

République du Cameroun

Paix - Travail - Patrie

Ministère de l'enseignement Supérieur

Republic of Cameroon

Peace– Work–Fatherland

Ministry of Higher Education



Institut Supérieur des Sciences et de Technologie

PROJET DE FIN DE CYCLE LICENCE

**DEVELOPEMENT D'UN CLIENT ANDROID DE
GESTION ACADEMIQUE BASE SUR LES SERVICES
WEB DANS LE CADRE DU PROJET UDM-ACADEMY**

**Rédigé et soutenu en vue de l'obtention d'une licence de technologie en Informatique
Réseaux et Télécommunication**

Par :

DJEUKAM TEUNJEU GEOGY ARSENE

Matricule : 13C054

Sous l'encadrement académique et professionnel de :

M. TAKOUDJOU ALEXIS

Année académique 2015-2016

Je dédie ce travail a :

- Ma mère Mme DJEUKAM TCHOUTA PELAGIE
- Mon père M. DJEUKAM TEUNJEU FLAUBERT
- Ma grande sœur DJEUKAM NJIKE KEVINE

REMERCIEMENTS

Nous remercions d'abord Dieu puissant de nous avoir permis d'aller jusqu'au bout de ce projet en nous accordant la santé.

Nous Profitons de l'occasion pour remercier les personnes suivant :

- ❖ **M. TAKOUDJOU ALEXIS** pour nous avoir éclairé jusqu'à la fin de ce projet
- ❖ **Dr NZALI JEAN PIERRE** pour ces nombreux conseils tout au long de notre formation
- ❖ **Le corps enseignement de la faculté de science et technologie** pour les cours dispensés
- ❖ **Toute l'équipe du projet UDM-ACADEMY (M. TAKOUDJOU, M. KUATE, M. TEUMAWE, M. KAMGANG, M. KEUMEZEU, M. NKENDEM, M. TCHASSEM, Mme. TAKOUDJOU)**
- ❖ **Mes petites sœurs (DJEUKAM BRENDA, DJEUKAM LOÏKA et DJEUKAM FLORA)**
- ❖ **DJOMKAM MYRTIELLE**
- ❖ **LES PETITES FILLES DE LA GRANDE FAMILLE MAGNI NDANKEU**
- ❖ **La famille LONKEU NJOUBOUSSI**
- ❖ **La famille BATCHOU**
- ❖ **La famille DIBABO**
- ❖ **M. KEMEKONG FRANCOIS**
- ❖ **M. AZOTSIE ANDERSON**
- ❖ **A tous mes amis**
- ❖ **Tous les étudiants de la FST ayant contribué de près ou de loin à l'aboutissement de ce travail**

SOMMAIRE

REMERCIEMENTS	2
SOMMAIRE	3
LISTES DES FIGURES	5
LISTE DES TABLEAUX	6
RESUME	7
ABSTRACT	8
INTRODUCTION	11
Chapitre I : CONTEXTE, PROBLEMATIQUE ET METHODOLOGIE DE TRAVAIL DU CLIENT ANDROID DE GESTION ACADEMIQUE	13
I.1 ETUDE DE L'EXISTANT	13
<i>I.1.1 Présentation de la structure [1].....</i>	<i>13</i>
<i>I.1.1 Déroulement des tâches</i>	<i>14</i>
<i>I.1.2 Problèmes rencontrés</i>	<i>16</i>
I.2 PRESENTATION DU PROJET	17
<i>I.2.1 Présentation du projet UDM-ACADEMY.....</i>	<i>17</i>
<i>I.2.2 Interaction avec les autres applications ou logiciel du projet</i>	<i>17</i>
<i>I.2.3 Présentation de fonctions de l'application</i>	<i>18</i>
<i>I.2.4 Méthodologie du projet et Méthode de travail</i>	<i>18</i>
I.3 Problématique et objectif.....	21
<i>I.3.1 Problématique</i>	<i>21</i>
<i>I.3.2 Objectif de projet.....</i>	<i>21</i>
<i>I.3.3 Les étapes</i>	<i>21</i>
Chapitre II : ETAT DE L'ART SUR LE DEVELOPEMENT ET L'INTEGRATION DES CLIENT ANDROID BASE SUR LES SERVICES WEB	23
II.1 Les applications mobiles.....	23
II.2 Présentation d'Android	23
<i>II.2.1 Définition et caractéristiques.....</i>	<i>23</i>
<i>II.2.2 Présentation des outils du développement ANDROID</i>	<i>23</i>
<i>II.2.3 Architecture de développement Android</i>	<i>24</i>
<i>II.2.4 Compilation.....</i>	<i>27</i>
<i>II.2.5 Présentation de la technologie JEE utilisée côté serveur [3].....</i>	<i>28</i>
II.2.6 Service Web	29
Chapitre III : ANALYSE ET CONCEPTION D'UNE SOLUTION ANDROID DE GESTION ACADEMIQUE.....	32

III.1	SPECIFICATION FONCTIONNELLE DE L'APPLICATION	32
III.2	SPECIFICATION GENERALE	32
III.2.1	<i>Environnement du logiciel</i>	32
III.2.2	<i>Description générale du logiciel</i>	33
III.2.3	<i>Contraintes opérationnelles</i>	33
III.2.4	<i>Contraintes de réalisation</i>	34
III.3	SPECIFICATION OBJET	35
III.3.1	<i>Les acteurs du système</i>	35
III.3.2	<i>Cas d'utilisation</i>	35
III.4	CONCEPTION DE L'APPLICATION	41
III.4.1	<i>Architecture</i>	41
III.4.2	<i>Conception détaillée des cas d'utilisation</i>	42
Chapitre IV	: MISE EN ŒUVRE ET RESULTATS	46
IV.1	<i>Environnement de travail</i>	46
IV.1.2	<i>Environnement de développement</i>	46
IV.2	Implémentation	46
IV.2.1	<i>Réalisation des interfaces utilisateurs</i>	46
IV.2.2	<i>Implémentation des services web</i>	50
IV.2.3	<i>Appel des web services à partir du client</i>	50
IV.2.4	<i>Test tu client</i>	50
IV.3	Intégration	51
IV.3.1	<i>Adaptation des services web</i>	51
IV.3.2	<i>Adaptation du client</i>	51
IV.3.3	<i>Test final</i>	51
	CONCLUSION ET PERSPECTIVES	55
	REFERENCE BIBLIOGRAPHIQUE	56
	ANNEXES	57

LISTES DES FIGURES

Figure 1:page une de l’inscription académique manuel	57
Figure 2:page deux de l’inscription académique manuel	58
Figure 3:La méthode 2TUP	20
Figure 4: architecture tout en un Android.....	25
Figure 5: architecture deux tiers Android	26
Figure 6: architecture trois tiers Android.....	27
Figure 7: interaction entre Protocol [5]	31
Figure 8: Diagramme de déploiement	33
Figure 9: diagramme de cas d’utilisation.....	36
Figure 10: diagramme de package.....	41
Figure 11:diagramme de composant.....	42
Figure 12:diagramme de classe inscription académique	43
Figure 13: diagramme de classe consultation des notes	44
Figure 14: interface de connexion	47
Figure 15: interface de menu	48
Figure 16: interface de lancement	49
Figure 17: interface de changement de mots de passe	50
Figure 18: résultat obtenu pour l'inscription	51
Figure 19:resultat obtenue pour l'inscription académique.....	52
Figure 20:consultation de note	53
Figure 21:interface de connexion	54

LISTE DES TABLEAUX

Tableau 1: liste des abréviations	10
Tableau 2:activités du projet	19
Tableau 3:comparaison éclipse et Android studio	24
Tableau 4:description textuelle de l'inscription administrative	37
Tableau 5:Diagramme de séquence de la réinscription.....	37
Tableau 6:description textuelle de l'inscription académique	38
Tableau 7:Diagramme de séquence d'inscription académique	38
Tableau 8: description textuelle de la connexion à l'application	39
Tableau 9:description textuelle du changement de mot de passe	39
Tableau 10:description textuelle de la consultations des notes de cours.....	40
Tableau 11:description textuelle de la consultations des notes d'UE	40
Tableau 12:liste des fonctions cote serveur	45

RESUME

Le client Android de gestion académique du projet UDM-ACADEMY reposant sur une architecture JEE côté serveur et sur une technologie Android côté client est une application client-serveur ayant pour but de faciliter la vie de l'étudiant sur le campus. Nous nous attardons sur l'inscription et la consultation des notes. En ce qui concerne l'inscription qui jusqu'à l'an passé se faisait manuellement, elle a été informatisée via une plateforme web. Cependant, bien qu'ayant résolu le problème de l'inscription manuel, cette plateforme web présente encore deux inconvénients majeurs dans la mesure où non seulement elle est très gourmande en ressource mais, il est difficile pour les étudiants de s'inscrire à partir de leur Smartphone car le rendu visuel de l'interface est difforme comparé à celui offert par l'ordinateur. De même, l'étudiant n'aura plus besoin de se déplacer en période de vacance pour entrer en possession de ses notes; ce qui avant constituait un problème à prendre très au sérieux car l'affichage de notes se fait uniquement sur le campus principal de l'université.

Dans le but de résoudre ce problème, nous utilisons l'environnement de développement intégré Android Studio qui nous permet de faire des interfaces attrayantes, de qualité et s'adaptant à plusieurs résolutions d'écrans afin de faciliter l'utilisation par l'étudiant. L'architecture utilisée nous amène à implémenter d'une part des services web représentant le métier de notre application et d'autre part un client Android qui consomme ces services web.

Le module Android de gestion académique dans sa première version offre les fonctionnalités d'inscription administrative, académique, de consultation des notes de cours et d'unités d'enseignements tout ceci avec une connexion à l'application comme préalable. Il est à rappeler que l'architecture utilisée ici est une architecture n-tiers.

Au terme de ce travail nous avons effectué des tests concluant sur les données réelles fournies par le service académique. Nous envisageons de développer des modules supplémentaires tels que le suivi des requêtes d'étudiant et l'accès aux actualités académiques.

Mot clefs: Android studio, UDM-ACADEMY, JEE, architecture n-tiers

ABSTRACT

The Android client academic management of the UDM-ACADEMY project based on JEE architecture server side and client-side technology Android is a client-server application that aims to facilitate student life on campus. We focus on registration and consultation notes. Regarding the registration which until last year was manual, it was computerized via a web platform. However, although having solved the problem of manual entry, this web platform still has two major drawbacks to the extent that it is not only a very demanding resource but it is difficult for students to register from their smartphone as the visual interface is deformed compared to that offered by the computer. Similarly, the student will no longer need to travel for holiday period to take possession of his marks; which before was a problem to be taken seriously because the display of marks is made only on the university's main campus.

In order to solve this problem, we use the integrated development environment Android studio that allows us to make attractive interfaces, attributes and adapting several screen resolutions in order to ease the use by the student. The architecture used leads to firstly implemented web services representing the profession of our application and also an Android client that consumes the web service.

The Android module academic management in its first version offers the administrative registration features, academic, consulting lecture notes and teaching units all with a prior application as the connection. It is recalled that the architecture used here is an n-tier architecture.

At the end of this work we have carried out tests concluded on actual data provided by the Registrar's Office. We plan to develop additional modules such as tracking student queries and access to academic news.

Keywords: Android Studio UDM-ACADEMY, JEE, n-tier architecture

LISTE DES ABREVIATIONS [4]

SIGLES	SIGNIFICATIONS	DESCRIPTIONS
2TUP	Two Tracks Unified Process	Processus de développement logiciel.
API	Application Programming Interface	Ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.
CC	Contrôle Continu	L'évaluation continue est une évaluation permanente qui s'exerce à tout moment de l'apprentissage. L'apprenant bénéficie d'un renforcement immédiat.
DAO	Data Access Object	Technique visant à regrouper les accès aux données persistantes dans des classes à part, plutôt que de les disperser.
EJB	Entreprise Java Beans	Composants logiciels côté serveur pour la plateforme de développement Java EE.
IDE	Integrated Development Environment (Environnement de développement intégré)	Ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels.
J2EE	Java 2 Entreprise Edition	Spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise.
JDBC	Java Data Base Connectivity	Middleware java d'accès aux données.
JNDI	Java Naming and Directory Interface	API Java de connexion à des annuaires, notamment des annuaires LDAP.
JPA	Java Persistence API	Interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java.
PHP	Hypertext Pre-processor	Langage de programmation 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 orienté objet comme C++.
SGBD	Système de Gestion de Bases de Données	Logiciel système destiné à stocker et à partager des informations dans une base de données.

SQL	Structured Query Language (langage de requête structurée)	Langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.
UdM	Université des Montagnes	L'Université privée à but non lucratif situé à Bangangté au Cameroun.
UE	Unité(s) d'Enseignement	Une formation est composée d'Unités d'Enseignement (UE) qui correspondent à un même domaine thématique de formation.
UML	Unified Modeling Language	Langage de modélisation graphique à base de pictogrammes.
XML	Extensible Markup Language	Langage informatique de balisage générique. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages ayant son vocabulaire et sa grammaire.

Tableau 1: liste des abréviations

INTRODUCTION

L'**Université des Montagnes** à sa création en 2000 était constituée de la **FSS** (Faculté des Sciences de la Santé) comportant 2 filières (Médecine et pharmacie), ensuite en 2001 s'est ajoutée la **FST** (Faculté des Sciences et Technologies) comportant les filières IMB (Instrumentation et Maintenance Biomédicale) et IR (Informatique et réseaux). Le nombre de filières augmente d'années en années dans chaque faculté ; ce qui fait qu'aujourd'hui on compte un effectif d'environ 2000 étudiants. Cet accroissement de l'effectif rend de plus en plus difficile la gestion académique (c'est-à-dire l'inscription des étudiants, la gestion de leurs notes, de leurs requêtes, de leurs dossiers académiques, et bien d'autres activités liées au cursus de leur formation), étant donné que cette gestion n'est que semi-automatique.

Pour pallier à ce problème, il serait moins coûteux en temps et en énergie pour l'**UdM** (Université des Montagnes), d'avoir un logiciel fiable, robuste, flexible et facile à utiliser et à maintenir, qui permettrait à ses différents services d'intégrer, traiter, harmoniser et produire les informations pertinentes de tous les étudiants. C'est ainsi que le logiciel **UDM-ACADEMY** a vu le jour. Ce **logiciel** est en cours de développement et est constitué d'un ensemble de modules interactifs à savoir la configuration, l'inscription, la gestion des notes, l'administration, la production des états, la gestion du profil étudiant, la gestion du paiement des droits universitaires, la numérisation du dossier étudiant, production des actes académiques, gestion des enseignants et des contrats, gestion des enseignants et des emplois de temps.

A cela nous ajoutons un client **ANDROID** qui offre à l'étudiant les fonctionnalités suivantes : Réinscription, inscription académique, consultation des notes. Les inscriptions se faisaient jusqu'ici en remplissant des formulaires retirés à l'université. L'accès aux notes se fait par consultations des affichages (pouvant être retiré après avoir passé une certaine période) au campus principal de l'université. L'accès aux notes était donc compliqué surtout dans les périodes de vacance où les étudiants ne sont pas sur le campus. La première difficulté en ce qui concerne l'inscription vient du fait que l'étudiant doit toujours aller au campus pour le faire, la seconde réside dans l'accès au formulaire d'inscription : les responsables en charge de les distribuer ne sont pas toujours disponibles pour le faire. Un client web d'inscription en ligne a été développé dans le cadre du projet UDM-PHOENIX mais cela

présente certaines faiblesses dans la mesure où l'accès est difficile car il est trop gourmand en ressource et il est difficile pour les étudiants d'accéder à toutes les fenêtres à partir de leurs téléphones car il y a un mauvais rendu visuel.

Le problème ainsi posé, nous avons pour objectif de développer un client ANDROID de gestion académique basé sur les services web interopérable avec le système existant (UDM-ACADEMY) et répondant au besoin de l'utilisateur (les étudiants). La suite de ce document présente de manière synthétique les différentes étapes qui ont conduit à la mise sur pied de notre client. Ce document est reparti en quatre principaux chapitres :

Le premier chapitre contient l'étude de l'existant, la présentation du projet et de la méthodologie de travail. Il sera question pour nous de situer le contexte de travail, présenter le projet et de ressortir la problématique. Le second chapitre concerne l'état de l'art, nous présenterons les différentes approches et techniques de mise en œuvre d'un client ANDROID. Le troisième chapitre présente l'analyse et la conception de notre client ANDROID de gestion académique qui sera basé sur les services web. Le quatrième chapitre présente la mise en œuvre. Nous terminerons par une conclusion et des perspectives en vue d'un éventuel approfondissement de ce travail.

Chapitre I : CONTEXTE, PROBLEMATIQUE ET METHODOLOGIE DE TRAVAIL DU CLIENT ANDROID DE GESTION ACADEMIQUE

I.1 ETUDE DE L'EXISTANT

I.1.1 Présentation de la structure [1]

L'**Université des Montagnes** est née d'une profonde réflexion sur l'état de l'enseignement supérieur au Cameroun, initiée depuis les années 1990 par des hommes et des femmes du monde universitaire, du monde des affaires et des différentes catégories professionnelles rassemblés au sein de l'**Association pour l'Education et le Développement (AED)**. Ce n'est qu'en 2000 qu'elle ouvre ses portes. Située dans la région de l'Ouest Cameroun, au niveau du département du **NDE**, l'**UdM** est subdivisée en 03 (trois) campus (Mfetum, Tchougouo, Banenkane). Sa devise est *Semper Altissimo Ascendere* (Toujours Rechercher l'Excellence).

L'**UdM** s'organise en 03 instituts fonctionnels:

- Institut Supérieur des Sciences de la Santé (**ISSS**)
- Institut Supérieur des Sciences et de Technologie (**ISST**)
- Institut d'Etudes Africaines (**IEA**);

L'**ISSS** est subdivisé aujourd'hui en 12 filières et spécialités:

- Médecine (**MED**);
- Pharmacie (**PHA**);
- Chirurgie Dentaire (**DEN**);
- Médecine Vétérinaire (**MVET**);
- Santé de la reproduction (**SAR**);
- Environnement et Changement Climatique (**ENCC**);
- Agroforesterie (**AGRF**);
- Agronomie (**AGRN**);
- Biologie Médicale (**BIO**);
- Imagerie Médicale (**IMG**);

- Kinésithérapie (KIN);
- Sciences Infirmières (SIN).

Les quatre premières préparent au doctorat et les autres à la licence.

Quant à lui,

L'**ISST** est également subdivisé en 8 filières et spécialités à savoir :

- Informatique, Réseaux et Télécommunications (IRT);
- Maintenance du Matériel Biomédicale (MB);
- Mathématique et Informatique Appliqués aux Finances (MIAF);
- Energie Renouvelable et Génie Climatique (ERGC);
- Génie Civil (GC);
- Génie Mécanique (GM);
- Génie Biomédical (GBM);
- Génie Informatique et Systèmes (GIS);

Les six premières préparent à la licence et les autres au master.

L'UdM est subdivisé en plusieurs services entre autres:

- La comptabilité
- Le service de programmation des enseignements
- Le service de la communication
- Le service des admissions et des affaires académiques

Notre stage a été effectué au service des admissions et des affaires académiques.

I.1.1 Déroulement des tâches

a. Réinscription

Les inscriptions se font chaque année, pendant une période bien précise (début d'année) et se font manuellement. Les étudiants doivent alors remplir un formulaire bâti sur trois grandes parties (**identification de l'étudiant, identité des parents, sponsors des études et autres informations**) tel que présenté dans l'annexes:

Il est important de noter que le processus d'inscription administratif inclut une vérification de l'état financier de l'étudiant. A l'issue de cette inscription, le montant total que l'étudiant doit payer au cours de l'année est initialisé. Dans le projet UDM-ACADEMY, il existe une plateforme web d'inscription qui a été mis en production au début de l'année académique 2016/2017.

a. Inscription académique

L'étudiant à la suite de sa réinscription administrative doit faire une inscription académique ; elle consiste à s'inscrire à un niveau de formation en vue de l'obtention d'un diplôme en choisissant les UE. L'inscription académique tient compte de l'année antérieure dans la mesure où les UE prioritaire sont celles qui n'ont pas été validées l'année d'avant. L'étudiant doit choisir un maximum de trente-quatre (34) crédits et un minimum de trente (30) crédits. On peut donc avoir à faire au cas exceptionnel où l'étudiant redouble un niveau dans ce cas; l'étudiant peut être amené à choisir moins de trente (30) crédits pour le semestre un (01) et éventuellement à faire une inscription anticipée pour le niveau suivant. Sur la plateforme web d'inscription citée plus haut on retrouve aussi l'inscription académique.

b. Accès aux notes

Au cours d'une année académique l'étudiant est donc amené à passer pendant les sessions d'évaluations les différentes UE choisies lors de son inscription académique. Les UE étant constituées de cours, le processus de passage des UE (unité d'enseignement) est donc le suivant : l'étudiant passe les cours de l'UE auxquels il obtient des notes. Un calcul est fait à l'aide des informations telles que les notes des cours, les poids des cours dans l'UE, le pourcentage du cours dans l'UE et la nature (examen ou CC) de la note pour déterminer la note sur 100 de l'étudiant sur l'UE ainsi que son grade et le point obtenu (noté sur 4). La suite se fait par affichage collectif au babillard du campus principal de l'université.

En ce qui concerne les notes de CC (contrôle continu), les étudiants entrent en possession de celles-ci soit à la remise de leur feuille par le professeur si celui-ci est permanent, soit par affichage des notes de la matière ou lors de l'affichage des UE. Ceci est donc un processus assez aléatoire dans la mesure où d'un enseignant à un autre, l'entrée en possession des notes par les étudiants peut se faire par différentes voies.

I.1.2 Problèmes rencontrés

a. Réinscription

Après avoir étudié de près le formulaire d'inscription, nous pouvons émettre l'observation suivante : l'étudiant est toujours appelé à remplir des champs qui ne changeront jamais (**le nom, le prénom, la date de naissance, le sexe, la nationalité, la première langue officielle, le nom et le prénom du père et de la mère leurs nationalités**) dû au fait que le même formulaire soit utilisé pour l'inscription des nouveaux étudiants que pour la réinscription. Nous pouvons aussi relever le problème d'accessibilité aux fiches d'inscription. En ce qui concerne l'inscription en ligne on constate des problèmes liés à l'accès au site car ce dernier demande beaucoup de ressources. Comme autre difficulté on constate que les étudiants n'ont pas un bon rendu visuel quand ils accèdent au site à partir de leurs smartphones.

b. Inscription académique

Cette inscription se fait en ligne depuis l'an passé et présente les mêmes problèmes que l'inscription administrative (en ligne). En plus, les publications tardives des résultats constitue un frein à son bon déroulement. Lorsque l'inscription était manuelle, le problème se situait au niveau de l'accessibilité aux fiches d'inscription car en début d'année, ces fiches sont distribuées à des heures précises (auxquelles les étudiants ne sont pas forcément tous disponibles ou sont en cours).

c. Accès aux notes

Ici l'accès aux notes est le véritable problème car l'affichage se fait dans un seul campus. Pendant les périodes de vacances, les étudiants n'étant pas sur le campus il peut arriver des incidents dans la mesure où on peut afficher et retirer des notes du babillard sans que les étudiants ne soient au courant.

I.2 PRESENTATION DU PROJET

I.2.1 Présentation du projet UDM-ACADEMY

UDM-ACADEMY est un projet issu de l'extension du projet UDM-PHOENIX ayant pour chef de projet M. TAKOUDJOU Alexis. Ce projet contient plusieurs fonctionnalités dont certaines sont en phase d'expérimentation, d'autres en cours de développement et d'autres déjà en production (le client web d'inscription évoqué plus haut).

Ce projet compte donc les modules suivants :

- Une application serveur ;
- Un client web ayant les modules suivants :
 - Inscription académique ;
 - Inscription administrative.
- Un client desktop ayant les modules suivants :
 - Gestion des notes ;
 - Gestion des paiements ;
 - Numérisation du dossier étudiant ;
 - Production et suivi des cartes d'étudiant ;
- Un client .NET ayant le module de programmation des enseignements ;
- Un client mobile Android ayant les modules suivant :
 - L'inscription administrative ;
 - L'inscription académique ;
 - La consultation des notes ;
 - La modification du mot de passe.

I.2.2 Interaction avec les autres applications ou logiciel du projet

Notre application étant une entité d'un projet existant, il est donc en interaction avec les modules suivants :

- Le client desktop ; en particulier les modules de configuration et de traitement des notes dans la mesure où ce sont les notes provenant du module traitement des notes qui sont affichées ;
- Le client web d'inscription car notre client mobile vient se greffer à ce module afin de pallier à ses manques.

I.2.3 Présentation de fonctions de l'application

Les fonctionnalités de notre client Android prévues à l'issue de nos objectifs sont :

- Connexion / déconnexion
- Réinscription
- Inscription académique
- Consultation des notes par unité d'enseignement
- Consultation des notes de cours
- Changement de mot de passe

I.2.4 Méthodologie du projet et Méthode de travail

a. méthodologie du projet

Afin de parvenir à ce résultat tout en respectant la méthodologie de développement des logiciels, le projet comprend les activités suivantes :

Activités	Durée en semaine
Etude préalable	2
Initiation du projet	1
Analyse et spécification	1
Conception	2

Implémentation du serveur d'application	1
Implémentation du client Android et Test	1
Intégration du serveur au serveur d'UDM-ACADEMY et adaptation de client Android	1
Test final	1
Total	11

Tableau 2:activités du projet

b. Méthode de travail [2]

Pour un bon suivi des activités, nous utilisons la méthode 2TUP qui fait partie de la famille des processus unifiés (UP). Cette famille est caractérisée par le fait qu'elle est itérative et incrémentale, guidée par la gestion des risques, les besoins utilisateurs et centrée sur l'architecture du logiciel. Une itération de la méthode 2TUP est constituée de 3 branches formant un « Y ». Les branches fonctionnelles et techniques se feront en parallèle. La troisième branche est le résultat de la fusion des deux premières branches. On peut donc la représenter comme suit :

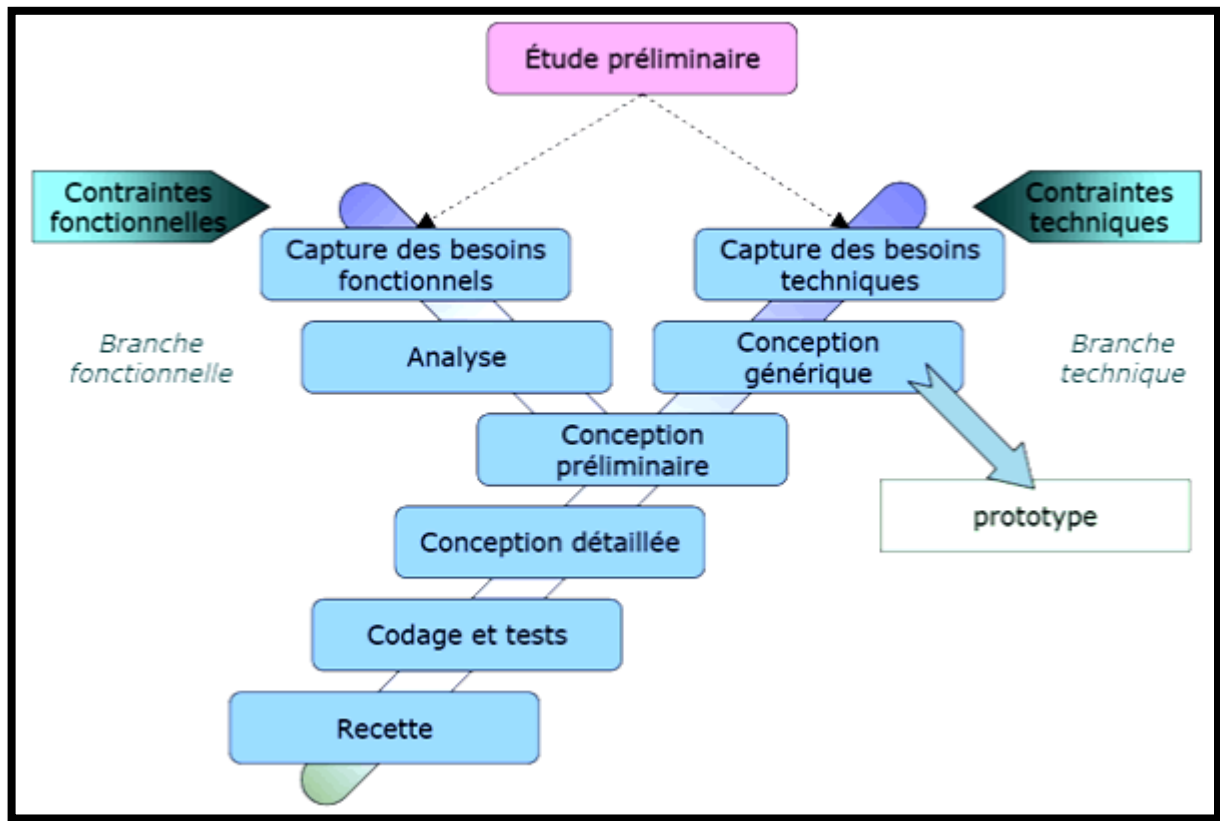


Figure 1: La méthode 2TUP

La gestion d'un tel processus est organisée d'après les 4 phases suivantes :

- **Étude préliminaire:** c'est ici qu'on évalue la valeur ajoutée du développement et la capacité technique à le réaliser (étude de faisabilité);
 - **Elaboration:** sert à confirmer l'adéquation du système aux besoins des utilisateurs et à livrer l'architecture de base;
 - **Construction :** sert à livrer progressivement toutes les fonctions du système;
- Transition:** déployer le système sur des sites opérationnels.

I.3 Problématique et objectif

I.3.1 Problématique

La réalisation de ce projet nous amène à nous interroger sur comment développer une solution de gestion académique dans ce contexte qui réponde au besoin de l'étudiant, qui s'arrime à la nouvelle technologie de l'information et de la communication embrassées par les étudiants et qui soit interopérable avec le système existant ; comment choisir l'environnement de développement intégré approprié pour ce type de travail ; comment choisir les outils de modélisation de la solution, développer et intégrer les modules serveur.

I.3.2 Objectif de projet

En fonction des besoins à satisfaire en vue de l'extension des fonctionnalités des logiciels à développer dans le cadre du projet UDM-ACADEMY, nous envisageons du point de vue général, de développer le client Android de gestion académique suivant une architecture JEE basée sur les services web et l'intégré au projet parent qui est UDM-ACADEMY. L'atteinte de nos objectifs ne pourra être possible qu'avec la bonne réalisation des étapes primordiales telles que la mise en place d'un environnement de développement intégré dont le fruit de la compilation est exécutable sur un téléphone ayant un système Android, la modélisation du système, le développement du client et du serveur ainsi que l'intégration du module serveur.

I.3.3 Les étapes

L'atteinte de nos objectifs se fera au travers des étapes suivantes :

- Faire une étude de l'existant,
- Faire Ressortir la spécification des besoins de l'utilisateur,
- Mettre en œuvre des interfaces de qualité,
- Faire la revue de la littérature et effectuer les choix technologiques de l'heure,
- Construire un modèle de donnée intégrable au modèle existant,
- Faire les services web qui nous permettrons de réaliser les fonctionnalités

- Consommer ces services web à partir du client Android,
- Tester le fonctionnement du client sur des données réelles,
- Intégrer notre modèle au modèle existant,
- Adapter les services web et le client au modèle final,
- Faire les tests Intégrations,
- Produire la documentation du client.

Chapitre II : ETAT DE L'ART SUR LE DEVELOPEMENT ET L'INTEGRATION DES CLIENT ANDROID BASE SUR LES SERVICES WEB

II.1 Les applications mobiles

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un smartphone ou un appareil ayant un système d'exploitation Windows. Nous distinguons plusieurs systèmes d'exploitation pour mobile, nous citons entre autre WINDOWS MOBILE, IOS APPLE, BLACKBERRY OS et ANDROID [7]. De tous ces systèmes il est à noter que le plus répandu en Afrique est Android en raison de son prix et des facilités qu'il offre.

II.2 Présentation d'Android

II.2.1 Définition et caractéristiques

Android est un logiciel de type système d'exploitation mobile comme « Windows » est un système d'exploitation sur PC. Il fait tourner votre mobile.

Une application Android est un logiciel pouvant fonctionner dans appareil muni d'un système Android.

II.2.2 Présentation des outils du développement ANDROID

Le développement sous Android est possible sous plusieurs environnements de développement intégré tels que : ANDROID STUDIO ECLIPSE OU NETBEANS. Cependant, il est aussi possible d'utiliser la technologie apache cordova qui nous permet de faire un développement hybride : web et mobile. Après avoir étudié les environnements ECLIPSE et ANDROID STUDIO

Nous pouvons dresser le tableau suivant.

	ADT (Eclipse) 2013	Android Studio 2.1
Facilité d'installation	Moyen	Simple
Langue	Nombreuses	Anglais
Performance	Peut être lourd	Rapide
Système de build	Ant	Gradle
Génération de variante et de multiple APK	Non	Oui
Android Code completion et refactoring	Base	Avancé
Editeur d'interface graphique	Oui	Oui
Signature d'APK et gestion de Keystore	Oui	Oui
Support NDK	Oui	Oui
maintenance et évolution	Non	Oui
occupation de l'espace sur le disque	Faible	Grande
plus grande version d'Android	4.4	7

Tableau 3: comparaison eclipse et Android studio

Nous avons donc choisi ANDROID STUDIO 2.1 sortie en Aout 2016.

II.2.3 Architecture de développement Android

a. Tout en un

Dans cette architecture on a un exécutable Android contenant à la fois la couche vue, la couche donnée et la couche application. La couche donnée est représentée ici par un SGBD (Système de Gestion de Base de Données) embarqué tel que SQLite ou Realm.

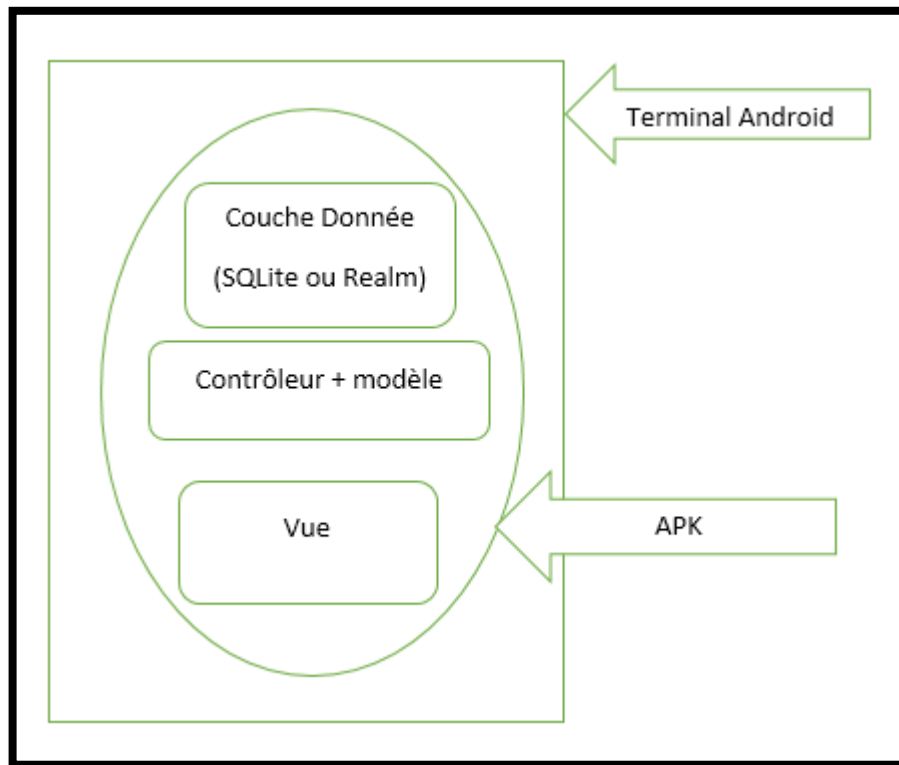


Figure 2: architecture tout en un Android

b. Client-serveur deux tiers

Dans cette architecture on a un exécutable Android contenant uniquement la couche vue et la couche application. Dans ce cas la couche donnée est logée sur un serveur de base donnée accessible via le réseau grâce à JDBC (Java Data Base Connectivity).

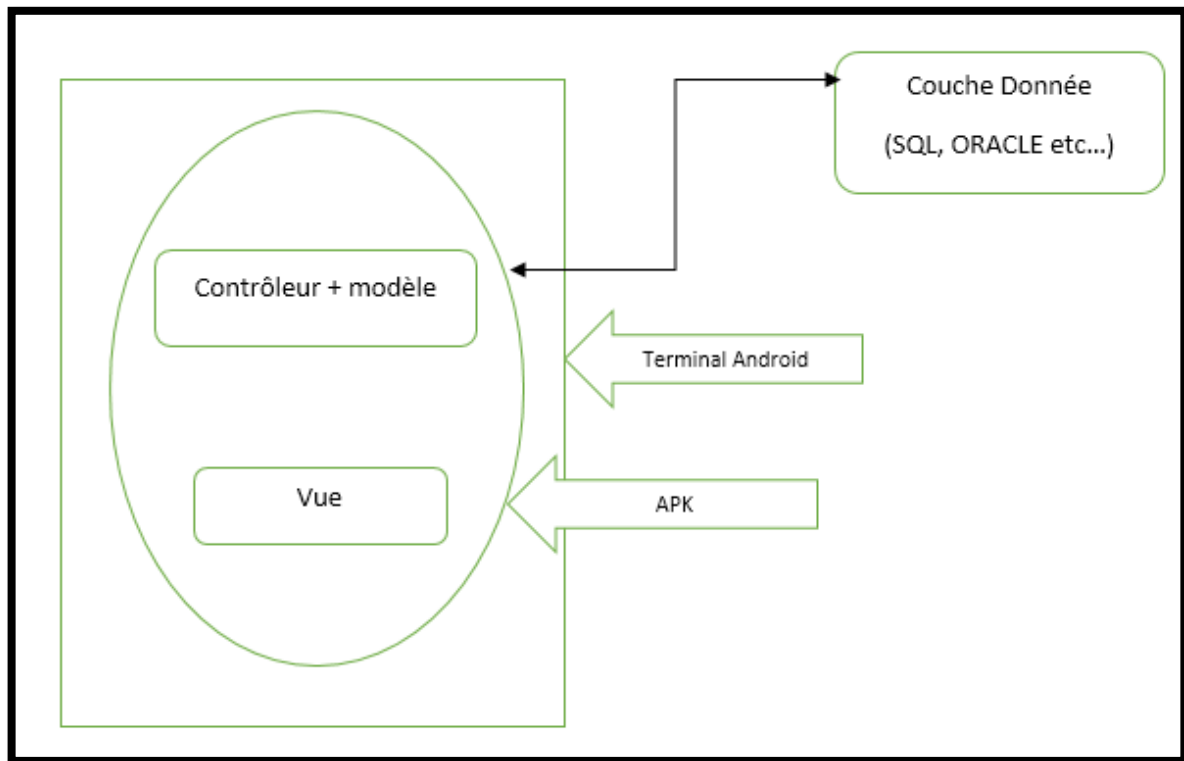


Figure 3: architecture deux tiers Android

c. Client-serveur trois tiers

Dans cette architecture on a un exécutable contenant essentiellement les vues, leurs contrôleurs et l'implémentation des méthodes permettant d'accéder aux services web accessible sur le réseau grâce à L'API KSOAP2 dans le cas du Protocol SOAP.

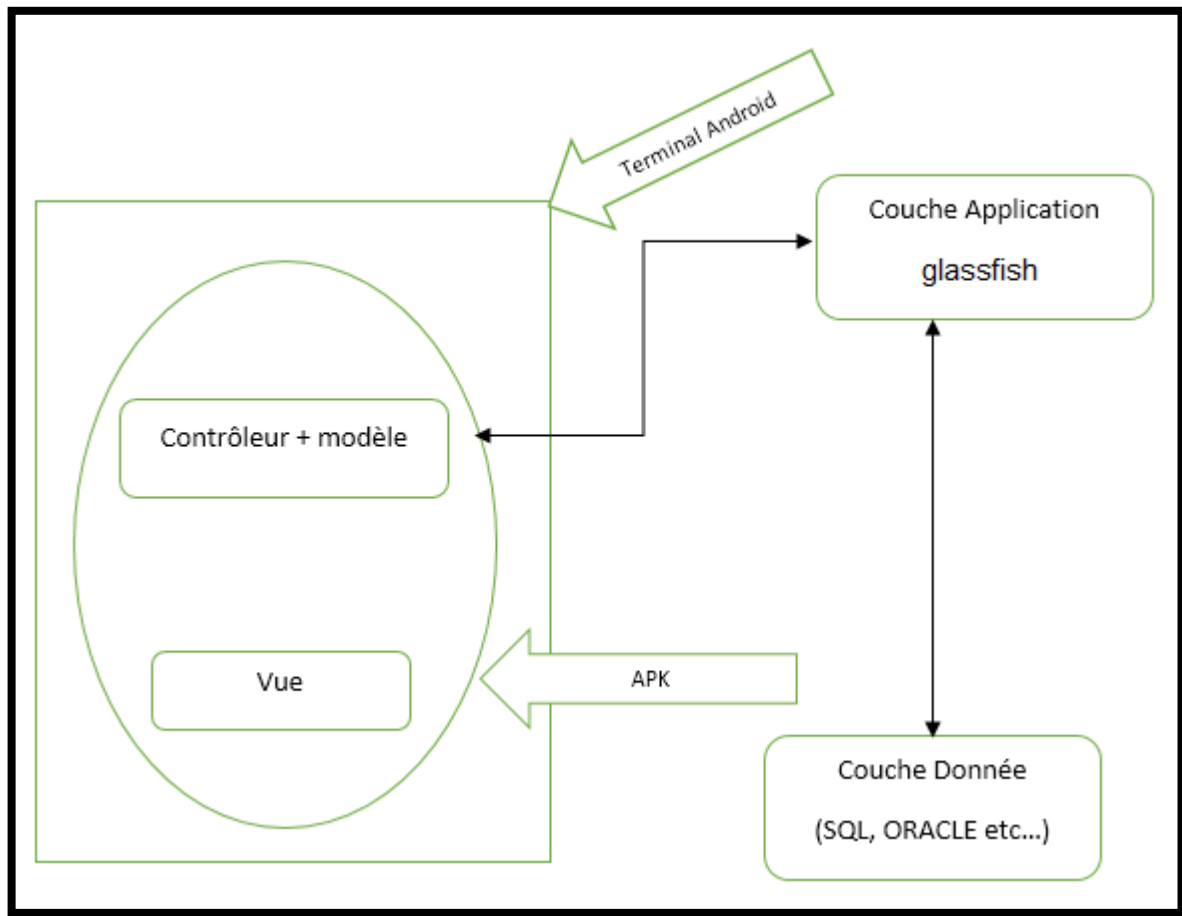


Figure 4: architecture trois tiers Android

Différemment du web ou desktop le développement mobile fait intervenir une notion de compilation importante car il est question de développer dans un environnement et de compiler sur un autre.

II.2.4 Compilation

En informatique, la **compilation** désigne le processus de transformation d'un programme écrit dans un langage lisible par un humain en un programme exécutable par une machine. De manière plus générale, il s'agit de traduire un programme écrit dans un langage source en un programme écrit dans un langage cible. [6]

Dans notre cas on a à faire à une compilation croisée car les programmes exécutables créés doivent être compréhensibles avec un Smartphone ayant un système ANDROID d'où l'utilisation d'un téléphone portable dans notre environnement de développement.

Il est important de noter que nous développerons avec la version 4.0.3 d'ANDROID mais que les téléphones pouvant supporter notre application sont des téléphones ayant une API comprise entre 14 (version 3.2.2 : honeycomb) et 23 (version 6.0.1 : marshmallow qui est l'avant dernier version d'ANDROID) incluses.

II.2.5 Présentation de la technologie JEE utilisée côté serveur [3]

a. Définition de JEE

JEE est une plateforme de développement d'application s'appuyant sur le langage Java. Les architectures JEE sont utilisées essentiellement pour l'élaboration d'applications présentant une architecture complexe. Ainsi, la mise en place de ce type d'application nécessite l'intervention de plusieurs acteurs et l'utilisation de plusieurs composants. D'où une exigence accrue en matière d'interopérabilité et d'interconnexion qui s'avèrent souvent difficile à mettre en place. Ce type d'application nécessite également des données hétérogènes (souvent de différents formats) pouvant provenir de différentes applications externes.

Le développement d'une telle application doit tenir compte des composants logiciels préexistants mais également des évolutions futures envisageables (changement de base de données, autres types de clients, changement de logique métier, etc.). Les applications JEE sont typiquement utilisées dans le cadre des architectures distantes (type client/serveur).

b. Atout de JEE

L'utilisation de JEE pour développer et exécuter une application présente plusieurs avantages :

- une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement;
- la possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, etc;

- la possibilité de choisir les outils de développement et le serveur d'application utilisés qu'ils soient commerciaux ou libres;

- JEE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend d'une part des besoins auxquels doit répondre l'application d'autre part des compétences dans les différentes API de JEE.

L'architecture d'une application se découpe idéalement en au moins trois tiers :

- **la partie cliente** : c'est la partie qui permet le dialogue avec l'utilisateur. Elle peut être composée d'une application mobile, d'une application web ou une application de bureau;
- **la partie métier** : c'est la partie qui encapsule les traitements (dans des EJB ou des JavaBeans);
- **la partie donnée** : c'est la partie qui concerne le stockage des données.

II.2.6 Service Web

a. Définition de service web

Un service web est un programme informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone. [4]

b. Caractéristique des services web

La technologie des services web repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur web.

Un service web possède les caractéristiques suivantes :

- il est accessible via le réseau ;
- il dispose d'une interface publique (ensemble d'opérations) décrite en XML ;
- ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire ;
- il communique en utilisant des messages XML, ces messages sont transportés par des protocoles Internet (généralement HTTP, mais rien n'empêche d'utiliser d'autres protocoles de transfert tels : SMTP, FTP, BEEP, etc.) ;
- l'intégration d'application en implémentant des services web produit des systèmes faiblement couplés, le demandeur du service ne connaît pas forcément le fournisseur.

Ce dernier peut disparaître sans perturber l'application cliente qui trouvera un autre fournisseur en cherchant dans l'annuaire.

c. Technologie utilisée pour accéder aux services web sous Android

L'accès aux services web se fait via la technologie telles que :

- **REST**

REST (Representational State Transfer) est une architecture de services Web. Élaborée en l'an 2000 par Roy Fielding, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd et d'autres travaux fondamentaux, REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

- **SOAP**

SOAP (Simple object Access Protocol) est un protocole standard de communication. C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes. Les services web circulent sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

- WSDL

WSDL (Web Services Description Language) est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

- UDDI

UDDI (Universal Description, Discovery and Integration) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients

L'interaction entre tous ceux-ci peut se voir comme suit

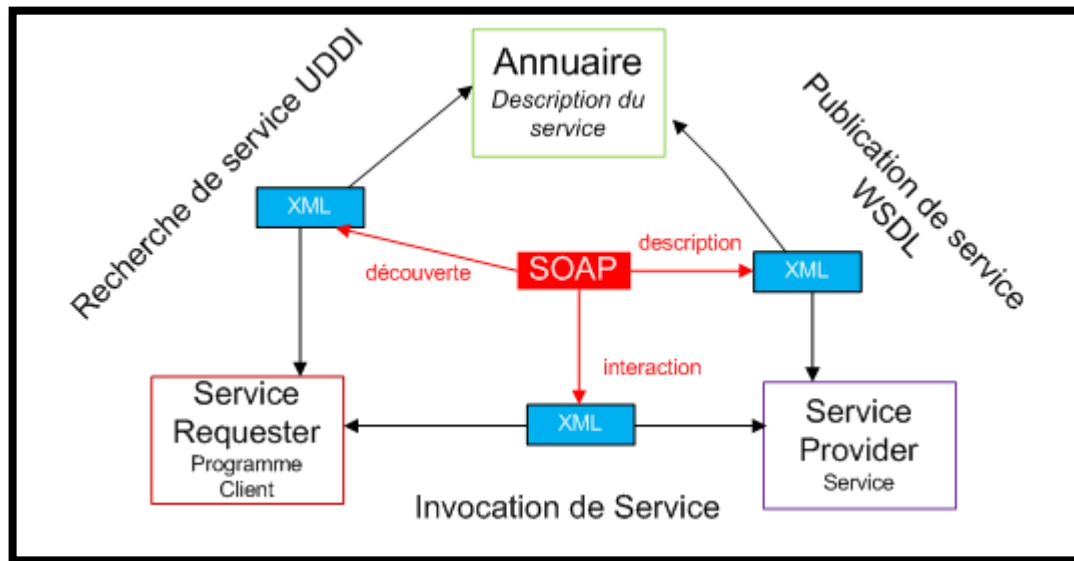


Figure 5: interaction entre Protocol [5]

Chapitre III : ANALYSE ET CONCEPTION D'UNE SOLUTION ANDROID DE GESTION ACADEMIQUE

III.1 SPECIFICATION FONCTIONNELLE DE L'APPLICATION

Les grandes fonctionnalités du système sont :

- Connexion / déconnexion
- Réinscription
- Inscription académique
- Consultations des notes d'unité d'enseignement
- Consultations des notes de cours
- Changement de mot de passe

III.2 SPECIFICATION GENERALE

III.2.1 Environnement du logiciel

Pour pouvoir utiliser notre application chaque utilisateur devra avoir à sa disposition :

- Un téléphone avec un système ANDROID ou sera installé l'application ;
- Une connexion qui lui permette d'atteindre le serveur de l'université ;
- Compte utilisateur fourni par l'université.

III.2.2 Description générale du logiciel

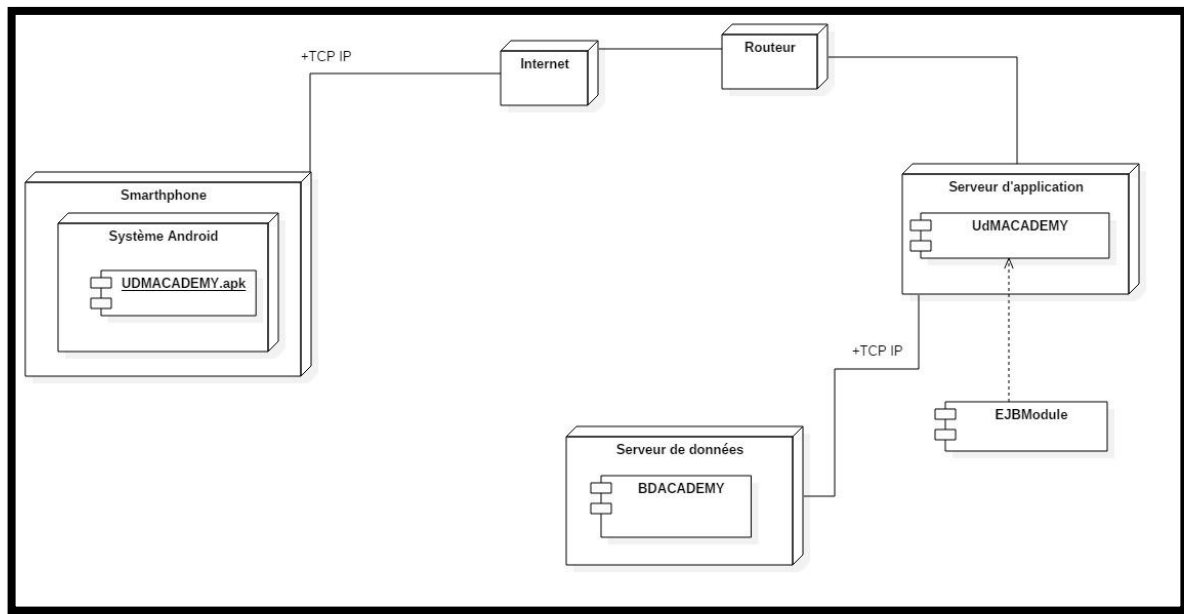


Figure 6: Diagramme de déploiement

III.2.3 Contraintes opérationnelles

a. Performances

Elles dépendront des capacités du téléphone de l'étudiant d'une part et du débit de la connexion qu'il possède d'autre part.

b. installation

L'installation se fera sur les téléphones de chaque étudiant de l'université et une fois connecté sur internet, il pourra jouir des fonctionnalités offertes par l'application.

III.2.4 Contraintes de réalisation

a. Contrainte d'évolution

Notre application offre un bon nombre de fonctionnalités mais qui ne saura être suffisant pour l'étudiant. Dans le futur nous envisageons développer un module qui permettra aux étudiants d'accéder à leur emploi de temps collectif et de suivre leurs requêtes.

b. Contrainte de développement

- **Matériel**

Pour mener à bien ce travail on aura besoin du matériel ci-après

- Une machine qui fera office de serveur de base de données
- Une machine fera office de serveur d'application
- Une machine où sera développé le client ANDROID
- Enfin un téléphone ANDROID pour les tests

- **Les langages**

Les langages utilisés sont :

- **Le JAVA** : utilisé pour faire le modèle et le contrôleur de notre application et aussi pour développer le serveur ;
- **JQUERY** : utilisé pour les requêtes du serveur d'application vers le serveur de données ;
- **Le XML** : utilisé pour générer les vues de notre application ;
- **Le SQL** : utilisé pour faire les manipulations directes dans la base de données ;

- **Les logiciels**

Les logiciels utilisés sont les suivants :

- **WIN 'Design** : pour nos différents diagrammes ;
- **STARUML** : pour le diagramme ;
- **ARGOUML** : pour le diagramme ;
- **SQL Serveur** : qui sera notre **SGBD** ;
- **NETBEANS** ou sera développé notre serveur d'application ;
- **ANDROID STUDIO** ou sera développé notre client ANDROID.

III.3 SPECIFICATION OBJET

III.3.1 Les acteurs du système

Notre système étant plus orienté du côté de l'étudiant le seul acteur ici est donc :

- Etudiant

III.3.2 Cas d'utilisation

Les différents cas d'utilisation primaires sont :

- Se connecter ;
- Modifier son mot de passe ;
- Consulter ses notes ;
- Faire réinscription administrative ;
- Faire inscription académique ;

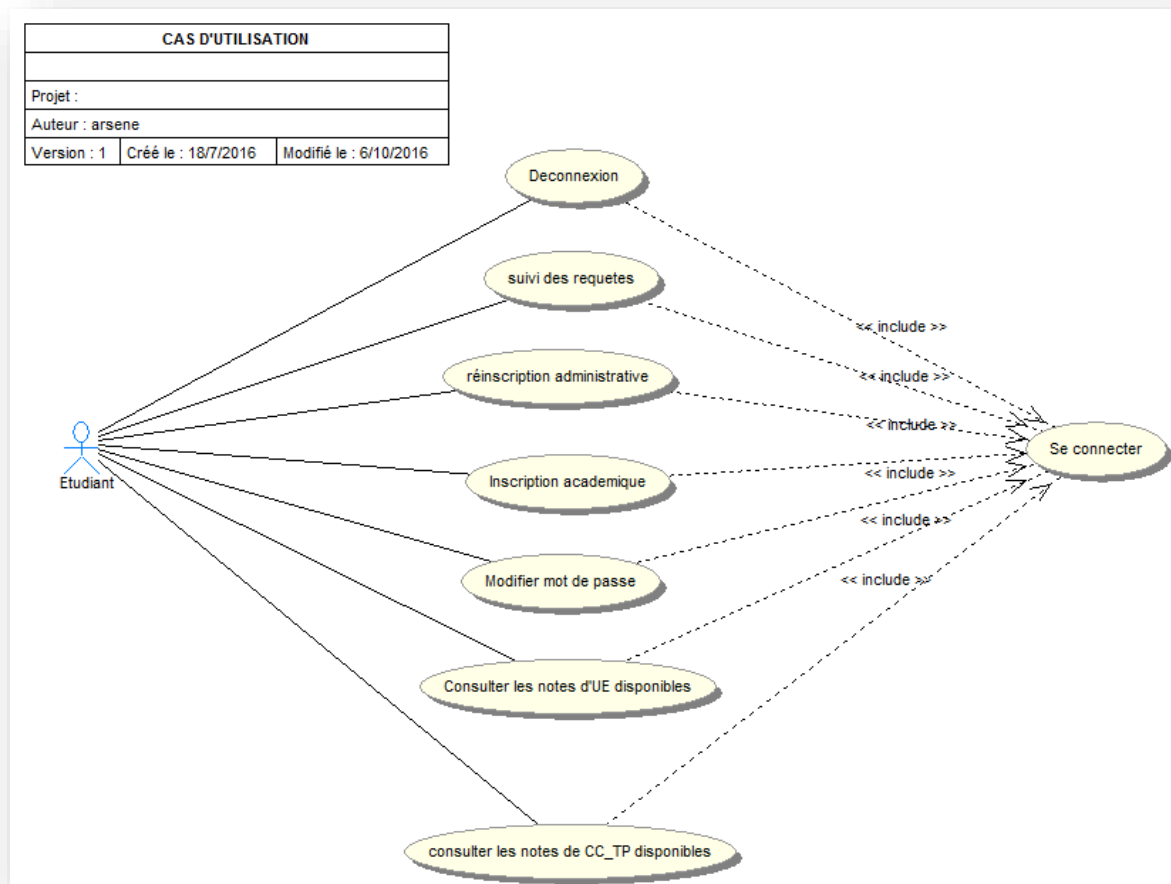


Figure 7: diagramme de cas d'utilisation

a. Faire réinscription administrative

Titre	Se réinscrire
Acteur	Etudiant
Description sommaire	Permet à l'étudiant d'entrer ses informations personnelles, les renseignements sur son sponsor, ses parents et de les soumettre au système afin de valider sa réinscription pour une nouvelle année.
Précondition	L'étudiant se connecte
Post-condition	L'inscription est effectuer et l'étudiant est déconnecté.
Scénario nominal	<ol style="list-style-type: none"> 1. Il clique sur l'onglet de réinscription 2. Le système affiche le formulaire des informations liées à l'étudiant

	<ol style="list-style-type: none"> 3. l'étudiant remplit les champs et clique sur suivant 4. le système vérifie si les champs sont bien remplis 5. Le système affiche le formulaire des informations liées aux parents 6. l'étudiant remplit les champs et clique sur suivant 7. le système vérifie si les champs sont bien remplis 8. Le système affiche le formulaire des informations liées au sponsor 9. l'étudiant remplit les champs et clique sur suivant 10. le système vérifie enregistre les informations et crée l'inscription de l'étudiant
Alternative	L'étudiant est déjà inscrit

Tableau 4: description textuelle de l'inscription administrative

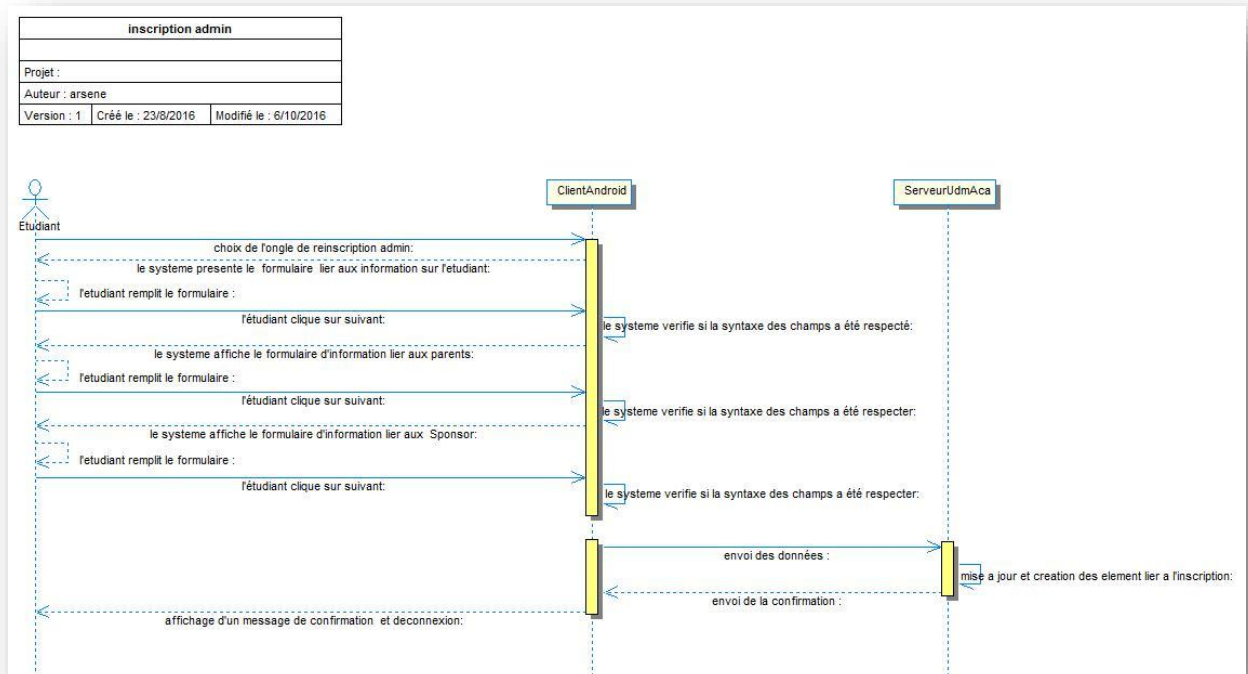


Tableau 5: Diagramme de séquence de la réinscription

b. Inscription académique

Titre	Consulté ses notes
Acteur	Etudiant
Description sommaire	Permet à l'étudiant de cocher les unités d'enseignement qu'il pourra effectuer durant l'année
Précondition	<ol style="list-style-type: none"> 1. L'étudiant doit s'authentifier 2. L'étudiant doit avoir déjà fait son inscription académique
Post-condition	L'étudiant reçoit un message de confirmation
Scénario nominal	<ol style="list-style-type: none"> 1. L'étudiant clique sur l'onglet inscription académique 2. Le système affiche les différentes unités d'enseignement par semestre 3. L'étudiant coche les unités auxquelles il souhaite postuler et soumet le formulaire 4. Le système vérifie si le total de crédit est réglementaire et valide l'inscription
Alternative	Le nombre de crédit est supérieur au nombre max

Tableau 6: description textuelle de l'inscription académique

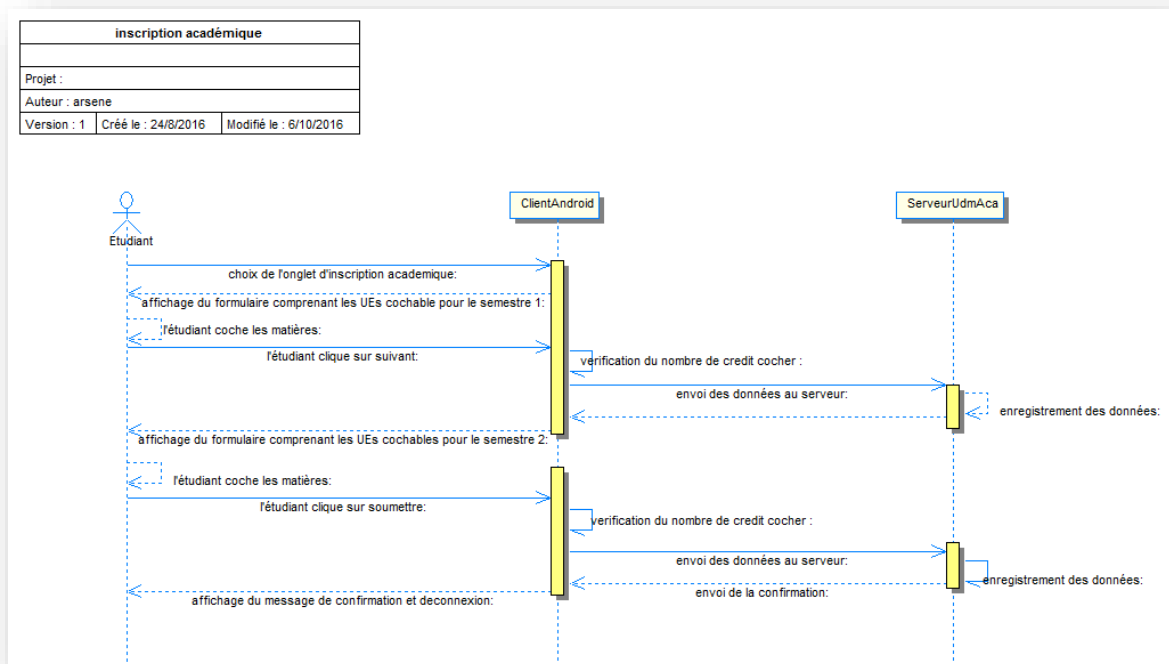


Tableau 7: Diagramme de séquence d'inscription académique

c. Se connecter

Titre	Se connecter
Acteur	Etudiant
Description sommaire	Permet à l'étudiant de s'authentifier a fin d'accédé aux différentes fonctionnalités de l'application
Précondition	Lancer l'application
Post-condition	La page d'accueil et les menus sont présentés
Scénario nominal	<ol style="list-style-type: none"> 1. L'étudiant entre les paramètres de connexion 2. Le système vérifie les paramètres de connexion créer une variable de session et présent la page d'accueil
Alternative	Les paramètres de connexion ne sont pas corrects : un message d'erreur est affiché

Tableau 8: description textuelle de la connexion à l'application

d. Modifier son mot de passe

Titre	Modifier mot de passe
Acteur	Etudiant
Description sommaire	Permet à l'étudiant de modifier son mot de passe
Précondition	L'étudiant doit authentifier
Post-condition	Le mot de passe est modifier et l'étudiant est déconnecté.
Scénario nominal	<ol style="list-style-type: none"> 1. Il clique sur l'onglet de modification du mot de passe 2. Le système affiche un formulaire de trois champs 3. L'utilisateur entre l'ancien mot de passe, le nouveau mot de passe le confirme et soumet le formulaire 4. Le système vérifie que l'ancien mot de passe est correct et que les deux autres sont identiques 5. Le système enregistre le nouveau de passe et déconnecte l'étudiant
Alternative	Si l'ancien mot de passe est incorrect ou si les deux nouveaux de passe ne sont pas identique : un message d'erreur est envoyé

Tableau 9: description textuelle du changement de mot de passe

e. Consulter ses notes de cours

Titre	Consulter ses notes
Acteur	Etudiant
Description sommaire	Permet à l'étudiant de consulté ses notes de cours déjà disponibles
Précondition	<ol style="list-style-type: none">1. L'étudiant doit s'authentifier2. L'étudiant doit être inscrit3. Doit être en règle financièrement
Post-condition	Les notes de l'étudiant sont affichées pour visualisation
Scénario nominal	<ol style="list-style-type: none">1. Il clique sur l'un des onglets de visualisation des notes2. Le système affiche les notes de cours disponible sous forme de liste
Alternative	Les notes ne sont pas disponible un message est donc affiché pour signifier cela

Tableau 10: description textuelle de la consultations des notes de cours

f. Consulter ses notes d'UE

Titre	Consulté ses notes
Acteur	Etudiant
Description sommaire	Permet à l'étudiant de consulté ses notes d'UE déjà disponibles
Précondition	<ol style="list-style-type: none">1. L'étudiant doit authentifier2. L'étudiant doit être inscrit3. Doit être en règle financièrement
Post-condition	Les notes de l'étudiant sont affichées pour visualisation
Scénario nominal	<ol style="list-style-type: none">1. Il clique sur l'un des onglets de visualisation des notes2. Le système affiche les notes d'UE disponible sous forme de liste
Alternative	Les notes ne sont pas disponible un message est donc affiché pour signifier cela

Tableau 11: description textuelle de la consultations des notes d'UE

III.4 CONCEPTION DE L'APPLICATION

III.4.1 Architecture

Le diagramme de package de notre application se présente comme suit

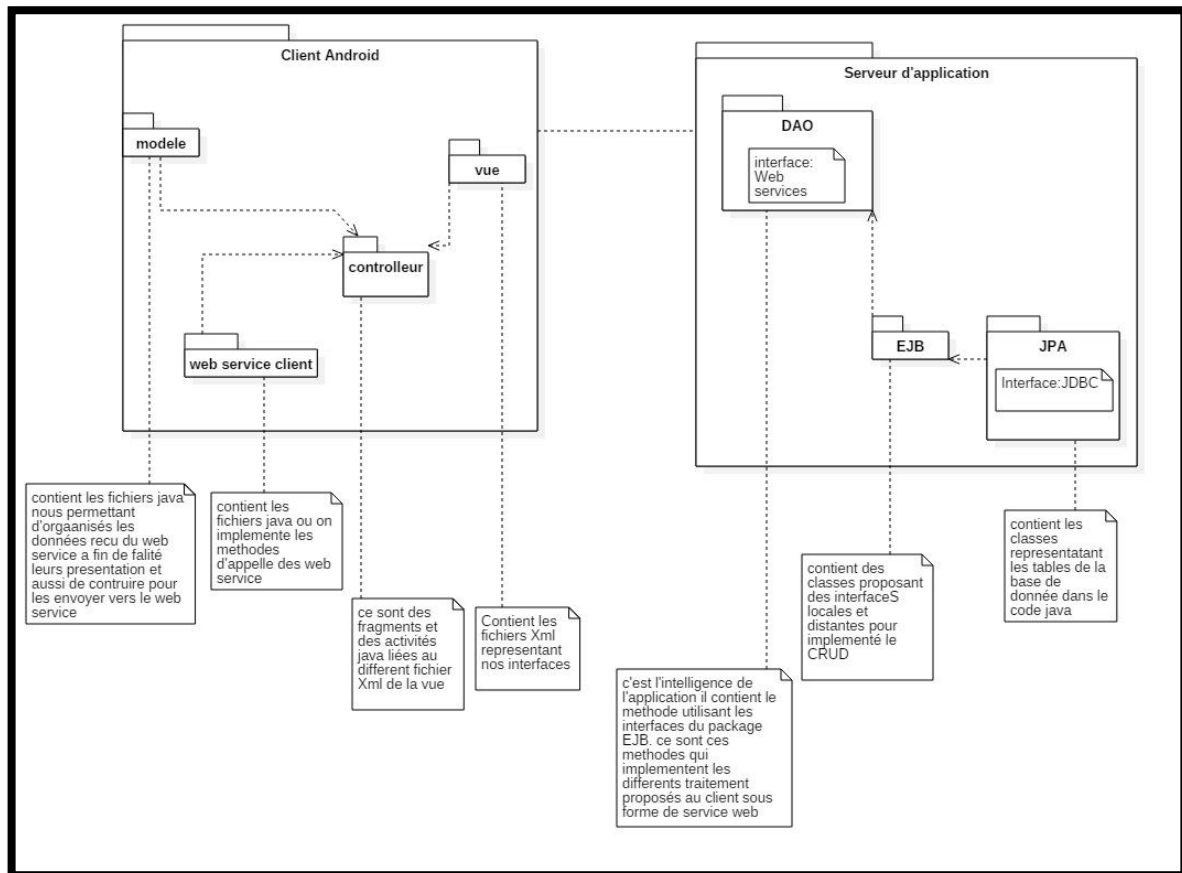


Figure 8: diagramme de package

Nos différents composants sont représentés comme suit

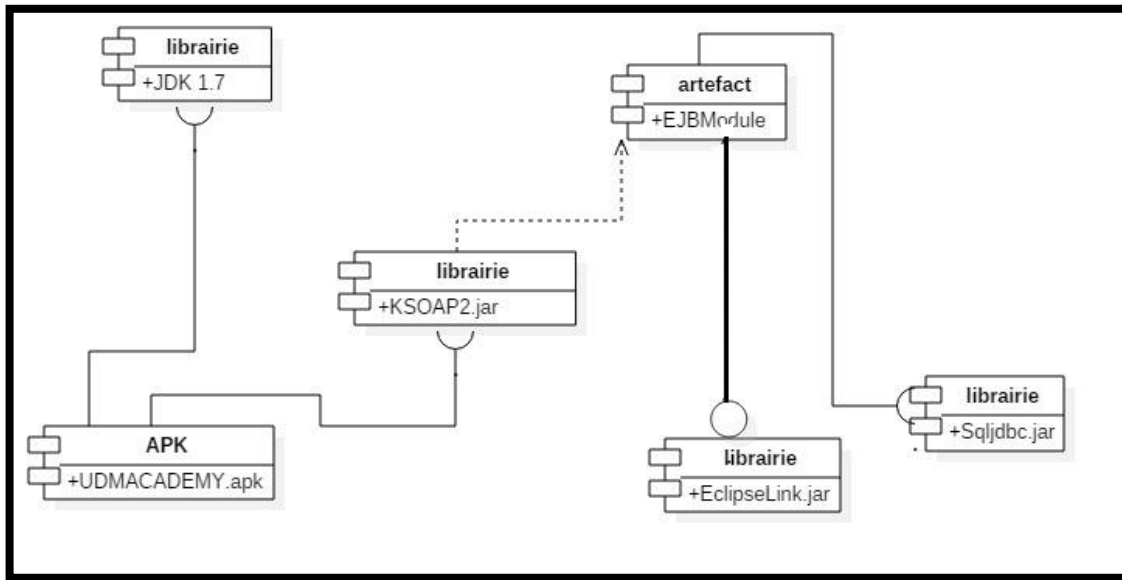


Figure 9:diagramme de composant

III.4.2 Conception détaillée des cas d'utilisation

a. Diagramme de classe par module

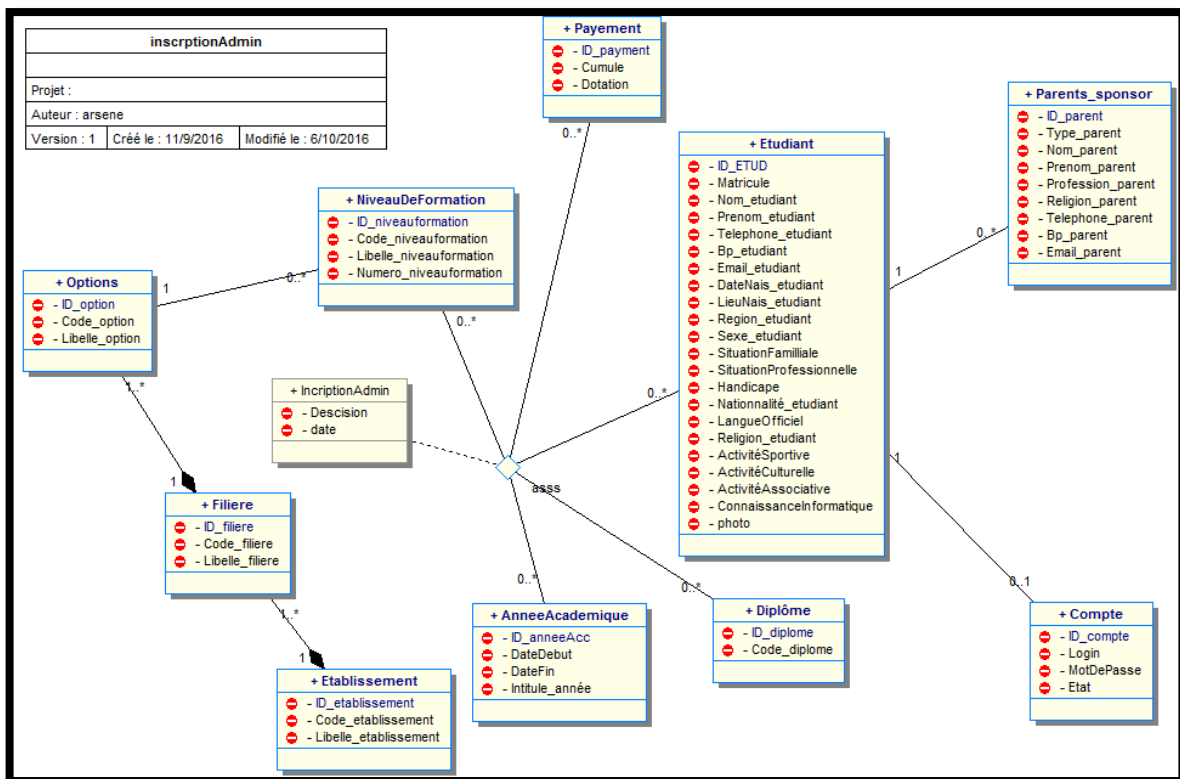


Figure 14 : diagramme de classe inscription administrative

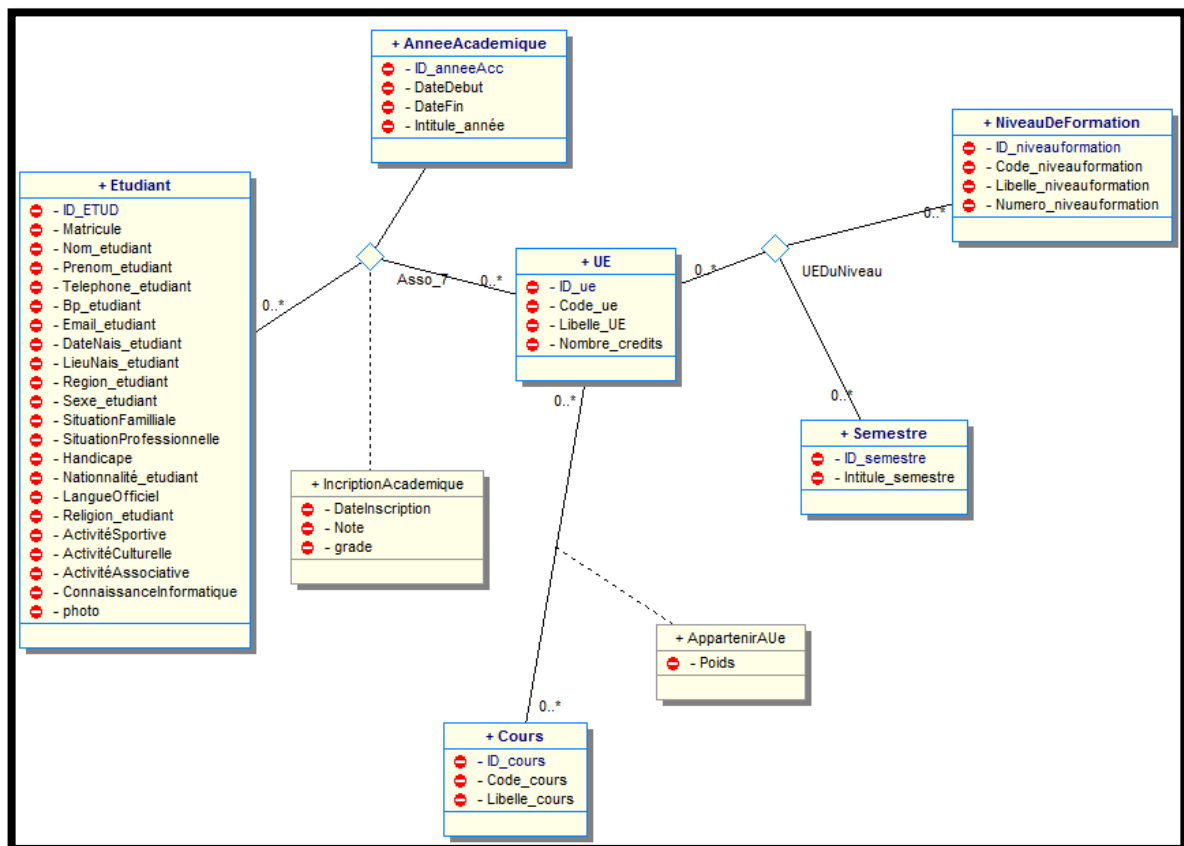


Figure 10:diagramme de classe inscription académique

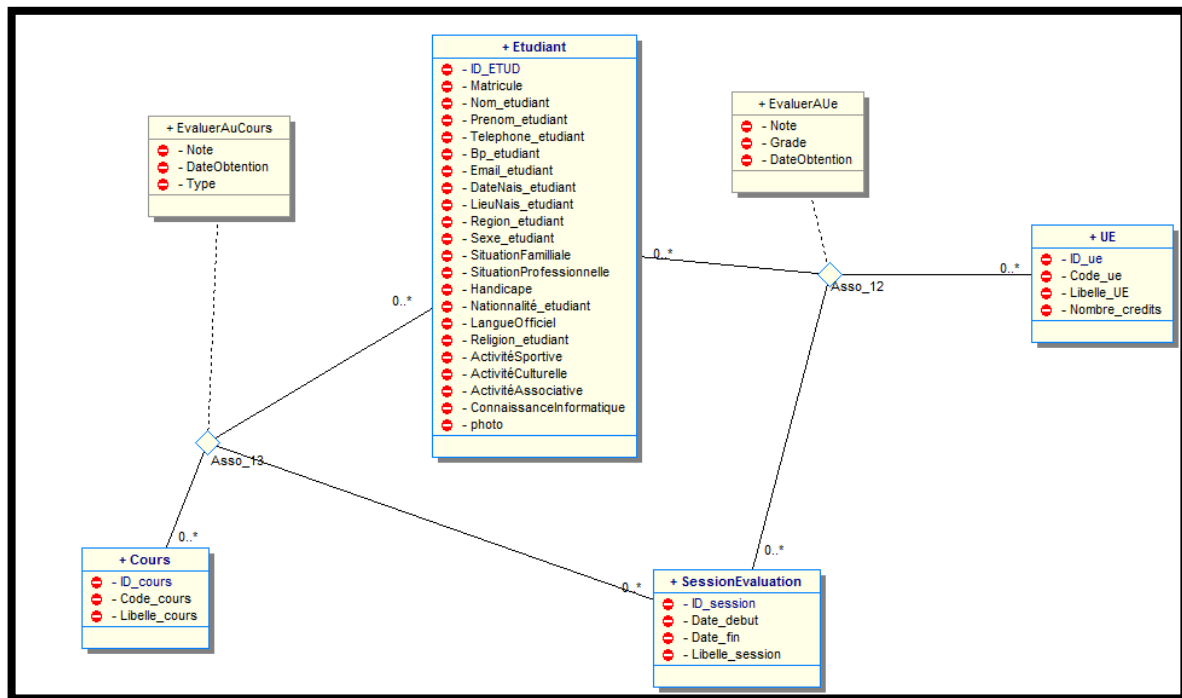


Figure 11: diagramme de classe consultation des notes

b. Description des différentes méthodes utilisées

Les méthodes **set** et **get** de chaque classe sont créées automatiquement à l’aide de l’environnement de développement que nous utilisons (NetBeans).

Les méthodes suivantes sont quelques grandes méthodes appelées par notre client Android, implémentées dans la couche métier et sont présentées au niveau de la couche application sous forme de services web.

Nom	Entrées	Sorties	Description
getAnneeEnCour	--	Une année académique	Permet d’avoir l’année académique courante
getLastInscription	Identifiant de l’étudiant	Une inscription	Permet d’avoir la dernière inscription de l’étudiant

CheckInscriptionAdmin	Identifiant de l'étudiant et identifiant de l'année en cour	Booleen	Permet de savoir si l'étudiant est déjà inscrit
DoInscription	Identifiant de l'étudiant	Booleen	Permet d'inscrire un étudiant
InscriptionAca	Identifiant de l'étudiant et l'identifiant de l'UE	Booleen	Permet d'inscrire un étudiant a une UE
GetNoteCours	Identifiant de l'étudiant	Liste de notes de cours	Permet d'avoir les différentes notes de cours de l'étudiant pour une année
GetNoteUE	Identifiant de l'étudiant	Liste de notes d'UE	Permet d'avoir les différentes notes de cours de l'étudiant pour une année
getInscription	Identifiant de l'étudiant	L'inscription d'un etudiant	Permet d'avoir l'inscription d'un étudiant pour une année académique

Tableau 12:liste des fonctions cote serveur

Chapitre IV : MISE EN ŒUVRE ET RESULTATS

IV.1 Environnement de travail

IV.1.2 Environnement de développement

- Client : ANDROID STUDIO 2.1
- Test : Smartphone avec un système ANDROID Version 4.4.2 marque htc
- Environnement de développement du module serveur : NETBEANS
- Serveur de base de données : SQLSERVEUR 2008

IV.2 Implémentation

L'implémentation proprement dite de notre solution s'est faite en plusieurs étapes qui sont :

IV.2.1 Réalisation des interfaces utilisateurs

A cette étape, il est question pour nous de monter des interfaces attractives, facilitant la manipulation de l'utilisateur et s'adaptant à tous les écrans.

Les figures ci-après présentent les résultats obtenus après avoir monté les interfaces utilisateurs. Nous avons capturé quelque une à savoir l'interface de lancement, de connexion, les menus, et changement de mot de passe

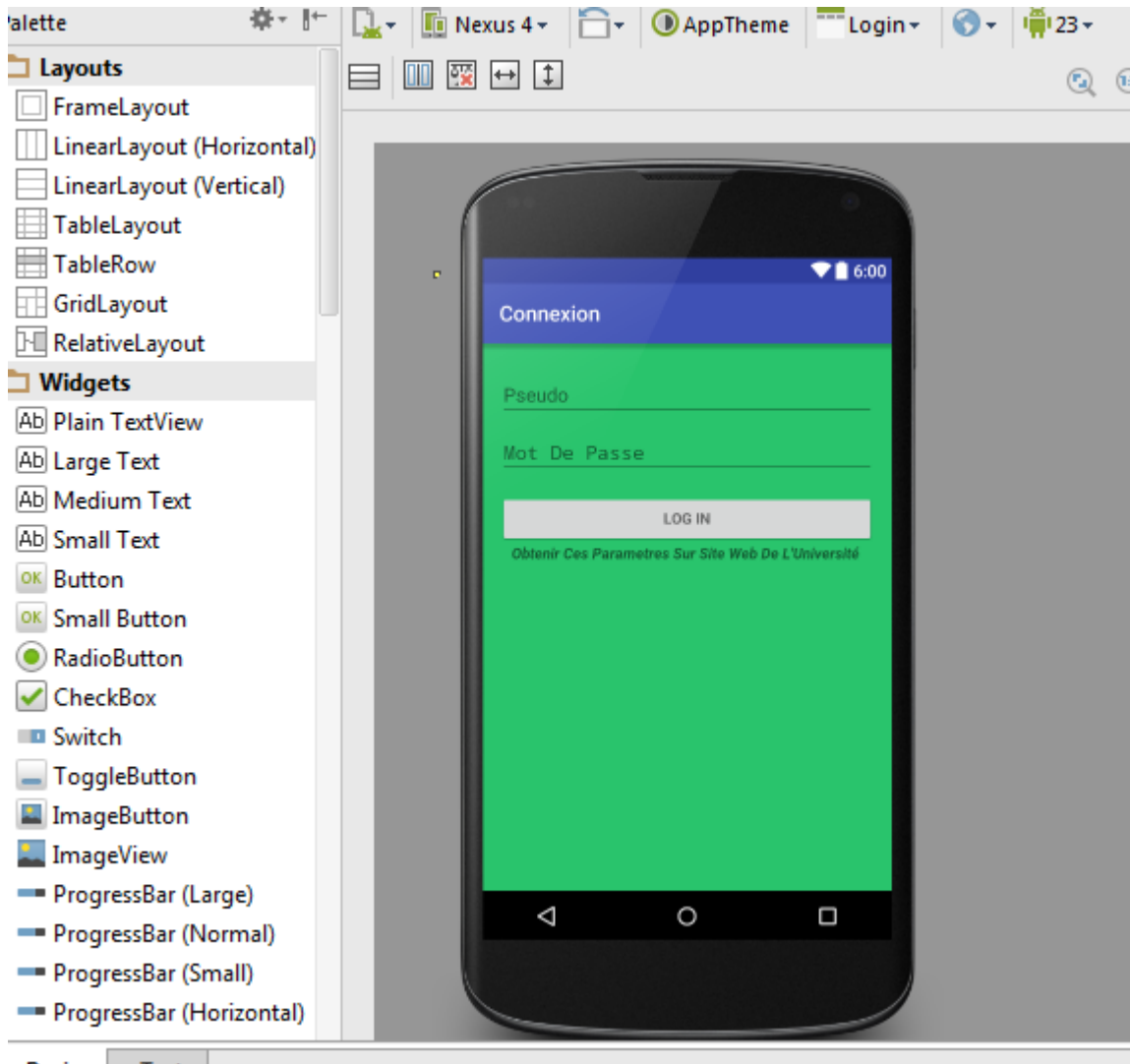


Figure 12: interface de connexion

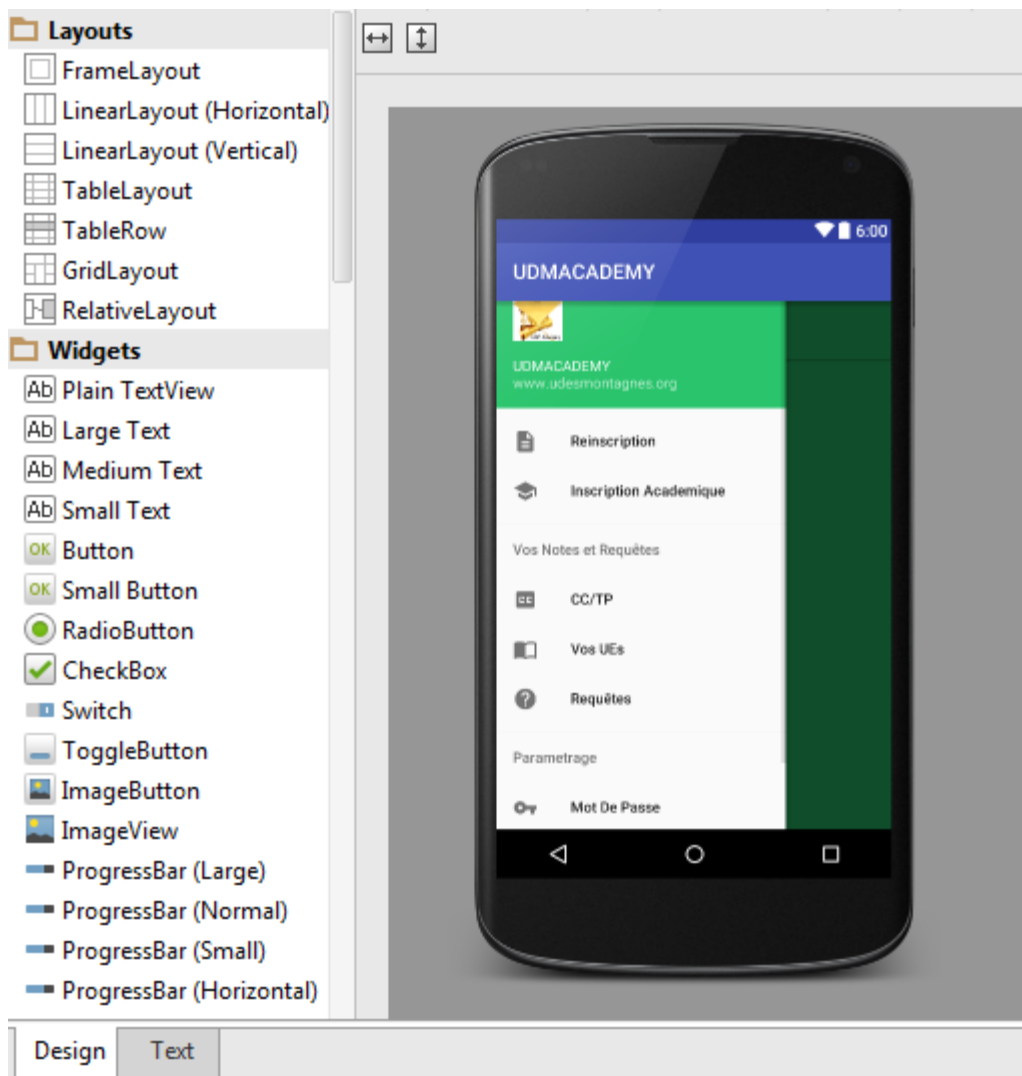


Figure 13: interface de menu

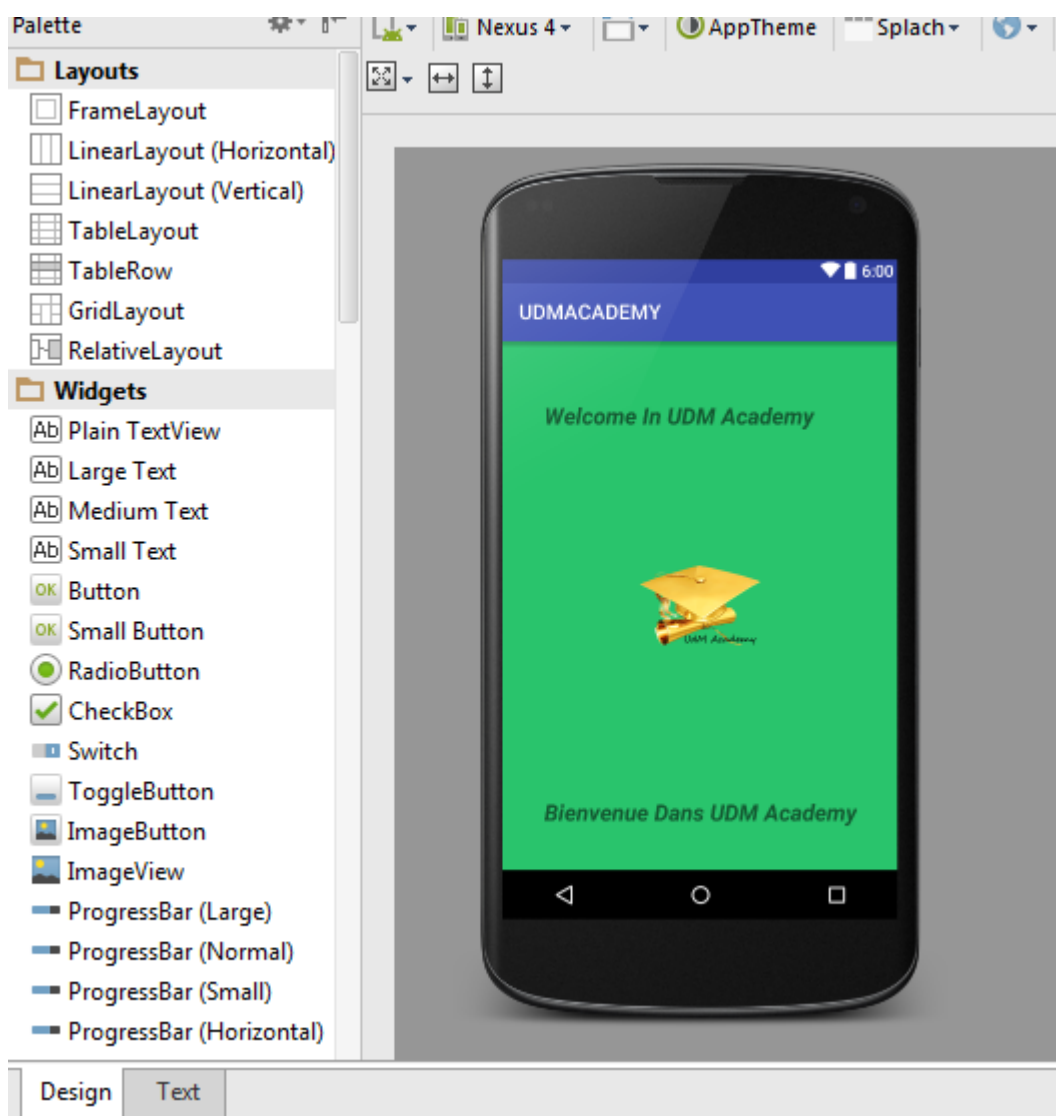


Figure 14: interface de lancement

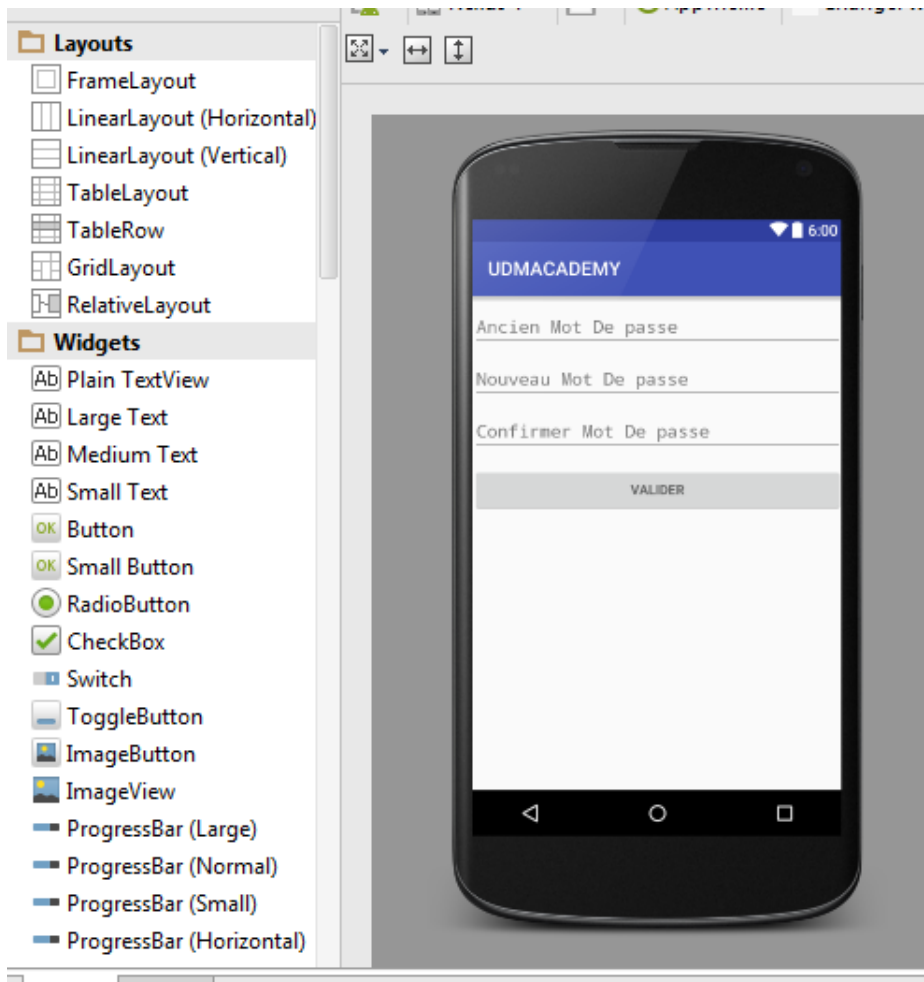


Figure 15: interface de changement de mots de passe

IV.2.2 Implémentation des services web

Ici il est question d'implémenter toutes les fonctionnalités de notre système sous forme de service web afin que ceux-ci puissent être appelés sur le client.

IV.2.3 Appel des web services à partir du client

Il est question ici d'ajouter du métier à nos interfaces ; de faire communiquer notre application avec la base de données par le biais des services web écrit plus haut.

IV.2.4 Test tu client

Ceci consiste à tester notre application avec les données réelles correspondant à notre modélisation.

IV.3 Intégration

IV.3.1 Adaptation des services web

Il est question ici d'adapter notre serveur d'application à la base de données existante dans le projet UDM-ACADEMY.

IV.3.2 Adaptation du client

Il est question ici de faire des petits réajustements de notre client Android par rapport aux changements apportés sur le serveur d'application et enfin de le lier au serveur du système définitif : UDM-ACADEMY

IV.3.3 Test final

Enfin nous avons procédé au test sur les données réelles du système.

IV.3.4 Résultats obtenu

- **Inscription administrative**

L'image qui suit nous présente les différentes interfaces montées pour l'inscription administrative. Elle se fait en trois étapes comme représenté sur ces interfaces.

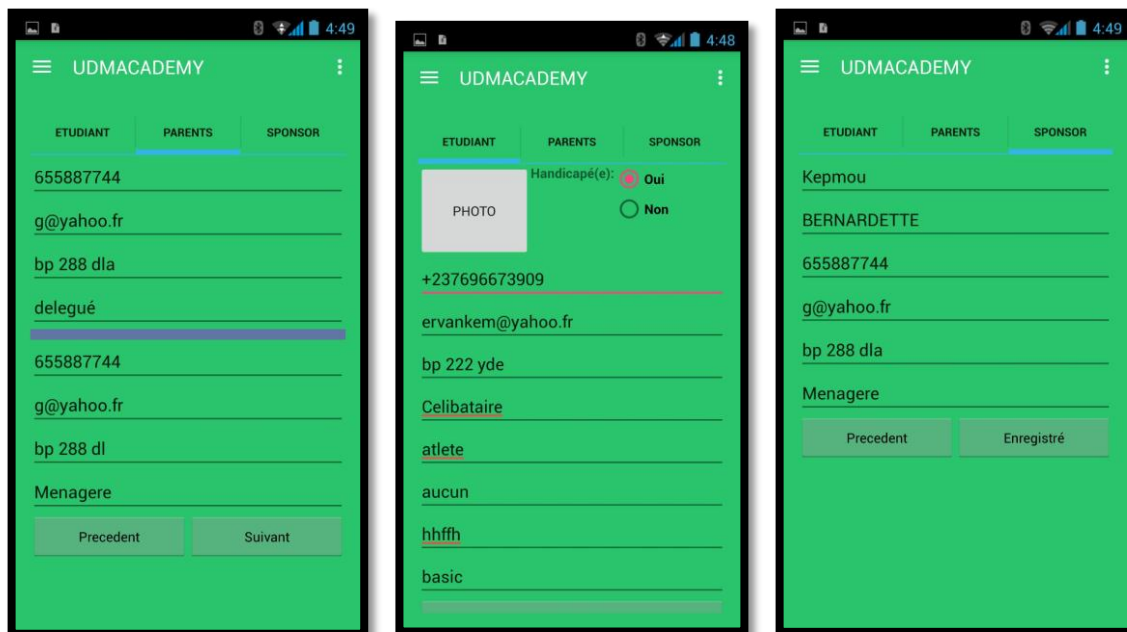


Figure 16: résultat obtenu pour l'inscription

- **Inscription académique**

L'image ci-après présente les interfaces d'inscription académique. Elle se fait en deux étapes représentant les deux semestres d'une année académique.

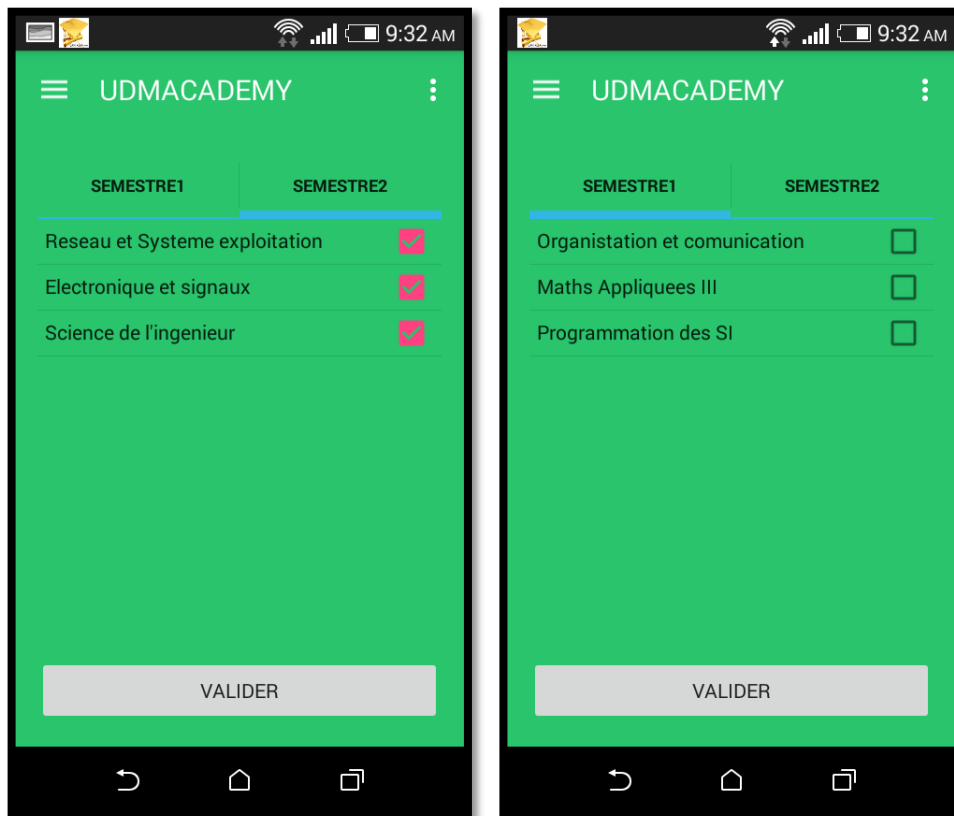


Figure 17: resultat obtenue pour l'inscription académique

- **Consultations des notes**

L'image ci-après présente l'interface de consultation de notes. Nous avons les notes de cours disposées sous forme de listes d'éléments.

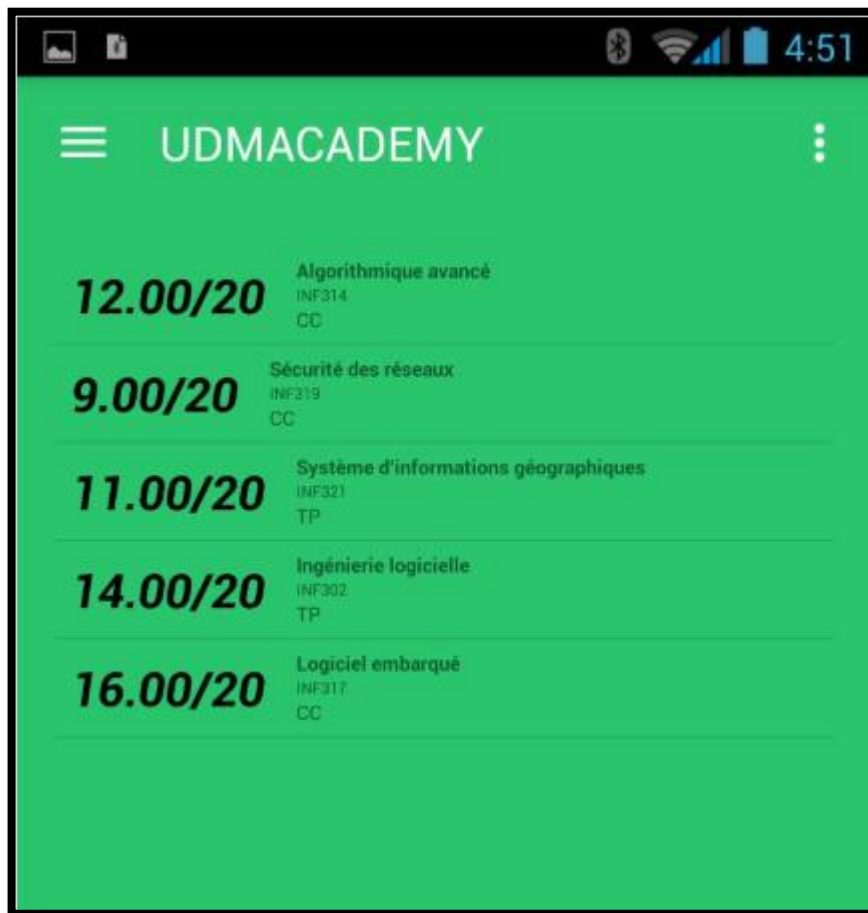


Figure 18:consultation de note

- **La connexion à l'application**

Ici nous avons l'interface de connexion à l'application contenant un champ login et un champ mot de passe.

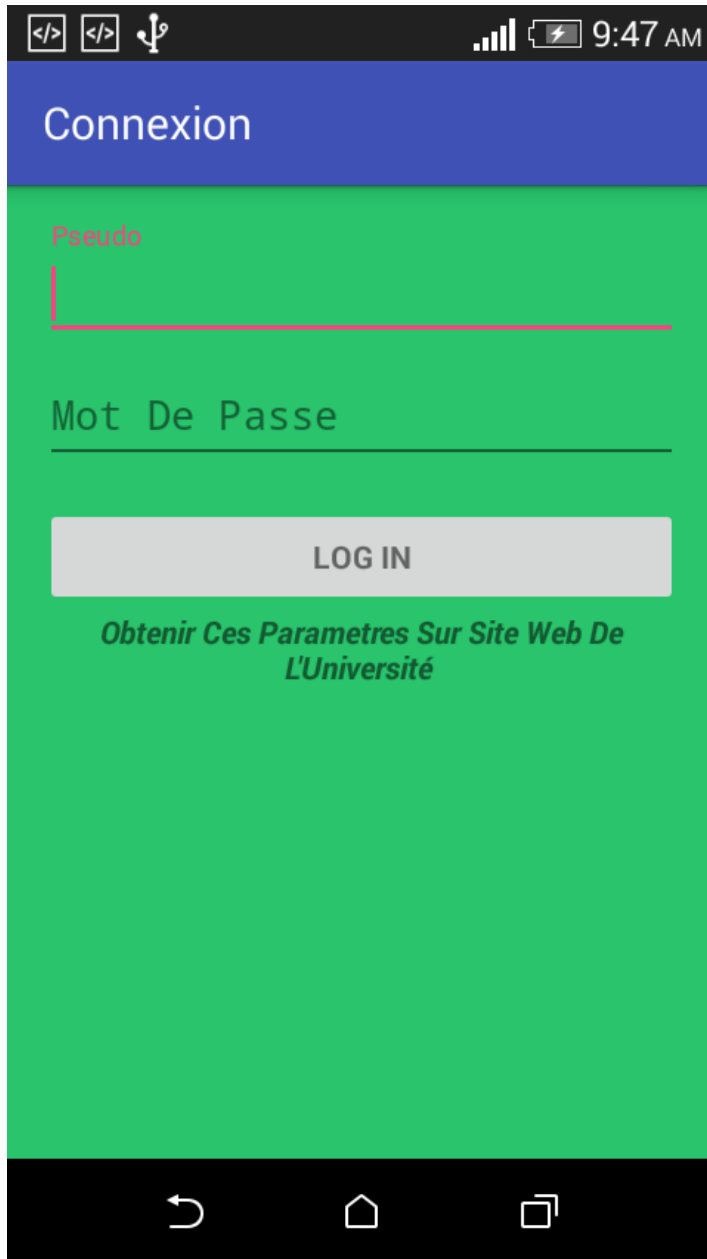


Figure 19:interface de connexion

CONCLUSION ET PERSPECTIVES

En conclusion, l'Université des Montagnes fonctionne avec un système dans lequel l'accès aux notes par les étudiants est compliqué et où l'inscription ne l'a pas moins malgré le fait qu'elle se face en ligne, car l'accès au site demande beaucoup de ressources et les étudiants étant toujours connectés sur leurs téléphones ne peuvent pas exploiter pleinement ce site car le rendu visuel est médiocre sur téléphone.

L'étude menée dans ce document montre les étapes et les outils qui nous ont permis d'implémenter le module ANDROID de gestion académique du projet UDM-ACADEMY pour la résolution de ces problèmes, et de maîtriser un ensemble d'outils et technologies à savoir Netbeans, Android Studio, MsSql Serveur, Win'Design, StarUML et de langages SQL, JQuery, Java, XML. A l'aide de ces outils, nous avons pour objectifs de recueillir les informations, d'étudier des solutions et choisir la plus appropriée, d'implémenter et intégrer (au projet UDM-ACADEMY) un client ANDROID de gestion académique basé sur les services web. Nous obtenons donc une application Android ayant les fonctionnalités suivantes :

- Faire une inscription administrative ;
- Faire une inscription académique ;
- Consulter ses notes de contrôle continu et de travaux pratiques ;
- Consulter ses notes aux unités d'enseignements ;
- Modifier le mot de passe de son compte.

Notre application facilitera la vie à l'étudiant en lui permettant de s'inscrire facilement à l'aide de son téléphone où qu'il soit à condition d'avoir une connexion internet et aussi de consulter ses notes. Comme perspectives, nous pourrons ajouter la gestion des requêtes et la consultation des actualités académique.

REFERENCE BIBLIOGRAPHIQUE

- [2] **Object-oriented analysis and Design using UML, edition 1.0, 2009**
- [2] **UML par le Pratique PASCAL ROQUE, 5e edition Eryrolles, 2001**
- [1] **Guide de l'étudiant Université des Montagnes 2015/2016,**
- [3] **open class room article : présentation de JEE**
- [4] **Wikipédia**
- [5] **open classe room article : les web service**
- [6] **Wikipédia.org/wiki/compilation**
- [7] **Wikipédia.org/wiki/Application_mobile**

ANNEXES



Université des Montagnes

DIRECTION DE L'ADMINISTRATION ACADEMIQUE
DEPARTMENT OF ACADEMIC ADMINISTRATION

SERVICE DES ADMISSIONS ET DES AFFAIRES ACADEMIQUES
ADMISSIONS AND ACADEMIC AFFAIRS OFFICE

FICHE D'INSCRIPTION ADMINISTRATIVE
ADMINISTRATIVE REGISTRATION FORM

ANNÉE ACADEMIQUE 2015/2016
ACADEMIC YEAR 2015/2016

☐ Première inscription

☐ Réinscription

N° Matricule

(Photo)

Cadre réservé à l'administration <i>Official use only</i>		Visa du service des Admissions et des Affaires Académiques <i>Remarks of the Admissions and Academic affairs office</i>		Cadre réservé à l'administration <i>Official use only</i>	
N° matricule <i>Registration number</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>	<input type="checkbox"/> Concours interne <i>Internal examination</i> <input type="checkbox"/> Concours national <i>National examination</i> <input type="checkbox"/> Sur titre <i>Other examination</i>	Niveau <i>Level</i> <div style="border: 1px solid black; width: 40px; height: 15px; margin-top: 5px;"></div>	<input type="checkbox"/> Nouveau <input type="checkbox"/> Redouble <input type="checkbox"/> ASUP <input type="checkbox"/> A Déterminer	Référence photo numérique <i>Digital photo reference</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>	
Etablissement ¹ Diplôme ¹ Filière ¹ Spécialité/Option ¹ 					
IDENTIFICATION DE L'ETUDIANT					
Nom(s) et Prénom(s) <i>Surname and first names</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>					
N° Téléphone <i>Telephone Number</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		B.P. <i>P.O.Box</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		E-mail <i>E-mail address</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>	
Date de naissance <i>Date of Birth</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		Lieu <i>Place</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>		Région d'origine ¹ <i>Region</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>	
Sexe <i>Sex</i> <input type="checkbox"/> Masculin <i>Male</i> <input type="checkbox"/> Féminin <i>Female</i>		Situation familiale <i>Marital status</i> <input type="checkbox"/> Marié(e) <i>Married</i> <input type="checkbox"/> Célibataire <i>Single</i>		Situation d'emploi <i>Employment status</i> <input type="checkbox"/> Employé(e) <i>Employee</i> <input type="checkbox"/> Sans Emploi <i>Unemployed</i>	
Nationalité <i>Nationality</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		1 ^{ère} langue officielle <i>1st official language</i> <input type="checkbox"/> Français <i>French</i> <input type="checkbox"/> Anglais <i>English</i>		Religion <i>Religion</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>	
ADRESSES DES PARENTS					
Nom et prénom du père <i>Father's name</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>			Profession <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>		
Téléphone <i>Telephone number</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		B.P. <i>P.O.Box</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		Ville <i>Town</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>	
Région ¹ <i>Region</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		Nationalité <i>Nationality</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		E-mail <i>Mailing address</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>	
Nom et prénom de la mère <i>Mother's name</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>			Profession <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>		
Téléphone <i>Telephone number</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		B.P. <i>P.O.Box</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		E-mail <i>Mailing address</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>	
ADRESSES DU SPONSOR DES ETUDES					
Nom du tuteur/Sponsor <i>Guardian/Sponsor's name</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>			Profession <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>		
Téléphone <i>Telephone number</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		B.P. <i>P.O.Box</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		Ville <i>Town</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>	
Nationalité <i>Nationality</i> <div style="border: 1px solid black; width: 100px; height: 15px; margin-top: 5px;"></div>		E-mail <i>Mailing address</i> <div style="border: 1px solid black; width: 100%; height: 15px; margin-top: 5px;"></div>			
AUTRES INFORMATIONS					
Activités sportives pratiquées/ <i>Involvement in sports/games</i> (Palmarès et performances/ <i>honours and past performances</i>) Activités culturelles pratiquées/ <i>Involvement in cultural activities</i> (Indiquer les références/ <i>State references</i>) Connaissances en informatique/ <i>Knowledge on computer sciences</i> Autres activités de type associatif/ <i>Other cultural activities</i>					

Figure 20: page une de l'inscription académique manuel

CURRICULUM STUDIORUM

ENSEIGNEMENT SECONDAIRE ET SUPERIEUR / SECONDARY AND HIGHER SCHOOL

Année <i>Year</i>	Etablissement <i>School</i>	Classes <i>Class</i>	Résultats ² <i>Results</i>	Diplôme obtenu <i>Certificate obtained</i>
2014 / 2015				
2013 / 2014				
2012 / 2013				
2011 / 2012				
2010 / 2011				
2009 / 2010				
2008 / 2009				
2007 / 2008				
2006 / 2007				

PERFORMANCES DE L'ANNEE DERNIERE POUR LES ANCIENS / LAST YEAR CURSUS FOR CURRENT STUDENTS

Filière /Niveau <i>Course of studies /Level</i>	Moyenne Pondérée Semestrielle <i>Semester GPA</i>		Moyenne Pondérée Cumulée à jour <i>Cumulated GPA</i>
	Sem.1	Sem. 2	

INFORMATIONS RELATIVES AU CONCOURS NATIONAL

- 1°) Quel était votre centre d'examen ? _____
- 2°) Quelle université avez-vous choisie ? _____
- 3°) Quel était votre numéro de table ? _____
- 4°) Quelle filière avez-vous choisie ? _____

*Je soussigné, certifie sur l'honneur que les informations que j'ai fournies dans cette fiche sont authentiques.
I certify, on my honour, that all the information I have given herein is authentic.*

Fait à Bangangté, le _____
Done at Bangangté, on the _____

Signature de l'étudiant
Student's signature

--

Partie réservée à l'administration / *Official use only*
INFORMATIONS SUR LE BACCALAUREAT, LE GCE AL NATIONAL OU DIPLOME EQUIVALENT RECONNU PAR LE MINESUP

Série _____ Code _____ Mention _____ Année _____

☐ Baccalauréat / GCE AL Camerounais
 Centre/S-Centre _____ Matricule _____ N° Jury _____

☐ Diplôme obtenu à l'étranger
 Equivalence Disponible ☐ Oui ☐ Non Référence _____ Date _____

Date de dépôt de la demande (si équivalent non disponible) _____

² Résultats/Results = Admis(e)/Pass OU/et Echoué(e)/Fail.