

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

Université Abderrahmane Mira, Bejaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle

**Conception et réalisation d'une Application Mobile sous
Android pour la gestion de la vie estudiantine de
l'Université de Bejaïa
Cas : Département d'Informatique**

En vue de l'obtention du diplôme de licence en informatique spécialité Systèmes informatiques

Présenté par :

BENSLIMANE Amine et RADJEF Melissa

Encadré par :

M. AISSANI Sofiane

Devant le jury composé de :

M. YAIYCI

L. BACHIRI

Promotion 2019

Remerciements

Nous tenons à chaleureusement remercier notre encadrant pour le temps qu'il a consacré à diriger notre travail, pour ses précieux conseils, sa modestie et sa disponibilité.

Nous remercions également les membres du jury pour la peine qu'ils se sont donné pour lire notre travail et pour les précieuses remarques et critiques qu'ils y apporteront.

Mais aussi, l'ensemble de la communauté universitaire et en particulier les enseignants du département d'Informatique pour tout le savoir qu'ils nous ont légué.

Comment oublier de remercier notre famille ? Eux qui nous ont soutenus moralement et financièrement durant tout ce travail.

Merci à tous !

- Amine & Melissa

Dédicaces

On dédie ce modeste travail à

*Nos parents qui nous ont soutenus et encouragés tout le
long de notre cursus.*

Nos frères et sœurs

Nos collègues

Nos ami(e)s

Table des Matières

Table des matières	I
Liste des Abréviations	II
Table des Figures	III
Liste des Tableaux	IV
Introduction générale	1
I. Introduction et étude préliminaire	2
I.1 Introduction	2
I.2 Présentation du projet	2
I.3 Cahier des charges	2
I.3.1 Contexte et problématique	2
I.3.2 Finalités	3
I.3.3 Objectifs de l'application	3
I.3.4 Cible de l'application	3
I.3.5 Type d'application	3
I.3.6 Périmètre du projet	4
I.4 Méthodologie de conception	4
I.4.1 Processus unifié (UP)	4
I.5 Langage de modélisation	5
I.5.1 Présentation d'UML	5
I.6 Etude préliminaire	5
I.6.1 Identification des acteurs	5
I.6.2 Diagramme de contexte du système	5
I.6.3 Diagramme des relations entre acteurs	6
I.7 Besoins fonctionnels	6
I.7.1 Identification des cas d'utilisation	6
I.7.2 Diagramme des cas d'utilisation	7
I.7.2.1 Diagramme des cas d'utilisation générale	7
I.7.2.2 Diagramme des cas d'utilisation de l'acteur « Utilisateur »	7
I.7.2.3 Diagramme des cas d'utilisation de l'acteur « Etudiant »	8
I.7.2.3 Diagramme des cas d'utilisation de l'acteur « Administrateur »	9
I.8 Besoins non fonctionnels	11
I.9 Conclusion	11
II. Analyse et Conception	12
II.1 Introduction	12
II.2 Analyse des besoins	12

II.2.1	Diagramme de séquence	12
II.2.2	Diagramme de séquence du cas d'utilisation « Se connecter ».....	12
II.2.3	Diagramme de séquence du cas d'utilisation « Accès consultations ...	13
II.2.4	Diagramme de séquence du cas d'utilisation « Se déconnecter ».....	15
II.2.5	Diagramme de séquence du cas d'utilisation « Gérer Enseignant »....	15
II.2.5.1	Diagramme de séquence du cas d'utilisation « Ajout d'un enseignant »	15
II.2.5.2	Diagramme de séquence du cas d'utilisation « Modification d'un enseignant »	16
II.2.5.3	Diagramme de séquence du cas d'utilisation « Suppression d'un enseignant ».....	17
II.2.5.4	Diagramme de séquence du cas d'utilisation « Gestion d'un enseignant »	18
II.3	Présentation du diagramme de classe.....	18
II.4	Présentation des classes et de leurs attributs.....	19
II.5	Règles de passage au modèle relationnel	20
II.5.1	Règle 1 : Transformation des classes.....	20
II.5.2	Règle 2 : Transformation des attributs.....	20
II.5.3	Règle 3 : Transformation des associations plusieurs à plusieurs.....	20
II.6	Le model relationnel	21
II.7	Conclusion	21
III.	Réalisation	22
III.1	Introduction.....	22
III.2	Présentation de l'environnement de conception	22
III.2.1	Visual Paradigm	22
III.3	Présentation de l'environnement de développement	22
III.3.1	Android Studio	22
III.3.2	SQLite	23
III.4	Outils de développement	23
III.4.1	SDK	23
III.4.2	JVM	23
III.4.3	JRE	23
III.4.4	JDK	23
III.4.5	AVD	24
III.5	Langages de programmation	24
III.5.1	Java	24
III.5.2	XML	24
III.6	Présentation de l'application	24
III.6.1	Nom de l'application	24
III.6.2	Logo de l'application	24
III.7	Présentation des interfaces	25
III.7.1	Interface de choix de session	25
III.7.2	Interface d'authentification	26
III.7.3	Interface d'inscription	26
III.7.4	Interface d'accueil Administrateur	27
III.7.5	Interface gestion d'un enseignant	28
III.7.6	Interface ajout d'un enseignant	28
III.7.7	Interface suppression d'un enseignant	29

III.7.8 Interface modifier un enseignant	31
III.7.9 Session Etudiant/ Utilisateur	32
III.7.10 Boutons « Se déconnecter » et « Home »	33
II.8 Conclusion.....	33
Conclusion générale	34
Bibliographie	
Résumé	

Table des Figures

Figure I.1 :	Déroulement du processus unifié (UP)	4
Figure I.2 :	Diagramme de contexte du système	5
Figure I.3 :	Diagramme des relations entre acteurs	6
Figure I.4 :	Diagramme principal des cas d'utilisation	7
Figure I.5 :	Diagramme de cas d'utilisation de l'acteur Utilisateur	8
Figure I.6 :	Diagramme de cas d'utilisation de l'acteur Etudiant	9
Figure I.7 :	Diagramme de cas d'utilisation de l'acteur Administrateur	10
Figure II.1 :	Diagramme de séquence du cas d'utilisation « Se connecter »	13
Figure II.2 :	Diagramme de séquence du cas d'utilisation « Accès aux consultations proposées »	14
Figure II.3 :	Diagramme de séquence du cas d'utilisation « Se déconnecter »	15
Figure II.4 :	Diagramme de séquence du cas d'utilisation « Ajout d'un enseignant »	16
Figure II.5 :	Diagramme de séquence du cas d'utilisation « Modification enseignant »	17
Figure II.6 :	Diagramme de séquence du cas d'utilisation « Suppression enseignant »	17
Figure II.7 :	Diagramme de séquence général du cas d'utilisation « Gestion enseignant »	18
Figure II.8 :	Diagramme de classe du système à réaliser	19
Figure III.1 :	Visual Paradigm	22
Figure III.2 :	Android Studio	22
Figure III.3 :	SQL Lite	23
Figure III.4 :	Logo de l'application à réaliser	25
Figure III.5 :	Choix de la session	25
Figure III.6 :	Authentification	26
Figure III.7 :	Inscription	27
Figure III.8 :	Interface Administrateur	27
Figure III.9 :	Gestion des enseignants	28
Figure III.10:	Ajout d'un enseignant	29
Figure III.11:	Boîte dialogue modifier/ supprimer enseignant	30
Figure III.12 :	Liste des enseignants actualisée	30
Figure III.13 :	Anciennes informations enseignant	31
Figure III.14 :	Modification informations enseignant	31
Figure III.15 :	Liste enseignant mise à jour	32
Figure III.16 :	Interface d'accueil Etudiant	32
Figure III.17 :	Interface d'accueil Utilisateur	33
Figure III.18 :	Bouton « Déconnexion »	33
Figure III.19 :	Bouton « Home »	33

Liste des Tableaux

Tableau I.1 :	Différents acteurs du système	5
Tableau I.2 :	Identifications des cas d'utilisations	6
Tableau II.1 :	Présentation des classes de l'application à réaliser	19

Liste des Abréviations

AVD	Android Virtual Device
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
OMT	Object Modeling Technique
OOSE	Object Oriented Software Engineering
SDK	Software Development Kit
SGBD	Système de Gestion de Base de Données
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
UML	Unified Modeling Language
UP	Unified Processus
XML	Extended Markup Language

Introduction générale

Les technologies mobiles ne cessent d'évoluer et de prendre une place de plus en plus considérable au sein de notre routine. À tel point qu'avoir un dispositif mobile est devenu banal ou même incontournable.

L'objectif de ce travail est de concevoir et réaliser une application sous Android, installable sur des terminaux mobiles, qui permettra aux étudiants du département d'informatique de l'université de Bejaïa d'avoir accès, depuis leur dispositif mobile, à différentes informations en temps réel de façon simple et organisée.

En effet, le monde de l'université est tellement vaste que les étudiants s'y perdent. Le concept de tutorat a justement été inventé pour établir des « mentors » pour les étudiants et les guider à mieux connaître ce nouveau monde mais ceci étant théorique, qu'en est-il de la pratique ? L'application du tutorat est inexistante.

Ce travail offrira aux étudiants d'informatique l'opportunité d'être indépendants et assurera une bonne circulation des informations les plus importantes pour vivre à l'aise leur vie estudiantine.

Ce rapport détaille toutes les phases –théorique et pratique – de notre projet qui est composé de trois chapitres disposés comme suit :

Le premier chapitre, « Introduction et étude préliminaire » introduira et soulignera les périmètres de notre projet, la problématique que nous traitons à travers ce dernier, notre cahier des charges ainsi que notre analyse des besoins. Il est suivi d'une spécification des besoins détaillée par le diagramme général de cas d'utilisation suivis des diagrammes de séquence les plus importants.

Le deuxième chapitre « Conception » concernera les profondeurs de la conception du projet avec notamment le diagramme de classe.

Le troisième et dernier chapitre « Réalisation » définira les outils utilisés et mis en pratique pour réaliser le système décrits au premier et deuxième chapitre. Et comportera un aperçu des interfaces les plus importantes de notre application mobile.

I. Introduction et étude préliminaire

I.1 Introduction

Ce chapitre est destiné à l'étude préliminaire de notre projet qui consiste à présenter ce dernier mais aussi à clarifier la problématique qui nous a poussés à entreprendre ce projet et ainsi repérer et décrire les besoins du système à réaliser.

I.2 Présentation du projet

Le projet s'intitule « Conception et réalisation d'une application mobile sous Android pour la gestion de la vie estudiantine de l'université de Bejaïa, cas : Département d'informatique ». Il sera dédié aux étudiants du département d'informatique comme guide ou même de véritable compagnon tout au long de leur cursus universitaire où l'application assurera une bonne fluidité et transmission d'informations qui garantira à l'étudiant d'être autonome et indépendant.

I.3 Cahier des charges

I.3.1. Contexte et problématique

La vie estudiantine peut être parfois terrible si l'on n'est pas organisé, informé, ou même dirigé.

Le concept de tutorat a vite fait surface, mais le constat est tel qu'il n'est pas une solution aux multiples problèmes que rencontrent les étudiants de l'Université de Bejaia.

Les étudiants, en particulier ceux de premières années, sont littéralement perdus dans ce nouveau monde qu'est l'Université. Entre quand prendre le bus universitaire, où se situe leur faculté, leur département ? Où est le bloc X ou Y ? Où est le bureau de mon enseignant ?

Toutes ces questions ont été posées au moins une fois, et n'ont pas eu de réponses, ou du moins une réponse claire et rapide.

↳ D'où, la nécessité de créer une application qui, en temps réel, peut guider, conseiller et surtout aider l'étudiant tout au long de sa vie estudiantine.

I.3.2. Finalités

À la fin de ce projet, les étudiants pourront consulter diverses informations précieuses telles que leur emploi du temps, la carte de l'université, la liste des enseignants...etc.

Ils seront avec cette application, indépendants, et aborderont, de la meilleure façon possible, l'année universitaire.

I.3.3 Objectifs de l'application

Les objectifs représentent l'ensemble des actions à réaliser pour atteindre la finalité à partir du contexte actuel.

Ces derniers sont :

- Une interface destinée à l'étudiant où il pourra consulter diverses informations proposées par l'application :
 - o Emploi du temps
 - o Horaire des transports universitaires
 - o Informations sur sa formation (description de son socle universitaire, modules, coefficients ...etc.)
 - o La liste des enseignants du département d'informatique avec plusieurs informations (Nom, prénom, email, numéro bureau ...etc.)
- Une interface pour les utilisateurs (sans authentification) où ils pourront effectuer plusieurs consultations
- Une interface administrateur à partir de laquelle s'établira la gestion de l'application (gestion des horaires transports, des affichages etc.)

I.3.4 Cible de l'application

Notre application portera sur la vie estudiantine et donc, ciblera tous les étudiants ainsi que les enseignants de l'université de Bejaïa. Mais en premier lieu, nous viserons exclusivement les étudiants du domaine Informatique, la généralisation pour le reste des étudiants étant analogue.

I.3.5 Type d'application

Les applications étant de trois types :

- Application Desktop (Bureau)
- Application Web (Forcément Client/Serveur)
- Application Mobile

Remarque : une application web peut être une application mobile

Vu les besoins exprimés plus haut, l'application doit être accessible à tout moment et partout par l'étudiant, on se penchera donc vers une application mobile. En laissant en perspective, l'initiative aux étudiants de refaire ce concept sous forme d'application web ou de bureau.

I.3.6. Périmètre du projet

Les études à l'université de Bejaia se font majoritairement en français, nous supposons alors que les étudiants seront beaucoup plus à l'aise avec cette langue, c'est pour cela que nous décidons en premier lieu d'utiliser le Français pour la création de notre application, avec une possibilité d'extension notamment en Tamazight, en Anglais ou en Arabe dans le futur.

I.4 Méthodologie de conception

Nous avons choisi le Processus Unifié (UP) comme méthodologie de conception.

I.4.1 Processus unifié (UP) :

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adaptée à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise. [1]

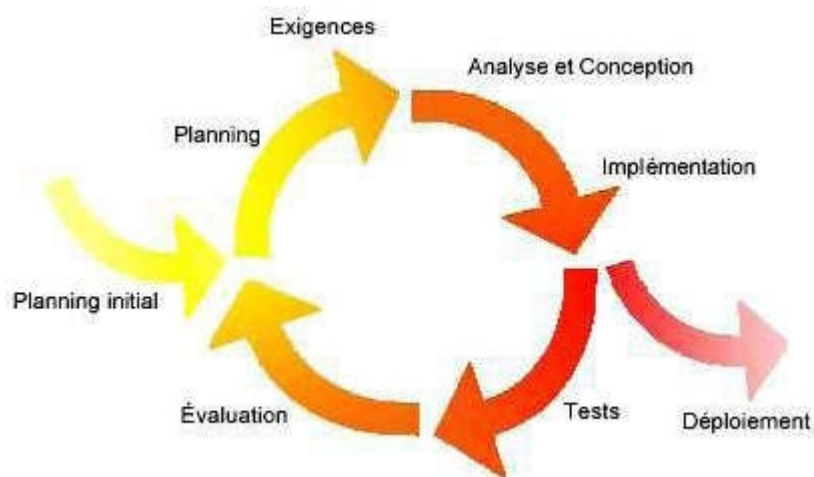


Figure I.1 : Déroulement du processus unifié (UP)

I.5 Langage de modélisation

Pour mener à bien ce projet, il est indispensable de modéliser le système, nous choisissons UML comme norme de modélisation.

I.5.1 Présentation d'UML :

UML, Unified Modeling Language, langage de modélisation objet unifié est une démarche orientée objet. Elle est née de la fusion de trois méthodes orientées objet Booch, OMT Object Modeling Technique et OOSE Object Oriented Software Engineering, conçues respectivement par Grady Booch, James Rumbaugh et Ivar Jacobson. [2]

I.6 Etude préliminaire

I.6.1 Identification des acteurs

Le tableau I.1 représente les différents acteurs interagissant avec le système :

Acteurs
Administrateur
Etudiant
Utilisateur

Tableau I.1 : Différents acteurs du système

I.6.2 Diagramme de contexte du système

Nous modélisons ci-dessous les acteurs qui interagissent avec le système que nous souhaitons réaliser :

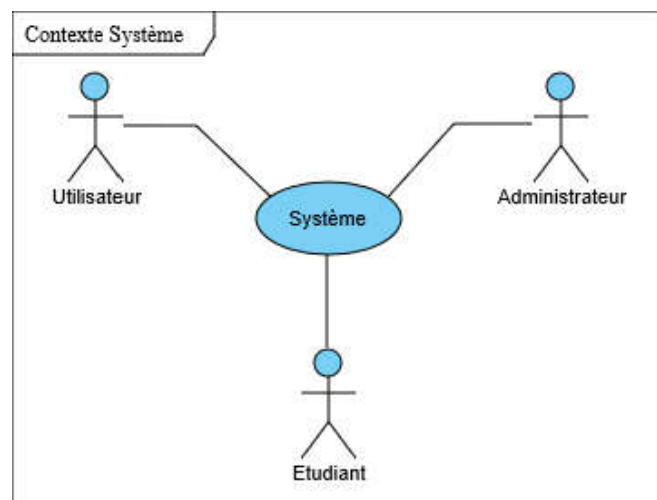


Figure I.2 : Diagramme de contexte du système

I.6.3 Diagramme des relations entre acteurs

Nous modélisons dans la figure I.3 la relation d'héritage entre les différents acteurs du système à réaliser.

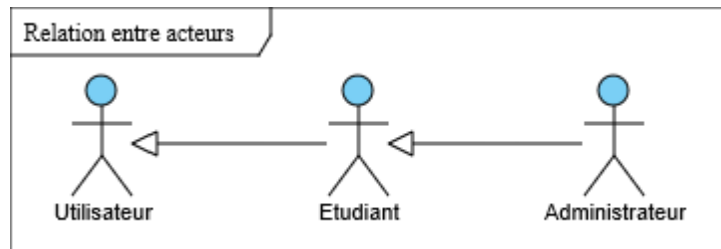


Figure I.3 : Diagramme des relations entre acteurs

I.7 Besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du système. [3]

I.7.1 Identification des cas d'utilisation

Dans le système à réaliser, nous avons identifié les cas d'utilisations suivants :

N°	Cas d'utilisation		Acteurs
1	Se connecter		Administrateur/Etudiant
2	Se déconnecter		
3	S'inscrire		Etudiant
4	Consulter	Emploi du temps	Etudiant
		Socle/Formation	Etudiant/Utilisateur
		Carte	Etudiant/Utilisateur
		Liste enseignant	Etudiant/Utilisateur
		Estimation restaurant	Etudiant/Utilisateur
		Horaire transport	Etudiant/Utilisateur
		Actualités	Etudiant/Utilisateur
5	Gestion d'un étudiant	Suppression	Administrateur
6	Gestion des horaires transports	Ajout	Administrateur
		Modification	
		Suppression	
7	Gestion d'un enseignant	Ajout	Administrateur
		Modification	
		Suppression	
8	Gestion des actualités	Ajout	Administrateur
		Modification	
		Suppression	

Tableau I.2 : Identification des cas d'utilisations

I.7.2 Diagramme des cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. [4]

I.7.2.1 Diagramme des cas d'utilisation générale :

Le diagramme principal des cas d'utilisation est représenté par la figure I.4 comme suit :

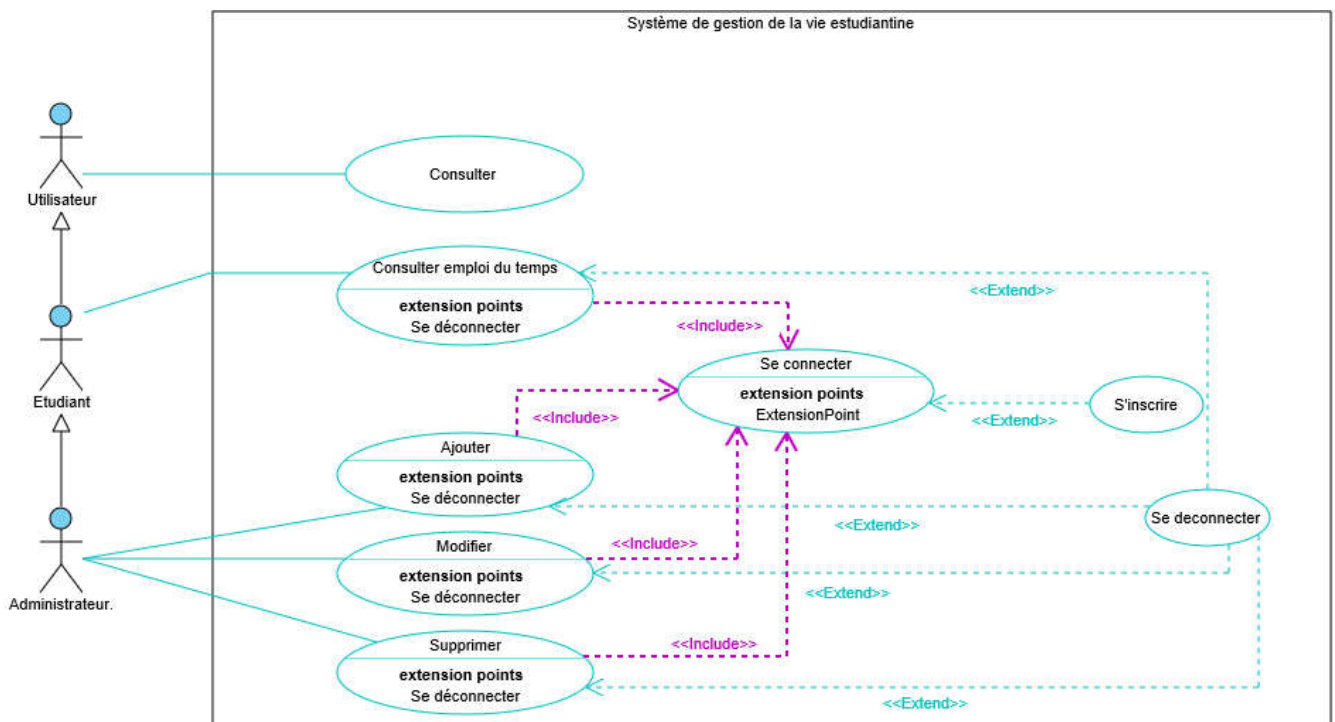


Figure I.4 : Diagramme principal des cas d'utilisation

I.7.2.2. Diagramme des cas d'utilisation de l'acteur « Utilisateur » :

La figure I.5 représente le diagramme des cas d'utilisations de l'acteur « Utilisateur » :

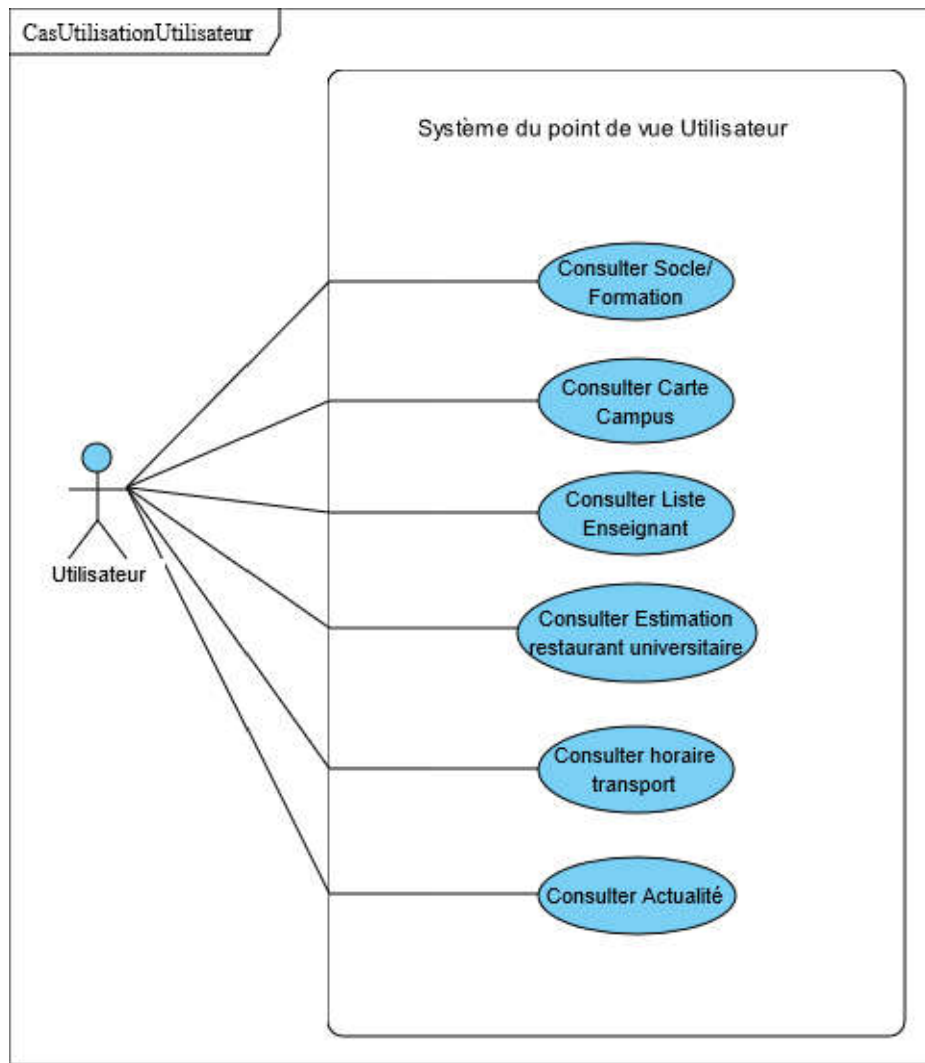


Figure I.5 : Diagramme de cas d'utilisation de l'acteur utilisateur

I.7.2.3. Diagramme des cas d'utilisations de l'acteur « Etudiant » :

La figure I.6 modélise le diagramme des cas d'utilisations de l'acteur « Etudiant » :

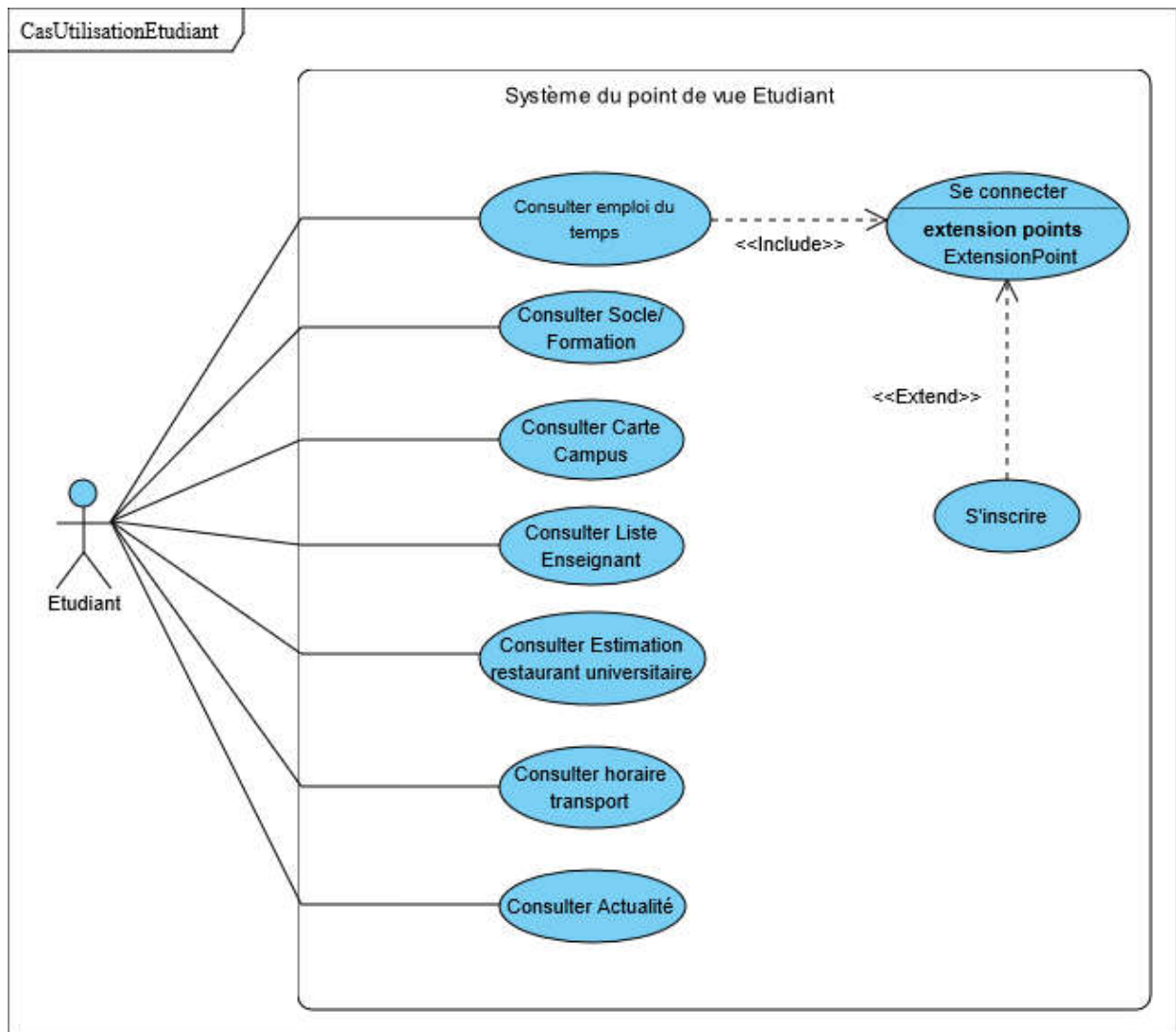


Figure I.6 : Diagramme des cas d'utilisations de l'acteur Etudiant

I.7.2.4. Diagramme des cas d'utilisations de l'acteur « Administrateur »

La figure I.7 modélise le diagramme des cas d'utilisation de l'acteur « Administrateur » :

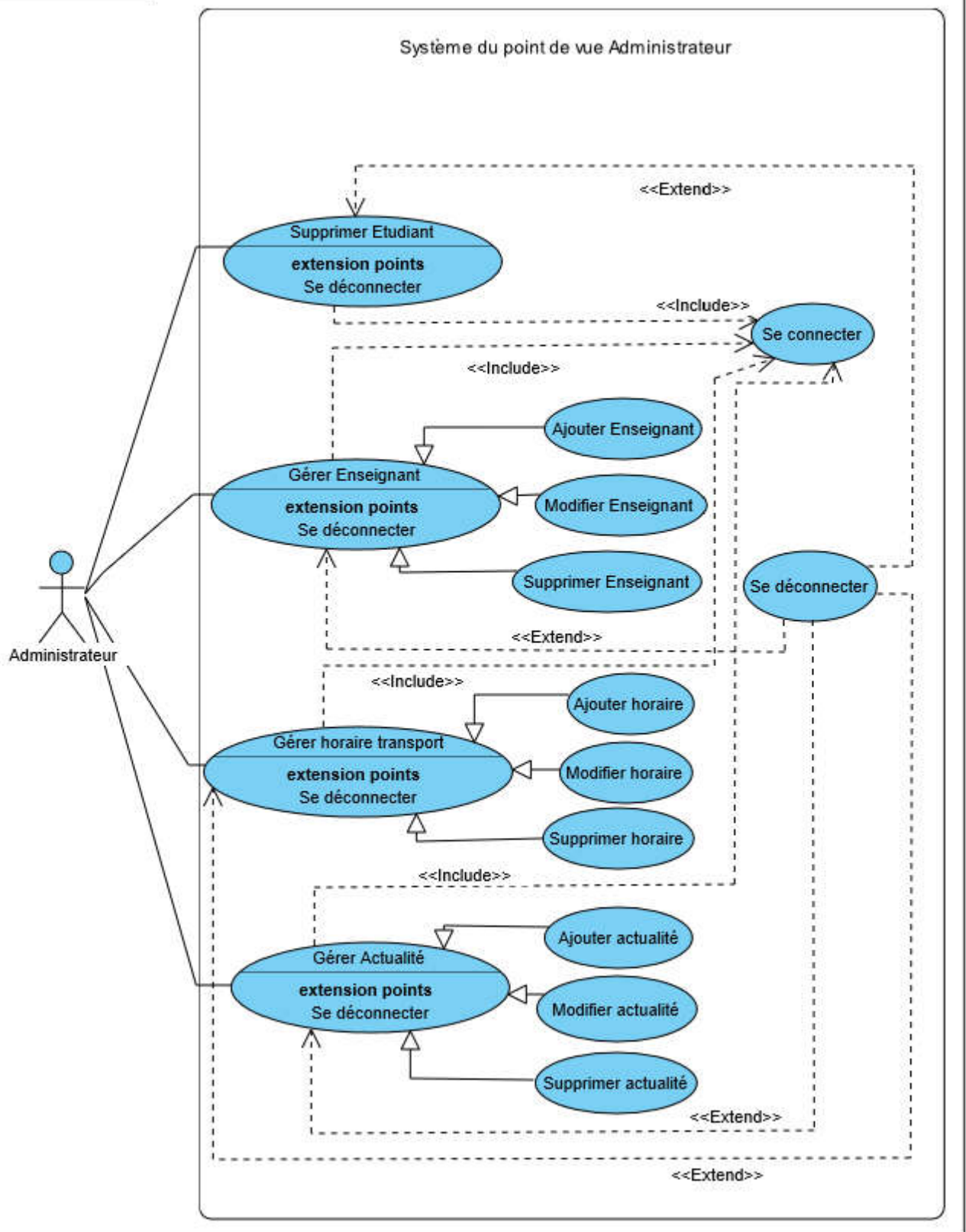


Figure I.7 : Diagramme des cas d'utilisation de l'acteur Administrateur

I.8 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 notre cas, l'application devra être extensible, c'est-à-dire qu'on pourra lui ajouter de nouvelles fonctionnalités et ainsi être mise à jour pour actualiser les informations qu'elle comporte.

L'application devra être légère en mémoire, c'est-à-dire qu'il faudra optimiser l'implémentation et ainsi réduire les coûts en mémoire.

Devra être ergonomique dans le sens large du mot, c'est-à-dire respecter les normes de design pour offrir à l'utilisateur la meilleure expérience possible.

La notion de sécurité est aussi très importante, les informations des utilisateurs doivent à tout prix resté confidentielles.

L'application devra être téléchargeable et utilisable par un maximum de dispositifs mobiles sous Android.

I.9 Conclusion

Durant ce chapitre, nous avons présenté notre projet ainsi que la problématique qui nous a poussés à l'entreprendre. Nous avons ensuite entamé notre étude préliminaire où nous avons identifié les acteurs du système à réaliser pour ensuite recueillir les besoins fonctionnels du système à travers des diagrammes de cas d'utilisations, mais aussi les besoins non fonctionnels dont doit répondre le système.

Nous entamerons dans le chapitre suivant, la phase analyse des besoins et conception du projet.

II. Analyse et Conception

II.1 Introduction

Dans ce chapitre, nous élaborerons une analyse des besoins détaillée ainsi qu'une conception convenable au système à réaliser. Cette dernière formera la base architecturale du projet et nous permettra d'avoir un aperçu du code d'implémentation et ainsi d'éviter les erreurs de programmation.

En premier lieu, nous modéliserons notre analyse des besoins à travers des diagrammes de séquence des cas d'utilisations les plus pertinents.

On décrira ensuite la traduction des besoins exprimés auparavant en les représentants à l'aide de classes, d'associations et d'attributs qui forment les concepts de base d'une conception orientée objet. Le but étant de produire le diagramme de classe du système à réaliser

Finalement, nous établirons le model relationnel, en spécifiant les règles de passage que nous avons utilisés pour passer du diagramme de classe vers ce dernier.

II.2 Analyse des besoins

Dans cette partie, on expose les diagrammes de séquences des cas d'utilisations les plus pertinents du système à réaliser.

II.2.1 Diagramme de séquence

Un diagramme de séquence est un diagramme UML (Unified Modeling Language) qui représente la séquence de messages entre les objets au cours d'une interaction. Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction. [5]

II.2.2 Diagramme de séquence du cas d'utilisation « Se connecter »

La figure I.8 modélise le diagramme de séquence du cas d'utilisation « Se connecter », le système d'authentification et le même pour l'acteur « Administrateur » et l'acteur « Etudiant ».

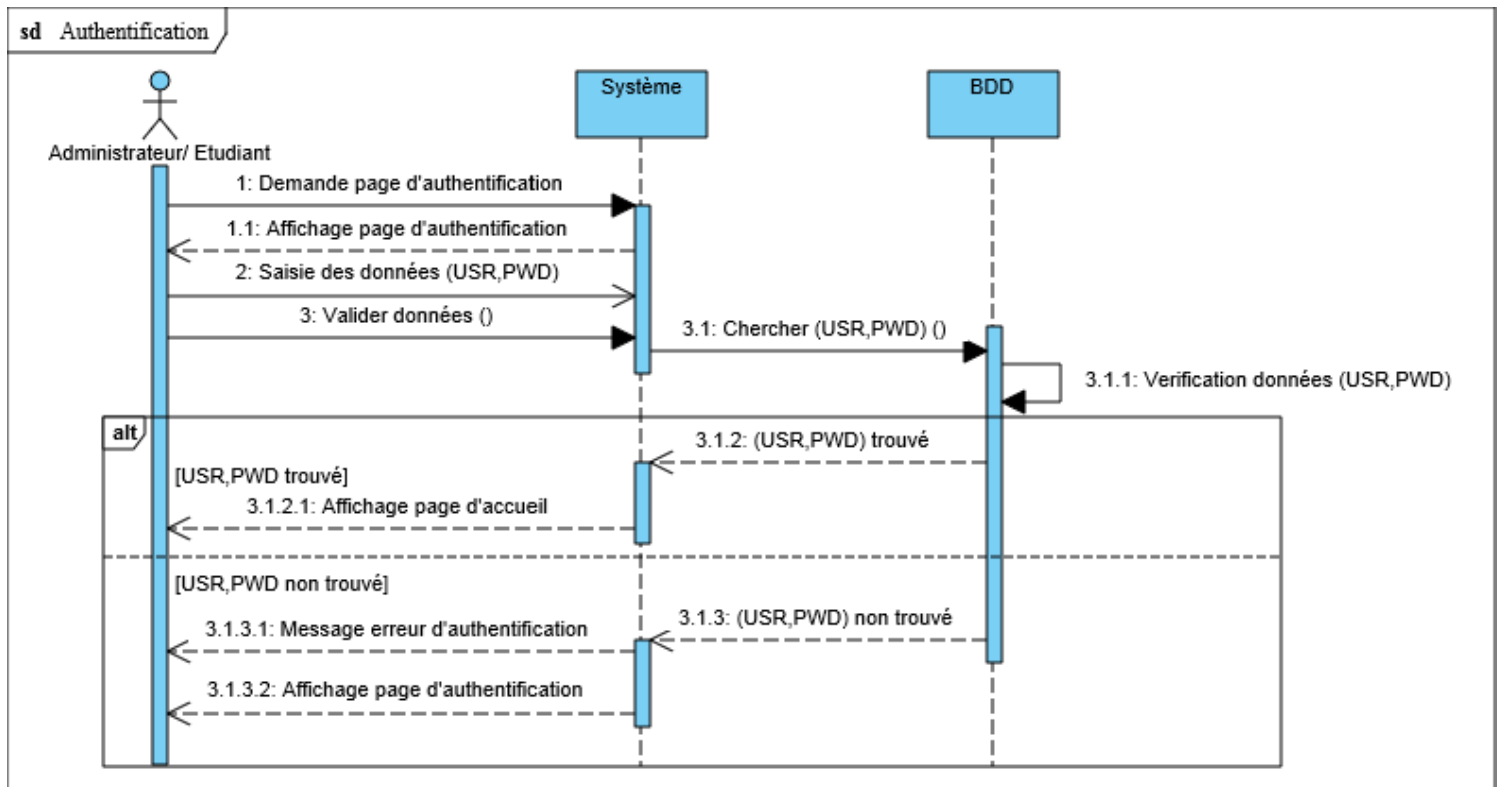


Figure II.1 : Diagramme de séquence du cas d'utilisation « Se connecter »

II.2.3. Diagramme de séquence du cas d'utilisation « Accès aux consultations »

La figure II.2 modélise le diagramme de séquence du cas d'utilisation « Accès aux consultations ». L'étudiant dispose ainsi d'un choix aisé d'espaces de consultations.

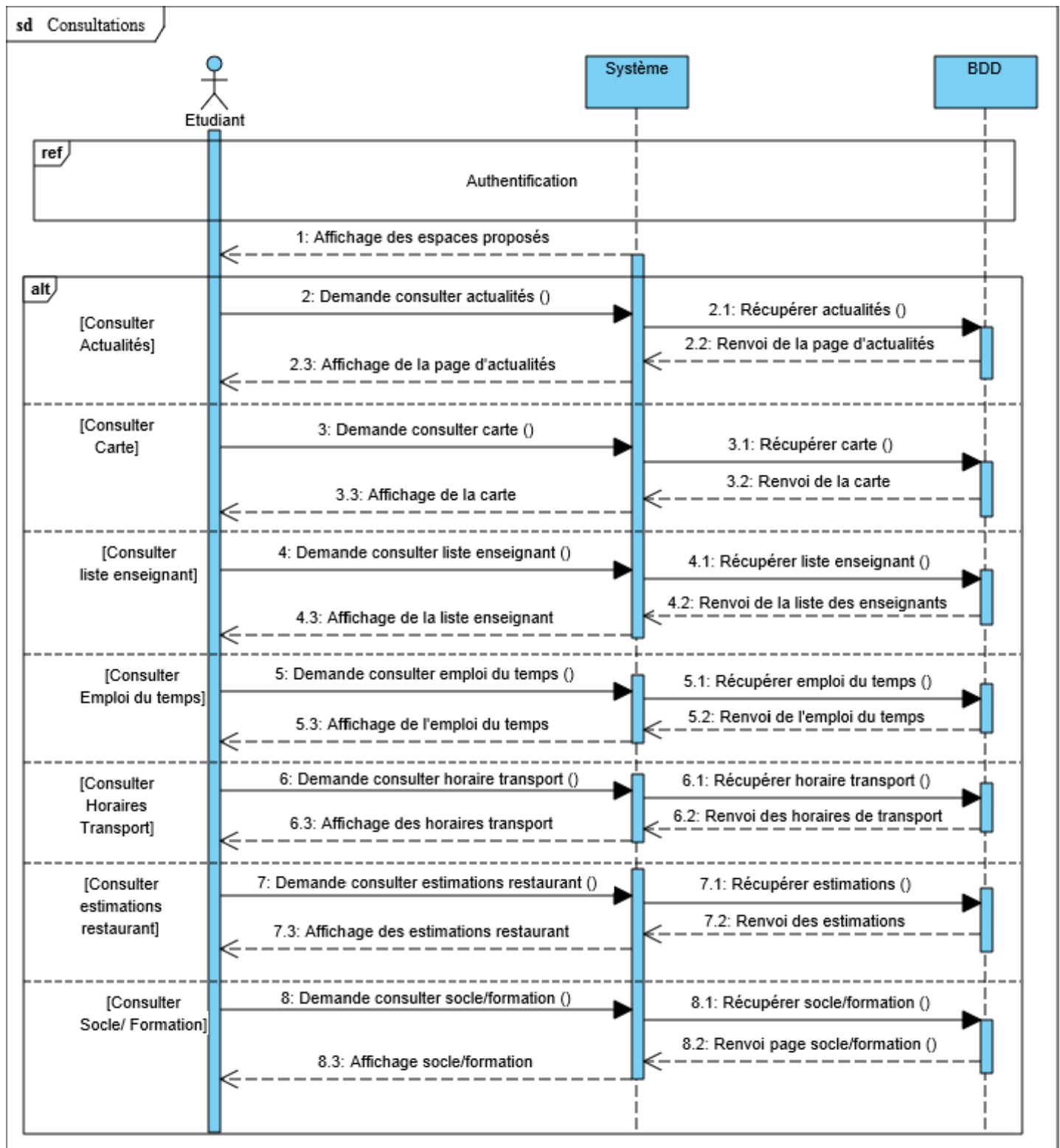


Figure II.2 : Diagramme de séquence du cas d'utilisation « Accès aux consultations proposées »

II.2.4 Diagramme de séquence du cas d'utilisation « Se déconnecter »

La figure I.10 modélise le diagramme de séquence du cas d'utilisation « Se déconnecter »

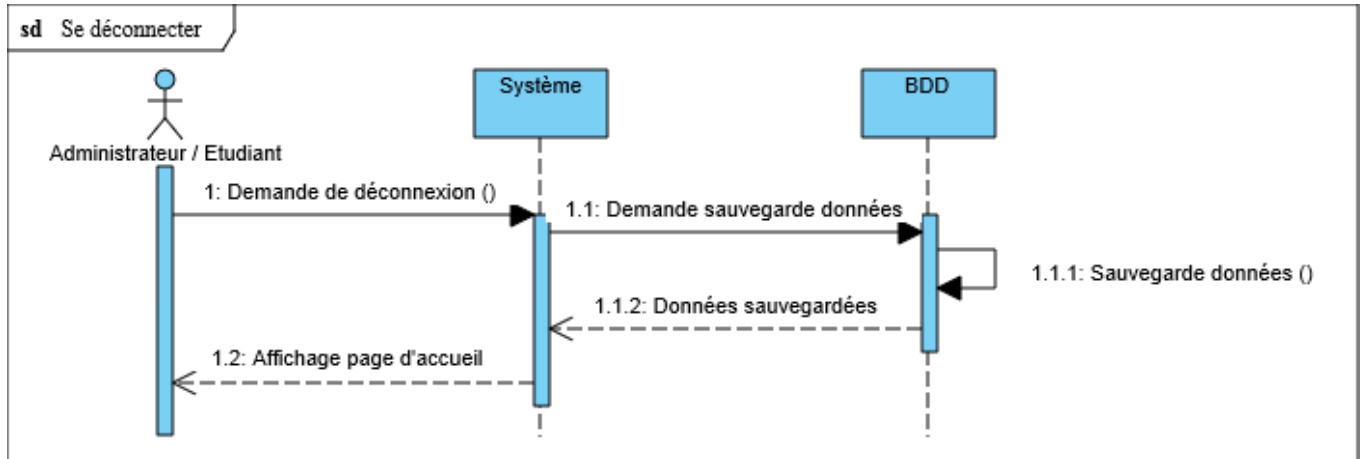


Figure II.3 : Diagramme de séquence du cas d'utilisation « Se déconnecter »

II.2.5 Diagramme de séquence du cas d'utilisation « Gérer Enseignant »

La gestion d'un enseignant est seulement faite par l'administrateur et est divisée en trois cas d'utilisation : « Ajouter enseignant », « Modifier enseignant » et « Suppression d'un enseignant ».

On présentera donc trois diagrammes de séquence pour gérer un enseignant.

II.2.5.1. Diagramme de séquence du cas d'utilisation « Ajout d'un enseignant »

La figure II.4 représente le diagramme de séquence du cas d'utilisation « Ajout d'un enseignant » :

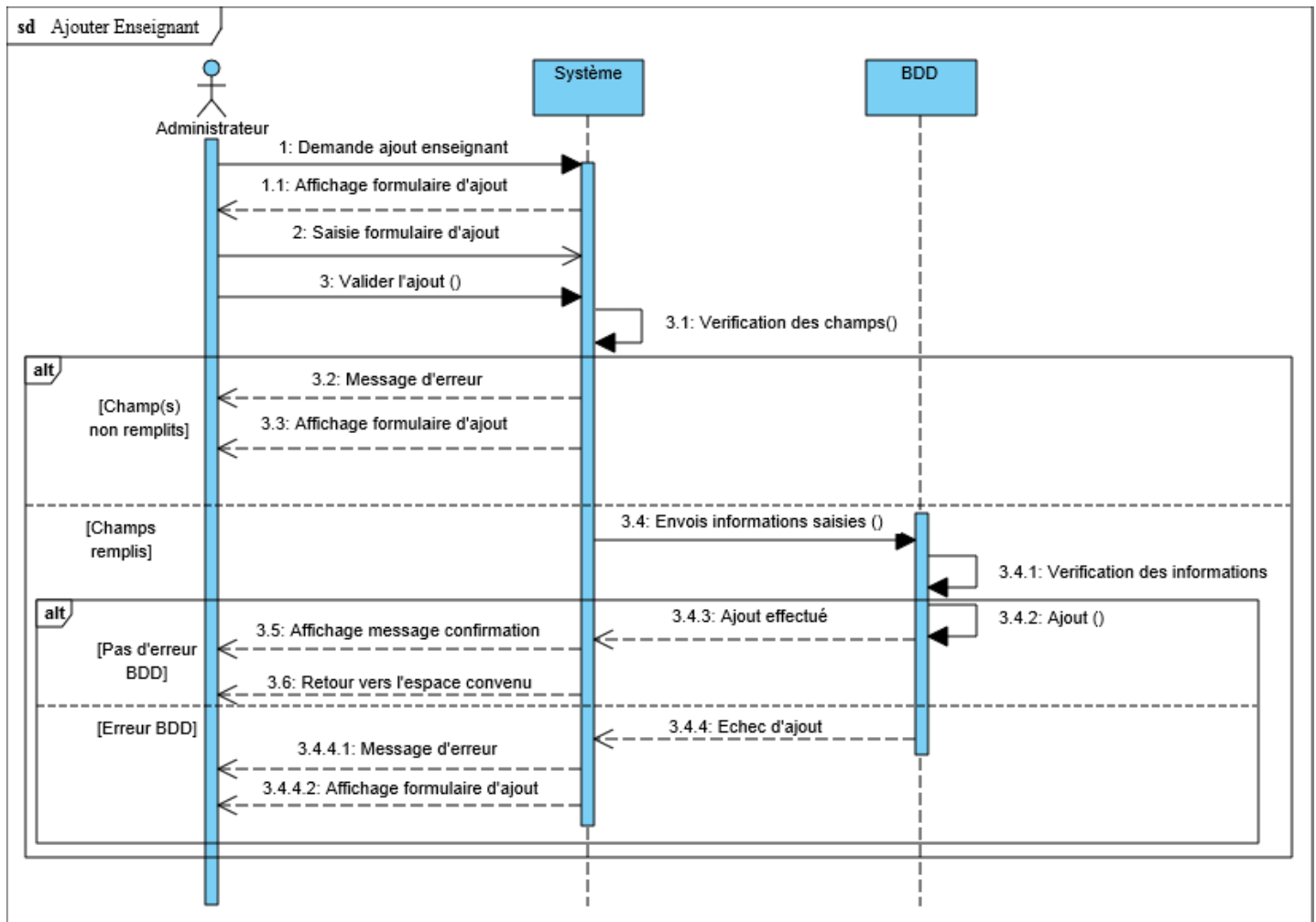


Figure II.4 : Diagramme de séquence du cas d'utilisation « Ajout d'un enseignant »

II.2.5.2 Diagramme de séquence du cas d'utilisation « Modification d'un enseignant »

La figure II.5 modélise le diagramme de séquence du cas d'utilisation « Modification d'un enseignant »

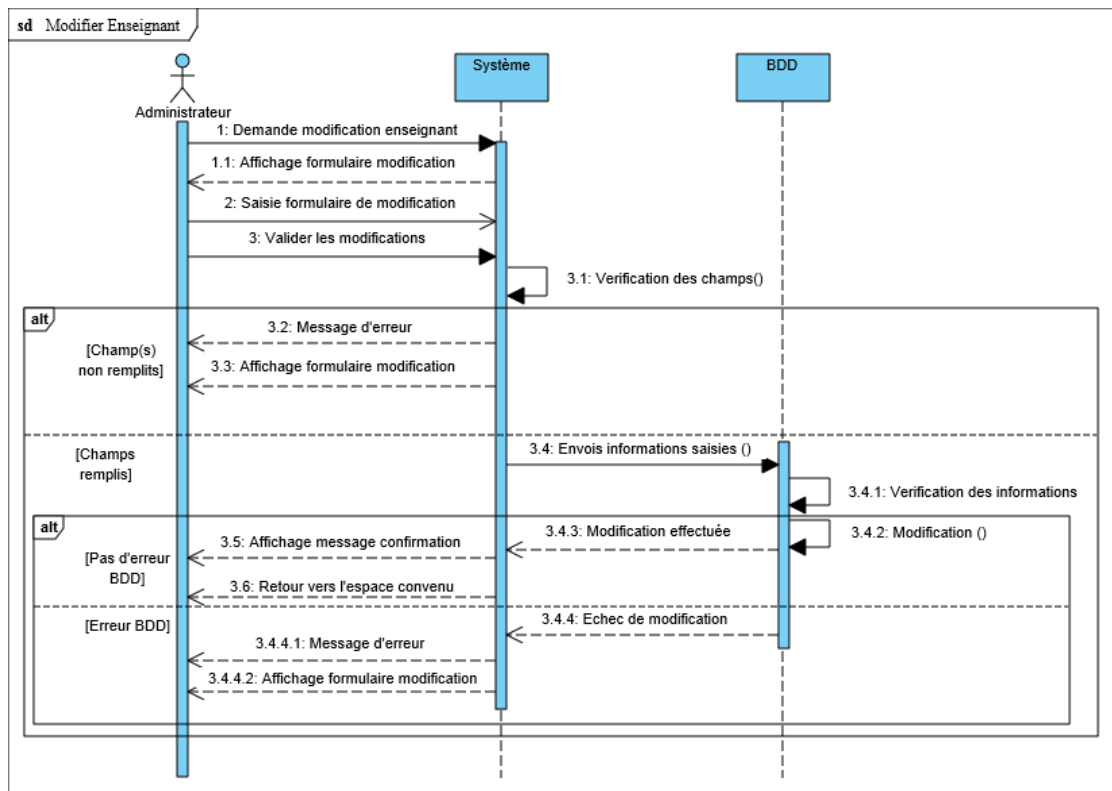


Figure II.5 : Diagramme de séquence de diagramme de séquence du cas d'utilisation « Modifier Enseignant »

II.2.5.3 Suppression d'un Enseignant

La figure II.6 modélise le diagramme de séquence du cas d'utilisation « Suppression d'un enseignant »

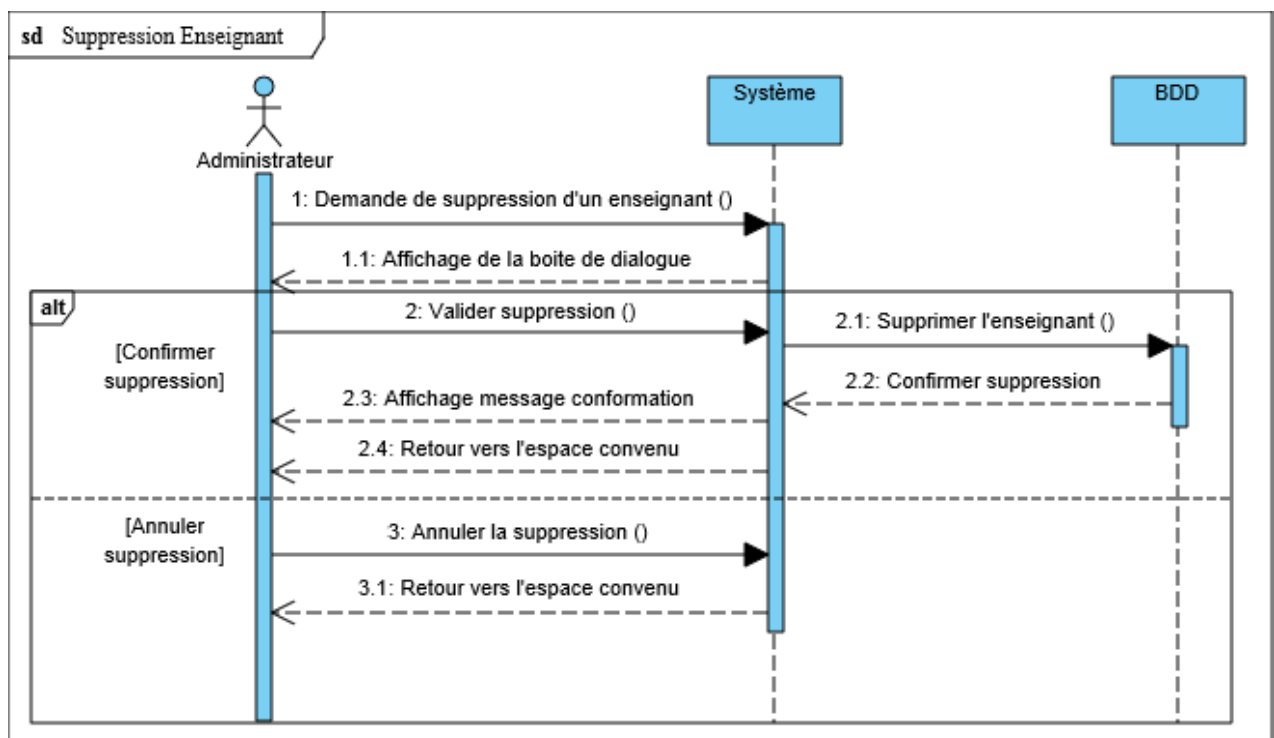


Figure II.6 : Diagramme de séquence du cas d'utilisation « Suppression d'un enseignant »

II.2.5.4 Diagramme de séquence du cas d'utilisation « Gestion d'un enseignant »

Après avoir présenté le diagramme de séquence des cas d'utilisation d'ajout, modification et suppression d'un enseignant, la figure II.7 les regroupe pour obtenir le diagramme de séquence du cas d'utilisation « Gestion d'un enseignant »

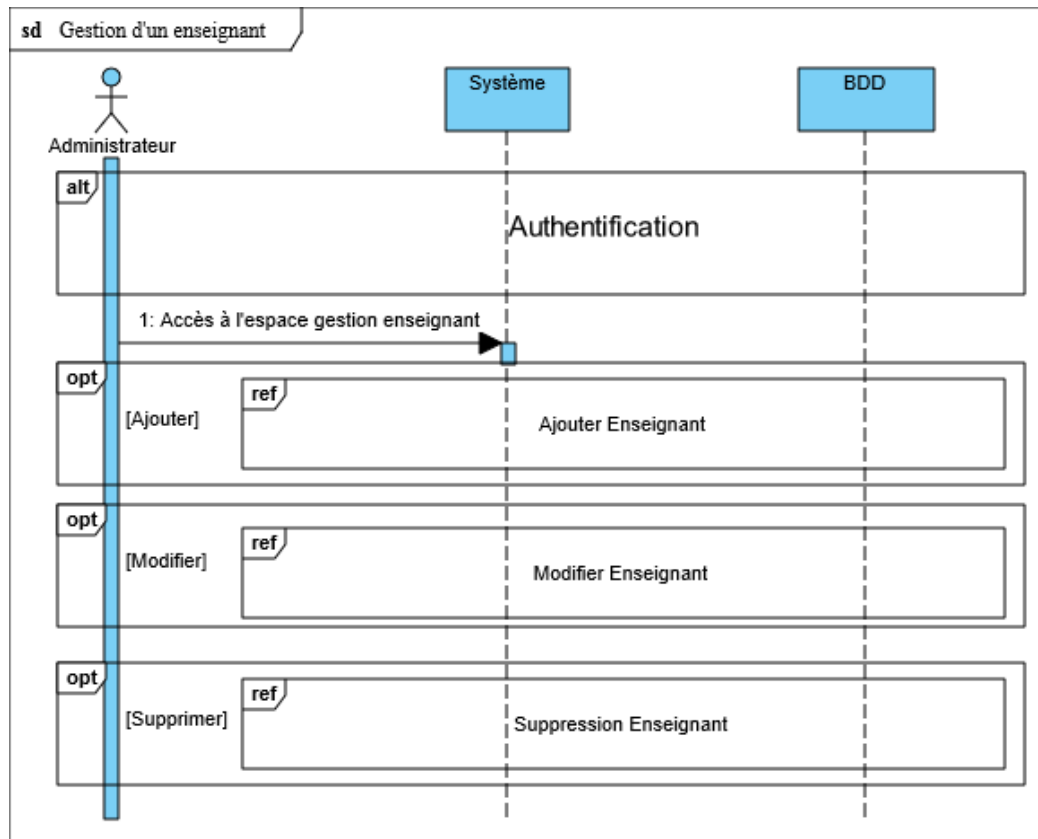


Figure II.7 : Diagramme de séquence général du cas d'utilisation « Gestion d'un enseignant »

II.3 Présentation du diagramme de classe

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. [6]

La figure II.8 suivante modélise le diagramme de classe du système à réaliser

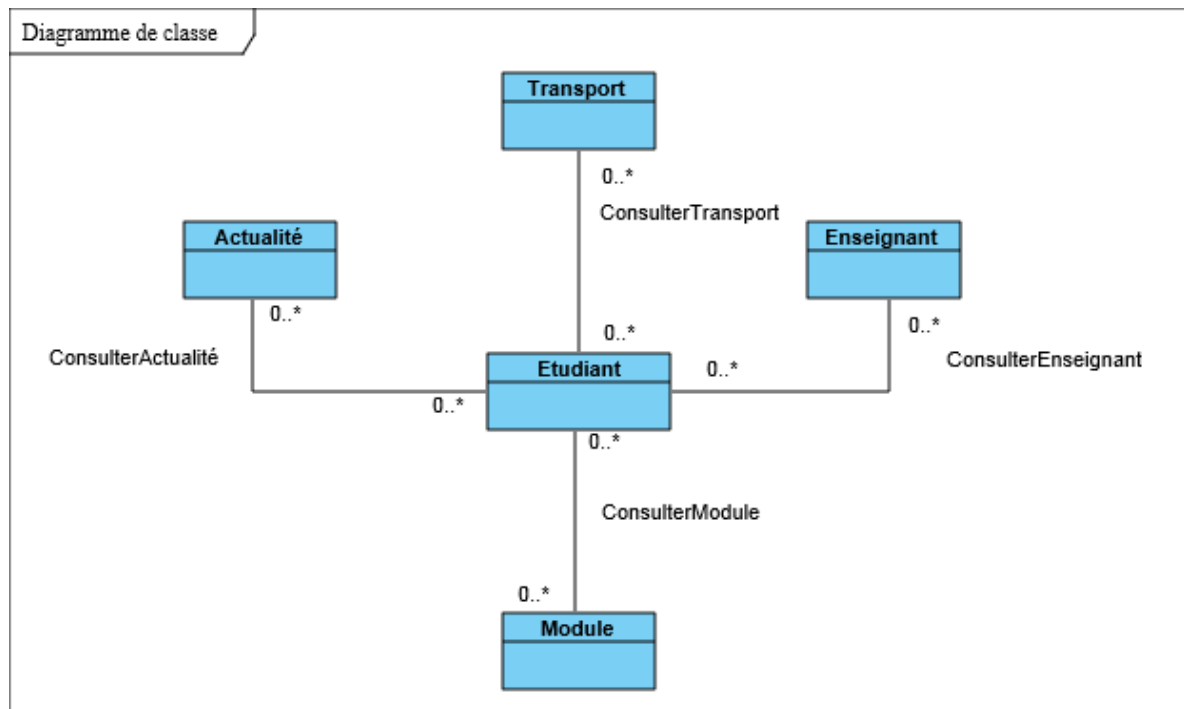


Figure II.8 : Diagramme de classe du système à réaliser

II.4 Présentation des classes et de leurs attributs

Nous présentons pour chaque classe, ses attributs, leur définition et leur type.

Classe	Attributs	Définition de l'attribut	Type
Etudiant	idEtudiant	Identifiant de l'étudiant	Numérique (entier)
	userEtud	Pseudonyme de l'étudiant	Alphanumérique
	psdEtud	Mot de passe de l'étudiant	Alphanumérique
	nivEtud	Niveau de l'étudiant	Alphanumérique
	specEtud	Spécialité de l'étudiant	Alphabétique
	secEtud	Section de l'étudiant	Alphabétique
	grpEtud	Groupe de l'étudiant	Alphabétique
Enseignant	idEnseignant	Identifiant de l'enseignant	Numérique
	nomEns	Nom de l'enseignant	Alphabétique
	prenomEns	Prénom de l'enseignant	Alphabétique
	numBrEns	Numéro du bureau de l'enseignant	Numérique
	emailEns	Adresse Mail de l'enseignant	Alphanumérique

Module	codeMod	Identifiant du module	Alphanumérique
	nomMod	Nom du module	Alphanumérique
	coefMod	Coefficient du module	Numérique
	credMod	Crédit du module	Numérique
	resuMod	Résumé du module	Alphanumérique
Transport	idTrans	Identifiant de l'horaire de transport	Numérique
	depTrans	Adresse de départ du transport	Alphanumérique
	desTrans	Adresse d'arrivée du transport	Alphanumérique
	heureTrans	Heure de départ du transport	Alphanumérique
Actualité	idActu	Identifiant de l'actualité	Numérique
	nomActu	Nom de l'actualité	Alphanumérique
	intituleActu	Intitulé de l'actualité (Contenu)	Alphanumérique

Tableau II.1 : Présentation des classes de l'application à réaliser

II.5 Règles de passage au modèle relationnel

Nous présentons ici uniquement les règles que nous avons utilisées pour produire le modèle relationnel à partir du diagramme de classe

II.5.1 Règle 1 : Transformation des classes

Pour chaque classe non abstraite, on crée une relation dont le schéma est celui de la classe ; la clé primaire de cette relation est une des clés de la classe. [7]

II.5.2 Règle 2 : Transformation des attributs

Pour chaque attribut élémentaire et monovalué d'une classe, on crée un attribut correspondant dans la relation. [7]

II.5.3 Règle 3 : Transformation des associations plusieurs à plusieurs :

Pour chaque association binaire de type M:N : on crée une nouvelle relation composée de clés étrangères vers chaque relation associée et dont la clé primaire est la concaténation de ces clés étrangères. [7]

II.6 Le modèle relationnel

Nous présentons le modèle relationnel du système à réaliser :

Etudiant	(<u>idEtudiant</u> , userEtud, psdEtud, specEtud, secEtud, grpEtud).
Enseignant	(<u>idEnseignant</u> , nomEns, prenomEns, numBrEns, emailEns).
Module	(<u>codeMod</u> , nomMod, coefMod, credMod, resuMod).
Transport	(<u>idTrans</u> , depTrans, desTrans, heureTrans).
Actualité	(<u>idActu</u> , nomActu, intituleActu).
ConsulterEnseignant	(<u>#idEtudiant, #idEnseignant</u>).
ConsulterModule	(<u>#idEtudiant, #codeMod</u>).
ConsulterTransport	(<u>#idEtudiant, #idTrans</u>).
ConsulterActualité	(<u>#idEtudiant, #idActu</u>).

II.7 Conclusion

Dans ce chapitre, nous avons réalisé l'étude conceptuelle de notre application. Nous avons modéliser les échanges d'informations entre l'utilisateur et le système à travers des diagrammes de séquence des cas d'utilisation les plus pertinents. Nous avons ensuite introduit la conception en modélisant le diagramme de classe du système puis on a défini les règles de passage vers le modèle relationnel et on a conclu par ce dernier. On obtient donc un aperçu de la structure de la base de données du système à réaliser.

Le prochain chapitre portera sur les technologies utilisées pour aboutir au résultat voulu.

III. Réalisation

III.1 Introduction

Nous poursuivons donc notre travail avec la dernière étape : la réalisation du système. Nous présentons ici l'environnement de développement logiciel, les outils de développement ainsi que les langages de programmation. Nous terminons par présenter le résultat obtenu avec les interfaces les plus importantes de notre application.

III.2 Présentation de l'environnement de conception

III.2.1 Visual Paradigm

Visual Paradigm est un logiciel de création de diagrammes dans le cadre d'une programmation. Tout en un, il possède plusieurs options permettant une large possibilité de modélisation en UML. Tous les diagrammes que comporte ce rapport ont été réalisés à l'aide de cet outil.



Figure III.1 : Visual Paradigm

III.3 Présentation de l'environnement de développement

III.3.1 Android Studio

Android Studio est un nouvel environnement pour développement et programmation entièrement intégré qui a été récemment lancé par Google pour les systèmes Android. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé. [8]

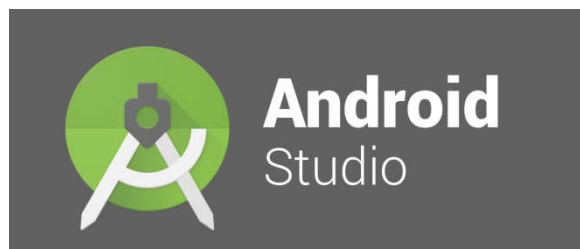


Figure III.2: Android Studio

III.3.2 SQL Lite

SQL Lite est une bibliothèque écrite en langage C qui contient un moteur de base de données relationnelle qui implémente le langage SQL. Son avantage par rapport à d'autre est de ne pas reproduire le schéma client-serveur habituel mais d'être directement intégrée aux programmes. Ce qu'on appelle couramment « base de donnée interne ».



Figure III.3 : SQL Lite

III.4 Outils de développement

On présente ici les divers outils de développement que l'on a utilisé.

III.4.1 SDK

Les applications Android sont développées en Java, mais un appareil sous Android ne comprend pas le Java tel quel, il comprend une variante du Java adaptée pour Android. Un SDK, un kit de développement dans notre langue, est un ensemble d'outils permettant de développer pour une cible particulière. Le SDK Android est donc un ensemble d'outils que met à disposition Google afin de vous permettre de développer des applications pour Android. [9]

III.4.2 JVM

JVM est une machine abstraite. C'est ce qu'on appelle une machine virtuelle car elle n'existe pas physiquement. C'est une spécification qui fournit un environnement d'exécution dans lequel le bytecode Java peut être exécuté. [10]

III.4.3 JRE

JRE est un ensemble d'outils logiciels utilisés pour développer des applications Java. Il est utilisé pour fournir l'environnement d'exécution. C'est l'implémentation de JVM. Cela existe physiquement. Il contient un ensemble de bibliothèques et d'autres fichiers que JVM utilise au moment de l'exécution. [10]

III.4.4 JDK

Le kit de développement Java (JDK) est un environnement de développement logiciel utilisé pour développer des applications et des applets Java. Cela existe physiquement. Il contient JRE + development tools. Le JDK contient une machine virtuelle Java privée (JVM)

et quelques autres ressources telles qu'un interpréteur / loader (java), un compilateur (javac), un archiveur (jar), un générateur de documentation (Javadoc), etc. [10]

III.4.5 AVD

AVD (Android Virtual Device) est un émulateur de terminal sous Android, il simule le lancement d'un terminal virtuel représentant sur la machine du développeur l'écran du téléphone doté du système d'exploitation Android. C'est un outil incontournable pour le développement mobile, il permet au développeur de voir à quoi ressemblera son application sur le futur matériel.

III.5 Langages de programmation

Nous présentons les langages de programmation utilisés pour réaliser notre application :

III.5.1 Java

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisé pour le développement d'applications d'entreprises et mobiles. [11]

III.5.2 XML

Le langage XML (eXtended Markup Language) est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de document que pour la transmission de données entre applications.

Le langage XML dérive de SGML (Standard Generalized Markup Language) et de HTML (HyperText Markup Language). Comme ces derniers, il s'agit d'un langage orienté texte et formé de *balises* qui permettent d'organiser les données de manière structurée. [12]

III.6 Présentation de l'application

III.6.1 Nom de l'application

Notre application portera le nom de « **StudApp** ». « Stud » pour Student (étudiant en Anglais), « App » pour Application.

III.6.2 Logo de l'application

Nous présentons dans la figure III.4 le logo de notre application. La lettre « E » a été choisie pour illustrer « étudiant » ou « étude » dans le logo.



Figure III.4 : Logo de l'application à réaliser

III.7 Présentation des interfaces

Nous consacrons cette partie à la présentation des interfaces principales de notre application mobile StudApp.

III.7.1 Interface de choix de session

Avant toute chose et dès lors que l'application se lance, on doit pouvoir offrir le choix à l'utilisateur de choisir entre deux sessions possibles : session étudiant et session utilisateur. Cette étape est illustrée dans la figure III.5 suivante :

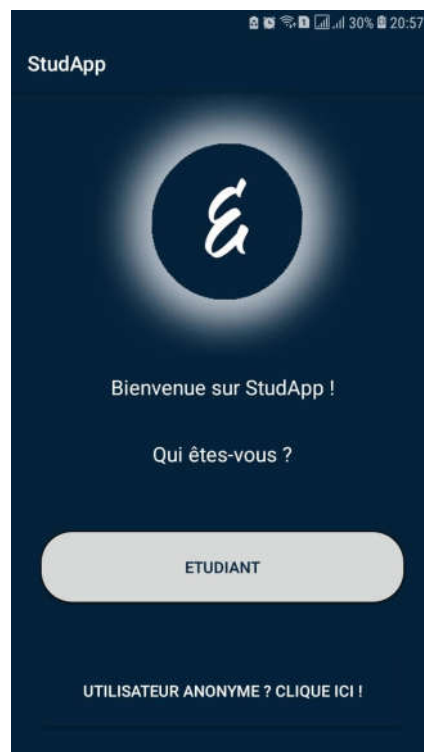


Figure III.5 : Choix de la session

III.7.2 Interface d'authentification

Les étudiants et les administrateurs doivent passer par l'étape d'authentification. Ils doivent donc saisir leur login et leur password, si les informations sont manquantes, on affiche un message d'erreur d'informations manquantes, sinon si les informations sont fausses, on affiche un message d'erreur d'informations fausses. Si par contre les informations saisies sont correctes, l'utilisateur accède à l'interface d'accueil de l'application.

Cette étape est illustrée dans la figure III.6 suivante :

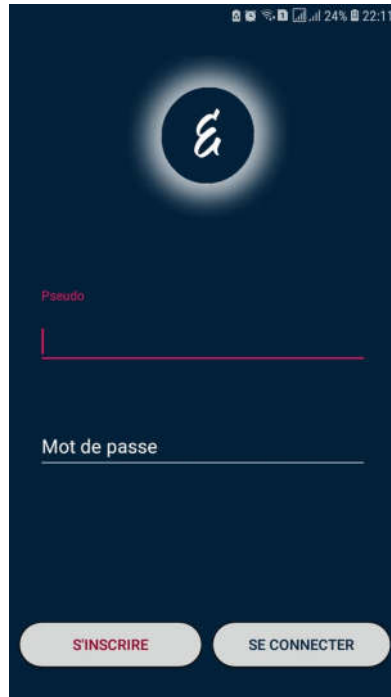


Figure III.6 : Authentification

III.7.3 Interface d'Inscription

Les étudiants qui utilisent l'application pour la première fois n'auront bien évidemment pas encore de compte, ils doivent en créer un via le bouton « S'inscrire » présent dans l'interface d'authentification (voir figure III.6). Cette étape consiste à s'approprier un login, un mot de passe et à saisir diverses informations (Spécialité, section et groupe de l'étudiant) comme montré dans la figure III.7 ci-dessous :

Figure III.7 : Inscription

III.7.4 Accueil Administrateur

Lorsque le duo (Login,Psd) indique qu'un Administrateur s'est connecté, l'interface de gestion de l'application s'affiche comme le montre la figure III.8 suivante :

Figure III.8 : Interface Administrateur

III.7.5 Gestion d'un enseignant

A titre d'exemple, cliquons sur « Gestion d'un enseignant », l'interface suivante apparaît :

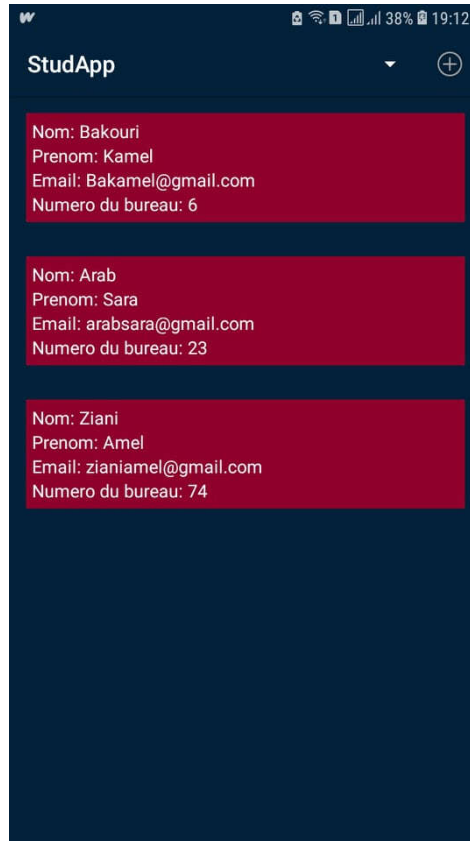


Figure III.9 : Gestion des enseignants

On a trois enseignants sur notre base de données. Pour accéder à l'interface d'ajout, il suffit à l'administrateur de cliquer sur le petit « + » en haut à droite de l'écran. L'interface d'ajout s'affiche.

III.7.6 Ajout d'un enseignant

Pour ajouter un enseignant, il suffit de remplir le formulaire adéquat contenant le nom, prénom, l'email ainsi que le numéro de bureau de l'enseignant comme le détaille la figure III.10 ci-dessous :

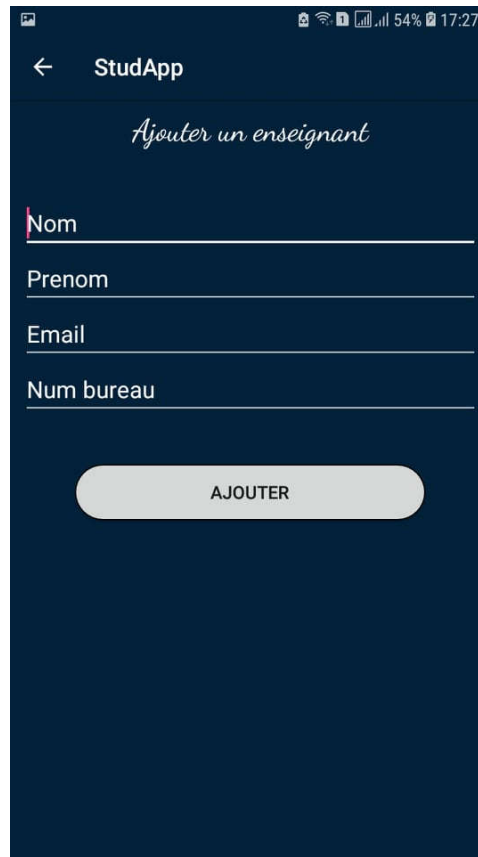
The image shows a mobile application interface for adding a teacher. At the top, there's a dark blue header with a back arrow and the text 'StudApp'. Below this, the title 'Ajouter un enseignant' is written in a light blue, cursive font. The form consists of four text input fields with light blue borders, labeled 'Nom', 'Prenom', 'Email', and 'Num bureau' from top to bottom. At the bottom of the form is a light blue rounded rectangular button with the text 'AJOUTER' in dark blue capital letters. The status bar at the very top shows various icons and the time '17:27'.

Figure III.10 : Ajout d'un enseignant

L'administrateur n'a plus qu'à remplir le formulaire et de cliquer sur « Ajouter ». L'interface mise à jour de la liste des enseignants est affichée.

Remarque : On peut annuler l'ajout en cliquant sur le flèche en haut à gauche de l'écran.

III.7.7 Suppression d'un enseignant

A titre d'exemple, l'enseignante « Arab Sara » part en retraite, on souhaite la retirer de la liste des enseignants. On se place donc à l'interface montrée dans la figure III.9 et on maintient le clic sur le cadre rouge entourant sa case.

Une boîte de dialogue s'affiche :

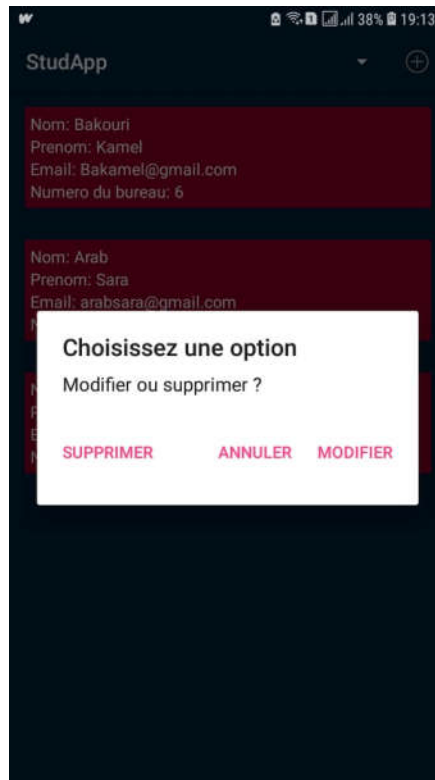


Figure III.11 : Boite dialogue modifier/ supprimer enseignant

Dans notre cas, il s'agit d'une suppression, on clique donc sur « Supprimer », voici le la nouvelle liste des enseignants après cette manipulation :

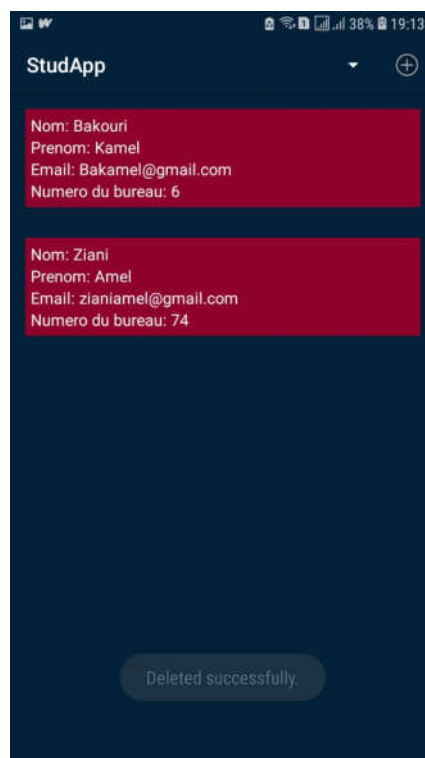
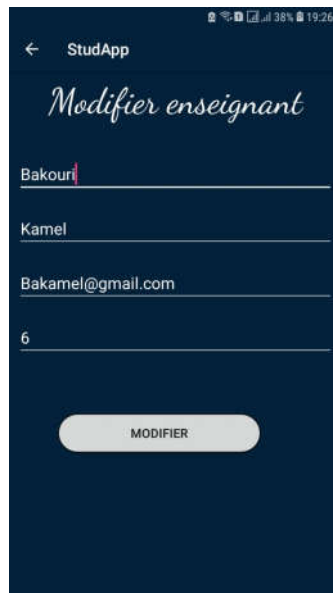


Figure III.12 : Liste des enseignants actualisée

III.7.8 Modifier un enseignant

Toujours à titre d'exemple, l'enseignant « Bakouri Kamel » a oublié le mot de passe de son compte email, son nouveau compte est : « bakourikamelnew@gmail.com ».

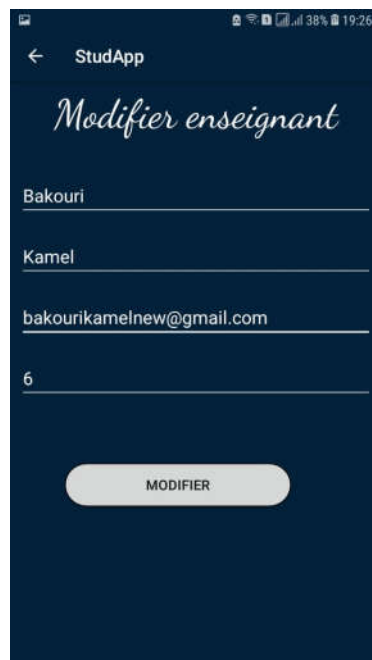
Depuis l'interface illustrée dans la figure III.12, on maintient notre clique sur la case de cet enseignant, la boîte de dialogue modifier/supprimer s'affiche comme dans la figure III.11, cette fois ci, on clique sur « Modifier », voici l'interface de modification :



The screenshot shows a mobile application interface titled 'StudApp' with a back arrow. The main heading is 'Modifier enseignant'. Below it, there are four input fields containing the following text: 'Bakouri', 'Kamel', 'Bakamel@gmail.com', and '6'. At the bottom of the form is a grey button labeled 'MODIFIER'.

Figure III.13 : Anciennes informations enseignant

On saisie les nouvelles informations comme présenté dans la figure III.14



This screenshot is similar to Figure III.13, but the email field has been updated to 'bakourikamelnew@gmail.com'. The other fields ('Bakouri', 'Kamel', '6') and the 'MODIFIER' button remain the same.

Figure III.14 : Modification informations enseignant

En validant les informations avec le bouton « Modifier », on obtient la liste des enseignants modifiée illustrée dans la figure III.15 ci-dessous :

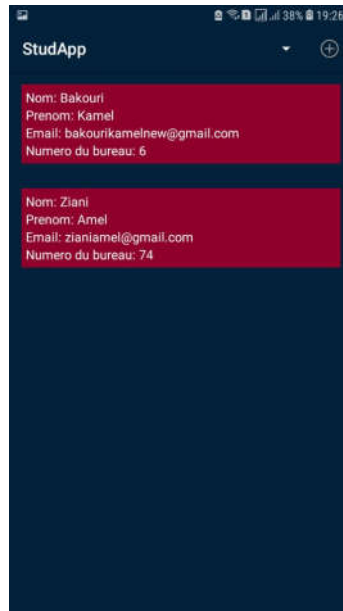


Figure III.15 : Liste enseignants mise à jour

III.7.9 Session Etudiant / Session Utilisateur

La session étudiant diffère de la session utilisateur en un seul point : En effet, l'étudiant peut consulter l'emploi du temps de son groupe qu'il a précisé lors de son inscription. Voici l'interface d'accueil de la session « Etudiant » :



Figure III.16 : Interface d'accueil Etudiant (Accès aux consultations)

Voici l'interface d'accueil de la session « Utilisateur » :



Figure III.17 : Interface d'accueil Utilisateur (Accès aux consultations)

III.7.10 Boutons « Se déconnecter » et « Home »

Partout dans l'application et durant la session « Administrateur » et « Etudiant », se promènent deux boutons « Se déconnecter » et « Home » qui permettent respectivement la déconnexion de l'application et le retour vers l'accueil, et ça, de n'importe quelle interface.



Figure III.18 : Bouton « Déconnexion »



Figure III.19 : Bouton « Home »

III.8 Conclusion

La dernière partie de ce travail s'achève. On a donc de ce fait présenté les divers outils avec lesquels nous avons mené ce projet. De Visual Paradigm qui nous a offert un formalisme de modélisation UML jusqu'à Android Studio, l'IDE de développement en passant par les différents outils (JRE, JDK...etc.) et ainsi les langages de programmations utilisés. Finalement nous avons conclu ce chapitre en présentant les interfaces les plus pertinentes de notre application mobile.

Conclusion générale

La croissance en nombre des dispositifs mobiles de nos jours ne cesse d'être exponentielle.

A travers ce travail, nous avons réalisé une application mobile « StudApp » qui sert les étudiants du département d'Informatique de l'université de Bejaia.

Le contexte qui nous a inspiré à entreprendre ce projet était que les étudiants n'obtiennent aucun encadrement ni aucun véritable guide pour recueillir les informations dont ils ont besoin.

L'objectif de ce travail a donc été de simplifier la vie estudiantine des étudiants du département d'informatique en garantissant une bonne circulation d'informations pertinentes grâce à la simplicité, la fluidité et l'ergonomie des interfaces de l'application.

Nous avons commencé notre projet par une brève introduction où nous y avons présenté notre projet ainsi que la problématique qui a suscité ce dernier. Nous avons tracé les périmètres du système à réaliser en listant ses divers objectifs, préciser notre méthodologie de conception (UP) ainsi que le formalisme de modélisation UML. Mais aussi vaguement décrits les futures interfaces dont se compose actuellement notre application.

Par ailleurs, dans le même chapitre, nous avons établi une étude préliminaire pour identifier les différents acteurs dont se compose le système à réaliser, suivi de la spécifications des besoins fonctionnels à travers des diagrammes de cas d'utilisation en finissant par l'analyse des besoins en présentant les divers diagrammes de séquences.

Ensuite, nous avons accordé tout un chapitre pour la conception de notre projet en établissant le diagramme de classe, en décrivant les attributs de chaque classe et en traduisant ce même diagramme vers le model relationnel en précisant bien sûr, les règles de passages utilisées.

Nous avons donc eu un aperçu de quoi aura l'air notre base de données pour ensuite réaliser le tout en présentant les divers outils utilisés et les interfaces obtenues lors du dernier chapitre consacré à la réalisation.

Ce travail nous a permis d'accentuer nos compétences ainsi que nos connaissances que ce soit en conception ou en programmation. Mais aussi forger l'esprit de coopération pour atteindre un but commun.

En guise de perspective, « StudApp » peut-être enrichie par :

- L'inclusion d'un système de réinitialisation du mot de passe (par email).
- Un système de messagerie entre utilisateurs.
- L'extension de l'application en intégrant les différents départements existants de l'Université de Bejaïa.

Bibliographie

- [1] : Sophnouille, UP : Unified Process, <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/>, 2004, consulté le 2019-06-12
- [2] : Alain Fernandez, Qu'est-ce que UML ?, <https://www.piloter.org/projet/methode/uml.htm>, Novembre 2018, consulté le 2019-06-12.
- [3] : Wilfried-Erisco MVOU-OSSIALAS, Conception et développement d'une application de gestion d'une bibliothèque, https://www.memoireonline.com/02/09/1973/m_conception-et-developpement-dune-application-de-la-gestion-dune-bibliotheque5.html, 2003, consulté le 2019-06-11.
- [4] : P. Roques, Modéliser un site e-commerce, Les cahiers du programmeur, Eyrolles, Paris, 2002.
- [5] : IBM Knowledge Center, Diagrammes de séquence, https://www.ibm.com/support/knowledgecenter/fr/SSRTLW_9.6.0/com.ibm.xtools.sequence.doc/topics/cseqd_v.html, consulté le 2019-06-11.
- [6] : Tutoriel sur les diagrammes de classes UML, <https://www.lucidchart.com/pages/fr/tutoriel-sur-les-diagrammes-de-classes>, consulté le 2019-06-11.
- [7] : Georges Gardarin, Bases de données, Eyrolles, Paris, 2003.
- [8] : Android Studio, <https://android-studio.fr.uptodown.com/windows>, consulté le 2019-06-11.
- [9] : F. Espiau, Créez des applications pour Android, <https://openclassrooms.com/fr/courses/2023346-creez-des-applications-pour-android/2023748-installation-et-configuration-des-outils>, 2019, consulté le 2019-06-11.
- [10] : Différence entre JDK, JRE et JVM, <https://waytolearnx.com/2018/11/difference-entre-jdk-jre-jvm.html>, Novembre 2018, consulté le 2019-06-11.
- [11] : J.M. Doudoux, Développons en Java, <https://www.jmdoudoux.fr/java/dej/chap-presentation.htm>, 2018, consulté le 2019-06-11.
- [12] : O. Carton, L'essentiel de XML, <https://www.irif.fr/~carton/Enseignement/XML/Cours/Annexes/introduction.html>, Octobre 2015, consulté le 2019-06-11.

Résumé

Au bout de ces trois années de licence, nous avons eu pour projet de fin de cycle de concevoir et de réaliser une application mobile sous Android pour la gestion de la vie estudiantine des étudiants du département d'Informatique de l'université de Bejaia. Afin de parvenir à notre but, nous avons utilisé plusieurs technologies parmi lesquelles on cite : La modélisation UML pour décrire les fonctionnalités de notre application, SQLite en tant que base de données interne, Android Studio comme environnement de développement, JAVA et XML comme langages de programmation. On ressort de ce projet avec une importante expérience qui nous a permis d'améliorer nos compétences dans le domaine de la conception et de la programmation.

Mots clés : Application Mobile, UP, UML, SQLite, Android Studio, Java, XML.

Abstract

In terms of our curriculum in computer science license, we were having for project to design and implement a mobile application for managing the daily life of computer science's students of the University of Bejaia. To reach that goal, we used many sophisticated tools and technologies which are: The UML Modeling to describe our application's functionalities, SQLite as an interne database, Android Studio as our development environment, Java and XML as programming languages. With this project, we gained a massive experience which allowed as to improve our abilities in the field of design and programming.

Keywords: Mobile Application, UP, UML, SQLite, Android Studio, Java, XML.