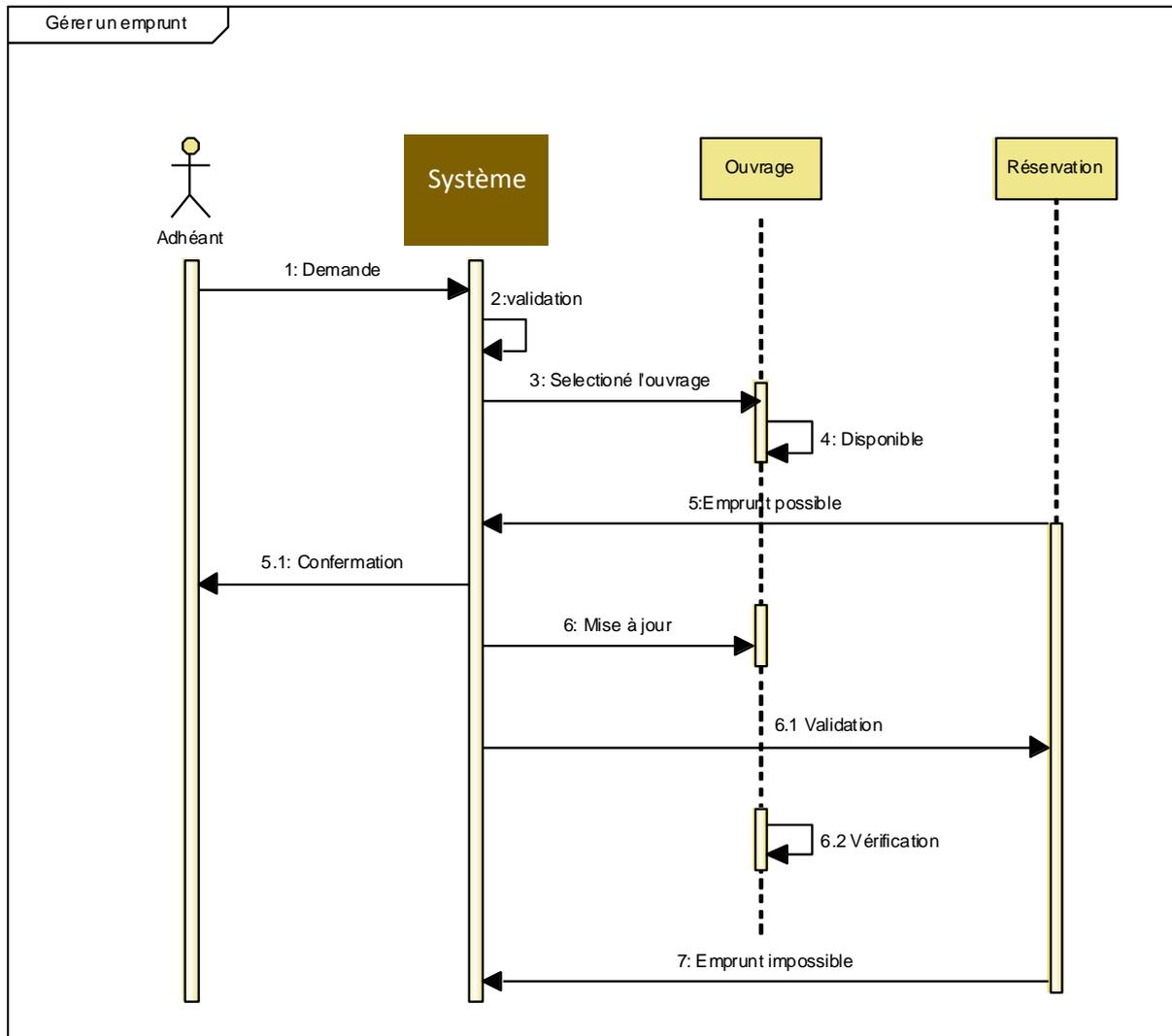


Figure 3.8 : Diagramme de séquence de cas d'utilisation « Gérer des emprunts ».

3.5.4 Diagramme de séquence de cas d'utilisation « Gérer les ouvrages » :

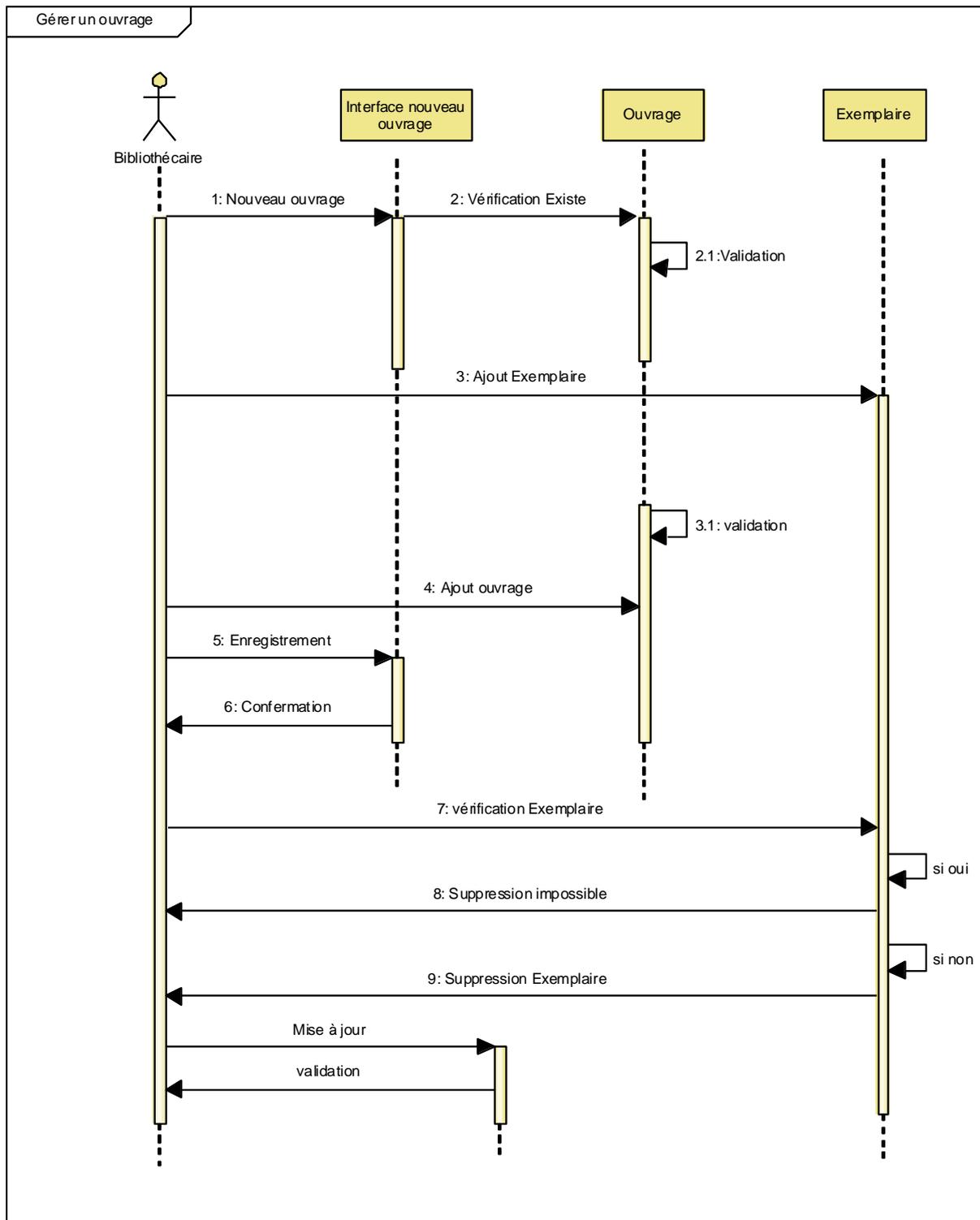


Figure 3.9 : Diagramme de séquence de cas d'utilisation « Gérer des ouvrages ».

3.5.5 Diagramme de séquence de cas d'utilisation « Gérer les pénalités » :

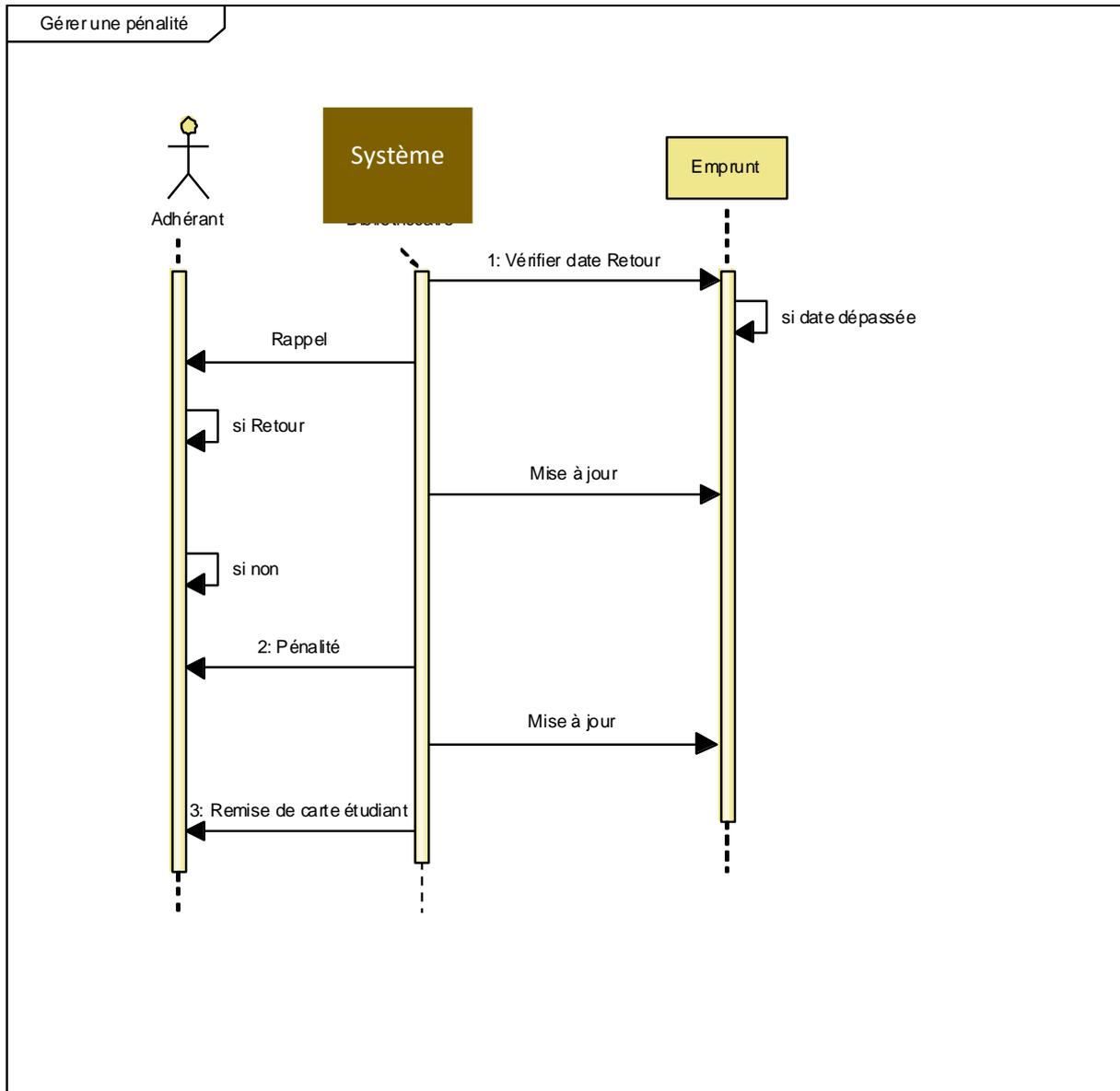


Figure 3.10 : Diagramme de séquence de cas d'utilisation « Gérer les pénalités ».

3.5.6 Diagramme de séquence de cas d'utilisation « Gérer les recherches » :

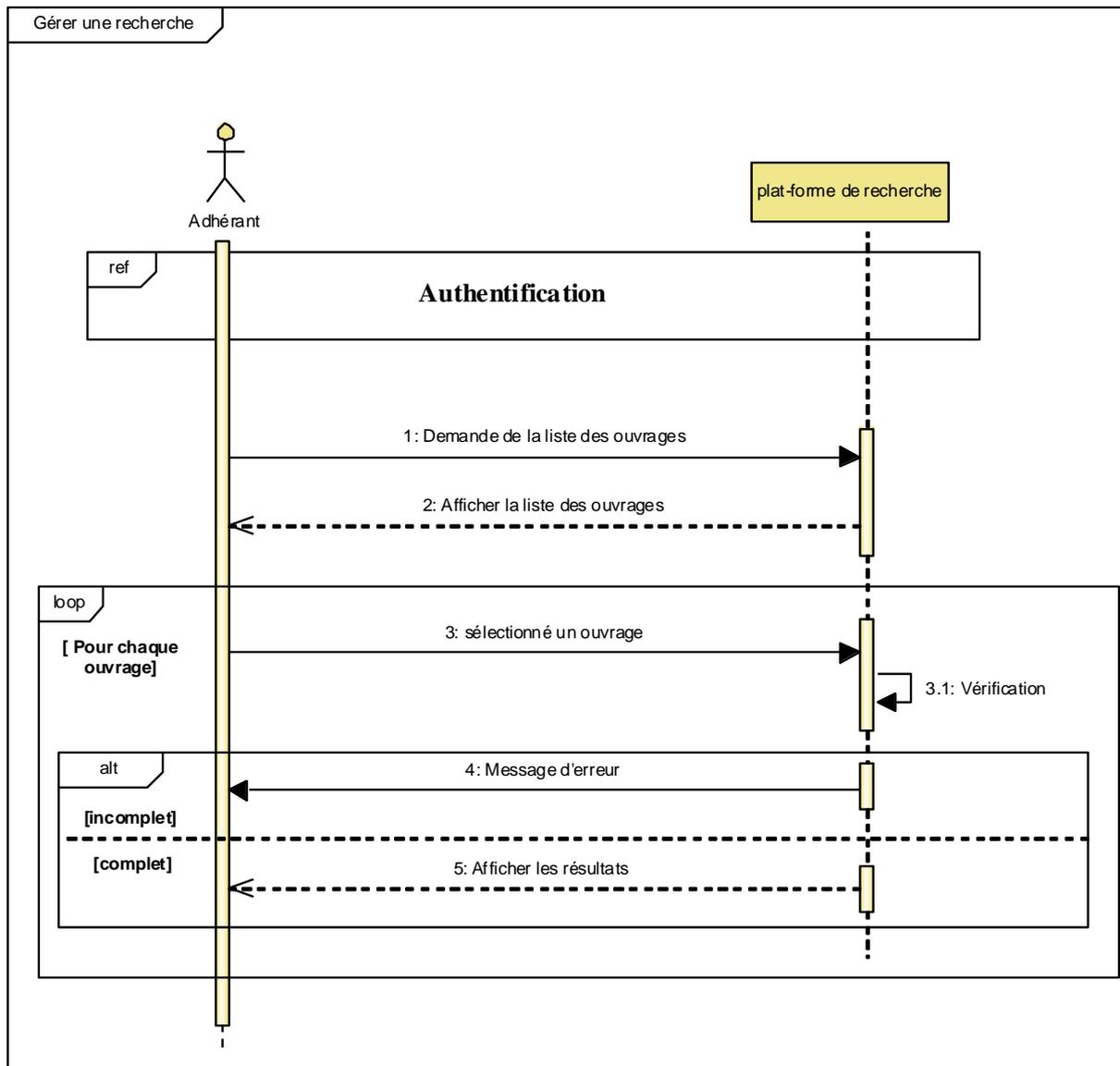


Figure 3.11 : Diagramme de séquence de cas d'utilisation « Gérer les recherches ».

3.5.7 Diagramme de séquence de cas d'utilisation « Gérer les Adhérent » :

La gestion d'un nouvel adhérent (en tant qu'étudiant ou enseignant...). Nous faisons appel à un scénario de création d'un adhérent. Le système retourne une réponse une fois l'enregistrement est fait, car cette séquence est invoquée lors de l'ajout d'un nouvel adhérent.

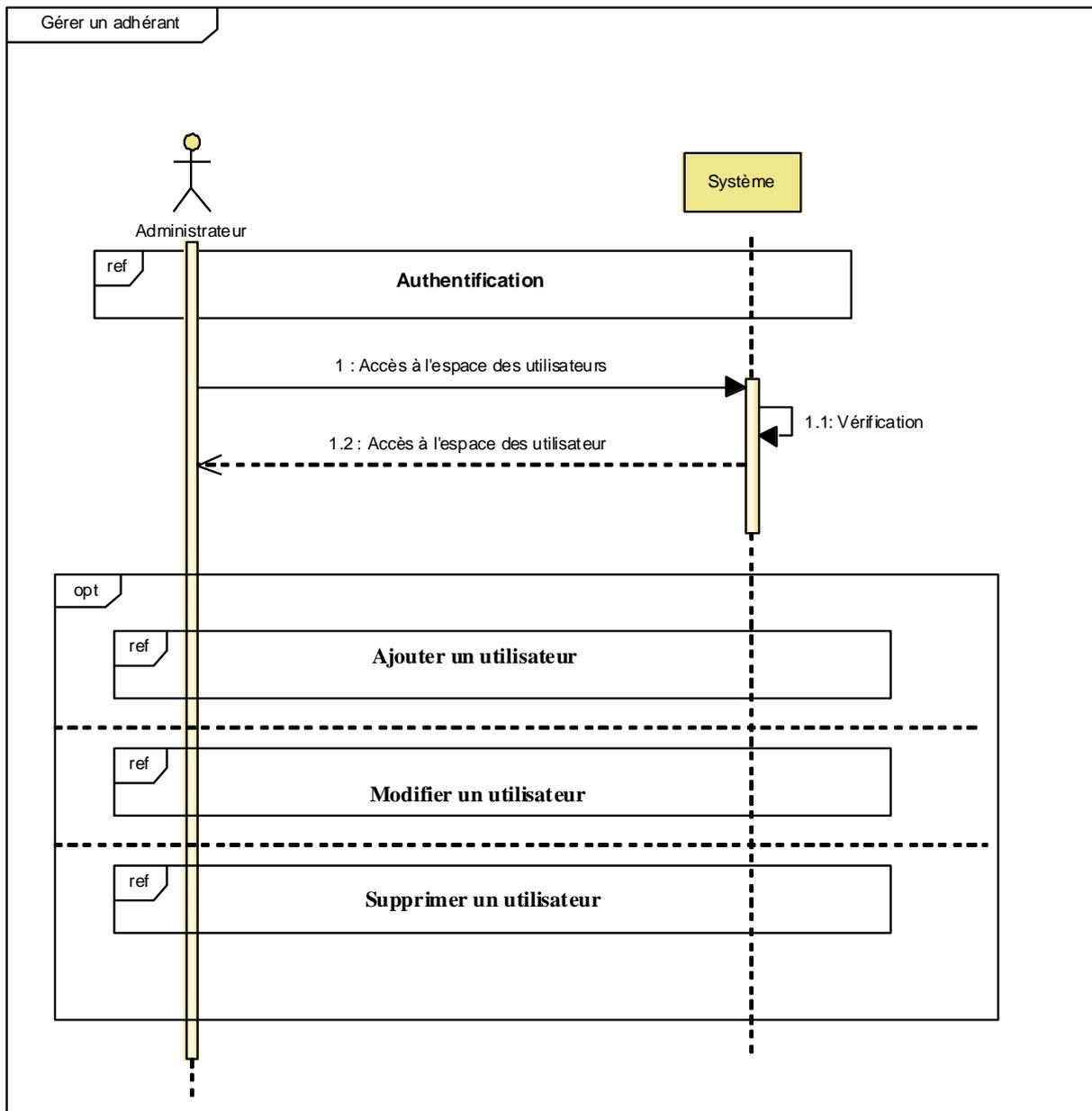


Figure 3.12 : Diagramme de séquence de cas d'utilisation « Gérer les Adhérent ».

3.5.8 Diagramme de séquence de cas d'utilisation « Gérer les restitutions » :

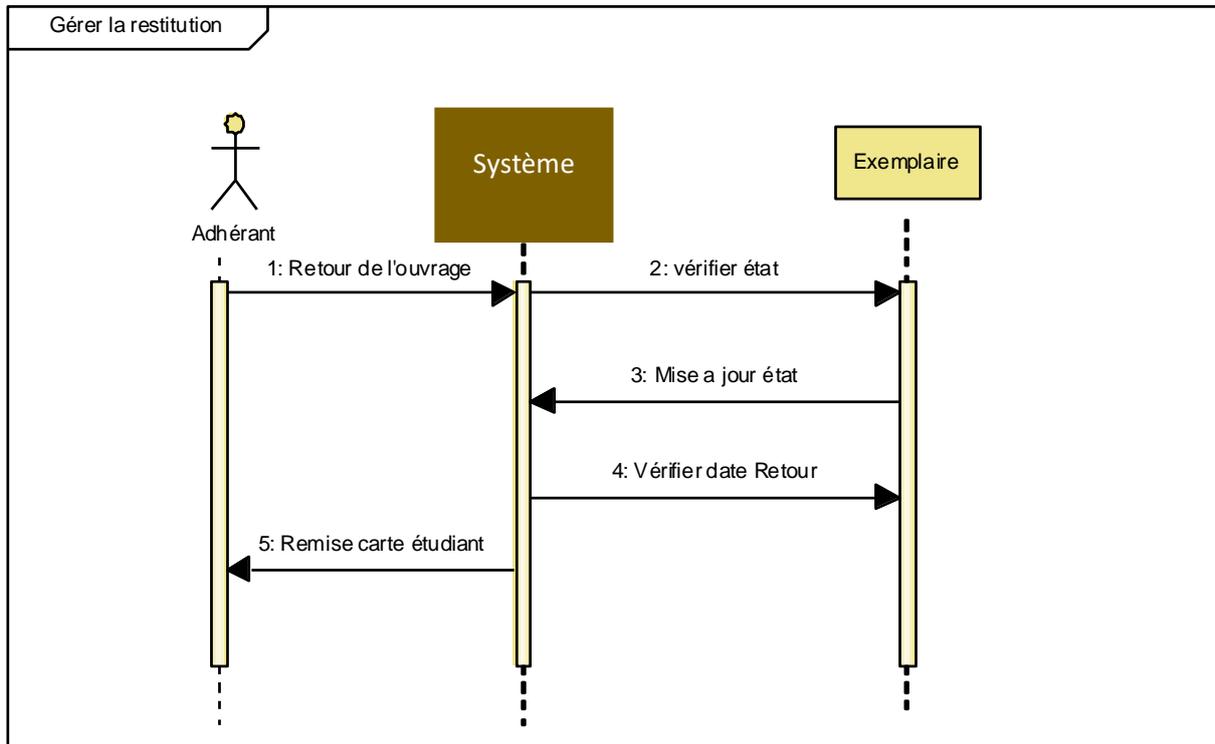


Figure 3.13 : Diagramme de séquence de cas d'utilisation « Gérer les restitutions ».

3.6 Diagramme de classe de l'application :

Le diagramme de classe exprime la structure statique du système en termes de classes et de relations. Le but du diagramme de classes est de modéliser les entités du système d'information. Ces informations sont regroupées ensuite sous forme de diagramme de classes (Figure 3.14) :

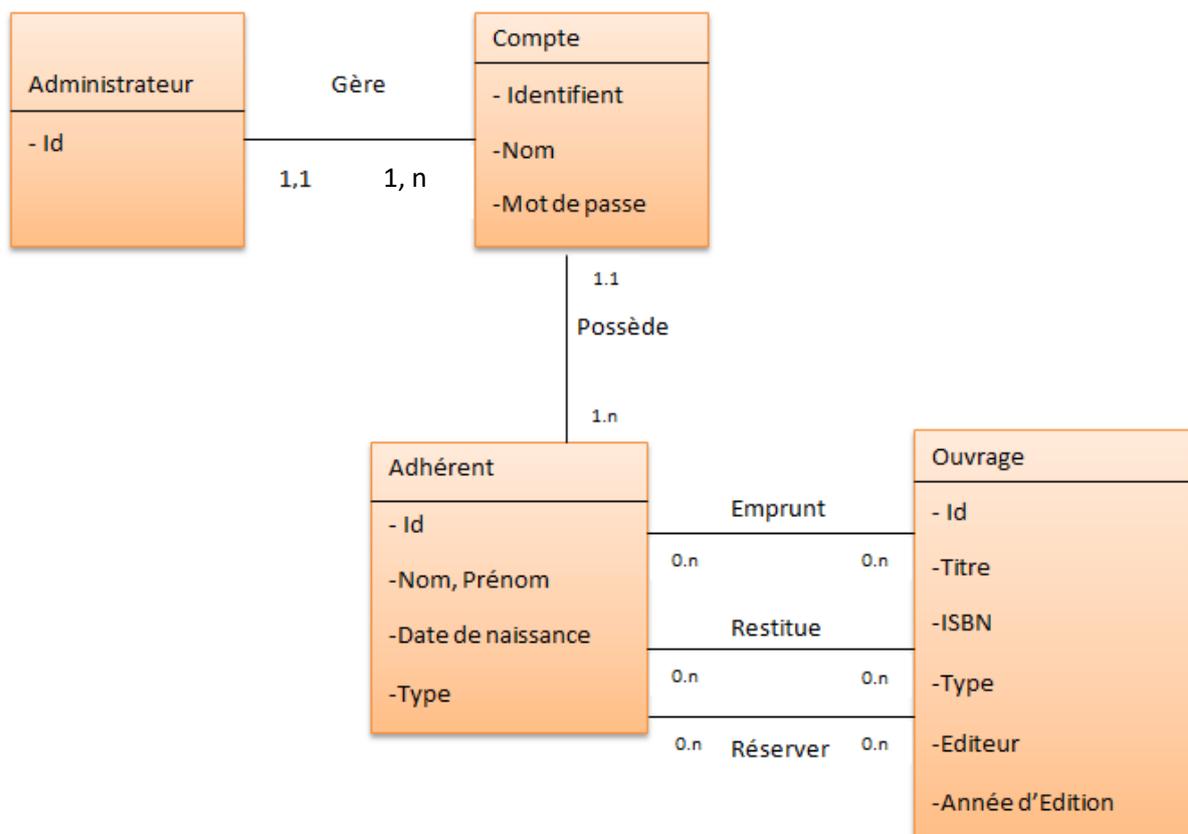


Figure 3.14: Diagramme de classe de l'application.

3.7 Modélisation

3.7.1 Présentation de MCD

Le **MCD (Modèle conceptuel des données)** est une représentation graphique de haut niveau qui permet facilement et simplement de comprendre comment les différents éléments sont liés entre eux à l'aide de diagrammes codifiés dont les éléments suivants font partie (Figure 3.15) :

- Les entités (1 rectangle = 1 objet).
- Les propriétés (la liste des données de l'entité).
- Les relations qui expliquent et précisent comment les entités sont reliées entre elles (les ovales avec leurs « pattes » qui se rattachent aux entités).
- Les cardinalités (les petits chiffres au-dessus des « pattes »).

Utilisé assez tôt en conception de base de données, le **MCD** évoluera ensuite vers les autres outils de Merise¹⁶, à savoir le MPD et le MLD.

Cet outil permet d'échanger entre informaticiens et non-informaticiens sur l'outil à informatiser. On peut ainsi à partir d'un MCD valider et préciser des règles qui s'appliqueront à la future base de données (d'après le MCD en image ci-dessus) :

- Une fiche est remplie par un et un seul chargé d'appui.
- Un contact peut avoir différentes qualités (dans ce modèle « utilisateur, salarié, porteur de projet ») en fonction des fiches (18).

¹⁶ Une méthode informatique dédiée à la modélisation qui analyse la structure à informatiser en termes de systèmes. Le gros avantage de cette méthode est qu'elle permet de cadrer le projet informatique et de « discuter » en se comprenant entre utilisateurs et informaticiens.

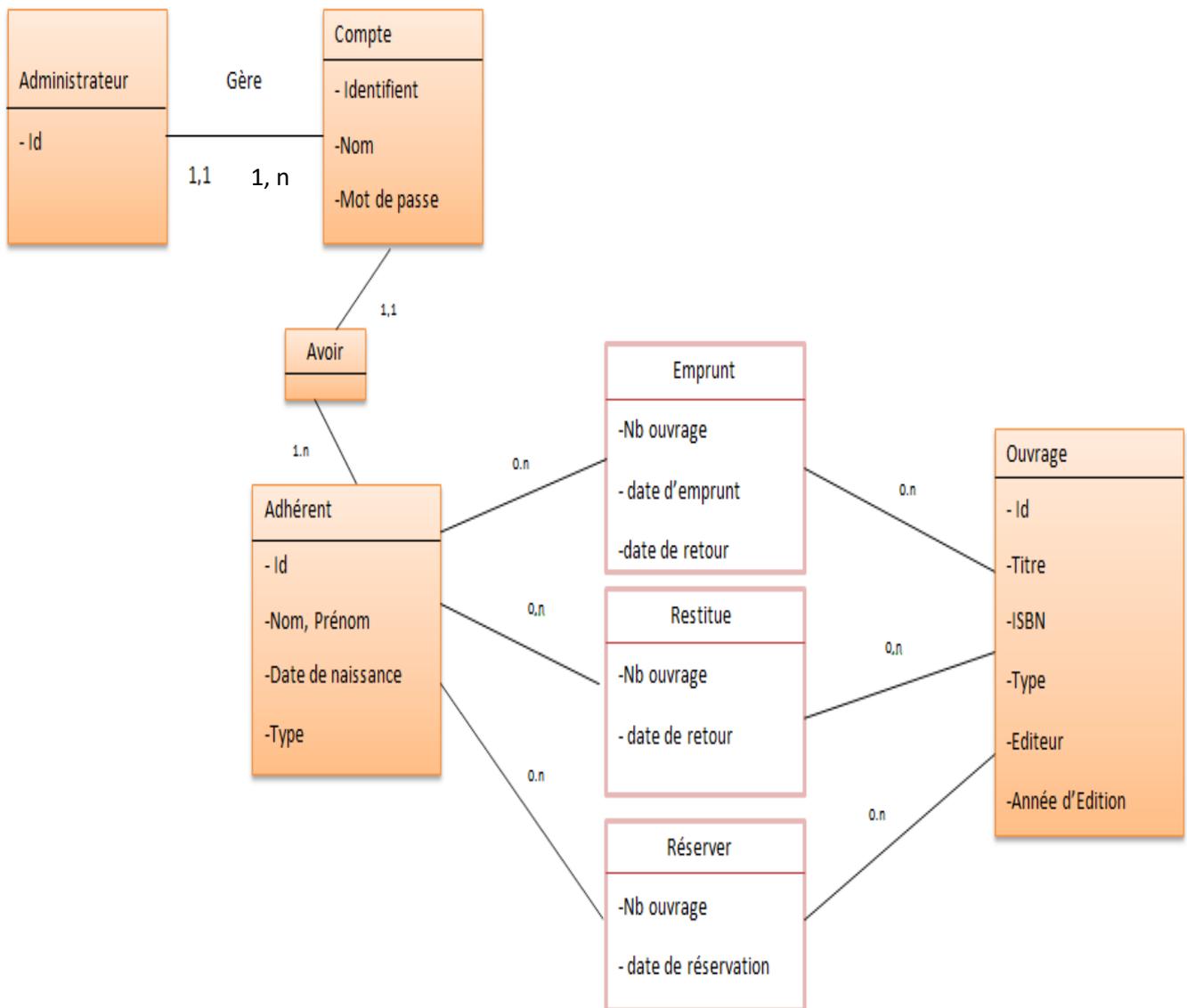


Figure 3.15 : Modèle conceptuel de données.

3.7.2 Diagramme entité association :

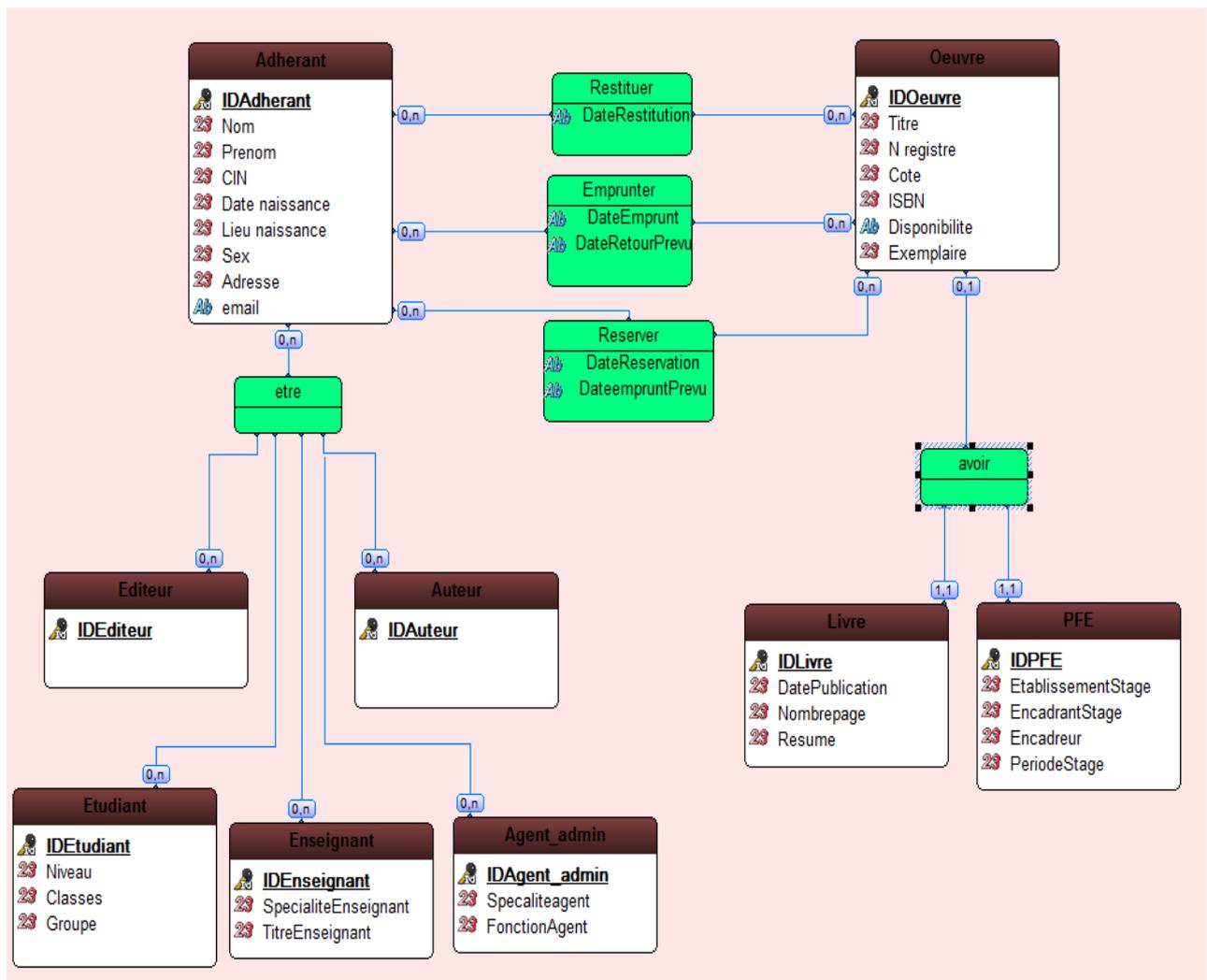


Figure 3.16 : Diagramme entité association.

3.7.3 Passage au modèle relationnel :

Le modèle relationnel est le modèle logique de donnée qui correspond à l’organisation des données dans les bases de données relationnelles. Un modèle relationnel est composé de relations, encore appelée table. Ces tables sont décrites par des attributs aux champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses. L’identifiant d’une relation est composé d’un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée (19).

Conclusion

Tout au long de ce chapitre nous avons présenté notre cas d'études dans la première partie, nous avons décrit la problématique, à laquelle notre projet doit répondre, ce qui nous a permis d'énumérer les différents besoins fonctionnels et non fonctionnels de l'application. Il a été d'une importance cruciale surtout pour la définition, la compréhension et la manipulation de différentes tâches portant chaque rubrique de l'application. Aussi dans la deuxième partie nous avons mené une conception détaillée de notre plateforme selon une approche d'objet, nous avons recensé la description graphique des cas d'utilisation, en réalisant des diagrammes de séquence décrivant les scénarios nominaux des cas d'utilisations essentiels. En fin, nous avons réalisé la conception appropriée à notre application selon les concepts de base de diagramme de classe ainsi que les règles de modélisation. Ensuite, nous avons recensé les règles de passage de diagramme de classe vers le modèle relationnel qui nous permet d'avoir le schéma de la base de données de l'application à réaliser. Cela fait la base telle qu'on va garantir la fiabilité et l'efficacité de la face de réalisation des applications qui sera le thème du chapitre suivant.

Chapitre 4 : La mise en œuvre

Introduction

Dans ce chapitre, nous allons faire des choix sur la programmation. Nous sommes arrivés pratiquement à la fin du processus de développement. Nous commençons par la présentation de l'environnement de développement, les outils et langages de programmation, ainsi que les services web utilisés dans notre application. Enfin, nous terminons par la présentation des réalisations effectuées au cours de ce projet.

4 Mise en œuvre

4.1 Environnement de développement

4.1.1 Android Studio :

Android Studio est l'environnement de développement spécialement utilisé pour Android. Pour l'installer : <https://developer.android.com/sdk/installing/studio.html>



➤ **SDK Android :**

Le SDK Android signifie "Software développement Kit" ou kit de développement contient tous les outils nécessaires pour créer une application Android. Il est disponible pour Windows, Mac, OS et Linux Chaque version du SDK contient :

- L'**AAPT** (Android Asset Packaging Tool):

Cet outil sert à créer et à analyser des fichiers, ces fichiers contiennent le programme de l'application créée.

- L'**ADB** (Android Debug Bridge) :

Le but de cet outil est d'établir des connexions avec un téléphone Android ou un émulateur en facilitant l'accès à leur Shell. Ceci afin de parcourir son contenu ou d'exécuter des commandes. Cet outil sert aussi à transférer une application ou des fichiers sur un téléphone Android ou un émulateur.

- Le **DDMS** (Dalvik Debug Monitor Service):

Cet outil est utilisé pour le débogage de l'application et a plusieurs fonctionnalités :

- Faire des captures d'écrans.
- Voir les threads en cours d'exécution.
- Voir le Logcat.
- Connaître la liste des processus en cours d'exécution sur l'appareil.
- Simuler l'envoi de messages et d'appels.
- Simuler une localisation, etc.

Le SDK contient aussi un système permettant de créer et de gérer les émulateurs ainsi que la documentation pour chaque version d'Android et des exemples pour les différentes API (21).

4.1.2 WampServer :

WAMP est l'**abrégié de Windows, Apache, MySQL, PHP**. Il est dérivé de LAMP, qui signifie Linux, Apache, MySQL et PHP. Comme son nom l'indique, alors que **LAMP est utilisé sur des serveurs Linux, WAMP est utilisé sur des serveurs Windows**. WordPress n'étant généralement pas installé sur les serveurs Windows, WAMP est devenu populaire parmi les développeurs en tant que méthode d'installation de WordPress sur leurs ordinateurs personnels.



Il peut être nécessaire, plutôt que d'installer votre site WordPress sur votre hébergement Web, de l'installer dans un premier temps sur votre ordinateur. Cependant, le PHP, par défaut ne peut être affiché sans logiciel adéquat. Wamp est un ensemble de service permettant d'afficher votre site en local (sans être connecté à internet).

De ce fait, WAMP facilite le travail des développeurs. De plus, il est possible de tester de nouvelles fonctionnalités et nouveaux plugins sans risque. Cependant, une fois le site Web créé, il est nécessaire de choisir un service d'hébergement Web afin de mettre en ligne ce dernier (23).

4.1.3 PhpMyAdmin :

PhpMyAdmin est un logiciel libre écrit en PHP qui a pour mission de s'occuper de l'administration d'un serveur de base de données MySQL ou MariaDB. Il est possible d'utiliser phpMyAdmin pour réaliser la plupart des tâches d'administration, ceci incluant la création de base de données, l'exécution de demandes, et l'ajout de comptes utilisateur.

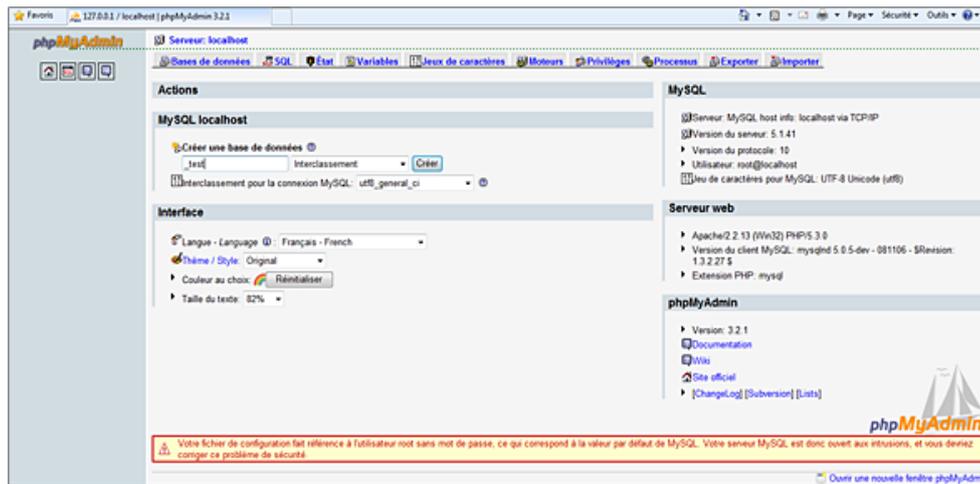


Figure 4.1 création d'une base de données sur PhpMyAdmin.

Beaucoup de personnes ont des difficultés à comprendre le concept de gestion des utilisateurs dans phpMyAdmin. Quand un utilisateur se connecte à phpMyAdmin, son nom d'utilisateur et mot de passe sont passés directement à MySQL. PhpMyAdmin ne gère aucun compte utilisateur lui-même (à part permettre de manipuler les informations des comptes utilisateurs de MySQL); tous les utilisateurs doivent être des utilisateurs MySQL valides (24).

4.1.4 Apache :

Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde. Le nom officiel est Serveur Apache HTTP et il est maintenu et développé par Apache Software Foundation.



Il permet aux propriétaires de sites web de servir du contenu sur le web, d'où le nom « serveur web ». C'est l'un des serveurs web les plus anciens et les plus fiables avec une première version

sortie il y a plus de 20 ans, en 1995. Quand quelqu'un souhaite visiter un site web, il saisit un nom de domaine dans la barre d'adresse de son navigateur. Puis le serveur web fournit les fichiers demandés en agissant comme un livreur virtuel.

Chez Hostinger¹⁷, infrastructure d'**hébergement web** utilise Apache en parallèle avec NGINX qui est un autre serveur web populaire. Cette configuration particulière nous permet d'obtenir le meilleur des deux. Cela améliore bien les performances du serveur en compensant les faiblesses d'un logiciel avec la force de l'autre (25).

➤ Fonctionnement du serveur web Apache

Apache n'est pas un serveur physique mais plutôt un logiciel qui s'exécute sur un serveur. Son travail consiste à établir une connexion entre un serveur et les navigateurs des visiteurs du site web (Firefox, Google Chrome, Safari, etc.) tout en délivrant des fichiers entre eux (structure client-serveur). Apache est un logiciel multiplateforme, il fonctionne donc à la fois sur les serveurs Unix et Windows.

Lorsqu'un visiteur souhaite charger une page sur un site web, son navigateur envoie une requête à votre serveur et Apache renvoie une réponse avec tous les fichiers demandés (texte, images, etc.). Le serveur et le client communiquent via le protocole http et Apache est responsable de la communication fluide et sécurisée entre les deux machines.

Apache est hautement personnalisable, car il a une structure basée sur des modules. Les modules permettent aux administrateurs de serveur d'activer ou de désactiver des fonctionnalités supplémentaires. Des modules pour la sécurité, la mise en cache, la réécriture d'URL, l'authentification par mot de passe et encore plus (25).

4.2 Langage de développement

4.2.1 JAVA

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux

¹⁷ Hostinger.fr est l'une des meilleures sociétés d'hébergement web. Elle est la marque mondiale la plus en expansion dans cette industrie.

superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts.

La dernière version de Java comprend d'importantes améliorations en matière de performances, de stabilité et de sécurité pour les applications Java exécutées sur votre ordinateur. L'installation de cette mise à jour gratuite garantit que les applications Java sont toujours exécutées de manière sécurisée et efficace (26).

4.2.2 PHP

PHP : Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP. PHP est un langage impératif orienté objet. Il s'agit d'un langage de script interprété côté serveur.

Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc (27).

➤ Fonctionnement de PHP

Même s'il peut être utilisé en ligne de commande, PHP est principalement associé à un serveur Web utilisant le protocole HTTP dans le cadre d'une architecture client/serveur.

Un serveur Web en architecture trois tiers est composé d'un système d'exploitation, un serveur HTTP, un langage serveur et enfin un système de gestion de base de données (SGBD), cela constituant une plate-forme.

Dans le cas de PHP comme langage serveur, les combinaisons les plus courantes sont celles d'une plateforme LAMP (pour Linux Apache MySQL PHP) et WAMP (Windows Apache MySQL PHP).

Une plate-forme WAMP s'installe généralement par le biais d'un seul logiciel qui intègre Apache, MySQL et PHP, par exemple EasyPHP et WampServer. Il existe le même type de logiciels pour les plates-formes MAMP (Mac OS Apache MySQL PHP), à l'exemple du logiciel MAMP (Figure4.1) (27).

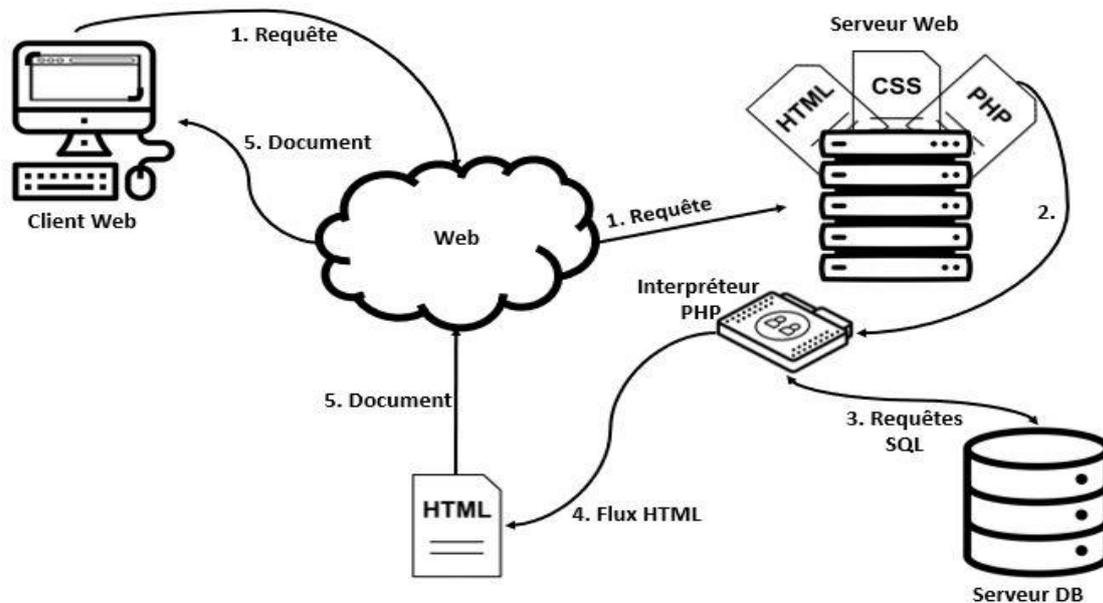


Figure 4.2 : Fonctionnement du PHP.

1. Le client, le plus souvent un navigateur web, envoie une requête HTTP au travers d'une URL vers un serveur.
2. Le serveur identifie la page à renvoyer :
 - S'il s'agit d'un document pouvant être envoyé immédiatement, il l'envoie.
 - S'il s'agit d'un document nécessitant une interprétation, comme le PHP, il va d'abord le traiter.
3. Le document PHP est alors interprété en faisant éventuellement appel à des données externes provenant d'un serveur de bases de données. L'interprétation de la page PHP produit un document, souvent une page HTML, fourni au serveur.
4. L'interprétation de la page PHP produit une page HTML de résultat fournie au serveur.
5. Le serveur Web renvoie ce document, cette page, au client pour affichage.

4.2.3 SQL

Le langage SQL (*Structured Query Language*) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est aujourd'hui supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel que *Access* ou par les produits plus professionnels tels qu'Oracle¹⁸. Il a fait l'objet de plusieurs normes ANSI/ISO dont la plus répandue aujourd'hui est la norme SQL2 qui a été définie en 1992.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution. Le langage SQL propose un langage de requêtes ensembliste et assertionnel. Néanmoins, le langage SQL ne possède pas la puissance d'un langage de programmation : entrées/sorties, instructions conditionnelles, boucles et affectations. Pour certains traitements il est donc nécessaire de coupler le langage SQL avec un langage de programmation plus complet.

De manière synthétique, on peut dire que SQL est un langage relationnel, il manipule donc des tables (*i.e.* des relations, c'est-à-dire des ensembles) par l'intermédiaire de requêtes qui produisent également des tables (28).

4.2.4 XML

Le XML, pour Extensible Markup Language, désigne un langage informatique utilisé, entre autres, dans la conception des sites Web et pour faciliter les échanges d'informations sur Internet. Ce langage de description a pour mission de formaliser des données textuelles. Il s'agit, en quelque sorte, d'une version améliorée du langage HTML avec la création illimitée de nouvelles balises.

Comme le langage HTML, le XML permet la mise en forme de documents via l'utilisation de balises. Développé et standardisé par le World Wide Web Consortium à la fin des années 1990, il répondait à l'objectif de définition d'un langage simple, facile à mettre en application.

¹⁸ Un système de gestion de base de données relationnelle (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné entre autres, de Bob Miner et Ed Oates.

Le XML se classe dans la catégorie des langages de description, il n'est ni un langage de programmation, ni un langage de requêtes. Il est donc naturellement utilisé pour décrire des données en s'appuyant sur des balises et des règles personnalisables.

Avec la généralisation de la connexion HTTP, le XML a encore gagné en popularité en devenant une solution habituelle pour créer un nouveau protocole (29).

➤ Fonctionnement d'un fichier XML

La syntaxe du XML repose sur une chaîne de caractères structurée en deux niveaux :

- Un pour le lecteur humain et un autre pour la machine.

Un document XML prend la forme d'un arbre, dont le tronc sert de support à différents types d'éléments appelés "nœuds", comme des textes, attributs, commentaires, éléments, etc. Les débutants sont parfois découragés parce que la moindre erreur dans la forme fait sortir le document du format XML et casse toute la chaîne de traitement (29).

4.3 Les services Web

4.3.1 Définition

Les services Web sont des applications Web standard (XML, SOAP, HTTP etc.) ouvertes qui interagissent avec d'autres applications Web dans le but d'échanger des données. Les services Web peuvent convertir vos applications existantes en applications Web.

Il avait des différents livres et différentes organisations fournissent des définitions différentes aux services Web :

- Les services Web sont des applications autonomes, modulaires, distribuées et dynamiques qui peuvent être décrites, publiées, localisées ou invoquées sur le réseau pour créer des produits, des processus et des chaînes d'approvisionnement. Ces applications peuvent être locales, distribuées ou basées sur le Web. Les services Web sont construits selon des standards ouverts tels que TCP / IP, HTTP, Java, HTML et XML.

- Un service Web est un logiciel qui se rend disponible sur Internet et utilise un système de messagerie XML normalisé. XML est utilisé pour coder toutes les communications vers un service Web. Par exemple, un client invoque un service Web en envoyant un message XML, puis il attend pour une réponse XML correspondante. Comme toutes les communications sont en XML, les services Web ne sont liés à aucun système d'exploitation ou langage de programmation (30).

4.4 Description des Application

4.4.1 Interfaces de l'application Android

Dans ce qui suit, nous allons présenter les interfaces de notre application Android (Figure 4.2)

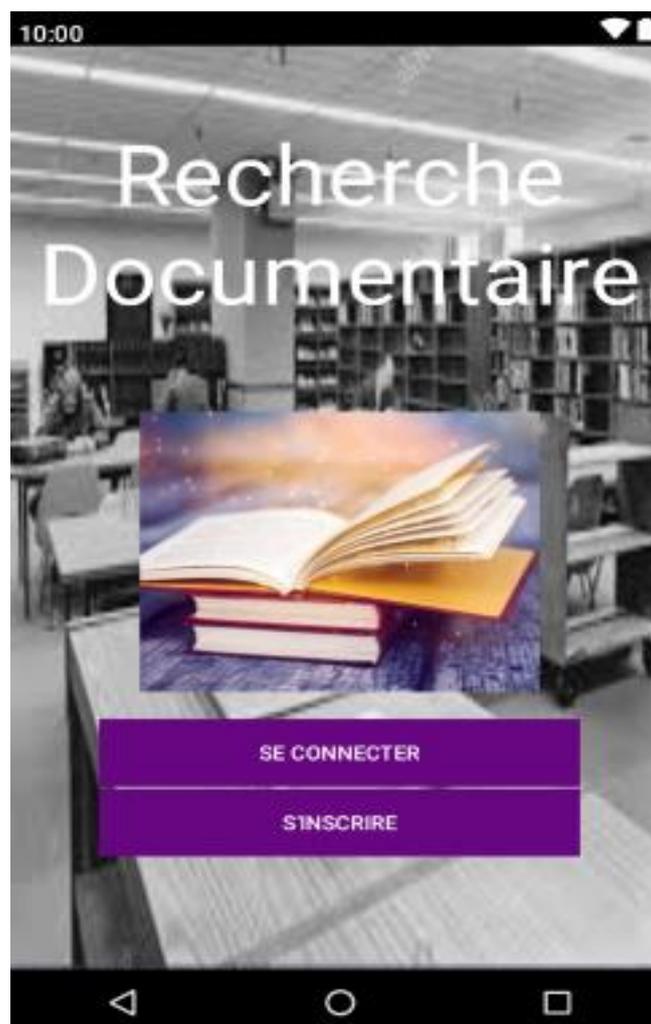


Figure 4.3 : Interface d'accueil de l'application.

4.4.1.1 Interface "authentification d'utilisateur"

Presque toute manipulation de logiciel ou d'application (web, mobile) débute par une authentification qui sera le cas dans cette plateforme. Chaque utilisateur doit être en mesure de s'authentifier en entrant son login et son mot de passe.

Dans le cas échant, chaque nouvel utilisateur doit enregistrer sa candidature par l'interface d'enregistrement pour bénéficier des services (Figure 4.3).



Figure 4.4: Interface authentification d'utilisateur.

4.4.1.2 Interface d'identification

Chaque nouvel utilisateur doit d'identifier pour accéder a sa propre interface d'inscription (Figure 4.4).

4.4.1.3 Interface d'Inscription « Etudiant »

Sur cette interface chaque nouvel utilisateur doit enregistrer sa candidature alors il doit remplir le formulaire d'inscription. On a choisi comme exemple l'inscription pour étudiant (Figure 4.5).

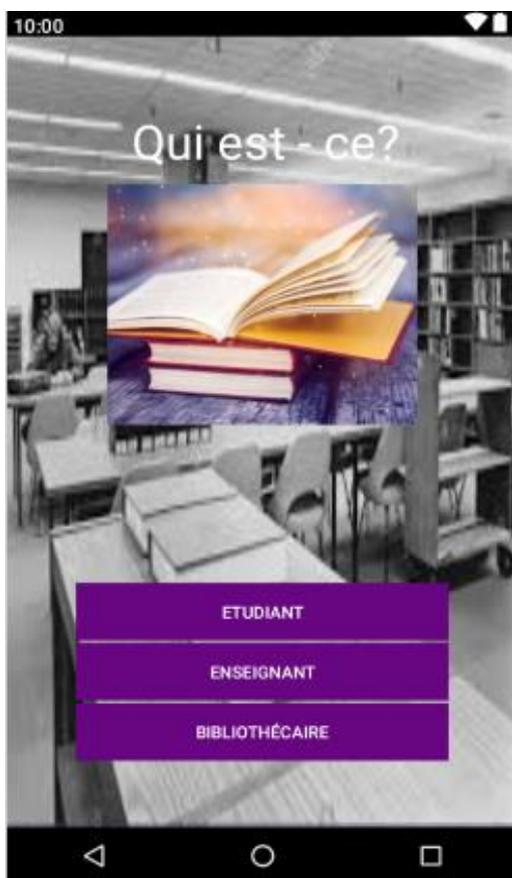


Figure 4.5: Interface identification.

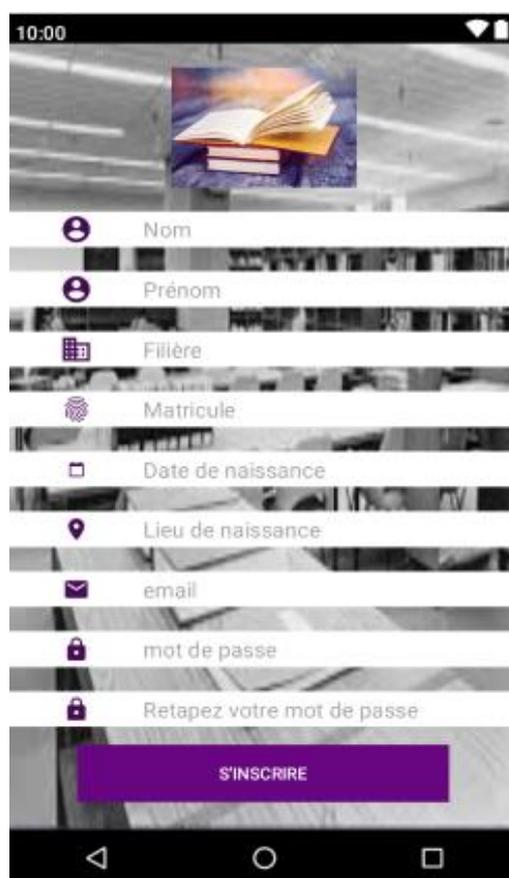


Figure 4.6: Interface d'inscription pour étudiant.

4.4.1.4 Interface recherche de document « PFE »

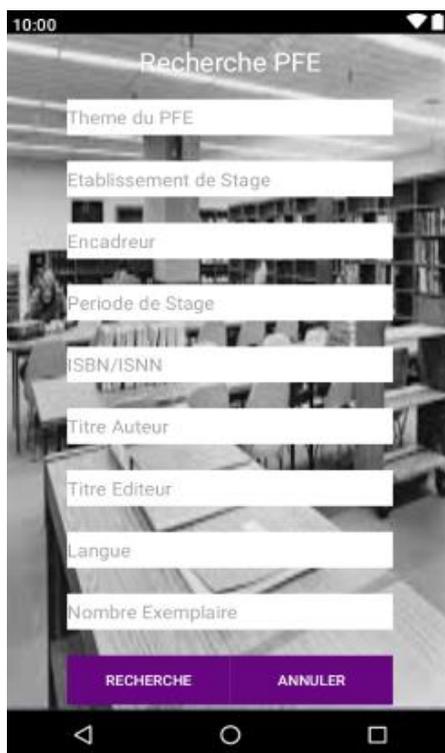


Figure 4.7: Interface recherche de document « PFE ».

4.4.1.5 Interface recherche de document « PFC »

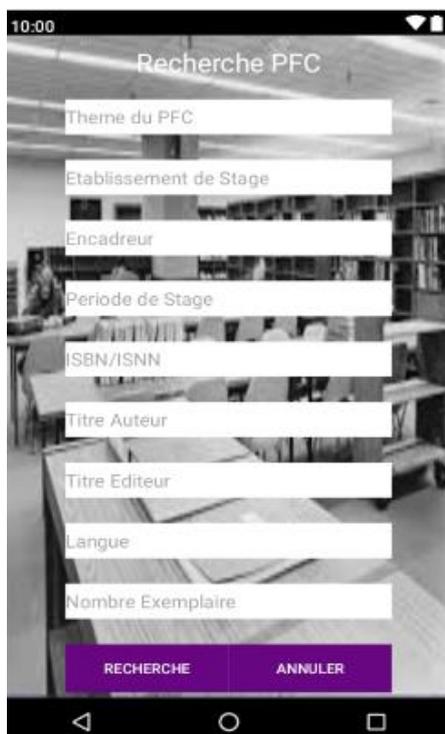


Figure 4.8: Interface recherche de document « PFC ».

4.4.1.6 Interface Catalogue de bibliothèque

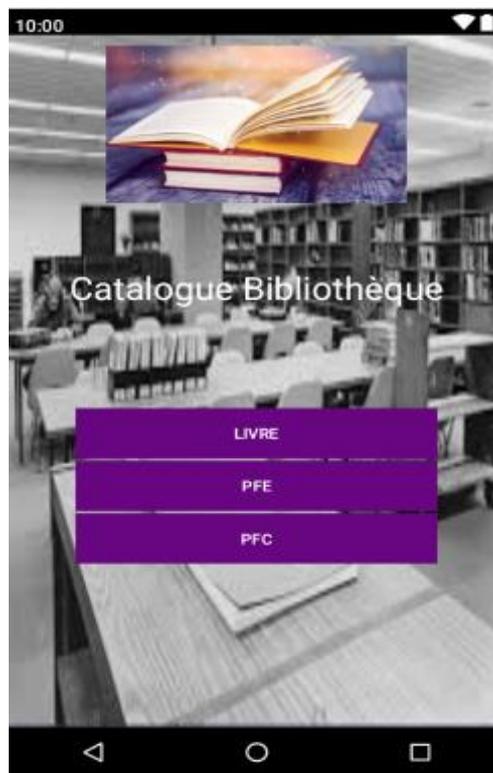


Figure 4.9: Interface Catalogue de bibliothèque.

4.4.1.7 Interface de recherche de documents « Livre »

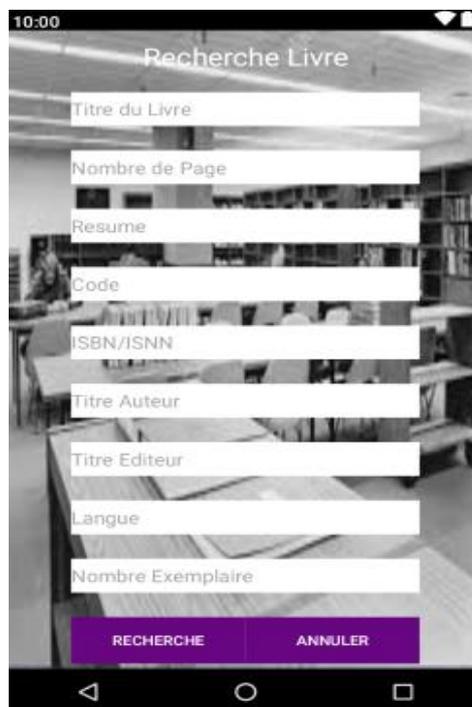
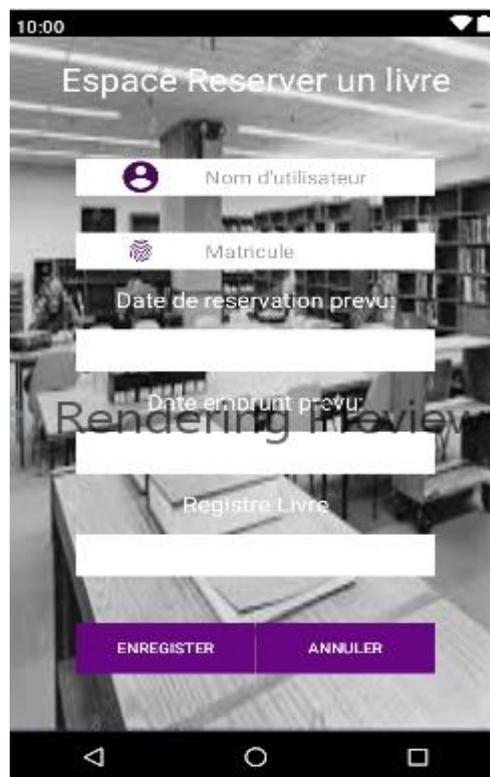


Figure 4.10: Interface de recherche de documents « Livre ».

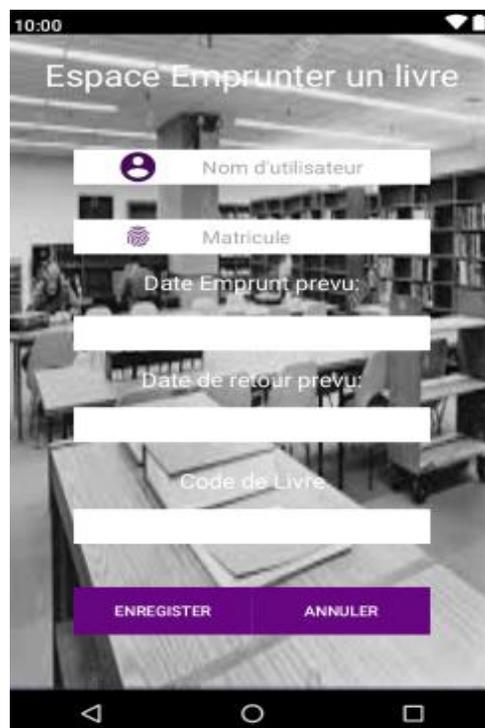
4.4.1.8 Interface réserver un livre



The screenshot shows a mobile application interface for reserving a book. The title at the top is "Espace Reserver un livre". Below the title, there are four input fields: "Nom d'utilisateur" (with a person icon), "Matricule" (with a book icon), "Date de reservation prévu:" (with a calendar icon), and "Date emprunt prévu:" (with a calendar icon). Below these fields is a "Registre Livre" label and another input field. At the bottom, there are two purple buttons: "ENREGISTER" and "ANNULER". The background is a grayscale image of a library interior.

Figure 4.11:Interface réserver un livre.

4.4.1.9 Interface emprunter un livre



The screenshot shows a mobile application interface for borrowing a book. The title at the top is "Espace Emprunter un livre". Below the title, there are four input fields: "Nom d'utilisateur" (with a person icon), "Matricule" (with a book icon), "Date Emprunt prévu:" (with a calendar icon), and "Date de retour prévu:" (with a calendar icon). Below these fields is a "Code de Livre" label and another input field. At the bottom, there are two purple buttons: "ENREGISTER" and "ANNULER". The background is a grayscale image of a library interior.

Figure 4.12:Interface emprunter un livre

4.4.1.10 Interface d'inscription « Enseignant ».



Figure 4.13:Interface d'inscription « Enseignant ».

A.1 Table utilisateur (Agent)

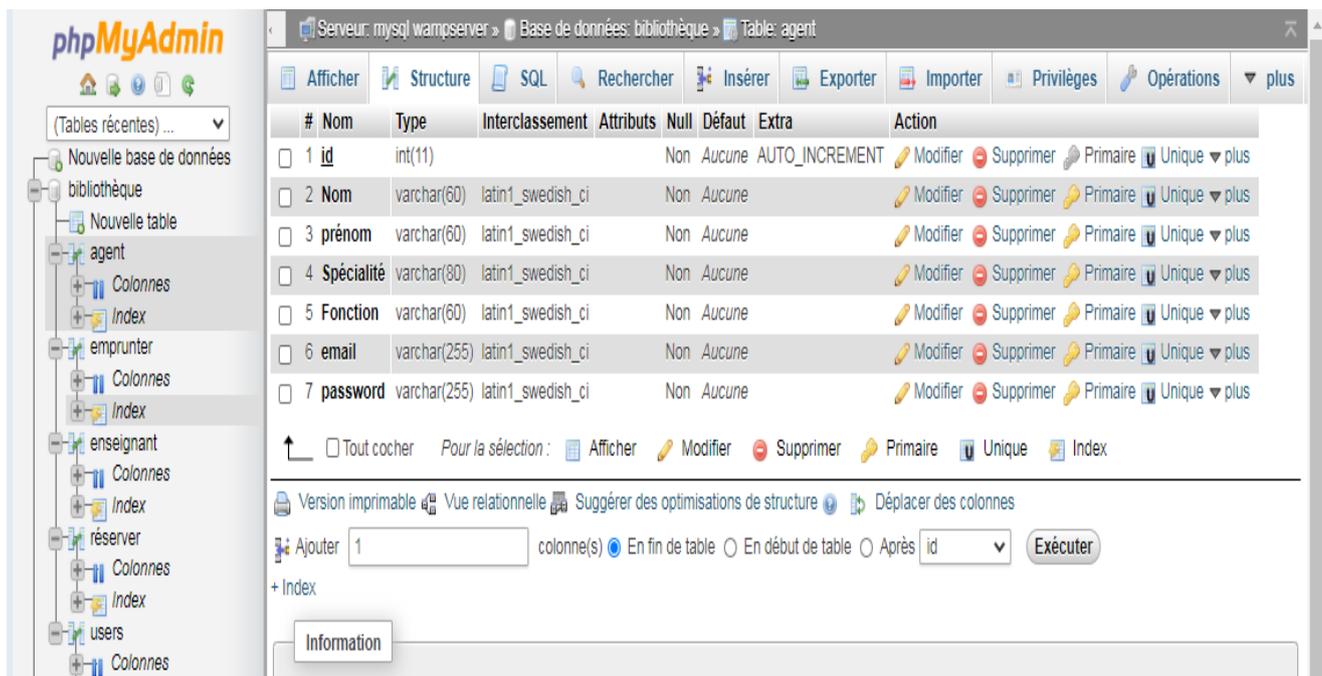


Figure 4.14 : table de données Agent.

A.2 Table de données pour Emprunt

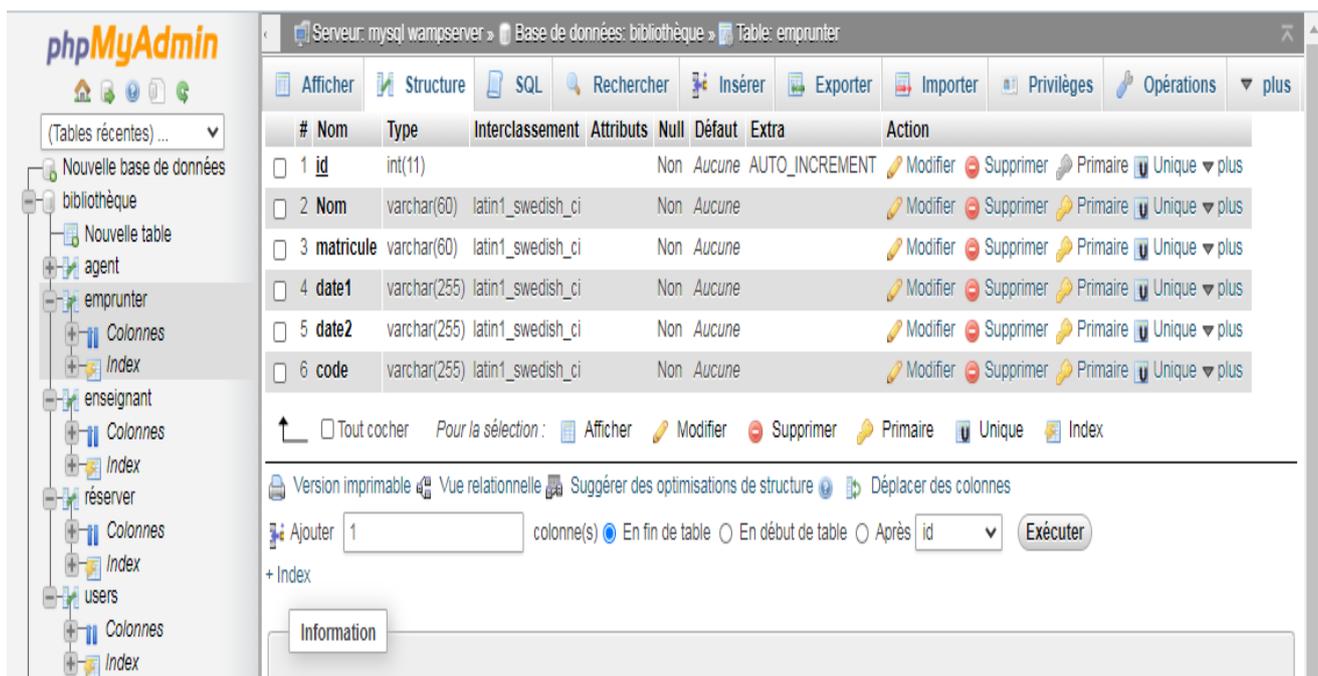


Figure 4.15 : table de données pour Emprunt.

Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application.

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et les langages de développement associés à notre système. En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.

CONCLUSION GENERALE

Dans ce mémoire, nous avons présenté les différentes étapes de la conception et la réalisation d'une application Android pour la recherche documentaire dans la bibliothèque de l'Université Abderrahmane Mira de Bejaia.

Afin de satisfaire les besoins des utilisateurs, nous avons commencé par la conception en utilisant le formalisme de Merise, ensuite la concrétisation de l'application sous l'environnement de programmation Android Studio qui est un environnement de développement pour des applications mobiles.

Les différentes tâches fixées ont été réalisées à partir de plusieurs hypothèses. Nous avons modélisé les opérations importantes en respectant les contraintes fixées et nous avons fait un bon nombre de retour en arrière au niveau des diagrammes afin de les conformer les uns les autres et d'avoir une vision au plus juste possible. Les diagrammes de séquences, cas d'utilisations, diagramme de classes font partie de notre analyse. Nous regrettons simplement de n'avoir pu faire de diagramme d'activité par manque de temps.

Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'améliorer nos connaissances et mes compétences dans le domaine de la programmation.

Cependant des perspectives d'améliorations de mon application restent envisageables, parmi eux, une meilleure gestion des emprunts, une meilleure gestion des livres (les livres ne seront plus perdus ou non rendu).

A la fin de ce modeste travail nous espérons avoir atteint la plupart des objectifs et fonctionnalités mentionnés dans la partie théorique, toutefois, d'autres améliorations peuvent être apportées concernant la partie ergonomique de l'application qui peut avoir des interfaces plus conviviales et intuitives.

Sachant qu'aucun travail n'est parfait, espérons bien avoir atteint le maximum des objectifs déjà fixés et pour le reste seront pour une future étude avec une amélioration de ce qui est déjà fait.

Bibliographie

1. <http://www.univ-bejaia.DZ/presentation-universite>.
2. L'informatisation des bibliothèques universitaires algériennes : état des lieux et perspectives d'avenir, *par* Hakim BENOUMELGHAR-Université d'Alger - Magister en bibliothéconomie et sciences documentaires 2002.
3. http://www.univ-bejaia.dz/Fac_Technologie/bibliotheque/mission-et-activites.
4. <https://www.enseignementsup-recherche.gouv.fr/cid20545/les-bibliotheques-universitaires>.
5. L'informatisation des bibliothèques universitaires algériennes, *par* Hakim BENOUMELGHAR-Université d'Alger - Magister en bibliothéconomie et sciences documentaires 2002.
6. http://www.univ-bejaia.dz/Fac_Technologie/bibliotheque.
7. Définitions Marketing » L'encyclopédie illustrée du marketing. Définition-Applications-Mobile, Écrit par B. Bathelot, modifié le 08/03/2017.
8. <https://generationmobiles.net/autre/les-differe-nts-types-dapps-mobiles-leurs-avantages-et- inconvenients/consulter> le 07/06/2020.
9. <https://www.nomai.fr/guide-achat-smartphone/systeme-os,consulter> le 18/07/2020.
10. lexpansion.lexpress.fr/high-tech/systeme-android-google_1491838.html, consulter le 25/07/2020.
11. <https://www.supinfo.com/articles/single/6015-evolution-systeme-exploitation-mobile-android>.

Bibliographie

12. Définitions Marketing » L'encyclopédie illustrée du marketing. Définition-Applications-Mobile, Écrit par B. Bathelot, modifié le 08/03/2017.
13. <http://developer.android.com/reference/android/app/Activity.html>/consulter le 01/05/2020.
14. F. Juliard UML Unified Method Language, Journal Université de Bretagne Sud UFR SSI-IUP Vannes, 2001- 2002 : <https://wikimemoires.net/2019/12/les-processus-unifies-et-uml-presentation-duml>, Wiki Mémoires © 2011 - 2020.
15. Joseph Gabay et David Gabay, Mise en œuvre guidée avec études de cas, édition Dunod, Paris 2008. 9 Pascal ROQUES, UML 2 par la pratique étude de cas et exercices corrigés, ÉDITIONS EYROLLES, Septembre 2006, p16 : <https://wikimemoires.net/2019/12/les-processus-unifies-et-uml-presentation-duml/>.
16. Modélisation Conceptuelle de Données ; Une Démarche Pragmatique-Édition : *Patrick - Bergougnoux Pat* - 320 pages, 15 octobre 2019. ISBN10: B07YTDY1LB - ISBN13 : 9791069930872_ <https://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet#L1-4>.
17. <https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml>-consulter le 23/07/2020.
18. Merise et MCD : à propos de l'entité-type DATE, par fsmrel- <https://www.base-de-donnees.com/mcd/>.
19. R. P. Franck Vallée, UML 2 en action, de l'analyse des besoins à la conception. 4ème édition, EYROLLES, 2007'.
21. <https://www.supinfo.com/articles/single/2569-presentation-android>/consulter le 22/06/2020.
23. [joomla-3-le-livre-pour-tous-v5.pdf](https://blog.lws-hosting.com/divers/quest-ce-que-wamp-guide-convivial-pour-les-debutants) _<https://blog.lws-hosting.com/divers/quest-ce-que-wamp-guide-convivial-pour-les-debutants>.
24. © Copyright 2012 - 2020, The phpMyAdmin devel team Revision 93746f07 : <https://docs.phpmyadmin.net/fr/latest/intro.html#a-word-about-users>.

25. <https://www.hostinger.fr/tutoriels/quest-ce-quapache-serveur-web-apache/consulter> 23/01/2020.
26. https://www.java.com/fr/download/faq/whatis_java.xml-consulte le 05/07/2020.
27. <https://www.codingame.com/playgrounds/34845/le-php---les-bases-du-langage/introduction-consulte> le 19/07/2020.
28. Oracle 12c - Administrer une base de données Exercices et corrigés *de Claire Noirault*/[https://laurent-audibert.developpez.com/Cours-BD/?page=langage sql#L4-1-1a](https://laurent-audibert.developpez.com/Cours-BD/?page=langage%20sql#L4-1-1a).
29. [https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction/mis a jour](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction/mis-a-jour) le 08/01/2019.
30. <https://www.supinfo.com/articles/single/4350-comprendre-web-services>.
31. Big Data & Streaming_Le Traitement Streaming et temps réel des données en Big Data *de Juvénal Chokogoue* : <https://www.lebigdata.fr/base-de-donnees>.

ANNENES

A.I Base de données

Une base de données (que nous nommerons BDD par commodité) est une collection d'informations organisées afin d'être facilement consultables, gérables et mises à jour. Au sein d'une data base, les données sont organisées en lignes, colonnes et tableaux. Elles sont indexées afin de pouvoir facilement trouver les informations recherchées à l'aide d'un logiciel informatique. Chaque fois que de nouvelles informations sont ajoutées, les données sont mises à jour, et éventuellement supprimées.

Elles se chargent elles-mêmes de créer, de mettre à jour ou de supprimer des données. Elles effectuent également des recherches parmi les données qu'elles contiennent sur demande de l'utilisateur, et de lancer des applications à partir des données.

Les bases de données sont utilisées par de nombreuses entreprises dans toutes les industries. Elles sont notamment utilisées par les compagnies aériennes pour gérer les réservations. Elles sont utilisées pour la gestion de production. Pour les enregistrements médicaux dans les hôpitaux, ou encore pour les enregistrements légaux dans les compagnies d'assurances. Les bases de données les plus larges sont généralement utilisées par les agences gouvernementales, les grandes entreprises ou les universités (31).

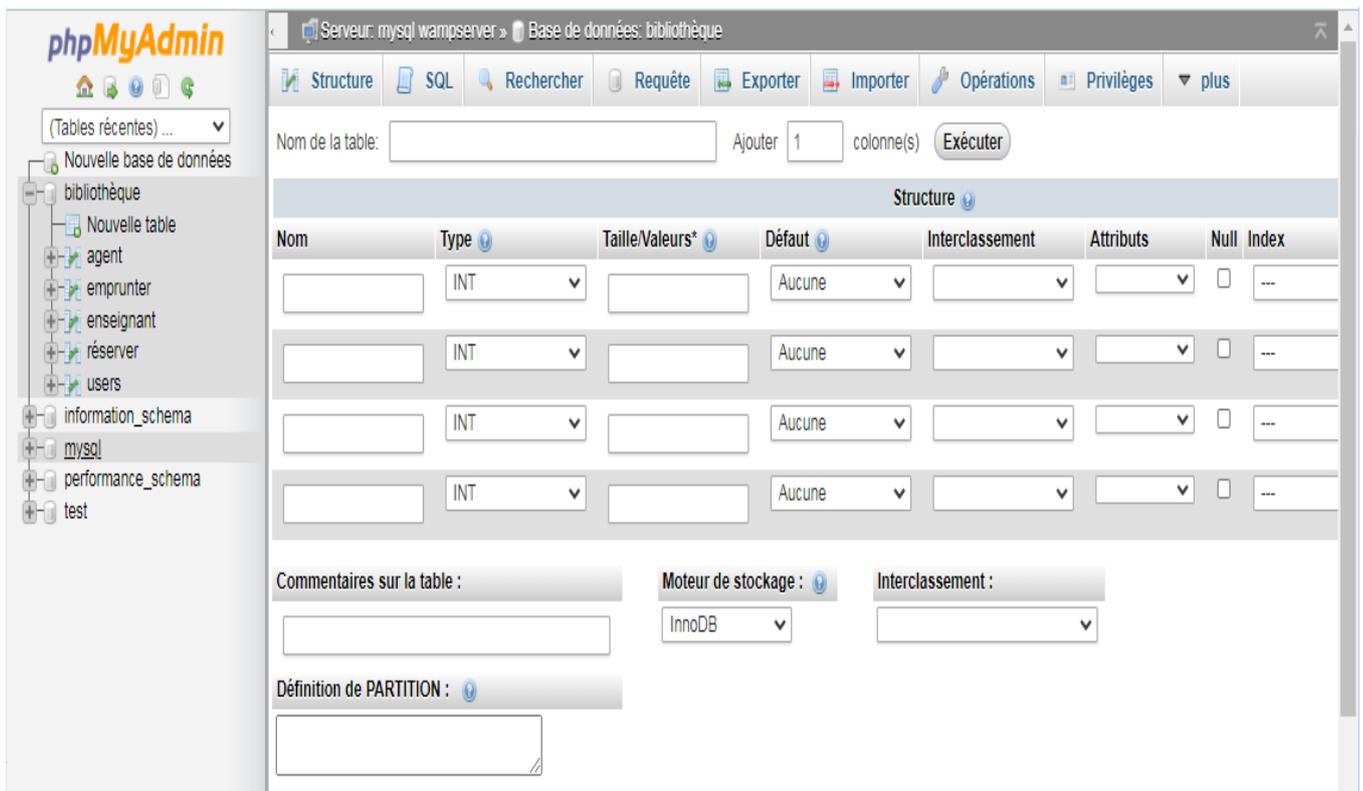


Figure 5.1 La Base de données.

A.1 Table utilisateur (Etudiant)

Seigneur: mysql wampserver > Base de données: bibliothèque > Table: users

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire plus
2	Nom	varchar(60)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
3	prénom	varchar(60)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
4	filière	varchar(60)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
5	matricule	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
6	date de naissance	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
7	lieu de naissance	varchar(60)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
8	email	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
9	password	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus

Tout cocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique Index

Ajouter 1 colonne(s) En fin de table En début de table Après id Exécuter

Figure 5.2 table de données Etudiants.

A.2 Table utilisateur (Enseignant)

Seigneur: mysql wampserver > Base de données: bibliothèque > Table: enseignant

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire plus
2	Nom	varchar(60)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
3	prénom	varchar(60)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
4	Spécialité	varchar(80)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
5	Titre Enseignant	varchar(60)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
6	email	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus
7	password	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire plus

Tout cocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique Index

Ajouter 1 colonne(s) En fin de table En début de table Après id Exécuter

Figure 5.3 table de données Enseignant.

➤ Le code pour l'Interface de l'application Android

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:background="@drawable/bibl">

            <TextView
                android:id="@+id/et_inscription"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center_horizontal"
                android:layout_marginTop="50dp"
                android:text="Recherche Documentaire"
                android:textSize="60dp"
                android:textColor="@color/cardview_light_background"
                android:visibility="visible" />

            <ImageView
                android:id="@+id/imageView2"
                android:layout_width="250dp"
                android:layout_height="200dp"
                android:layout_marginTop="60dp"
                android:scaleType="centerCrop"
                android:layout_gravity="center_horizontal"
                android:visibility="visible"
                app:srcCompat="@drawable/livre" />

        </LinearLayout>

    <RelativeLayout
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/btn_login"
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="400dp"
            android:background="#680681"
            android:textColor="ffffff"
            android:text="Se connecter"/>

        <Button
            android:id="@+id/btn_register"
            android:layout_width="300dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="450dp"
            android:background="#680681"
            android:textColor="ffffff"
            android:text="S'inscrire"/>

    </RelativeLayout>

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Figure 5.4 code de l'interface de l'application en Xml.

➤ Programmation en java

```
package com.example.espacemembre;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import Response.Listener;

public class MainActivity extends AppCompatActivity {

    private Button btn_login, btn_register;
    private SessionManager sessionManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        sessionManager = new SessionManager( context: this);
        if(sessionManager.islogged()){
            Intent intent= new Intent( packageContext: this, PremiereActivity.class);
            startActivity(intent);
            finish();
        }

        btn_register = (Button) findViewById(R.id.btn_register);
        btn_login = (Button) findViewById(R.id.btn_login);

        btn_register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), identificationActivity.class);
                startActivity(intent);
            }
        });

        btn_login.setOnClickListener((view) → {
            Intent intent = new Intent(getApplicationContext(),identificationActivity.class);
            startActivity(intent);
        });
    }
}
```

Figure 5.5 Programmation en java de l'application

➤ Le code pour l'Interface authentification d'utilisateur(login)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".loginActivity">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal"
            android:orientation="vertical"
            android:background="@drawable/bibl">

            <ImageView
                android:id="@+id/imageView2"
                android:layout_width="200dp"
                android:layout_height="100dp"
                android:layout_marginTop="20dp"
                android:layout_marginBottom="100dp"
                android:scaleType="centerCrop"
                android:visibility="visible"
                app:srcCompat="@drawable/livre" />
```

```
<EditText
    android:id="@+id/et_password"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginTop="20dp"
    android:layout_margin="20dp"
    android:drawableLeft="@drawable/ic_launcher_pass1"
    android:hint="mot de passe"
    android:imeOptions="actionNext"
    android:background="#ffffff"
    android:inputType="textPassword"
    android:singleLine="true" />
```

```
<EditText
    android:id="@+id/ed_password2"
    android:layout_width="fill_parent"
    android:layout_height="50dp"
    android:layout_marginTop="20dp"
    android:layout_margin="20dp"
    android:drawableLeft="@drawable/ic_launcher_pass1"
    android:background="#ffffff"
    android:hint="Retapez votre mot de passe"
    android:imeOptions="actionNext"
    android:inputType="textPassword"
    android:singleLine="true" />
```

```
<Button
    android:id="@+id/btn_login"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:background="#680681"
    android:textColor="#ffffff"
    android:text="Se connecter" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

</RelativeLayout>

<ProgressBar
    android:id="@+id/pb_loader"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="50dp"
    android:visibility="invisible" />

</LinearLayout>

</RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Figure 5.6 code de l'interface authentification en Xml.

➤ Programme de l'interface en java (authentification) :

```

package com.example.espacemembre;

import ...

public class LoginActivity<RequestQueue> extends AppCompatActivity {
    private EditText et_username, et_password, et_password2;
    private Button btn_send;
    private RequestQueue queue;
    private Myrequest request;
    private ProgressBar pb_loader;
    private Handler handler;
    private SessionManager sessionManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        // Récupération du message de l'inscription
        Intent intent = getIntent();
        if(intent.hasExtra( name: "REGISTER")){
            Toast.makeText( context: this, intent.getStringExtra( name: "REGISTER"), Toast.LENGTH_SHORT).show();
        }

        et_username= (EditText) findViewById(R.id.et_username);
        et_password= (EditText) findViewById(R.id.et_password);
        et_password2= (EditText) findViewById(R.id.et_password2);
        btn_send= (Button) findViewById(R.id.btn_send);
        pb_loader = (ProgressBar) findViewById(R.id.pb_loader);

        queue= (RequestQueue) volleysingleton.getInstance(this).getRequestQueue();
        request= new Myrequest( context: this, queue);
        handler = new Handler();
        sessionManager = new SessionManager( context: this);

        btn_send.setOnClickListener((OnClickListener) (view) -> {
            final String username = et_username.getText().toString().trim();
            final String password = et_password.getText().toString().trim();
            final String password2 = et_password2.getText().toString().trim();
            pb_loader.setVisibility(View.VISIBLE);
            if (username.length() > 0 && password.length() > 0 && password2.length() > 0) {
                handler.postDelayed((Runnable) () -> {

                    request.connection(username, password, password2, new LoginCallback() {
                        @Override
                        public void onSuccess(String id, String username) {
                            pb_loader.setVisibility(View.GONE);
                            sessionManager.insertUser(id,username);
                            Intent intent = new Intent(getApplicationContext(), ouvrage.class);
                            startActivity(intent);
                            finish();
                        }
                    })

                    @Override
                    public void onError(String message) {
                        pb_loader.setVisibility(View.GONE);
                        Toast.makeText(getApplicationContext(), message, Toast.LENGTH_SHORT).show();
                    }
                });
            }
            handler.postDelayed(
                },1000);
        }else{
            Toast.makeText(getApplicationContext(), text: "Veuillez remplir tous les champs",Toast.LENGTH_SHORT).show();
        }
    }
}

```

Figure 5.7 Programme de l'interface authentification en java.

➤ Programmation de l'interface en PHP (authentification)

```

File Edit Selection Find View Goto Tools Project Preferences Help
register.php x login.php x
1 <?php
2
3
4
5 $db= new PDO("mysql:host=localhost;dbname=bibliothèque","root","");
6 $results['error']= false;
7 $results['message']= [];
8
9 if(!empty($_POST)){
10
11     if(!empty($_POST['username']) && !empty($_POST['Password'])){
12
13         $username = $_POST['username'];
14         $password = $_POST['password'];
15
16         $sql = $db->prepare("SELECT* FROM users WHERE username = :username");
17         $sql->execute(["username"=> $username]);
18         $row = $sql->fetch(PDO::FETCH_OBJ);
19         if($row){
20             if(password_verify($password, $row->password)){
21                 $results["error"]= false;
22                 $results["id"]= $row->id;
23                 $results["username"]= $row->username;
24
25             }else{
26                 $results['error']= true;
27                 $results['message']= ['Nom utilisateur ou mot de passe incorrect'];
28             }
29
30         }else{
31             $results['error']= true;
32             $results['message']= ['Nom utilisateur ou mot de passe incorrect'];
33         }
34
35     }else{
36         $results['error']= true;
37         $results['message']= ['Veuillez remplir tous les champs'];
38     }
39
40
41     echo json_encode($results);
42 }
43
44
45 ?>
46

```

Figure 5.8 Programme de l'interface authentification en PHP.

➤ Programmation de l'interface en java (Etudiant)

```
package com.example.espacemembre;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import java.util.Calendar;
import java.util.Map;

import myrequest.Myrequest;
import myrequest.RegisterCallBack;

public class registerActivity<RequestQueue> extends AppCompatActivity {

    private Button btn_send;
    private EditText et_username, et_username2,et_email, et_password, et_password2,et_CIN, et_date, et_date2, et_Filière;
    DatePickerDialog.OnDateSetListener setListener;
    private ProgressBar pb_loader;
    private RequestQueue queue;
    private Myrequest request;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        btn_send = (Button) findViewById(R.id.btn_send);
        pb_loader= (ProgressBar) findViewById(R.id.pb_loader);
        et_username= findViewById(R.id.et_username);
        et_username2= findViewById(R.id.et_username2);
        et_email=findViewById(R.id.et_email);
        et_password= findViewById(R.id.et_password);
        et_password2=findViewById(R.id.et_password2);
        et_CIN= findViewById(R.id.et_CIN);
        et_date=findViewById(R.id.et_date);
        et_date2=findViewById(R.id.et_date2);
        et_Filière=findViewById(R.id.et_Filière);

        queue= (RequestQueue) volleysingleton.getInstance(this).getRequestQueue();
        request= new Myrequest( context: this, queue);
```

```
btn_send.setOnClickListener((view) -> {
    pb_loader.setVisibility(View.VISIBLE);
    String Nom = et_username.getText().toString().trim();
    String Prénom = et_username2.getText().toString().trim();
    String email = et_email.getText().toString().trim();
    String password = et_password.getText().toString().trim();
    String password2 = et_password2.getText().toString().trim();
    String CIN = et_CIN.getText().toString().trim();
    String date = et_date.getText().toString().trim();
    String date2 = et_date2.getText().toString().trim();
    String Filière = et_Filière.getText().toString().trim();
    if (Nom.length() > 0 && email.length() > 0 && password.length() > 0 && Prénom.length() > 0 && CIN.length() > 0 && date.length() > 0 && date2.length() > 0 && Filière.length() > 0) {
        request.register(Nom, Prénom, email, password, password2, CIN, date, date2, Filière, new RegisterCallBack() {
            @Override
            public void onSuccess(String message) {
                pb_loader.setVisibility(View.GONE);
                Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
                intent.putExtra("name", "REGISTER", message);
                startActivity(intent);
                finish();
            }

            public void inputError(Map<String, String> error) {
                pb_loader.setVisibility(View.GONE);
                if(error.get("email") != null){
                    et_email.setError(error.get("email"));
                }else {
                    et_email.setErrorEnable(false);
                }
                if(error.get("password") != null){
                    et_password.setError(error.get("password"));
                }else {
                    et_password.setErrorEnable(false);
                }
            }

        });
    }else {
        Toast.makeText(getApplicationContext(), "Veuillez remplir tous les champs",Toast.LENGTH_SHORT).show();
    }
});
}
```

Figure 5.9 Programmation en java de l'interface inscription étudiant.

➤ Programmation en PHP (Etudiant)

```

1 <?php
2
3 $db= new PDO("mysql:host=localhost;dbname=bibliothèque","root","");
4 $results['error']= false;
5 $results['message']= [];
6
7
8 if(isset($_POST)){
9
10     if(!empty($_POST['Nom']) && !empty($_POST['Prénom']) && !empty($_POST['Filière']) && !empty($_POST['Matricule']) && !
11         empty($_POST['date']) && !empty($_POST['date2']) && !empty($_POST['email']) && !empty($_POST['Password']) && !empty($_
12         _POST['Password2'])){
13
14         $Nom = $_POST['Nom'];
15         $Prénom = $_POST['Prénom'];
16         $Filière = $_POST['Filière'];
17         $Matricule = $_POST['Matricule'];
18         $date = $_POST['date'];
19         $date2 = $_POST['date2'];
20         $email = $_POST['email'];
21         $password = $_POST['password'];
22         $password2 = $_POST['password2'];
23
24         //vérification du Nom
25
26         if(strlen($Nom) < 2 ou !preg_match("/^[a-zA-Z0-9 _-]+$/", $Nom) if strlen($Nom) > 60){
27             $results['error']= true;
28             $results['message']['Nom']= "Nom invalide";
29         }
30     }
31
32     // vérification de l'email
33
34     if(!filter_var($email, FILTER_VALIDATE_EMAIL)){
35         $results['error']= true;
36         $results['message']['email']= "email invalide";
37     }else{
38         //vérifier que l'email n'existe pas
39     }
40
41
42     // vérification des password
43
44     if($password !== $password2){
45         $results['error']= true;
46         $results['message']['password']= "les mots de passes doivent etre identique";
47     }
48
49     if($results["error"] === false){
50
51         $password = password_hash($password, PASSWORD_BCRYPT);
52
53         //Insertion
54
55         $sql = $db->prepare("INSERT INTO users(Nom, email, password) VALUES(:".$Nom", " :email", " :password");
56
57         $sql->execute([":Nom"=> $Nom, ":email"=> $email, ":password"=> $password]);
58
59         if(!$sql){
60             $results['error']= true;
61             $results['message']= "Erreur lors de l'inscription";
62         }
63     }
64 }
65
66 }else {
67     $results['error']= true;
68     $results['message']= ["Veuillez remplir tous les champ"];
69 }
70
71 echo json_encode($results);
72 }
73
74
75
76 ?>

```

Figure 5.10 Programme en PHP d'interface inscription étudiant.

Résumé

Ce mémoire vise à développer une application mobile pour la recherche documentaire dans la bibliothèque, pour les comptes des étudiants et des enseignants de l'université Abderrahmane Mira de Bejaia. Pour aboutir à cette fin, nous avons effectué une étude conceptuelle de l'application. Cette dernière nous permettra, en effet d'accéder facilement à la réalisation de l'application en organisant nos idées et en structurant le processus de codage suivants les diagrammes illustrés par UML. L'application a été implémentée par divers technologies en partant essentiellement par l'étude conceptuelle de données. Notre système de gestion de base de données fut choisi est MySQL ainsi que le serveur apache. Sans oublier quelques services web tel que Android API par les quelles on a pu créer cette plateforme.

□ Mots Clés : Android Studio, Application mobile, Recherche documentaire, Système Android, service Web.

Abstract

This thesis aims to develop a mobile application for documentary research in the library, for the accounts of students and teachers of Abdurrahman Mira University in Bejaia. To achieve this, we performed a concept study of the application. The latter will allow us to easily access the realization of the application by organizing our ideas and structuring the coding process following the diagrams illustrated by UML. The application has been implemented by various technologies starting mainly from the conceptual study of data. Our database management system was chosen is MySQL as well as the apache server. Without forgetting some web services such as Android API by which we were able to create this platform.

□ Keywords: Android Studio, Mobile application, Documentary research, Android system, Web service.