

Université Abderrahmane Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique



Mémoire de fin de Cycle

*En vue de l'obtention du diplôme de Master professionnel en
Informatique*

Option : Administration et sécurité des réseaux

Thème :

Conception et Réalisation d'une Application

Android dédiée aux Étudiants de l'université de Béjaïa

Présenté par :

M^r. Boubekour CHERID ET M^r. Halim BOUHOUÏ

Soutenu le 30/06/2015 devant le jury composé de :

Président :	Mr A. SIDER	U.A/Mira Béjaïa
Examinatrice :	Mme B. GASMI	U.A/Mira Béjaïa
Promoteur :	Mr K. MEHAOUED	U.A/Mira Béjaïa

Béjaïa, juin 2015.

Dédicaces

Ce modeste travail est dédié :

A nos chers parents qui nous ont soutenus et encouragés durant

toute notre scolarité.

A nos frères

A nos enseignants

A nos amis(e)

A toutes les personnes qui nous ont apportés de l'aide.

BOB et HALIM

Remerciements

Nos premiers remerciements s'adressent à **Dieu** le tout puissant qui par sa bonté et sa miséricorde nous a permis d'avoir le courage, la foi et la volonté de mener à bien ce travail.

Nous tenons aussi à remercier Monsieur Kamal MEHAOUED notre encadreur qui ne nous a lésé d'aucune information, qui a été présent à tout moment de la réalisation de ce projet et surtout sans lequel ce modeste travail n'aurait jamais vu le jour, ainsi que les membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant d'examiner notre mémoire et de l'enrichir par leurs propositions.

Nous remercions également tous les professeurs qui ont contribués de près ou de loin à notre formation universitaire, sans oublier toute personne qui nous a aidés à mener à terme notre projet .Merci à tous.

Table des matières

Table des matières	I
Liste des figures	IV
Liste des tableaux	V
Liste des abréviations	VI
Introduction générale	1

Chapitre I : Etat de l'art

Introduction	4
I. Le Système Android	4
I.1 Les Smartphones	4
I.2 Le système d'exploitation Android	4
I.3 Les versions d'Android.....	4
II. Architecture Android	5
II.1 Linux Kernel	6
II.2 Android Runtime	6
II.3 Bibliothèques	7
II.4 Framework de développement	7
II.5 Application	7
III. Les Fonctionnalités	7
III.1 Le Design	8
III.2 La Sécurité	8
Conclusion	8

Chapitre II Présentation de l'organisme d'accueil

Introduction	10
I. Présentation de l'université de Béjaïa	10
II. Définition de E-Learning	10
III. Les moyens mis en œuvre pour la communication et information	10
III. Analyse critique de l'existant	11
Conclusion	11

Chapitre III : Présentation de l'application et méthodologie de Conception

Introduction	13
Partie I : Présentation de l'application	
I.1 Description de l'application	13
I.2 Architecture globale de l'application	13
I.3 Les fonctionnalités proposées par E-UBejaia	14
I.4 Décomposition du projet	15
I.4.1 L'application Android	15
I.4.1.1 Architecture de l'application Android	15
I.4.2 Une application web	16
I.4.2.1 Architecture de l'application web	16

Partie II : Méthodologie de conception

II.1 La méthode OOD de Grady BOOCH	16
II.2 Présentation des outils utilisés dans la méthode de Grady BOOCH	17
II.2.1 Classes et Objet	17
II.2.1.1 Les Objets	17
II.2.1.2 Les Classes	17
II.3 Relations entre objets	17
II.3.1 Les Liens	17
II.3.2 L'agrégation	18
II.4 Relations entre classes	19
II.4.1 L'association	19
II.4.2 L'héritage	19
II.4.2.1 L'héritage simple	19
II.4.2.2 L'héritage multiple	20
II.4.3 La relation entre class et objet	20
II.4.4 Le Polymorphisme	21
Conclusion	21

Chapitre IV : Analyse et conception

Introduction	23
I. Les acteurs du système	23
II. Capture des besoins fonctionnels	23
III. Les cas d'utilisation	24
III.1 Identification des cas d'utilisation	24
III.2 Description textuelle des cas d'utilisation	25
III.3 Relations entre cas d'utilisation	27
III.4 Représentation des diagrammes de cas d'utilisation	29
III.4.2 Diagramme de cas d'utilisation de l'administrateur	30
III.4.3 Diagramme Global de cas d'utilisation du système	31
IV. Diagramme de séquence	32
IV.2 Elaboration des diagrammes de séquence des cas d'utilisation	33
IV.2.1 Les diagrammes de séquence des cas d'utilisation de l'utilisateur	34
IV.2.2 Les diagrammes de séquence des cas d'utilisation de l'administrateur du système	39
V. Les Diagrammes de classe	40
VI. Le modèle relationnel	40
Conclusion	41

Chapitre V : Réalisation

Introduction	43
I. Outils et environnement de développement	43
I.1 IDE Eclipse	43
I.2 Le Plugin ADT (Android Développement Tools)	43
I.3 Software Development Kit (SDK)	43
I.4 MySQL	44
II. Les langages de programmation	44
II.1 Le langage JAVA	44

II.2 PHP	44
II.3 JSON (JavaScript Object Notation)	44
II.4 SQLite	45
II.4.1 Avantages de SQLite	45
II.4.2 Inconvénients de SQLite	45
II.5 XML	45
II.6 Les Bases de données web	46
III. Les étapes de développement	46
III.1 Création d'un AVD	46
III.2 Création d'un projet Android	47
III.3 Etapes de la mise en marche de notre application	48
IV Fonctionnement détaillé de l'application	49
IV.1 L'authentification	50
IV.2 Le menu principal de l'application	51
IV.3 Consultation Cours	51
IV.4 Consultation des emplois du temps	51
IV.5 Consultation de l'actualité	52
IV.6 Echanger	53
Conclusion	53
Conclusion Générale	54
Liste Bibliographique	55

Table des figures

Figure 1.1 : Les composants principaux du système d'exploitation Android	6
Figure 3.1 : Architecture de l'application E-UBejaia.....	14
Figure 3.2 : Architecture basique d'une application Android	15
Figure 3.3 : Représentation de liens	18
Figure 3.4 : Représentation d'une agrégation	18
Figure 3.5 : Héritage simple	20
Figure 3.6 : L'héritage multiple	20
Figure 4.1 : Représentation graphique de la relation d'inclusion	28
Figure 4.2 : Représentation graphique de la relation d'extension	28
Figure 4.3 : Représentation graphique de la génération	28
Figure 4.4 : Diagramme de cas d'utilisation de l'utilisateur	29
Figure 4.5 : Diagramme de cas d'utilisation de l'administrateur.....	30
Figure 4.6 : Diagramme Global de cas d'utilisation du système	31
Figure 4.7 : Diagramme de séquence du cas « Authentification » de l'utilisateur	34
Figure 4.8 : Diagramme de séquence du cas « Consultation des cours ».	35
Figure 4.9 : Diagramme de séquence du cas « Consultation de l'emploi du temps ».....	36
Figure 4.10 : Diagramme de séquence du cas « Consultation des actualités»	37
Figure 4.11 : Diagramme de séquence du cas « Echange ».	38
Figure 4.12 : « Diagramme de séquence du cas « Authentification » de l'administrateur.	39
Figure 4.13 : « Diagramme de classe du système.	40
Figure 5.1 : Principaux outils de communication	45
Figure 5.2 : Création d'un AVD android sous windows	46
Figure 5.3 : Emulateur android	47
Figure 5.4 : Création d'un projet Android sous eclipse	48
Figure 5.5 : La base de données E-UBejaia sous MySQL	49
Figure 5.6 : Interface d'authentification de l'application	50
Figure 5.7 : Menu principal de l'application	51
Figure 5.8 : Page d'actualités	52
Figure 5.9 : interface d'échange entre les émulateurs	53

Liste des tableaux

Tableau 1.1 : Historique des versions d'Android depuis sa création	5
Tableau 4.1 : Les cas d'utilisation de l'administrateur pour l'application android	24
Tableau 4.2 : Les cas d'utilisation de l'utilisateur pour l'application android	25
Tableau 4.3 : Description textuelle du cas d'utilisation : cas authentification.....	26
Tableau 4.4 : Description textuelle du cas d'utilisation : Consultation	26
Tableau 4.5 : Description textuelle du cas d'utilisation Exploitation de l'application.....	27

Liste des Abréviation

API : **A** pplication **P** rogramming **I** nterface

AVD : **A** ndroid **V** irtual **D** evices

EDI : **E** nvironnement de **D** éveloppement **I** ntégré

HTTP : **H** yper **T** ext **T** ransfer **P** rotocol

IHM : **I** nterface **H** omme **M** achine

JDBC : **J** ava **D** ata **B** ase **C** onnectivity

JDK : **J** ava **D** eveloppement **K** it

JSON : **J** ava **S** cript **O** bject **N** otation

OOD : **O** bject **O** riented Analysis and **D** esign

SDK : **S** oftware **D** eveloppement **K** it

SGBD : **S** ystème de **G** estion de **B** ases de **D** onnées

UML: **U** nified **M** odeling **L** anguage

XML : e **X** tensible **M** arkup **L** anguage

Introduction Générale

Le marché de la téléphonie portable connaît actuellement une véritable révolution, menée par la société Apple en commercialisant son premier téléphone intelligent : *l'iPhone*. Apple a su mettre en avant son produit en ajoutant au téléphone de nouvelles fonctionnalités et en créant de nouveaux besoins.

Suite à ça, le marché de la nouvelle gamme de la téléphonie mobile nommée « Smartphones » qui veut dire téléphone intelligent a connu un véritable essor dans lequel les acteurs habituels (Windows et Symbian) essaient de s'engouffrer.

Google, ayant réalisé le potentiel de ce marché, a décidé de s'y introduire en rachetant une startup travaillant sur un système d'exploitation ouvert pour terminal mobile en 2003 qui portait le nom de « Android ».

Le point fort de ce système est qu'il est open source. Tout individu désirant l'exploiter et l'améliorer est le bienvenu. Et cela est devenu très bénéfique à Google par l'apparition de ce système sur de nouveaux matériels tels que les montres connectées et les voitures et même sur les appareils électroménagers.

Dans le cadre de notre projet de développement d'une application mobile, nous étions menés à explorer ce nouveau système d'exploitation pour mobiles, Android.

Notre travail consiste à la conception et l'implémentation d'une application mobile sous Android qui facilitera aux étudiants l'accès à plusieurs fonctionnalités et services pour les aider durant leurs scolarités. Cela dans le cadre d'obtention du diplôme de Master Professionnel en Informatique Option Administration et Sécurité des Réseaux.

Ainsi, nous articulons notre travail autour de cinq chapitres :

Le premier chapitre s'intitule « **état de l'art** » où nous présenterons quelques concepts du système Android et ses fonctionnalités.

Dans le second chapitre titré « **Organisme d'accueil** » nous aborderons l'université Abderrahmane mira de Béjaïa qui est notre cas d'étude, par suite nous citerons les différents moyens mis en place par cette dernière pour assurer la communication avec les étudiants scolarisés à l'université. Nous essayerons d'analyser des méthodes et trouver une salutation conviviale aux failles existantes.

Le troisième chapitre nommé « **Présentation de l'application et méthodologies de conception** » est partagé en deux grandes parties : La première partie est consacrée à la présentation de l'application à développer, son architecture ainsi que ses fonctionnalités. Dans la seconde partie, nous aborderons la méthode orientée Objet de Grady BOOCH. Nommée OOD, qui sert à concevoir des applications et des systèmes d'information logiciels. Ses outils seront bien expliqués dans cette partie.

La phase Conception est présentée dans le chapitre quatre qui est « **Analyse et Conception** », il aura pour objectif la modélisation de l'application via le langage UML. En définissant tout d'abord les acteurs et les besoin fonctionnels de l'application. Nous acheminerons le travail à travers quelques diagrammes UML pour ensuite passer à l'étape de la réalisation.

Dans le cinquième et dernier chapitre, nous allons enfin passer à la « **réalisation** » de notre application, et dans le quel nous citerons les différents outils et langages de programmation ayant servi à l'implémentation du système, par la suite nous exposerons quelques interfaces et fonctionnalités de l'application.

Enfin, nous achèverons notre mémoire par une conclusion générale et quelques perspectives.

Chapitre I

Etat de l'art

Introduction

Dans ce tout premier chapitre, nous allons présenter l'environnement Android, l'architecture de son système *Open Source* ainsi que ses fonctionnalités. Ensuite nous aborderons ses composants et leurs utilités.

I. Le Système Android

Android est un système d'exploitation Open Source pensé pour les téléphones mobiles (Smartphones) et développé par Open Handset Alliance (OHA) sous l'autorité de Google et composé d'enivrantes 140 sociétés telles que Samsung. [1]

I.1 Les Smartphones

Un Smartphone, ou téléphone intelligent, est un téléphone mobile évolué disposant des fonctions d'un assistant numérique personnel, d'un appareil photo numérique et d'un ordinateur portable.[2]

I.2 Le Système d'exploitation D'Android

Android était développé par la startup Android Inc. en 2003, puis racheté par Google en 2005. Pour pouvoir réaliser un système complet, ouvert et gratuit dans le monde du mobile, une coalition de trente cinq (35) entreprises évoluant dans l'univers du mobile, dont Google, a été créé. Ce rassemblement se nomme l'Open Handset Alliance (OHA) et composé aujourd'hui de 80 membres. Cette alliance a pour but de développer un système open source : c'est-à-dire dont les sources sont disponible librement sur internet pour l'exploitation sur mobile, Android.

Android est à l'heure actuelle le système d'exploitation pour Smartphones et tablettes le plus utilisé. Les terminaux visés par Android inclus les téléphones portables, Netbook/Smartbook, tablettes multimédia, automobile, GPS, Réfrigérateur, Téléviseur.

I.3 Les versions D'Android

Android a débuté avec la sortie de la première version 1.0 en septembre 2008, il a connu plusieurs mises à jour depuis cette version. Elles servent à corriger des bugs et ajouter des nouvelles fonctionnalités.

Au moment de la rédaction de ce mémoire, la version d'Android la plus récente est la 5.1 (Lollipop). Sortie le 3 novembre 2014, *Android 5.1 Lollipop* est une évolution majeure d'Android qui propose de nombreuses modifications et nouveautés. Cette version étend sa disponibilité sur de nouveaux supports tels que la télévision, la voiture ou les montres connectées.

Le tableau suivant représente l'historique des versions d'Android depuis sa création :

Version	Nom	Date de sortie	Quelques évaluations
5.0	Lollipop	03/11/2014	Première fois sur Android TV, montres connectés et sur voitures.
4,4	Kit Kat	14/04/2014	Interface translucide, Framework pour imprimer, Framework pour la gestion des fichiers.
4,1	Jelly Bean	09/07/2012	Assistance vocale, accessibilité : mode gestuel 'Braille', WIFI-Direct service discovery, vsync timing
4.0	IceCream Sandwich	19/11/2011	WI-FI direct, Bluetooth Health Devie profile, Control over network data, Grid Layout.
3,2	Honeycomb	15/07/2011	Support des processeurs Qualcomm, Support des tablettes tactiles de 7 pouces
2,3	Gingerbread	06/12/2010	Support de la VoIP et SIP. Gestionnaire de téléchargement, support de plusieurs cameras.
2,2	Froyo	20/05/2010	Implémentation de JIT, partage de connexion USB.
2.0	Eclair	26/10/2009	Bluetooth, support de plus de taille d'écran.
1,6	Donut	15/09/2009	Google navigation (GPS gratuit)
1,5	Cupcake	30/04/2009	Envoi de vidéos vers YouTube et Picasa, rotation automatique
1,1	Banana bread	09/02/2009	Support pour sauvegarder les fichiers attachent aux MMS.
1.0	Apple pie	23/09/2008	Début de l'aventure Android.

Tableau 1.1 : Historique des versions d'Android depuis sa création

II. Architecture Android

La plateforme Android est composée de cinq couches principales, basée sur une architecture autour d'un noyau Linux, ce dernier offre des services système de base tel que la sécurité, la gestion de la mémoire, la gestion des processus. Un ensemble de bibliothèques écrites en C/C++ jouent le rôle d'intermédiaire entre le noyau et la machine virtuelle qui se trouve dans Android Runtime. Dans la couche au dessus, la couche logiciel appelée Framework écrite en Java et qui fait la mutualisation entre les applications Java. Enfin, à un

niveau plus supérieur se situe un ensemble d'applications sous forme de paquets APK (*Android Package*). [3]

La figure suivante offerte par Google illustre explicitement les différents composants de l'architecture Android.

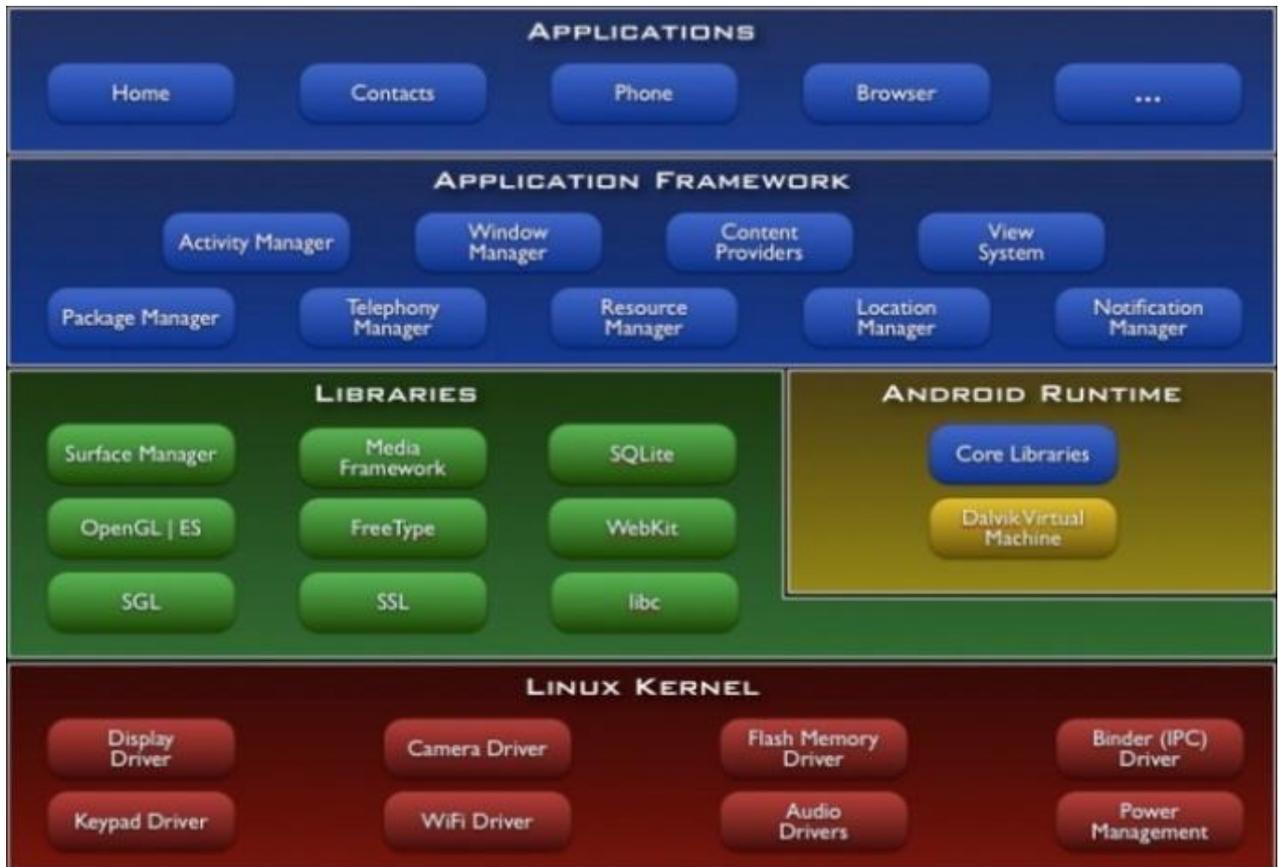


Figure 1.1 : Les composants principaux du système d'exploitation Android.

II.1 Linux Kernel

La couche la plus basse de l'architecture d'Android, elle est basée sur un Kernel linux 2.6 mais ce n'est pas linux. Il ne possède pas de système de fenêtrage natif (X Windows system). La glibc n'étant pas supportée, Android utilise une bibliothèque libc customisée appelée Bionic libc.

II.2 Android Runtime

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

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik. Dalvik a été écrit pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik exécute des fichiers dans l'exécutable Dalvik (. DEX), un format optimisé pour ne pas encombrer la mémoire. La machine virtuelle est la base de registres et fonctionne grâce aux classes compilées par un compilateur Java et transformées dans le format DEX.

La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

II.3 Bibliothèques

Android dispose d'un ensemble de bibliothèques C /C++ utilisées par les différents composants du système Android. Elles sont offertes aux développeurs à travers le Framework Android.

II.4 Framework de développement

Un Framework est une plateforme de développement ouverte, Android offre une possibilité aux développeurs de créer des applications extrêmement riches et innovantes. Les développeurs sont libres de profiter du matériels périphériques et information sur la localisation d'accès, exécuter des services d'arrière-plan, définir des alarmes, ajouter des notifications à la barre d'état.

II.5 Application

La couche la plus haute de l'architecture, Android est fourni avec un ensemble d'applications dont un client mail, une application SMS, un calendrier, un service de cartographie, un navigateur.etc.

III. Les Fonctionnalités

Android a été conçu pour intégrer au mieux les applications existantes de Google comme le service de courrier Gmail, l'agenda Google Calendar ou encore la cartographie Google Maps. Des nouvelles fonctionnalités rajoutées dans chaque nouvelle version d'Android. [4]

Nous citerons quelque nouvelles fonctionnalités de la version Android Lollipop 5.1 dans quelques points :

III.1 Le Design

Ce design est audacieux, coloré et se caractérise par des animations cohérentes et une interface multiplateforme s'adaptant à l'ensemble des appareils existants mais aussi à des appareils nouveaux (mobile, montre, TV, voitures).

III.2 La Sécurité

Le chiffrement du contenu de l'appareil est activé par défaut, ce qui protège les données de l'utilisateur en cas de perte ou de vol. La fonctionnalité Kill Switch a été ajoutée, permettant le blocage du téléphone à distance pour empêcher la réinitialisation d'usine. Le fonctionnement de SELinux a été renforcé et le déverrouillage intelligent est possible avec les appareils de confiance comme le bluetooth.

Conclusion

Dans ce chapitre, nous avons fait une étude de l'art sur Android tout en présentant un bref historique de son évolution, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et son architecture, à savoir les principaux composants du système.



Chapitre II

Présentation de l'organisme
D'accueil

Introduction

Dans ce deuxième chapitre, nous aborderons notre cas d'étude. Qu'est l'université de Béjaïa. Nous allons traiter spécifiquement, les méthodes mises en place par l'université pour assurer la communication et l'information entre l'administration et les étudiants. Puis nous essayerons d'établir un compte rendu sur l'efficacité de ces méthodes.

I. Présentation de l'université de Béjaïa

L'Université de Béjaïa, créée en octobre 1983, est un établissement public pluridisciplinaire. Elle compte aujourd'hui plus de 40 900 étudiants, 1442 enseignants et 1079 personnels techniques et administratifs. Elle se déploie aujourd'hui sur deux sites : Targa Ouzemmour et Aboudaou. Elle porte le nom du *Colonel Abderrahmane Mira*, Répartis sur huit facultés. [5]

II. Définition de E-Learning

Le E - Learning, ou apprentissage en ligne, est une méthode d'apprentissage à distance qui repose sur l'utilisation des nouvelles technologies qui offrent des moyens de communication et de transfert d'information à un coût très réduit et optimal via des supports électroniques (Cdrom, Internet, intranet, support multimédia etc.).

Afin de faciliter l'utilisation des ressources offerte par la nouvelle technologie dans le domaine d'apprentissage à distance, des applications web (LMS : Learning management system) ont été développées en se basant sur des méthodes d'apprentissage avancées dans le but de compléter l'enseignement présent. [6]

L'université de Béjaïa a opté pour la plate-forme Moodle pour mettre en place sa solution E-Learning.

III. Les moyens mis en œuvre pour la communication et information

L'université de Béjaïa a mis en place plusieurs outils pour assurer une bonne communication administration-étudiants. Parmi eux nous citons :

- **E-Learning de l'université de Béjaïa:** L'université de Béjaïa comme toute autre université digne de ce nom, aspire à un enseignement de qualité, tout en suivant la nouvelle technologie. La révolution informatique offre des moyens considérables permettant une bonne circulation d'information et facilitant la communication entre les différents acteurs de notre université. De ce fait, elle est mise en place par la Section E-Learning accessible depuis le net par l'adresse <http://e-learning.univ-bejaia.dz>. Elle est organisée sous forme de rubriques concernant des thèmes et cela dans une interface simple d'utilisation.[6]
- **Un Réseau INTRANET :** l'université de Béjaïa est dotée d'un réseau local intranet sur les deux campus.. Ce réseau est connecté à Internet pour assurer une connexion externe. Il est géré par le Centre des Systèmes et Réseaux d'Information, de Communication de Télé-enseignement et de l'Enseignement à Distance. Son but est d'assurer une connexion permanente entre des différents services de l'université tels que le service pédagogique et la scolarité. [7]
- **Autres moyens :** l'affichage classique, sur les tableaux d'affichage des départements et facultés, reste le plus utilisé à ce jour.

IV. Analyse critique de l'existant

Après analyse des différents systèmes de communication existant au sein de l'université de Bejaia, nous avons établis une liste de plusieurs anomalies qui sont :

- La difficulté d'accès à la plateforme E-Learning dû à la surcharge des serveurs.
- Les ressources sur la plateforme ne sont pas sauvant mises à jour.
- Le réseau intranet souffre de manque d'infrastructures et d'efficacité.
- Les informations affichées sur les tableaux d'affichage ont une durée de vie très coutes.

En effet, elles sont souvent arrachées par des étudiants mécontents.

Conclusion

Dans ce chapitre, nous avons présenté brièvement l'université de Béjaïa. Puis nous avons essayé d'établir un état des lieux sur les outils mit par cette dernière pour assurer une communication administration-étudiants. Notre analyse nous a permit de distinguer des failles dans les systèmes mit en place, c'est pour cela que nous suggérons une application mobile qui permettra de remédier au failles existantes.

Chapitre III

**Présentation de l'application et
Méthodologie de Conception**

Introduction

Dans ce chapitre, nous allons présenter l'application que nous allons développer, son architecture logicielle, ainsi que ses fonctionnalités. Puis, nous aborderons la méthode OOD de Grady BOOCH avec tous ses outils et concepts que nous utiliserons pour la conception de notre application.

Partie I : Présentation de l'application

I.1 Description de l'application

Le projet à réaliser consiste à développer une application mobile avec la plateforme android, celle-ci offrira aux étudiants de l'université de Bejaia plusieurs fonctionnalités et services pour les aider durant leurs scolarités. Le nom que nous avons choisi pour notre application est U-EBejaia.

L'application possède une interface moderne et ergonomique ce qui va faciliter l'accès aux étudiants pour la navigation et la manipulation de l'application, tout en proposant de nombreuses manipulations avancées.

Les ressources seront stockées dans une base de donnée, elles seront accessibles aux étudiants via cette application.

Enfin, un fort accent est mis sur l'interaction avec une application web conçu, permettant aux utilisateurs d'accéder facilement à l'application via internet.

I.2 Architecture globale de l'application :

E-UBejaïa est une application avec deux bases de données. Une base de données interne (SQLite) et une base de données externe (web) qui se trouve sur un serveur de bases de données distant. Cette architecture nécessite l'intégration d'un serveur web entre l'application client et le serveur de bases de données. D'où l'architecture à trois niveaux de notre application (architecture 3-tiers) et qui partagée entre:

- **Le client Android** : notre application android.
- **Application web** : permet d'accéder à la base de données de notre application depuis un ordinateur.
- **Le serveur de base de données** : fournit les données au serveur web.

- **Le serveur Web** : Vu que les données seront communiquées entre deux environnements hétérogènes, le rôle principal du serveur web est de gérer la communication entre le client Android et le serveur de base de données.

Voici un schéma de notre architecture qui explique la communication entre l'application Android, le serveur et l'application web. Nous allons décrire dans la partie suivante son fonctionnement.

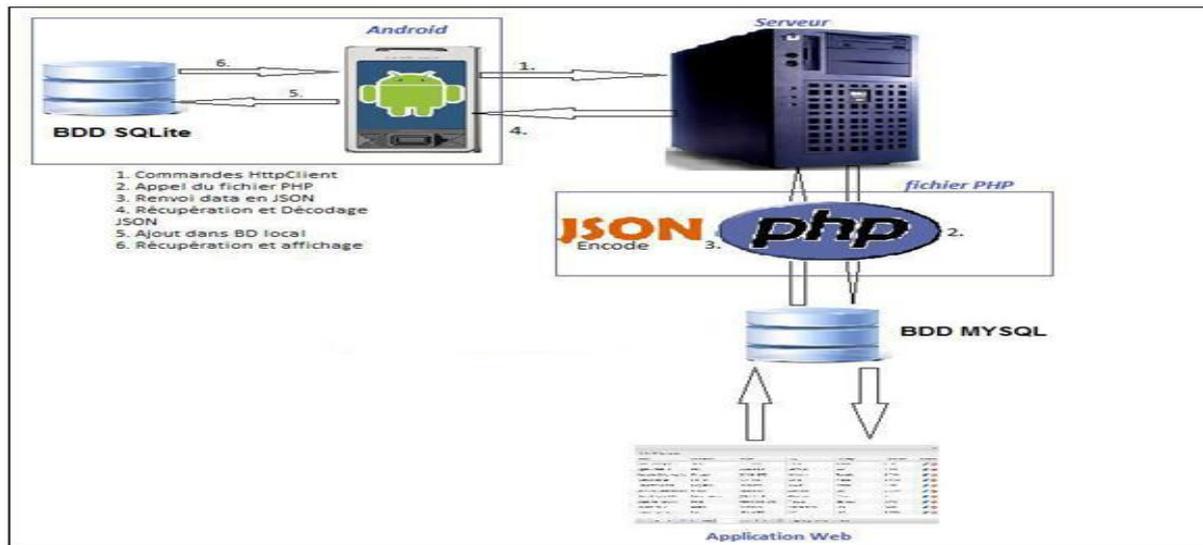


Figure 3.1 : Architecture de l'application E-UBejaia.

I.3 Les fonctionnalités proposées par E-UBejaia

Dans une aussi grande université telle que l'université de Bejaïa, il serait très difficile pour les étudiants d'accéder aux informations et à leurs cours. C'est pour cela, l'application U-EBejaia pourrait être désormais la solution intelligente et le guide pour les étudiants de l'université de Bejaïa.

Les fonctionnalités intégrées dans notre application sont :

- **Emplois du temps** : Les emplois du temps de toutes les filières et de tous les niveaux d'études seront affichés de façon détaillée.
- **Echanges** : Une plateforme de messagerie instantanée sera intégrée à notre application pour permettre aux étudiants d'échanger des messages courts entre eux.
- **Actualité** : un onglet de notre application est réservé pour l'espace d'actualités sur les différents campus de l'université.

- **Cours** : des résumés sur les cours de chaque module peuvent être téléchargés à partir de l'application.
- **Notifications** : Envoyer par les administrateurs pour informer les étudiants sur des séances de rattrapages, consultations etc.
- **Mettre le téléphone automatiquement sur vibreur pendant les heures des cours.**

I.4 Décomposition du projet

Notre projet se décompose en deux grandes parties:

I.4.1 L'application Android

C'est une application qui fonctionne sur un Smartphone qui interagit avec la base de données, elle implémente les fonctions de base, telles que la consultation de l'emploi du temps, le suivi de l'actualité des différentes activités de l'université, l'échange de message etc.

I.4.1.1 Architecture de l'application Android

Le langage XML est bien utilisé pour le stockage de document ainsi que pour la transmission des données entre applications. Le fichier XML permet la déclaration des éléments de l'interface de l'application comme les boutons et les champs de saisies, ainsi que les attributs des différentes variables.

Le fichier JAVA permet le contrôle des éléments déclarés dans le fichier XML. Il est directement lié à XML par une ligne de code, qui lui permet de chercher les éléments dans ce fichier XML.

Voici donc l'architecture basique de l'application Android, sans Base de données.

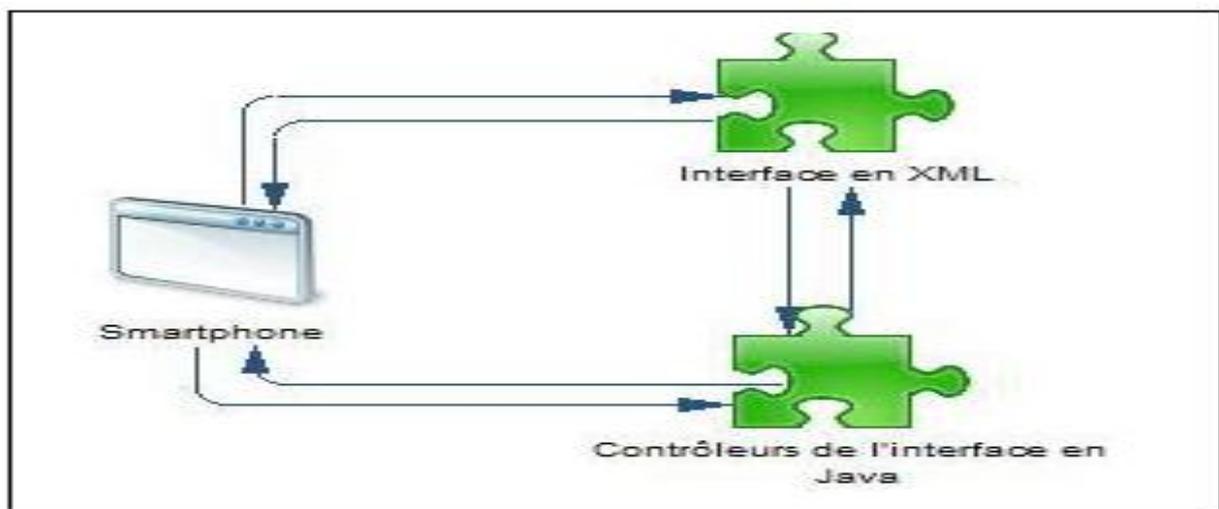


Figure 3.2 : Architecture basique d'une application Android

I.4.2 Une application web

Elle est hébergée sur un serveur applicatif distant qui communique avec l'application Android et la base de données en permanence, pour assurer des fonctions comme le chargement et la mise à jour des cours ainsi que des inscrits sur le site de l'application.

I.4.2.1 Architecture de l'application web

Nous allons réaliser une application web en PHP et HTML et MySQL comme SGBD, ce dernier étant parmi les plus utilisés avec PHP. Cette application nous permettra d'effectuer les tâches suivantes:

- L'authentification par l'administrateur.
- Mise à jours des cours et des emplois du temps.
- Mises à jour des actualités.
- Mise à jour des données des étudiants.

Partie II : Méthodologie de conception

II.1 La méthode OOD de Grady BOOCH

La méthode OOD de Booch est une technique utilisée en génie logiciel. C'est une méthode de conception orientée objets et qui possède tout les outils nécessaires à la modélisation objets. Cette méthode permet de faciliter l'implémentation de programmes dans des langages de programmation orientée objet, ainsi que de représenter les différentes phases du développement d'un projet. Pour les programmeurs, l'avantage est d'obtenir une modélisation d'une solution qui soit indépendante de son implémentation dans un langage particulier par exemple Java, C++, PHP...etc. Elle a été développée par Grady Booch avec la coopération Rational Software (qui fait maintenant partie d'IBM).

L'aspect de la notation de la méthode Booch a été remplacé maintenant par l'Unified Modeling Language (UML), qui dispose d'éléments graphiques à partir de la méthode Booch long avec des éléments de la technique objet de modélisation (OMT) et du génie logiciel orienté objet (OOSE).

La méthodologie de conception Booch, comme OMT, est largement utilisé et a une ample documentation et des outils de soutiens. [8]

II.2 Présentation des outils utilisés dans la méthode de Grady BOOCH [9]

II.2.1 Classes et Objet

Les classes et les objets sont considérés comme les éléments les plus importants dans la modélisation orientée objet de la méthode OOD de Grady Booch.

II.2.1.1 Les Objets

Un objet doit posséder les caractéristiques suivantes:

- Un état
- Un comportement
- Identité

II.2.1.2 Les Classes

Une classe est un ensemble d'objets qui partage les mêmes caractéristiques et les mêmes comportements. La classe déclare des attributs représentant l'état des objets et des méthodes représentant leur comportement.

II.3 Relations entre objets [9]

La relation entre deux objets quelconques englobe les hypothèses que chacun fait à propos de l'autre, y compris les opérations qui peuvent être effectuées et les résultats qui en découlent.

II.3.1 Les Liens

Le terme lien est conçu par Rumbaugh, qui le définit comme connexion conceptuelle physique entre les objets. Un objet collabore avec d'autres objets à travers ses liens. En d'autres termes, un lien désigne une association spécifique à travers laquelle un objet (le client) applique les services d'autres objets (le fournisseur), ou à travers laquelle un objet peut naviguer vers d'autres objets. Dans le contexte de la figure 3.3, Le contrôleur de débit agit comme un objet de contrôleur, l'afficheur agit comme un objet de serveur, et Valve agit comme un proxy.

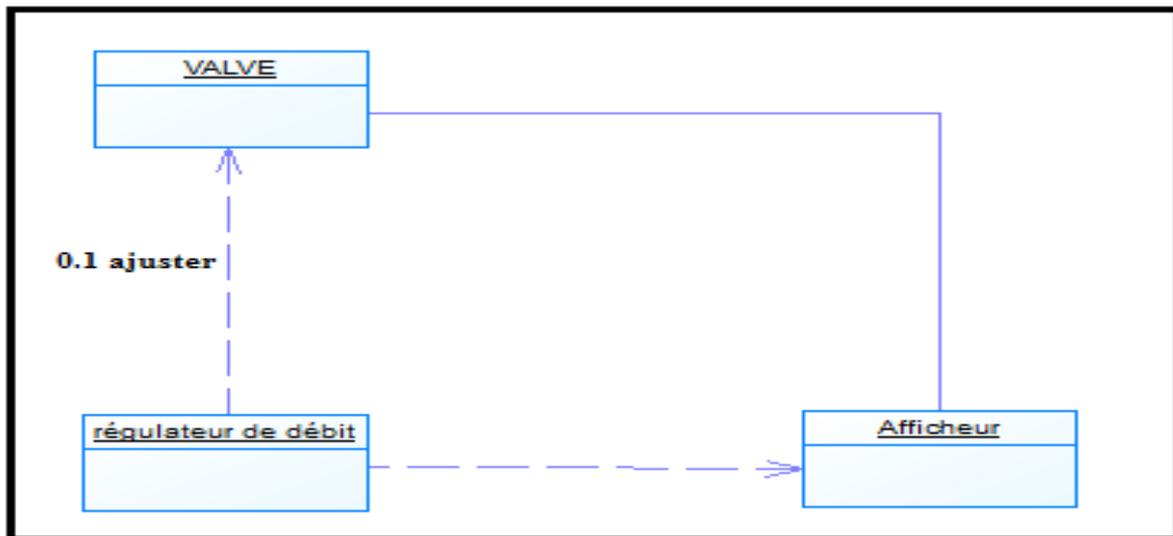


Figure 3.3 : Représentation de liens.

II.3.2 L'agrégation

Les liens sont conçus pour désigner des relations paires à paires entre les objets. Tandis que l'agrégation désigne toute une hiérarchie entre les objets, avec la possibilité de naviguer à partir de l'ensemble. En ce sens, l'agrégation est un type spécialisé d'association. Elle permet de définir une entité comme étant un assemblage de plusieurs sous-entités. Par exemple, comme le montre la Figure 3.4, l'objet Contrôleur de Température a un lien vers l'objet Température Rampe ainsi que l'objet Chauffage. L'objet Contrôleur de Température est donc l'ensemble des Chauffage est de ses parties.

Par voie de conséquence, un objet qui est une partie d'un autre objet a un lien vers son agrégat.

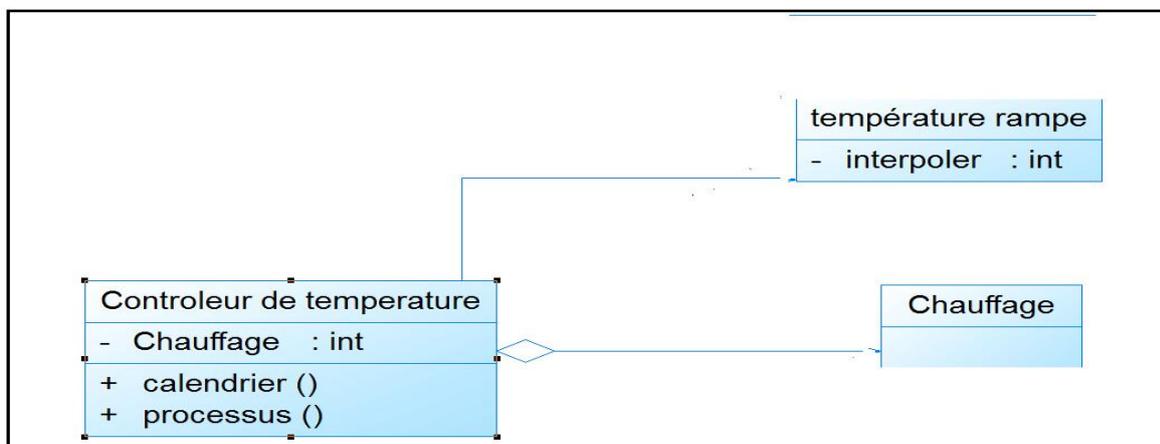


Figure 3.4: Représentation d'une agrégation.

II.4 Relations entre classes [9]

Nous utilisons les relations entre classes pour indiquer une sorte de partage entre les différentes entités, mais aussi pour spécifier certaines connexions symétrique entre classes. En tout, il existe trois types de base de rapports entre les classes. Le premier est la généralisation ou la spécialisation, désignant une relation " est une ". Par exemple : une voiture est un véhicule, ce qui signifie qu'une voiture est une sous classe de véhicule. Le deuxième est l'ensemble ce qui désigne une "partie de ", un écran fait partie d'un ordinateur, ce qui veut dire qu'un écran est une partie d'un ordinateur. Le troisième est l'association, ce qui décrit une certaine dépendance sémantique entre certaines classes.

II.4.1 L'association

Parmi ces différents types de rapports de classe, les associations sont les plus généraux mais aussi les plus faibles sémantiquement. L'identification des associations entre des classes est souvent une activité d'analyse et de conception au début, ensuite les associations faibles seront affinées et remplacées par des rapports de classes plus concrets.

II.4.2 L'héritage

L'héritage, peut-être le plus intéressant de ces relations sémantiques concrètes, il est utile pour exprimer les relations de généralisation et de spécialisation, cependant, l'héritage est un moyen insuffisant pour exprimer toutes les relations qui peuvent exister entre les abstractions clés dans un problème donné (domaine).

II.4.2.1 L'héritage simple

L'héritage simple est une relation entre classes qui définit le comportement ou la structure d'une seule succession dans le cas de l'héritage. Nous appelons la classe qui hérite d'une autre classe une sous-classe.

Dans l'exemple représenté dans la figure suivante, les classes Chat et Chien sont des sous-classes de la superclasse Animal. L'héritage dans cet exemple est simple, car les deux sous-classes héritent d'une seule superclasse qui est Animal.

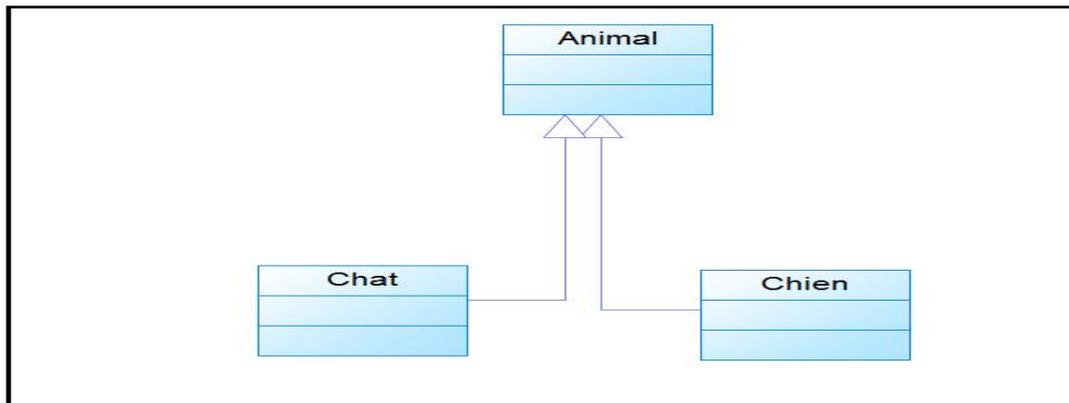


Figure 3.5 : Héritage simple.

II.4.2.2 L'héritage multiple

Contrairement à héritage simple, dans l'héritage multiple, une sous-classe hérite de plusieurs superclasses.

Dans l'exemple de la figure 3.6, la sous-classe Cheval hérite de deux superclasses qui sont Quadrupède et Herbivore. Qu'on peut interpréter que le cheval est un quadrupède et herbivore en même temps. Et nous déduisant que l'héritage multiple élargie le domaine de spécialisation des entités.

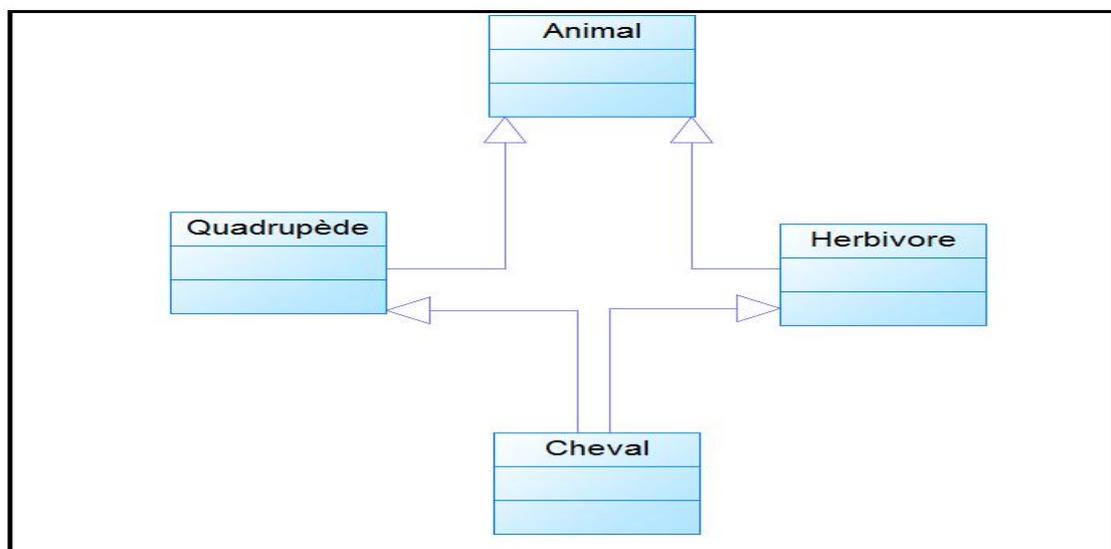


Figure 3.6 : L'héritage multiple.

II.4.3 La relation entre class et objet

Comme on la déjà définie, une class contient un ensemble d'attributs qui la caractérise, de ce fait un objet de cette class est construit a partir de l'assignation d'une valeur pour chaque attribut de cette class, exemple : la class employé contient les attributs nom, prénom, adresse,

numéro de sécurité social, pour construire un objet (instance) de cette class on affecte les valeurs Ali, Mohamed, Alger, 06125 respectivement aux attributs de la class employé.

II.4.4 Le Polymorphisme

En décortiquant le mot polymorphisme on peut déjà avoir une idée sur sa signification, ce mot est composé de deux parties, la première étant **Poly** qui signifie **plusieurs** et **morphisme** qui signifie **forme**. Autrement le Polymorphisme est la faculté que possède un objet pour changer de forme selon des contextes déferents.

Conclusion

Dans ce chapitre, nous avons d'abord abordé la définition globale de notre application où nous avons détaillé ses fonctionnalités ainsi que ses objectifs. Puis nous avons donné un bref aperçu sur la méthode OOD de Grady BOOCH que nous allons adopter pour la modélisation et la conception de notre application.



Chapitre IV

Analyse et Conception

Introduction

La conception est une étape préliminaire et primordiale qui doit précéder l'étape de développement de toute application informatique. Nous utiliserons l'aspect orienté objet pour décrire la conception de notre application à travers la méthode OOD de Grady BOOCH. Nous commencerons par capter les besoins fonctionnels de l'utilisation, puis, à travers différents diagrammes, nous achèverons la conception logique de l'application.

I. Les acteurs du système

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant ou en recevant des messages susceptible d'être porteurs de données.

Les acteurs qui interagissent avec le système de l'application à développer sont :

- **L'administrateur du système** : le rôle principal de l'administrateur est la gestion de l'application web et de la base de données, il s'occupe aussi de la mise à jour de l'application d'une façon dynamique.
- **L'utilisateur final de l'application Android** : son rôle est de naviguer sur l'application en consultant les différents services.

II. Capture des besoins fonctionnels

L'étape de capture des besoins fonctionnels, est la première phase de la conception. Elle permet de produire un modèle des besoins en se basant sur la nature de l'acteur. Elle évite le risque de produire un système inadapté aux utilisateurs. Le système doit répondre aux fonctionnalités suivantes :

- ✓ Le système doit offrir une interface optimale et simple à utiliser pour l'utilisateur et l'administrateur.
- ✓ Le système doit permettre à l'utilisateur de l'application android de consulter ses cours et son emploi du temps, ainsi que l'actualité de l'université, et de recevoir les notifications envoyées par l'administrateur.
- ✓ Le système offrira un espace d'échange entre les étudiants via internet.

- ✓ Le système doit permettre à l'administrateur de gérer les étudiants, les cours, les emplois du temps, et les actualités, via son application web qui les met à jours simultanément.
- ✓ Le système garantira un minimum de sécurité avec l'authentification.
- ✓ L'utilisateur pourra déclencher la fonction qui met le téléphone en mode vibreur pendant les heures des cours.

III. Les cas d'utilisation

Un cas d'utilisation est une fonctionnalité du système qui produit un résultat observable pour un utilisateur potentiel du système. Le cas d'utilisation regroupe une famille de scénarios ou chaque scénario est un traitement particulier du système. [11]

III.1 Identification des cas d'utilisation

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 de cas d'utilisation.

Numéro	Acteur	Cas d'utilisation	
01	Administrateur	Authentification	Connexion
			Déconnexion
		Gestion des cours	Ajout
			Modification
			Suppression
		Gestion des Emploie du temps	Ajout
			Modification
			Suppression
		Gestion des Actualités	Ajout
			Modification
			Suppression
		Gestion des étudiants	Ajout
			Modification
			Suppression

Tableau 4.1 : Les cas d'utilisation de l'administrateur pour l'application android

Numéro	Acteur	Cas d'utilisation	
02	Utilisateur	Authentification	Connexion
			Déconnexion
		Consulter les cours	
		Consulter les emplois du temps	
		Consulter les actualités	
		Echanger des messages	Envoyer des messages
			Recevoir des messages

Tableau 4.2 : Les cas d'utilisation de l'utilisateur pour l'application android

III.2 Description textuelle des cas d'utilisation

La description des cas d'utilisation est libre. Cependant, cette description prend souvent une forme rédigée qui convient mieux à la communication avec les utilisateurs. Des règles de structuration doivent être appliquées pour en faciliter l'expression, la compréhension et la cohérence.[12]

Pour exprimer les cas d'utilisations de notre système, nous avons choisi le formalisme suivant :

Description sommaire	
Titre	Authentification
But	Ce cas d'utilisation permet d'accéder à l'application.
Acteur	L'utilisateur/administrateur
Description de l'enchaînement	
Pré-condition	L'utilisateur saisit son login et son mot de passe
Enchaînement	<ol style="list-style-type: none"> 1. Une page d'authentification se manifeste 2. Saisie du login et du mot de passe 3. Le système vérifie l'authenticité des informations 4. Accès à l'application
Post-condition	
Alternative	Dans le cas où le login et/ou le mot de passe sont erronés, le système affiche un message d'erreur.
Besoin d'IHM	L'interface d'authentification de l'application.

Tableau 4.3 : Description textuelle du cas d'utilisation : cas authentification.

Description sommaire	
Titre	Consultation
But	Ce cas d'utilisation permet à l'utilisateur de consulter les cours, l'emploi du temps ou l'actualité
Acteur	L'utilisateur
Description de l'enchaînement	
Pré-condition	L'Authentification
Enchaînement	<ol style="list-style-type: none"> 1. Accès au cours/emploi du temps/Actualité. 2. Choix de l'objet de consultation.
Post-condition	
Alternative	
Besoin d'IHM	L'interface principale de l'application.

Tableau 4.4 : Description textuelle du cas d'utilisation : Consultation

Description sommaire	
Titre	Gestion de l'application
But	Ce cas d'utilisation permet à l'utilisateur de gérer l'application sur son Smartphone
Acteur	L'utilisateur
Description de l'enchaînement	
Pré-condition	L'Authentification, Connexion Internet
Enchaînement	<ol style="list-style-type: none"> 1. L'utilisateur entre dans le menu option 1. Il active l'option qui met le téléphone en mode vibreur pendant les heures des cours ou recevoir des notifications sur la scolarité.
Post-condition	
Alternative	
Besoin d'IHM	L'interface principale de l'application. Menu de l'application.

Tableau 4.5 : Description textuelle du cas d'utilisation Exploitation de l'application

III.3 Relations entre cas d'utilisation

Afin d'optimiser la formalisation des besoins en ayant recours, notamment à la réutilisation de cas d'utilisation, trois relations peuvent être utilisées entre cas d'utilisation : [13]

- **Relation d'inclusion « include »** : une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B. Elle est représentée graphiquement comme suit :

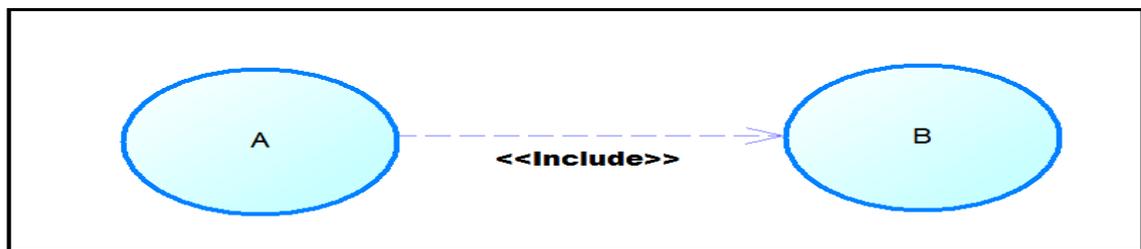


Figure 4.1 : Représentation graphique de la relation d'inclusion.

- **Relation d'extension « extend »** : une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B. Elle est représentée graphiquement comme suit :

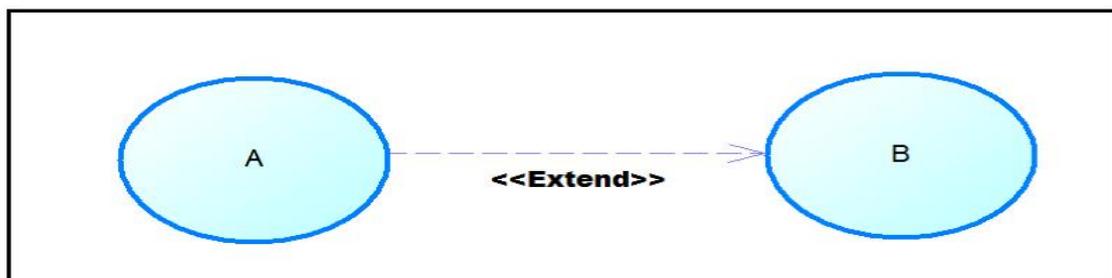


Figure 4.2 : Représentation graphique de la relation d'extension.

- **Relation de généralisation** : une relation de généralisation de cas d'utilisation peut être définie conformément au principe de la spécialisation-généralisation pour les classes. Elle permet de décrire les différentes variantes d'un cas d'utilisation.

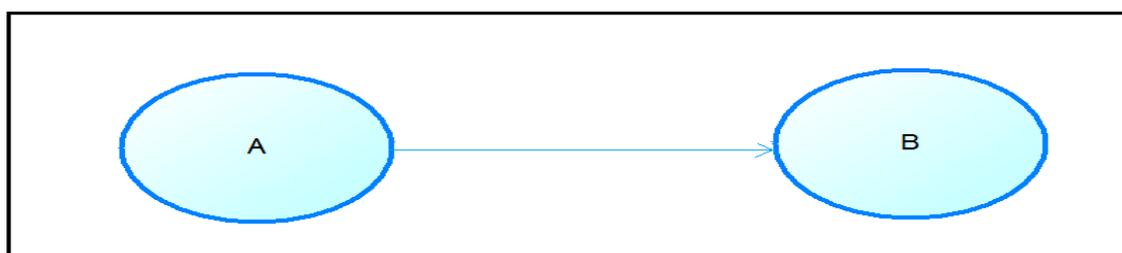


Figure 4.3 : Représentation graphique de la généralisation.

III.4 Représentation des diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les utilisations requises d'un système. Les principaux concepts de ces diagrammes sont les acteurs, cas d'utilisation et les relations entre eux. C'est donc un graphe d'acteurs et d'associations de communication entre ces derniers qui sert à représenter les besoins des utilisateurs par rapport au système. [14]

III.4.1 Diagramme de cas d'utilisation de l'utilisateur :

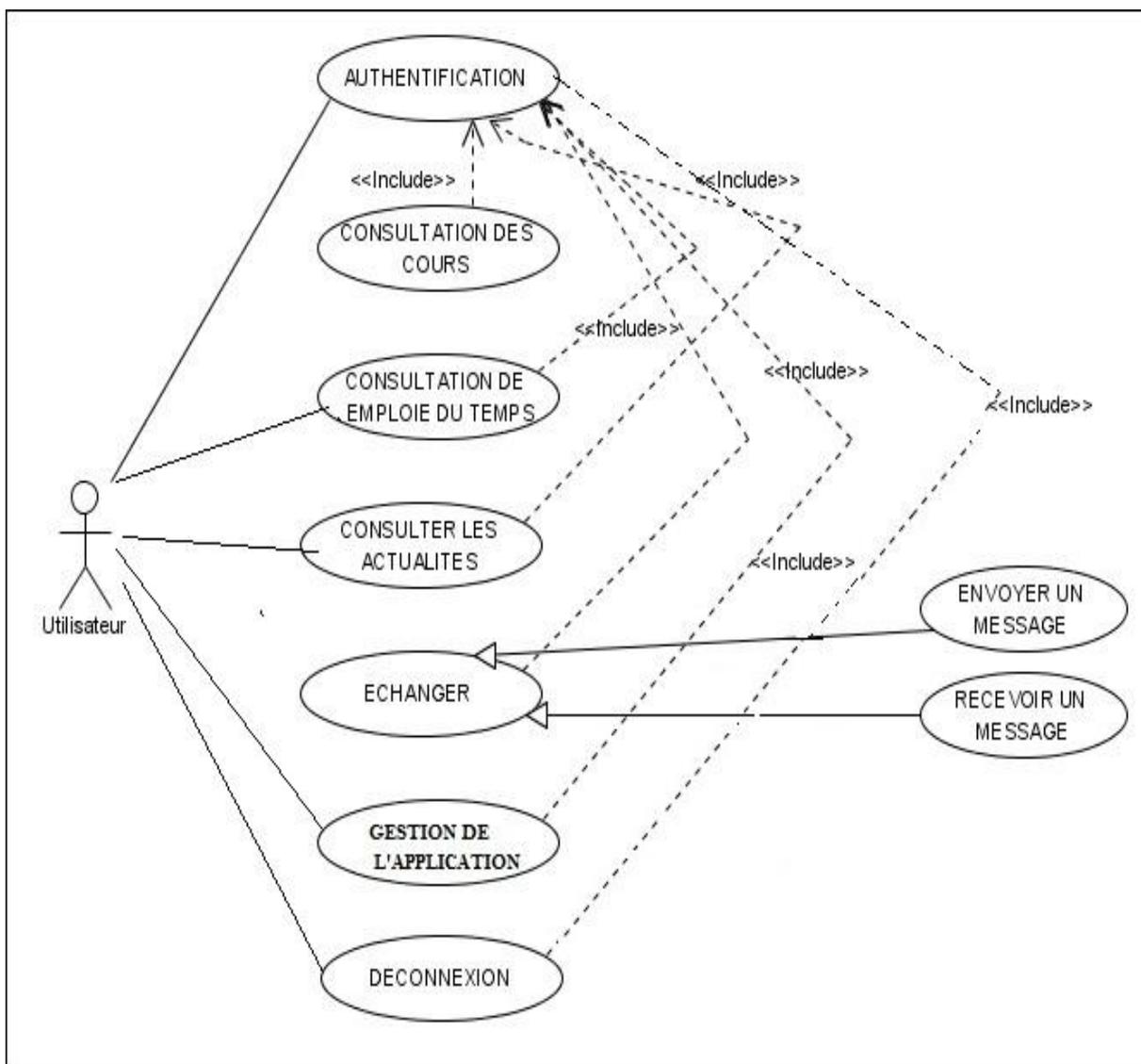


Figure 4.4: Diagramme de cas d'utilisation de l'utilisateur.

III.4.2 Diagramme de cas d'utilisation de l'administrateur

En plus du diagramme de cas d'utilisation de l'utilisateur l'administrateur a des privilèges représenté dans la figure qui suit :

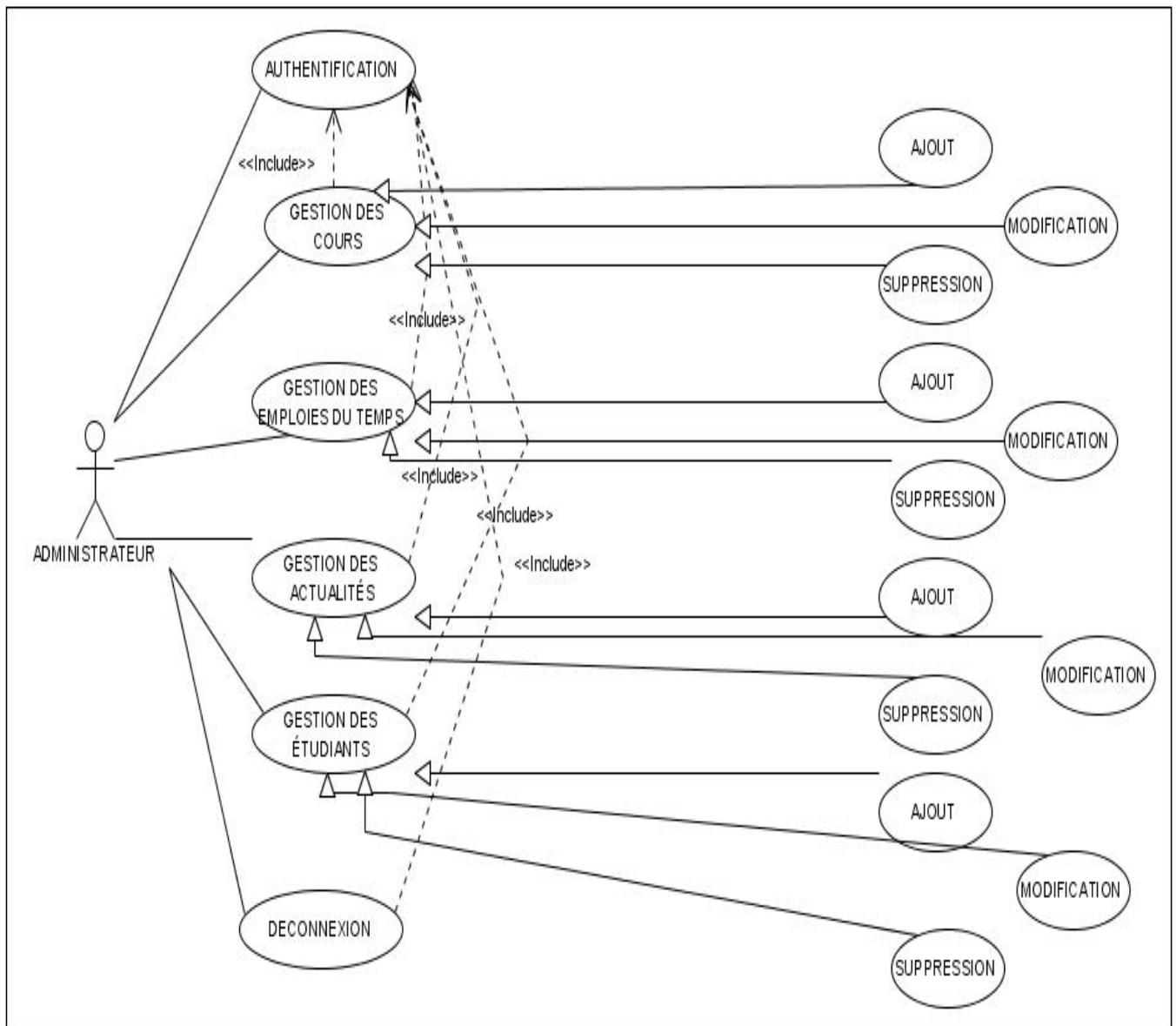


Figure 4.5: Diagramme de cas d'utilisation de l'administrateur.

III.4.3 Diagramme Global de cas d'utilisation du système

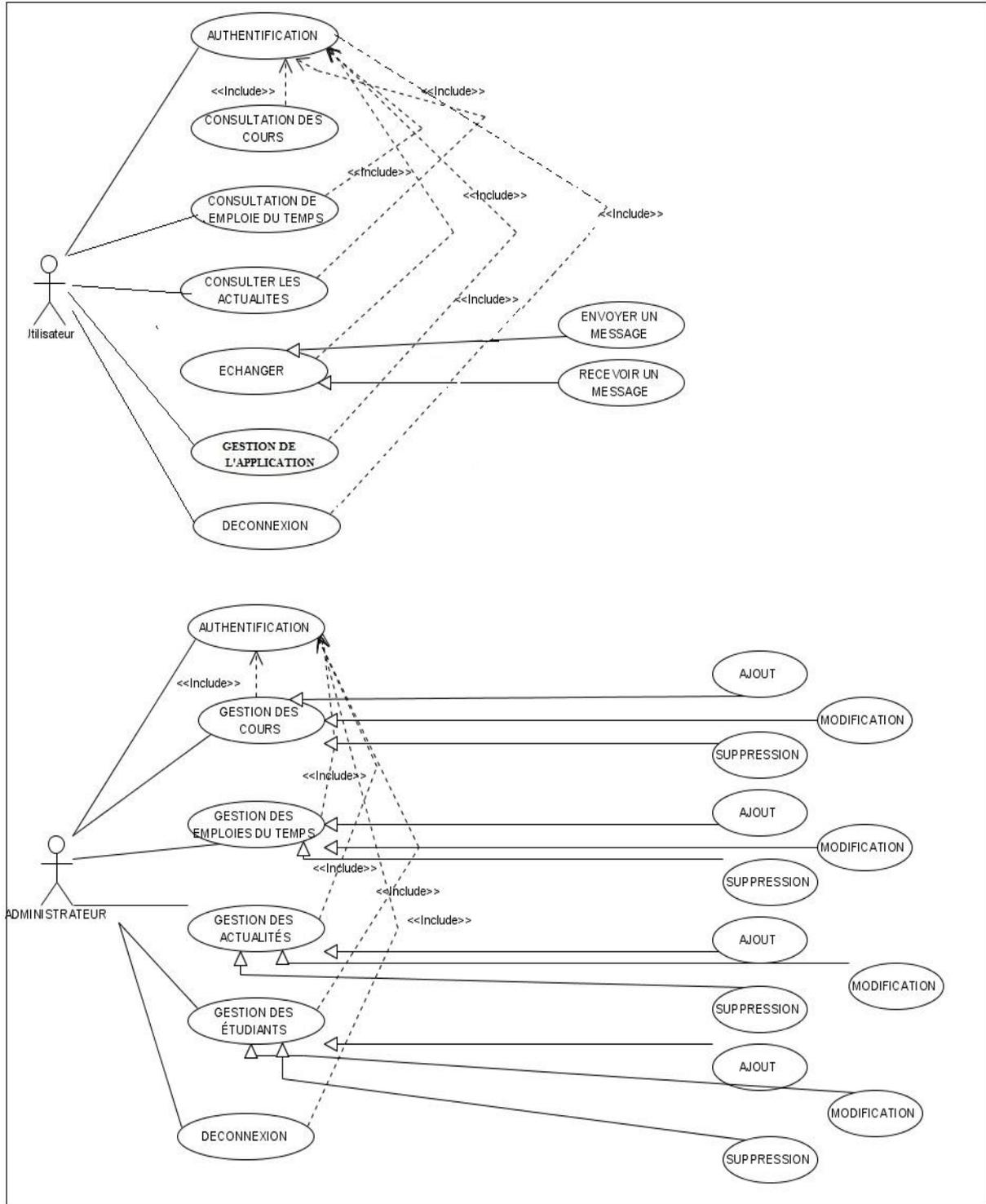


Figure 4.6: Diagramme Global de cas d'utilisation du système

IV. Diagramme de séquence

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

Dans un diagramme de séquence, deux types de messages peuvent être échangés:[15]

- **Les messages synchrones :** Dans ce cas, l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions, c'est une communication interactive. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin de l'exécution de l'opération destinataire du message.
- **Les messages asynchrones :** Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

Un fragment d'interactions est une partie du diagramme de séquence (délimitée par un rectangle) associée à une étiquette (dans le coin supérieur gauche). L'étiquette contient un opérateur d'interaction qui permet de décrire des modalités d'exécution des messages à l'intérieur du cadre. Les opérandes d'un opérateur d'interaction sont séparés par une ligne pointillée. Les conditions de choix des opérandes (éventuels) sont données par des expressions booléennes entre crochets ([]).

Les principales modalités sont les boucles, les branchements conditionnels, les alternatives, les envois simultanés, etc.

Il existe 12 opérateurs définis dans la notation UML 2.0, en ce qui concerne notre projet nous utiliserons uniquement les opérateurs suivants :

- **Opérateur alt :** L'opérateur "alt" désigne un choix, une alternative. Il représente deux comportements possibles : c'est en quelque sorte l'équivalent du SI...ALORS...SINON : donc, une seule des deux branches sera réalisée dans un scénario donné. La condition d'exécution d'une des deux branches (l'équivalent du SI) peut être explicite ou implicite. L'utilisation de l'opérateur 'else' permet d'indiquer que la branche est exécutée si la condition du alt est fausse.

- **Opérateur OPT :** L'opérateur "opt" désigne un fragment combiné optionnel comme son nom l'indique : c'est à dire qu'il représente un comportement qui peut se produire ou pas. Un fragment optionnel est équivalent à un fragment "alt" qui ne posséderait pas d'opérande else (qui n'aurait qu'une seule branche). Un fragment optionnel est donc une sorte de SI...ALORS.
- **Opérateur loop :** L'opérateur "loop" est utilisé pour décrire un ensemble d'interactions qui s'exécutent en boucle. En général, une contrainte appelée garde indique le nombre de répétitions (minimum et maximum) ou bien une condition booléenne à respecter.
- **Opérateur par :** L'opérateur "par" est utilisé pour représenter des interactions ayant lieu en parallèle. Les interactions des différents opérandes (les deux branches de notre opérateur ci-dessous) peuvent donc se mélanger, s'intercaler, dans la mesure où l'ordre imposé dans chaque opérande est respecté.

IV.2 Elaboration des diagrammes de séquence des cas d'utilisation

En se basant sur les éléments et concepts défini précédemment , nous allons présenter les diagrammes de séquence des cas d'utilisations les plus importants.

IV.2.1 Les diagrammes de séquence des cas d'utilisation de l'utilisateur

a. L'authentification: pour se connecter sur l'application, l'utilisateur doit s'authentifier.

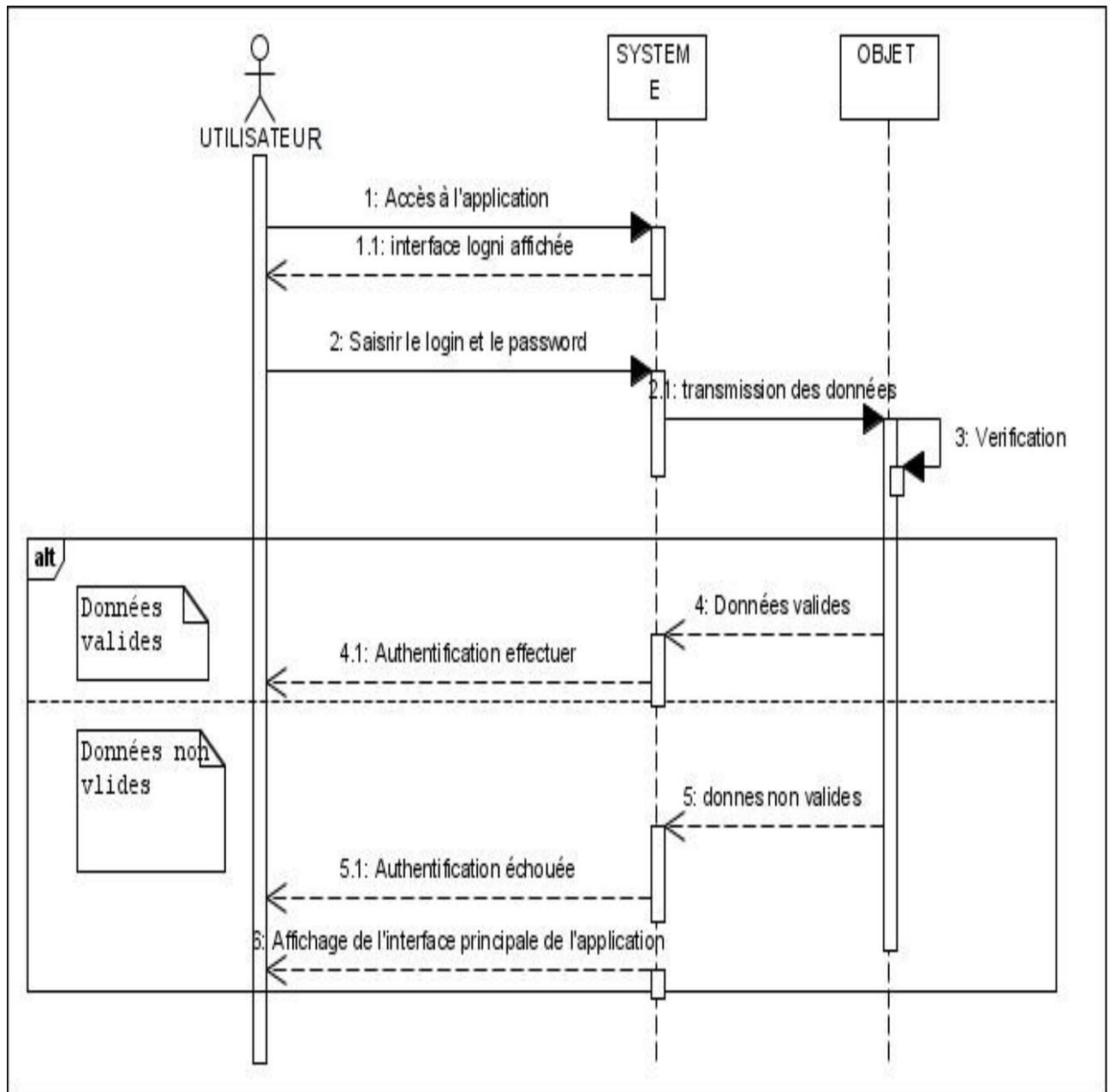


Figure 4.7: Diagramme de séquence du cas « Authentification » de l'utilisateur.

b. Consultation des cours : une fois authentifié, l'utilisateur peut accéder au menu général de l'application pour consulter les cours.

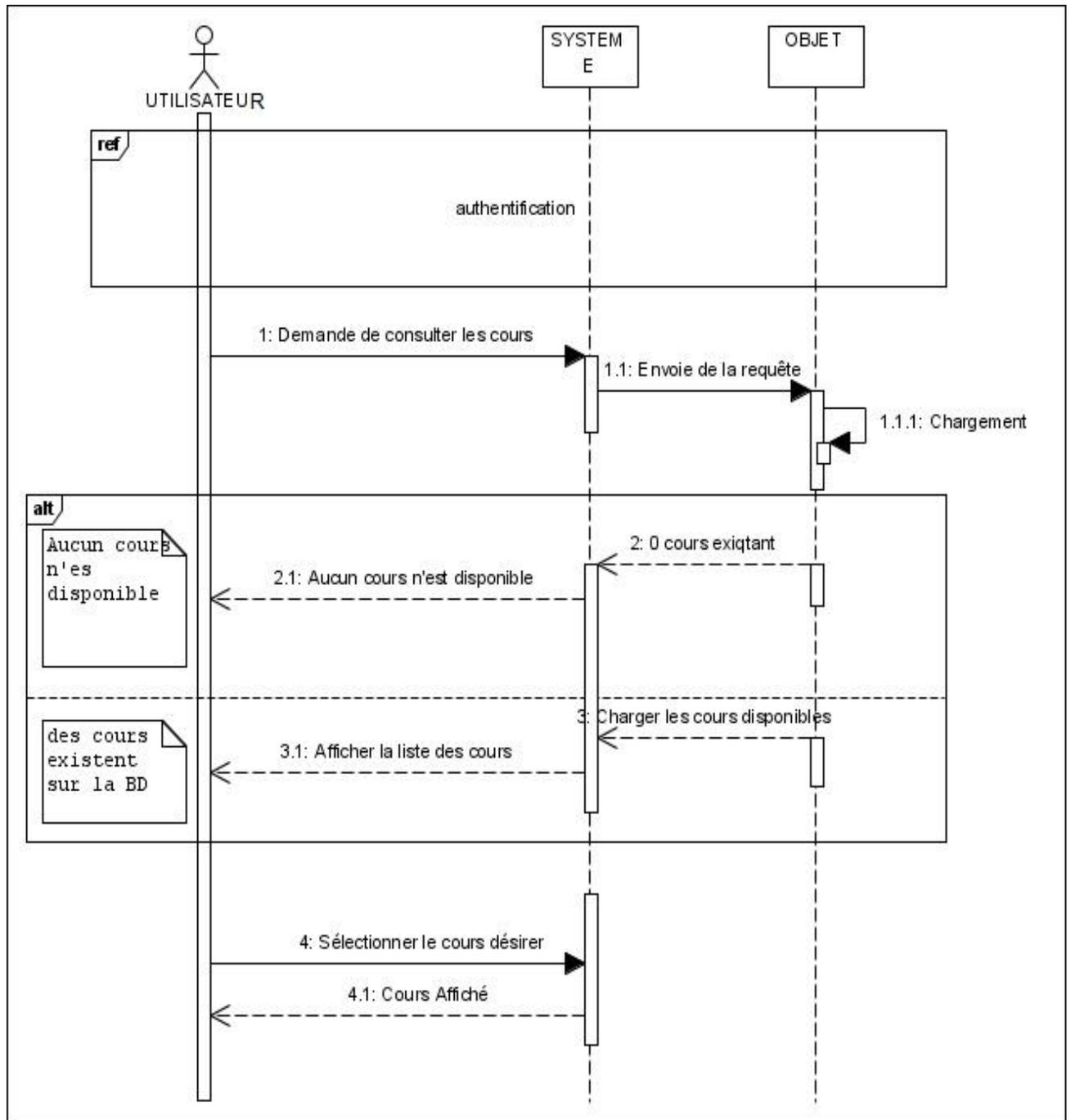


Figure 4.8: Diagramme de séquence du cas « Consultation des cours ».

c. Consultation de l'emploi du temps : de la même façon que les cours, l'utilisateur pourra vérifier son emploi du temps.

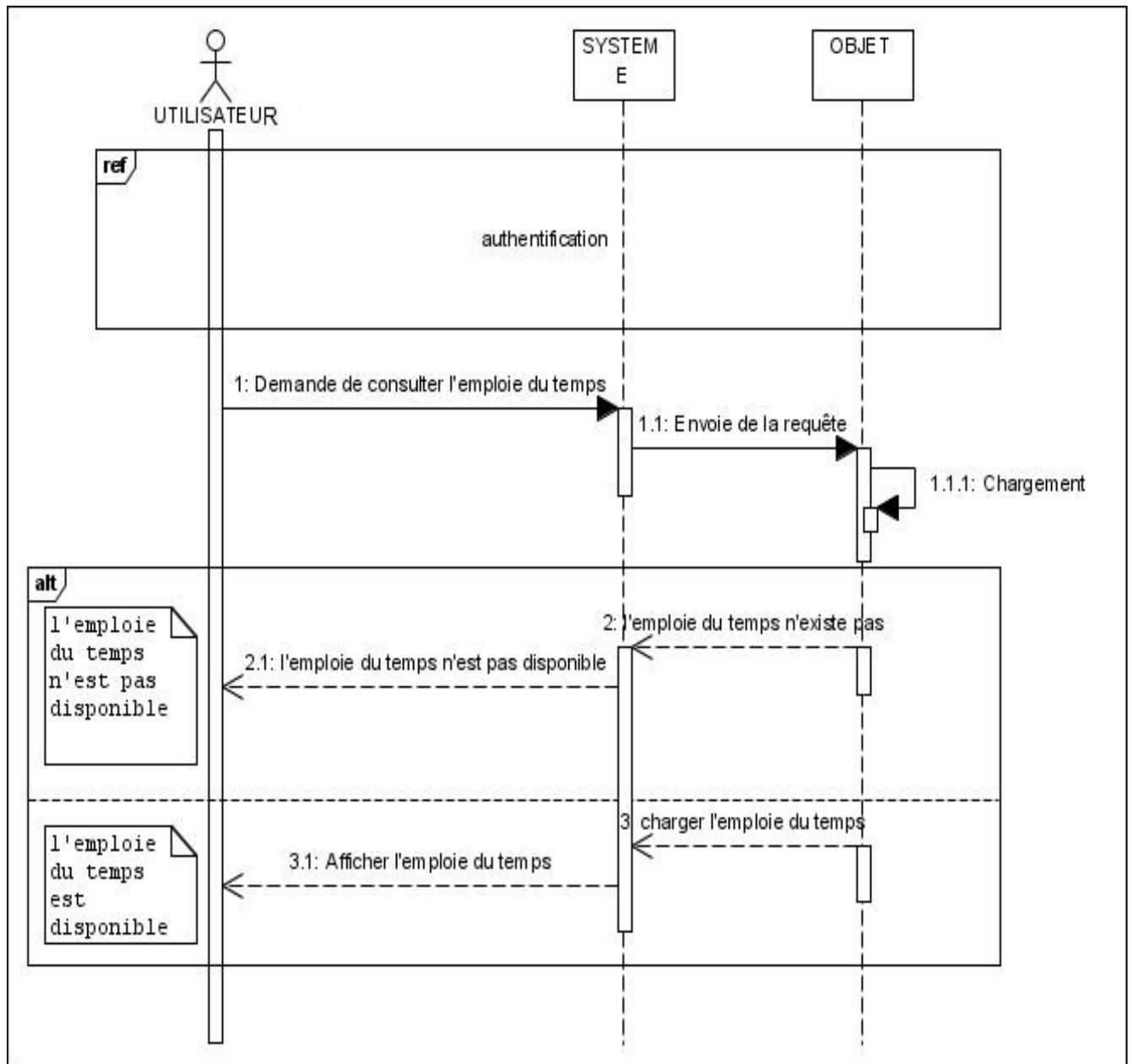


Figure 4.9: Diagramme de séquence du cas « Consultation de l'emploi du temps ».

d. Consultation de l'actualité : de la même manière, un utilisateur à l'accès à une page d'actualités via le navigateur de son Smartphone.

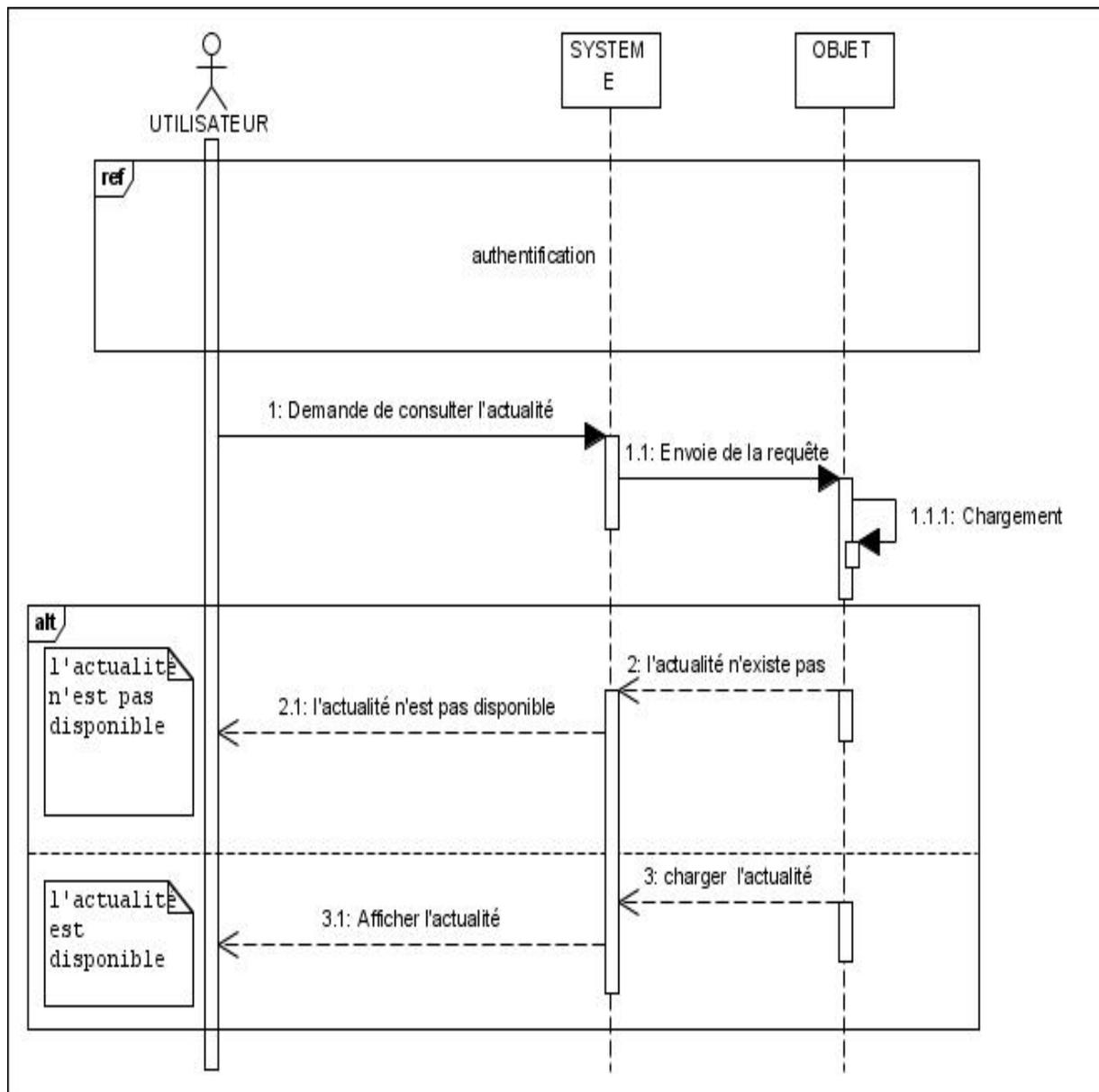


Figure 4.10: Diagramme de séquence du cas « Consultation des actualités».

e. Echanger : l'échange entre les utilisateurs connectés est une option de notre application, et cela nécessite une connexion au réseau internet.

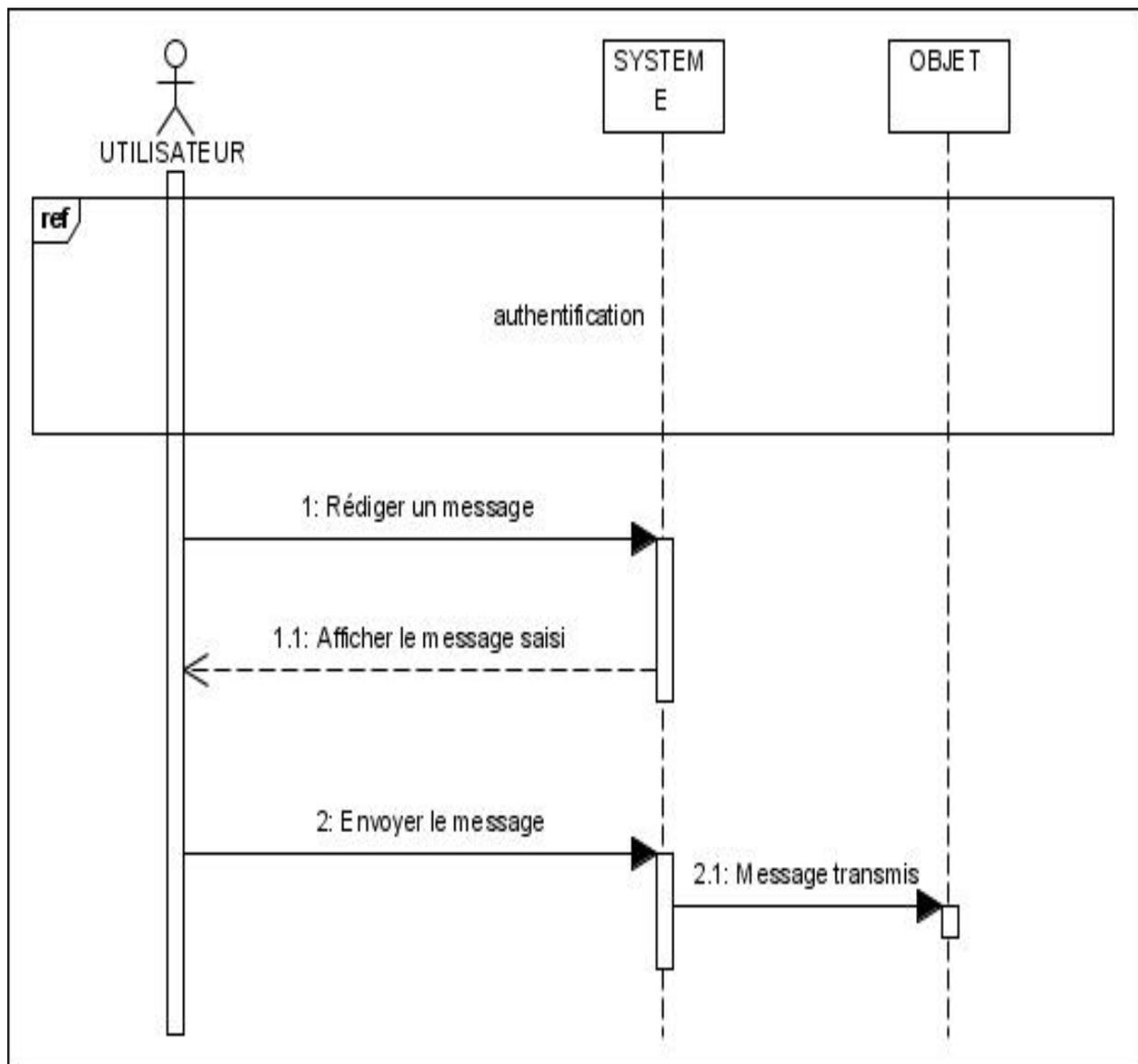


Figure 4.11: Diagramme de séquence du cas « Echange ».

IV.2.2 Les diagrammes de séquence des cas d'utilisation de l'administrateur du système

a. L'authentification: l'administrateur doit s'authentifier en insérant son nom d'utilisateur et son mot de passe pour se connecter à l'application, ce qui va lui permettre la gestion des fonctionnalités de l'application.

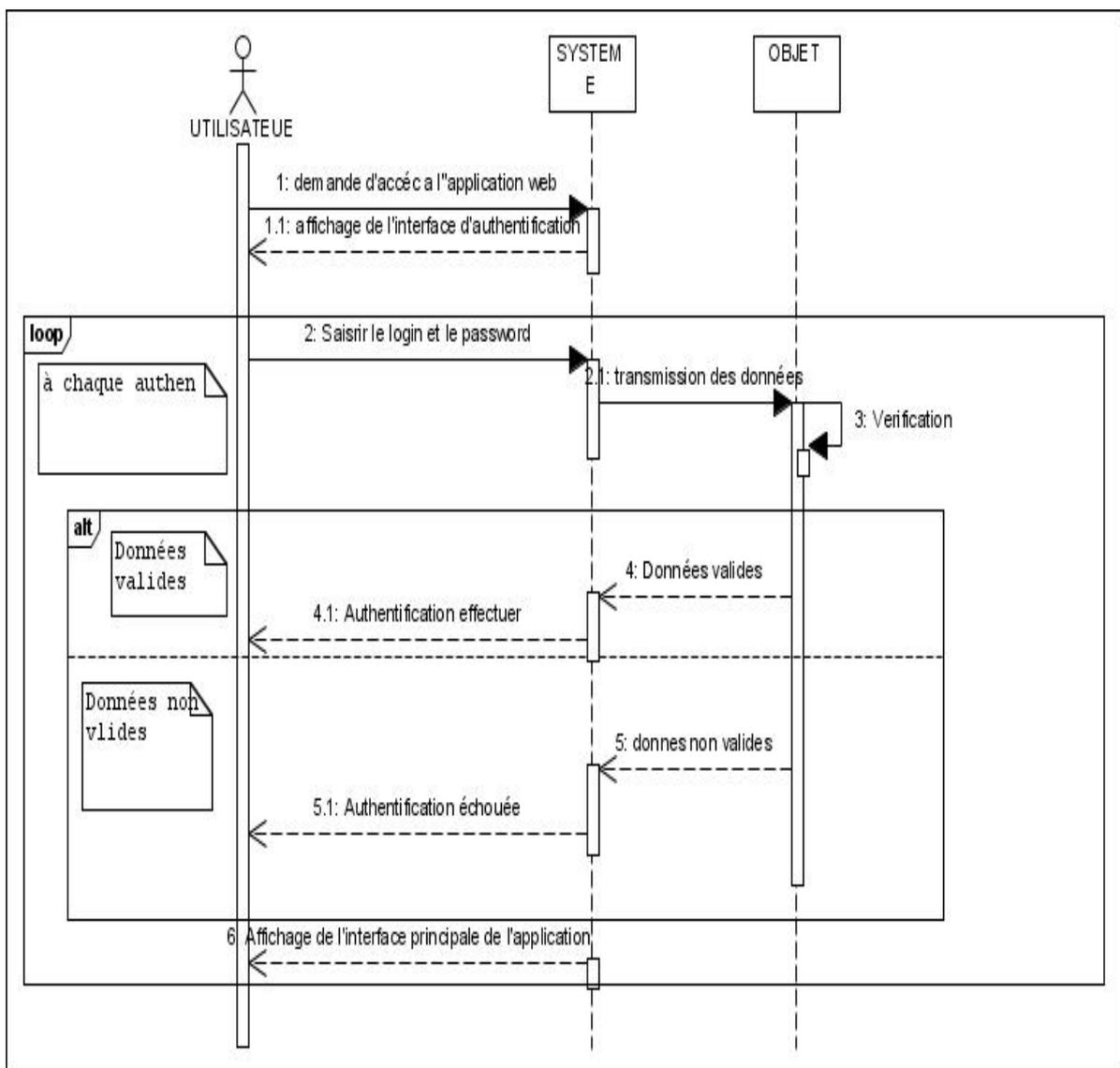


Figure 4.12: « Diagramme de séquence du cas « Authentification » de l'administrateur.

V. Le Diagramme de classe

Les diagrammes de classe identifient la structure des classes dans un système, y compris les propriétés et les méthodes de chacune d'elles. Ainsi que les relations entre classes. [16]

Les éléments qui les composent sont bien expliqués dans le chapitre précédent.

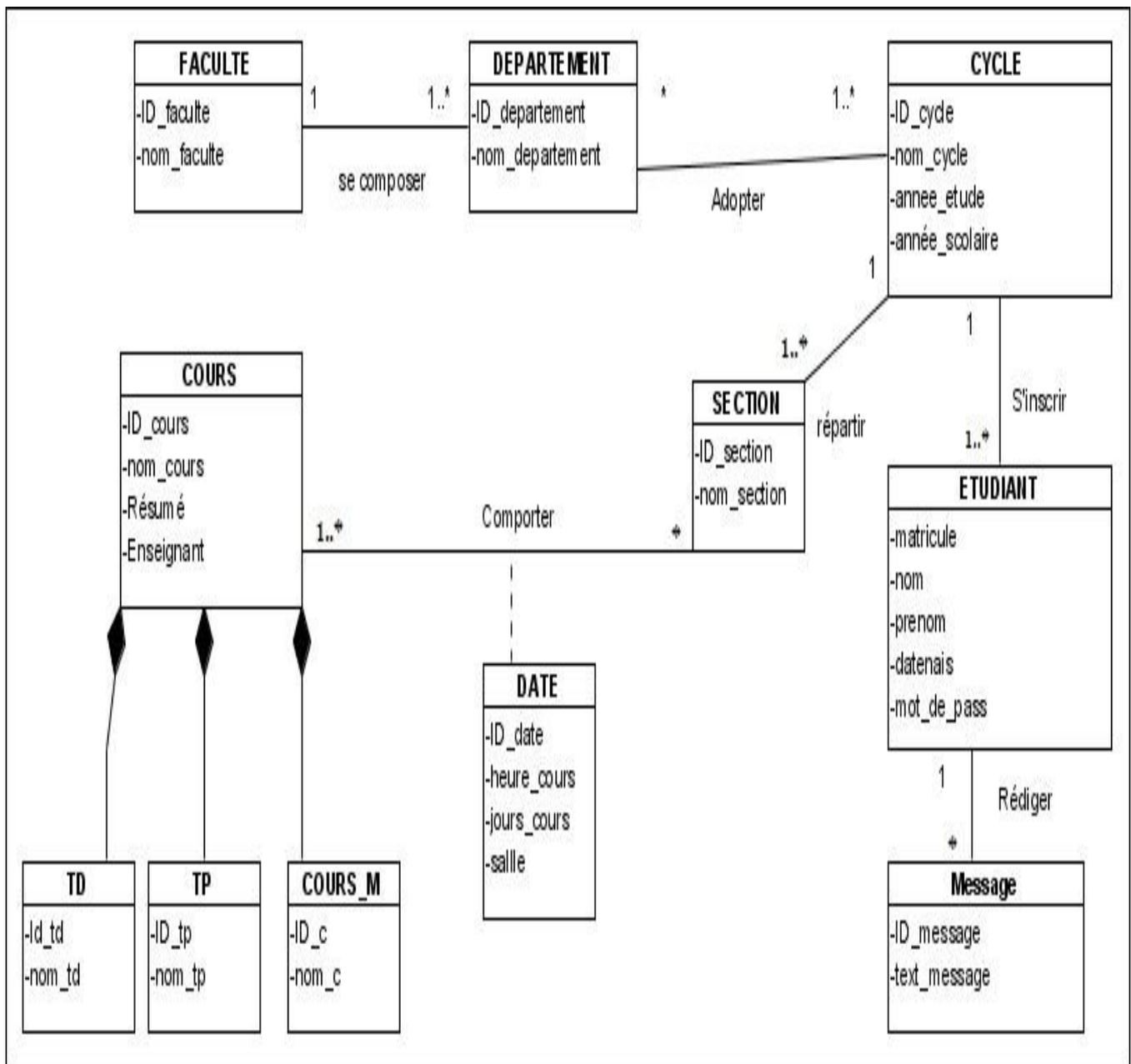


Figure 4.13: « Diagramme de classe du système.

VI. Le modèle relationnel

L'utilisation d'un SGBD (Système de Gestion de Bases de Données) impose un changement de représentation entre la structure des classes et la structure des données relationnelles :

Cours (**id_cours**, nom_cours, resume, enseignant) ;

Cours_m (**#id_cours**, **id_c**, nom_c) ;

Cycle (**id_cycle**, nom_cycle, annee_eude, annee_scolzire) ;

Date_Cours (**#id_date**, heure_cours, jours_cours, salle, **#id_section**, **#id_cours**) ;

Département (**id departement** _, nom_departement, **#id_faculté**) ;

Département_cycle (**#id_département**, **#id_cycle**) ;

Etudiant (**matricule**, nom, prenom, datenais, mot_de_pass) ;

Faculté (**id_faculté**, nom_faculté) ;

Message (id_message, text_message, **#matricule**) ;

Section (id_section, nom_section, **#id_cycle**) ;

Td (**#id_cours**, **id_td**, nom_td) ;

TP (**#id_cours**, **id_tp**, nom_tp) ;

Conclusion

Dans ce chapitre, nous avons fait une relation entre les besoins de notre application et les outils que propose la méthode OOD de Grady BOOCH à travers différents diagrammes, afin de donnée la meilleure conception possible pour notre application.



Chapitre V

Réalisation

Introduction

Après avoir présenté les fonctionnalités et les objectifs de notre application, nous allons aborder dans ce qui suit la phase de la réalisation de notre projet, celle-ci sera divisé en deux grandes parties, dans la première nous allons présenter les principaux outils et concepts que nous avons utilisés dans le développement, en ce qui concerne la deuxième partie nous présenterons des captures d'écran représentatifs de notre application avec des explications sur l'utilisation et les fonctionnalités pour chaque capture.

I. Outils et environnement de développement

I.1 IDE Eclipse [17]

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

I.2 Le Plugin ADT (Android Développement Tools) [18]

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

I.3 Software Development Kit (SDK) [19]

C'est un kit de développement basé sur le langage Java. Il s'agit des outils que Google a fourni pour interagir avec Android pour la réalisation des applications. Le SDK propose, en plus, un émulateur Android. Ce dernier permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version

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

I.4 MySQL [20]

MySQL est un système de gestion de base de données relationnel, un langage de requêtes vers les bases de données exploitant le modèle relationnel et utilise le langage SQL comme langage de requête. SQL est un langage de manipulation de bases de données mis au point dans la années 70 par IBM, il permet d'effectuer trois types de manipulations :

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

II. Les langages de programmation

II.1 Le langage JAVA

Java est un langage de programmation très utilisé, notamment par un grand nombre de développeurs professionnels, ce qui en fait un langage incontournable actuellement.

II.2 PHP [21]

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

II.3 JSON (JavaScript Object Notation)

Est un format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter de l'information structurée. Créé par Douglas Crockford, il est décrit par la RFC 4627 de l'IETF.



Figure 5.1 : Principaux outils de communication.

II.4 SQLite [22]

SQLite est un système de gestion de bases de données intégrée dans chaque appareil Android. L'utilisation d'une base de données SQLite sous Android ne nécessite pas de configuration ou d'administration de la base de données. On doit uniquement définir les instructions SQL pour créer et mettre à jour la base de données. Ensuite, celle-ci est gérée automatiquement par la plate-forme Android.

II.4.1 Avantages de SQLite

- Il ne nécessite pas de base MySQL sur le serveur, vous utilisez votre propre stockage, un fichier SQLite.
- On peut sauvegarder la base simplement en téléchargeant le fichier.
- On peut l'utiliser localement avec des programmes classiques écrit en C ou PHP.

II.4.2 Inconvénients de SQLite

- L'accès aux bases de données SQLite peut ralentir si la taille devient importante.
- Plusieurs utilisateurs ne peuvent modifier simultanément une base.
- L'extension n'est pas activée par tous les hébergements mutualisés.

II.5 XML

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

II.6 Les Bases de données web

Se sont des bases de données qui se trouvent sur des serveurs distants par rapport a notre application, ces bases de données ont l'avantage de permettre la sécurité et la gestion de nos données ainsi que le partage des données grâce a l'internet.

Une particularité importante des bases de données web est qu'un compte utilisateur est requis pour utiliser ces bases de données, autrement dit l'accès anonyme n'est pas pris en charge.

III. Les étapes de développement

Dans cette partie nous allons décrire les étapes de création de notre projet sur la plateforme android, et nous expliquerons la manière de le tester sur un émulateur android.

III.1 Création d'un AVD

Afin de tester notre application, nous allons utiliser l'émulateur Android. Il faudra donc créer un Android Virtual Device (AVD). Un AVD décrit les paramètres systèmes et les composants de notre émulateur.

Pour créer un AVD:

1. Nous lançons Eclipse
2. Nous allons sous « Window > Android SDK and AVD Manager »
3. Nous sélectionnons « Virtual Device » dans le panneau à gauche
4. Nous cliquons sur « New ». La boîte de dialogue « Create New AVD » apparaîtra.
5. Nous tapons le nom de notre AVD, « hello_avd » par exemple
6. Nous choisissons la cible (the target). La cible est la version de la plateforme Android SDK que nous avons téléchargé.

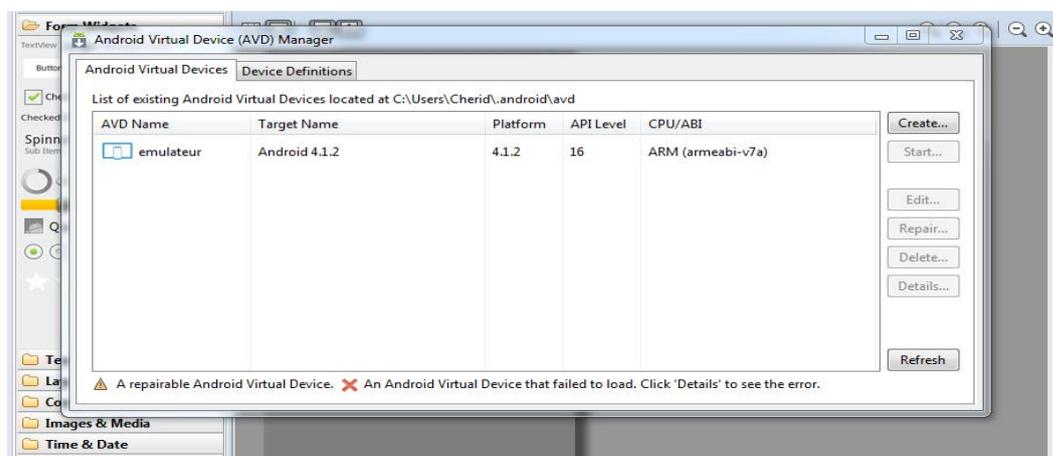


Figure 5.2 : Création d'un AVD android sous windows

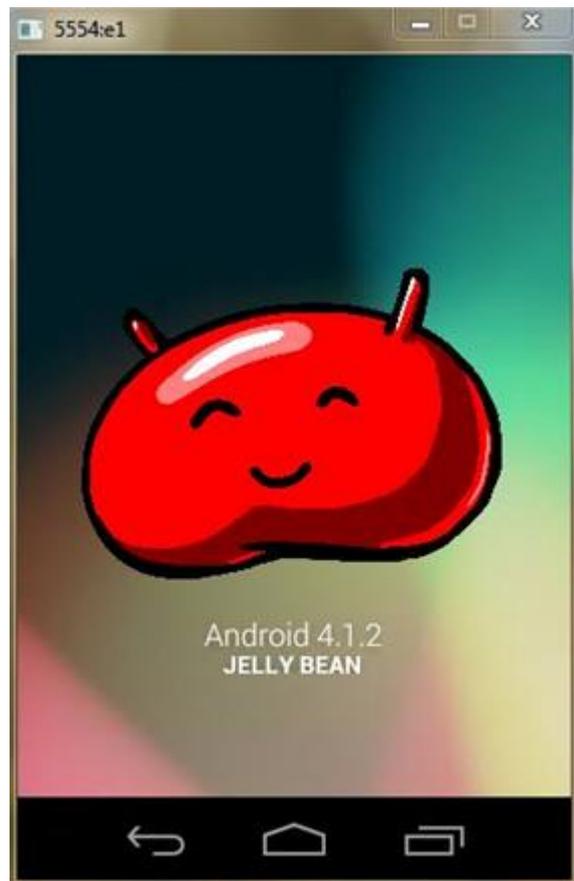


Figure 5.3 : Emulateur android

III.2 Création d'un projet Android

Après avoir créé un émulateur android, nous passons à la création du projet sous Eclipse. Nous lançons Eclipse et nous allons sous **File** -> **New**->**Project** et nous sélectionnons « **Android Project** ».

- Nous renseignons les détails à propos de notre projet comme suit :
 - **Project Name** : E-UBejaia
 - **Build Target** : android 4.2
 - **Application Name** : E-UBejaia
 - **Package Name** : com.masterbejaia.bobhalim
 - **Create Activity** : UBejaia

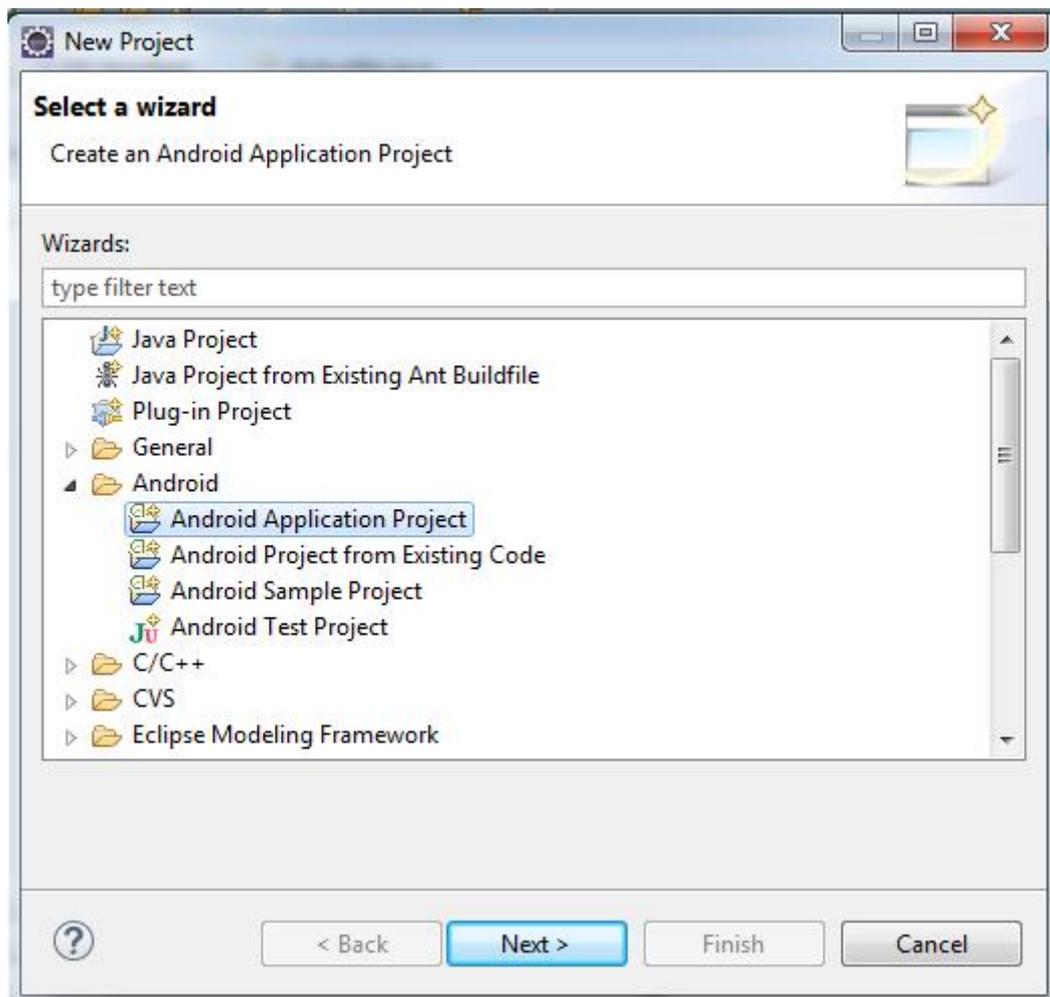


Figure 5.4 : Création d'un projet Android sous eclipse

III.3 Etapes de la mise en marche de notre application

- Téléchargement et installation de WampServer.
- Copier tous les fichiers existant sous le dossier Server vers un dossier dans le répertoire du serveur web (C:\wamp\www), ce dossier sera nommé par exemple android_im. Nous pouvons accéder à ce dossier par http://192.168.7.5/android_im/ (192.168.7.5 est l'adresse IP de l'ordinateur qui exécute Wampserver et Mysql).
- Créer la base de données E-UBejaia

Nous créons notre base de données 'e-ubejaia' dans laquelle nous travaillerons dans toute la suite en utilisant le formulaire dans la page <http://localhost/phpmuadmin>

Table	Action	Lignes	Type	Interclassement	Taille	Perte
admin	Afficher Structure Rechercher Insérer Vider Supprimer	-2	InnoDB	latin1_swedish_ci	16 KiB	-
cours	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	16 KiB	-
cours_ma	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	16 KiB	-
cycle	Afficher Structure Rechercher Insérer Vider Supprimer	-2	InnoDB	latin1_swedish_ci	16 KiB	-
date_cours	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	32 KiB	-
departement	Afficher Structure Rechercher Insérer Vider Supprimer	-2	InnoDB	latin1_swedish_ci	32 KiB	-
departement_cycle	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	32 KiB	-
etudiant	Afficher Structure Rechercher Insérer Vider Supprimer	-1	InnoDB	latin1_swedish_ci	48 KiB	-
faculte	Afficher Structure Rechercher Insérer Vider Supprimer	-2	InnoDB	latin1_swedish_ci	16 KiB	-
message	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	32 KiB	-
section	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	32 KiB	-
td	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	16 KiB	-
tp	Afficher Structure Rechercher Insérer Vider Supprimer	-0	InnoDB	latin1_swedish_ci	16 KiB	-
13 tables	Somme	9	InnoDB	latin1_swedish_ci	320 KiB	0

Figure 5.5 : La base de données E-UBejaia sous MySQL

IV Fonctionnement détaillé de l'application

Nous présenterons dans ce qui suit les différentes fonctionnalités de notre application, ainsi que des explications sur le rôle de chacune.

IV.1 L'authentification

La première étape consiste à s'authentifier. A travers une première vue de l'application, l'utilisateur est amené à fournir un nom d'utilisateur (Login) et un mot de passe.

Par exemple dans notre cas, le Login sera '11MI404' et le mot de passe '08/06/1992'

Une fois appuyé sur le bouton 'connexion', une requête est envoyée au serveur pour vérifier les données entrées par l'utilisateur, Dans le cas idéal, l'utilisateur est authentifié et une mise à jour du temps d'authentification est faite dans la base de données. Sinon un message d'erreur sera affiché pour demander à l'utilisateur de vérifier son login et mot de passe.

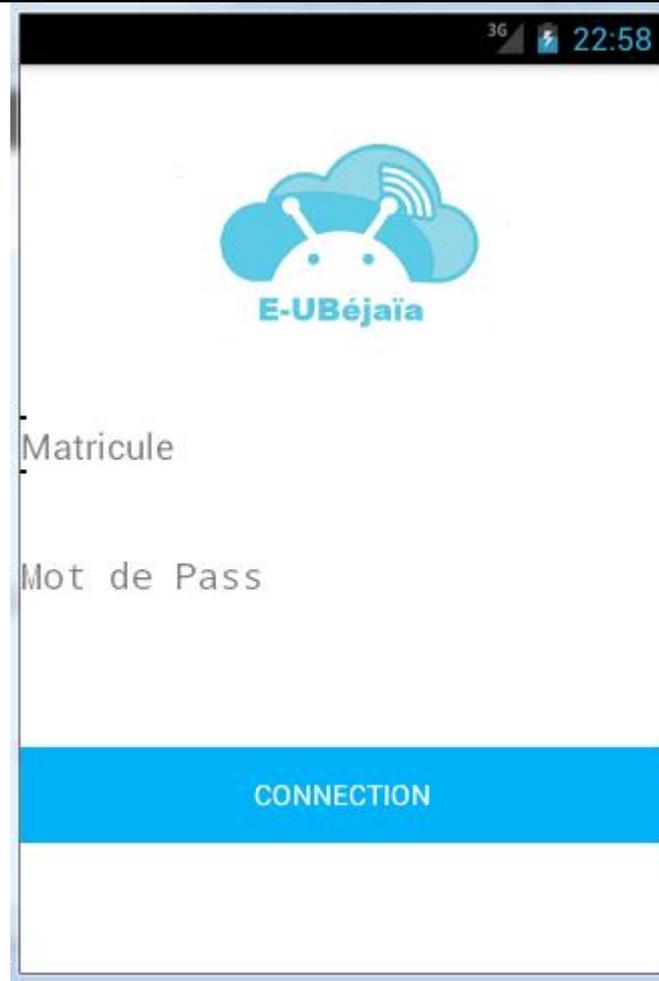


Figure 5.6 : Interface d'authentification de l'application

IV.2 Le menu principal de l'application

Après l'authentification, l'utilisateur aura un accès sur une vue qui contient le menu principal de l'application, ce dernier contient les différentes fonctionnalités que l'application propose (citées dans le chapitre trois).

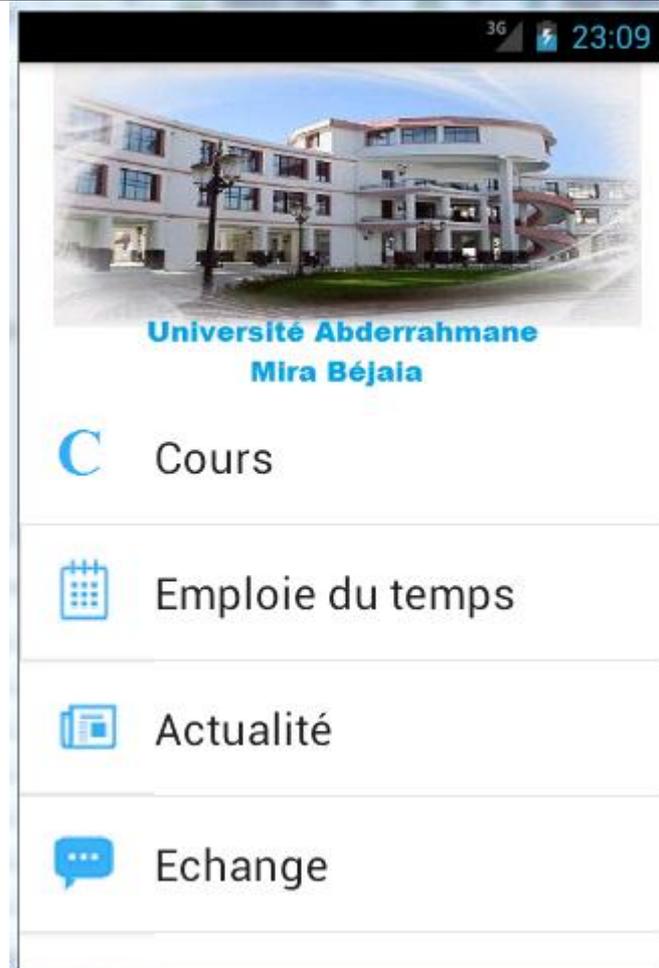


Figure 5.7 : Menu principal de l'application

IV.3 Consultation des cours

L'utilisateur a le privilège de consulter et de télécharger les cours qui le concerne via l'application E-UBéjaia.

IV.4 Consultation des emplois du temps

Cette fonctionnalité permet à l'utilisateur de consulter les différents emplois du temps de tous les cycles et de tous les niveaux.

IV.5 Consultation de l'actualité

L'utilisateur peut suivre l'actualité de l'université directement sur son téléphone.



Figure 5.8 : Page d'actualités

IV.6 Echanger

Cette partie permettra aux étudiants d'échanger des messages concernant leurs activités pédagogiques.

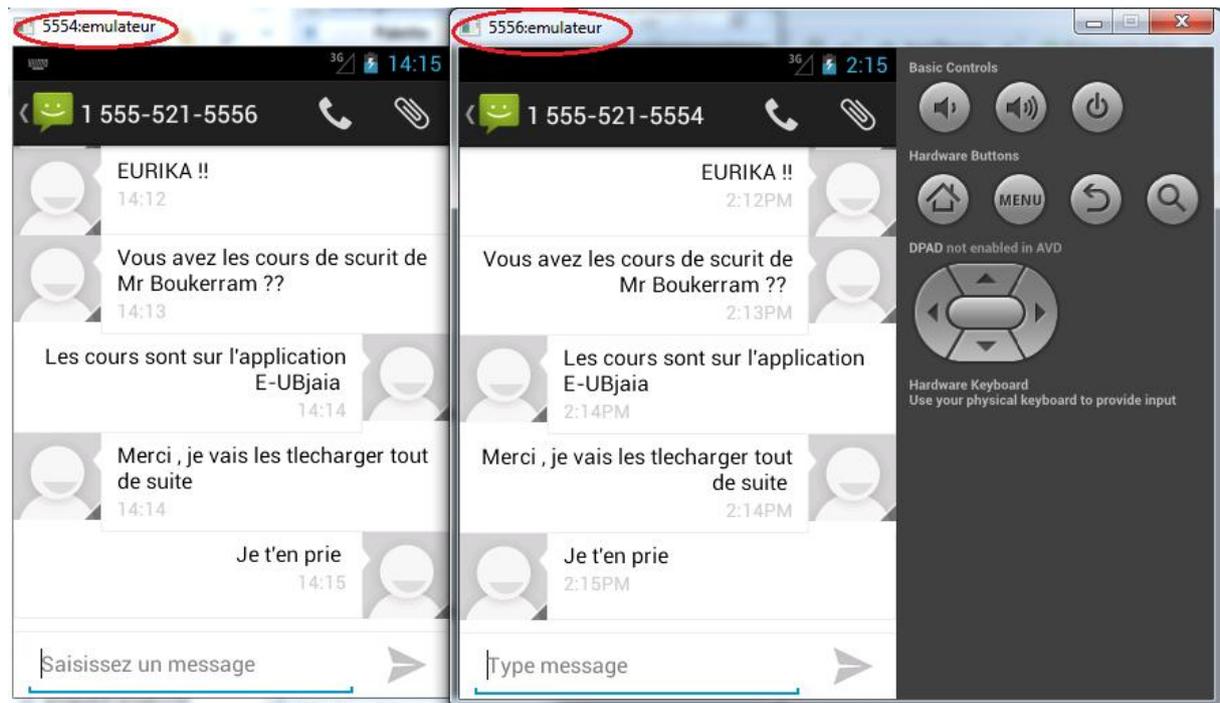


Figure 5.9 : interface d'échange entre les émulateurs

Conclusion

Dans ce dernier chapitre, nous avons enfin présenté la réalisation de notre application. Cette étape est très décisive car les phases ultérieures viennent enfin de porter leurs fruits. Dans un premier lieu nous avons présenté les différents outils que nous avons utilisés dans l'implémentation et la réalisation de notre projet, puis, nous avons décrit le fonctionnement de l'application U-EBejaia, tout en présentant des captures d'écrans pour chaque vue.

Conclusion générale

Nous sommes parvenus, par le biais de ce projet à réaliser une application mobile sous système android aux étudiants de l'université de Béjaïa pour faciliter l'accès à l'information durant leurs scolarité, et de diminuer la charge sur la plateforme d'enseignement en ligne ainsi que sur le personnel de l'université.

Pour cela, nous avons débuté notre projet par des généralités sur le système Android et ses fonctionnalités pour mieux cerner notre travail. L'élaboration d'un cahier de charge est notamment mise en place dans ce champ d'études afin de présenter notre projet d'une façon générale, la méthode de conception adoptée, les différents langages de programmation choisis, ainsi que les besoins fonctionnels et non fonctionnels.

Ce travail nous a permis d'acquérir une expérience personnelle et professionnelle très bénéfique. Ce fut une occasion pour notre groupe de se familiariser avec l'environnement du travail de la vie professionnelle, d'élargir et d'approfondir dans les connaissances et les apprécier aux diverses réalités du terrain.

En perspective, notre application pourrait devenir l'application officielle de l'université de Béjaïa en la généralisant sur toutes les facultés et tous les départements. En ajoutant d'autres fonctionnalités telles que les notifications des enseignants et la géo localisation des différents blocs et salles de l'université.

Liste Bibliographique

- [1] : BENBOURAHLA.N, Android 4 Les fondamentaux du développement, Informatique Technique, 2012 ;
- [2] : <http://www.futura-sciences.com> , Dernier accès avril 2015 ;
- [3] : <http://developer.android.com/guide/components/fundamentals.html> , dernier accès Avril 2015 ;
- [4] : <http://android.developer.com/lollipop>, dernier accès avril 2015 ;
- [5] : <http://www.univ-bejaia.dz/presentation-vrcdpo>, dernier accès Mai 2014 ;
- [6] : AREZKI.N, Centre des Systèmes et Réseaux d'Information et de Communication, de Télé-enseignement Section E – Learning;
- [7] : <http://www.univ-bejaia.dz/centre-des-systemes-et-reseaux-d-information>, dernier accès Mai 2015 ;
- [8] : <http://www.uni-klu.ac.at> , dernier accès Avril 2015 ;
- [9]: BOOCH.G, Object-Oriented Analysis and Design with Applications, 3eme edition, Edition Wesley;
- [11] :http://www.memoireonline.com/11/12/6484/m_Conception-et-realisation-dune-application-de-gestion-des-marches-par-appel-doffres-au-sein8.html, dernier accès Mai 2015
- [12] : J.GABY, UML2 analyse et conception, 2008, DUNOD.
- [13] : J. OTTELE, Analyse et conception d'une application informatique de gestion, LITECG Luxembourg, version9, ENI, .2009 ;
- [14] :http://docwiki.embarcadero.com/RADStudio/XE4/fr/D%C3%A9finition_des_diagrammes_de_cas_d%27utilisation_UML_2.0.
- [15] : D.CHOLLEY, Informatique pour les systèmes d'aide à la décision, LAVOISIER, 1999 ;
- [16] : M.GRIMALDY, modélisation UML diagrams Structurels, Février 2007 ;
- [17] : <http://www.techno-science.net/?onglet=glossaire&definition=517>, dernier accès : Juin 2015 ;
- [18] : <http://developer.android.com/tools/sdk/eclipse-adt.html>, dernier accès : juin 2015 ;
- [19] : <http://techterms.com/definition/sdk>, dernier accès : Juin 2015 ;
- [20] : C. SOUTOU, Apprendre SQL avec MySQL, Eyrolles, 2006 ;
- [21] : www.ed-productions.com/FTP/expose/Php-MySql-ASP.pdf, dernier accès Juin 2015 ;
- [22] : <http://www.finalclap.com/faq/180-sqlite-definition>, dernier accès juin 2015 ;

Résumé

Les étudiants de l'université Abderrahmane Mira de Béjaïa souffrent de la pénurie des moyens d'informations au sein de l'université. Le seul moyen existant reste la plateforme d'enseignement en ligne « E-Learning » qui est souvent inaccessible via internet à cause de problèmes locaux et la surcharge des serveurs de l'université.

C'est pour cela que nous avons pensé à une solution intelligente pour rendre la vie étudiante moins pénible pour les étudiants. Une application mobile sous Android accessible à tout le monde.

Pour la modélisation et la conception nous avons adopté la méthode OOD de Grady Booch et pour la phase de concrétisation nous avons utilisé le langage Java qui est le langage officiel de programmation pour Android. Ainsi que tous les outils que Google nous offre pour développer sur cette plateforme (SDK). Pour les bases de données nous avons choisi de travailler avec le SGBD MySQL.

Mots clés : OOD, Google, Android, SDK, Java, SGBD, MySQL

Abstract

Students from the University of Béjaïa suffer from the shortage of ways of information and communication within the university. The only existing means remains the online learning platform "E-Learning" which is often inaccessible via internet because of local problems and overload of university servers.

This is why we thought of a clever solution to make student life less painful for students. A mobile application on Android accessible for everyone.

For the modeling and design we adopted Grady Booch OOD method and the realization phase we used the Java language which is the official programming language for Android. And all the tools that Google offers us to develop on this platform (SDK). For databases we work with MySQL.

Keywords: OOD, Google, Android, SDK, Java, DBMS, MySQL