

SOMMAIRE

Introduction	3
A. ANDROID	4
I. Qu'est-ce qu'Android.....	4
1. Introduction.....	4
2. Les principales applications.....	5
a) La zone de notifications.....	5
b) Le home	6
c) Les applications téléphoniques	6
d) L'Android Market.....	7
e) Le clavier virtuel	7
II. Le marché	8
1. Évolution du marché des smartphones.....	8
2. Un même système d'exploitation pour une multitude d'appareils	8
3. Possibilités offertes par le SDK.....	9
4. Rémunération.....	10
a) Rémunération fixe	10
b) Rémunération par la publicité.....	11
c) Rémunération par abonnement.....	11
B. ANALYSE ET CONCEPTION	12
I. Présentation de l'organisme d'accueil	12
II. Etude des documents	12
III. Sondage	13
IV. Critiques et suggestions	14
V. Dictionnaires des données :	14
VI. MCD.....	15
VII. MLD	16
C. REALISATION	17
I. Présentation de l'environnement de développement.....	17
1. Langage.....	17
a) Java	17
b) Interface XML	17

2.	Mise en place de l'espace de travail.....	17
3.	Machine virtuelle.....	18
4.	Les outils mis à disposition par le SDK.....	18
a)	ADB (Android Debug Bridge).....	18
b)	L'émulateur	18
5.	Les différents types d'applications sous Android.....	19
a)	Activity.....	19
b)	Service	19
c)	BroadcastReceiver	19
d)	ContentProvider	19
6.	Quelques caractéristiques du SDK d'Android.....	20
a)	Création du projet simplifiée.....	20
b)	Architecture de base d'un projet	21
c)	Le fichier Android Manifest.xml	22
d)	Les ressources.....	22
e)	Les bases de données SQLite.....	24
f)	Les Listview	26
II.	Publication sur l'Android Market	26
III.	Application « UABBT App »	27
1.	Structure générale de l'application	27
2.	Page de chargement et bouton pour accéder a l'interface d'accueil	27
3.	Menu principal	28
4.	Les différentes interfaces de l'application	29
	Conclusion.....	36
	Liste des abréviations	37
	Bibliographie.....	38

Introduction

Dans le cadre de la préparation au diplôme de Licence L3, au sein de l'université Abou Bakr Belkaid Tlemcen département d'informatique, nous avons choisi comme thème de mémoire de fin d'étude de développer une application mobile sous Android, avec l'avancé et l'émergence des technologies mobiles, les développements embarqués sont de plus en plus demandés sur le marché, avoir un smartphone ou une tablette est devenu incontournable pour les jeunes de nos jours.

Dans cette optique, nous avons réfléchi de faire profiter notre université et nos étudiants de ces nouvelles opportunités.

Ainsi, nous avons envisagé de réaliser une plateforme sous Android pour remédier à toutes les contraintes pédagogiques que subit le service de scolarité dans sa gestion et faciliter le quotidien de l'étudiant. Cette application qui se voit contenir toutes les informations indispensables, par exemple connaître les résultats des différents examens, les délibérations, les affichages, les soutenances, et informer les étudiants des dernières actualités et événements ayant lieu au sein de leur université en général et leur département en particulier.

Objectif ciblé :

Notre travail consiste à développer une application mobile sous Android, permettant de répondre aux différents besoins d'un étudiant. La principale fonctionnalité est de les informer de toutes les nouvelles qui leurs sont utiles.

Au terme de ce projet, l'application devra être en mesure de délivrer tout cela en un simple clic.

Ce produit doit permettre aux utilisateurs d'avoir la possibilité :

1. D'accéder facilement et rapidement à l'information.
2. D'économiser le temps, l'espace et l'argent.
3. La disponibilité et la portabilité de l'information.

Notre mémoire peut ainsi être subdivisé en trois parties :

La première consistera à la présentation de l'environnement Android avec les outils de développement associés. La seconde partie sera consacrée à une analyse et une étude de l'existant. Enfin, la troisième et dernière partie fera l'objet du développement proprement dit de l'application.

A. ANDROID

I. Qu'est-ce qu'Android

1. Introduction

Android est un système d'exploitation pour smartphone, aussi appelé téléphone intelligent ou Ordiphone. Ce projet est né d'un consortium de 34 entreprises (en 2008), initié par la société Google le 5 novembre 2007 est appelé l'Open Handset Alliance ou OHA son but consistait à trouver une solution fiable pour concurrencer Apple avec l'iPhoneOS, Microsoft avec Windows Mobile, Nokia avec Symbian et Research In Motion avec Blackberry OS, en effet à cette époque la quasi-totalité des smartphones fonctionnait sur ces systèmes d'exploitation.

Cependant la principale différence d'Android face aux autres solutions est qu'il soit open source. Cette particularité le rend donc gratuit et personnalisable par les constructeurs et les opérateurs de téléphonie mobile. Malgré des personnalisations accrues en terme d'interface utilisateur et de fonctionnalité le cœur du système reste commun ce qui permet une interopérabilité des applications. De ce fait, si l'on ne prend pas en compte les difficultés liées aux différences matérielles des périphériques, un téléphone Samsung et un autre Sony-Ericsson fonctionnant tous les deux sur Android seront en mesure d'exécuter les mêmes applications sans aucune compilation supplémentaire.

Schéma : Voici un schéma très bref d'un système Android et de son rôle



Ce schéma ne permet pas d'expliquer en détail ce qu'est Android, mais d'avoir une vue d'ensemble de la solution. Il met l'accent sur la différence entre l'appareil, son système d'exploitation et ses applications.

En effet, bien qu'il soit courant de différencier la partie matériel et logiciel d'un ordinateur (par exemple, un ordinateur fonctionnant sous le système d'exploitation Microsoft Windows n'est pas construit par Microsoft), cette idée n'est pas encore démocratisée pour ce qui concerne les appareils mobiles.

2. Les principales applications

Le système Android tout seul ne rend pas l'appareil réellement utilisable. Il fonctionne donc étroitement avec plusieurs applications, provenant de Google ou d'autres éditeurs et permettant une utilisation complète d'un smartphone. Ce type d'architecture donne aussi la possibilité aux développeurs et le choix à l'utilisateur final de remplacer les applications de base de leur appareil.

Parmi ces applications, nous détaillerons quelques exemples.

a) La zone de notifications

La zone de notification n'est pas une application comme les autres puisqu'elle ne permet pas son remplacement. Elle informe l'utilisateur des différentes notifications invoquées par les applications par exemple la réception d'un SMS.



Barre de notification pliée

Lorsque l'utilisateur glisse son doigt sur cette barre afin de la faire coulisser vers le bas, elle affiche les détails des notifications et permet, lors d'une pression sur ces notifications, l'ouverture de l'application concernée. Bien que le SDK donne la possibilité au développeur de cacher cette barre lors de l'affichage de leurs applications, cette fonctionnalité est rarement utilisée puisqu'elle ne facilite pas l'expérience utilisateur. En effet, à part pour quelques cas précis comme la lecture de vidéo ou les jeux, cette barre ne gêne pas de par son apparence plutôt discrète et très pratique puisqu'elle permet de voir d'un simple coup d'œil les différents événements nécessitant l'action de l'utilisateur.

b) Le home

Le home est l'application principale du système. Elle peut s'apparenter au bureau d'un système d'exploitation pour ordinateur. L'équipe de Google ayant développé le système a aussi développé un home sous licence open source. Ce home est représenté par trois bureaux coulissants de gauche à droite sur lesquels il est possible d'y placer des raccourcis vers les applications installées et des widgets, sorte d'interface minimaliste d'application.



Home d'origine avec un widget représentant une horloge et la météo

c) Les applications téléphoniques

Sur Android, les applications pour toutes les utilités téléphoniques sont aussi remplaçables. C'est aussi vrai pour le répertoire des contacts, que pour le dialer ou la gestion des SMS et MMS.

Ainsi, il existe plusieurs applications gratuites pour remplacer l'application d'origine qui gère les SMS et les MMS. Ces applications fournissent plusieurs avantages comme une personnalisation accrue de l'apparence, la gestion des MMS intégrés dans les fils de discussions ou encore la possibilité de dicter le message et de le lire par TextToSpeech.

d) L'Android Market

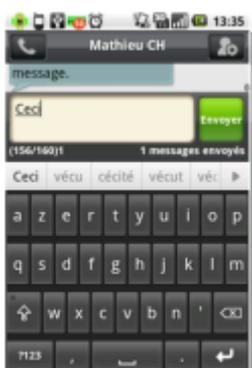
Afin que les développeurs puissent mettre à disposition de manière simple et efficace leurs applications aux utilisateurs du monde entier, Google a développé l'Android Market.

Cette application, similaire à l'AppStore d'Apple pour l'iPhone, est le portail d'application par défaut des téléphones Android. Elle n'est pas Open source, il n'est donc pas légal de vendre un appareil ou publier une version d'Android avec l'Android Market sans l'autorisation de Google.

Le Market permet la mise à disposition d'applications gratuites ou payantes et de les parcourir par catégorie. Chaque application affiche une icône, un nom d'application, une description, des captures d'écrans et les commentaires des utilisateurs.

Pour ce qui est du mode de paiement utilisé par Google pour l'Android Market, il s'agit sans trop de surprises de Google Checkout, un concurrent direct de Paypal pour le paiement en ligne.

e) Le clavier virtuel



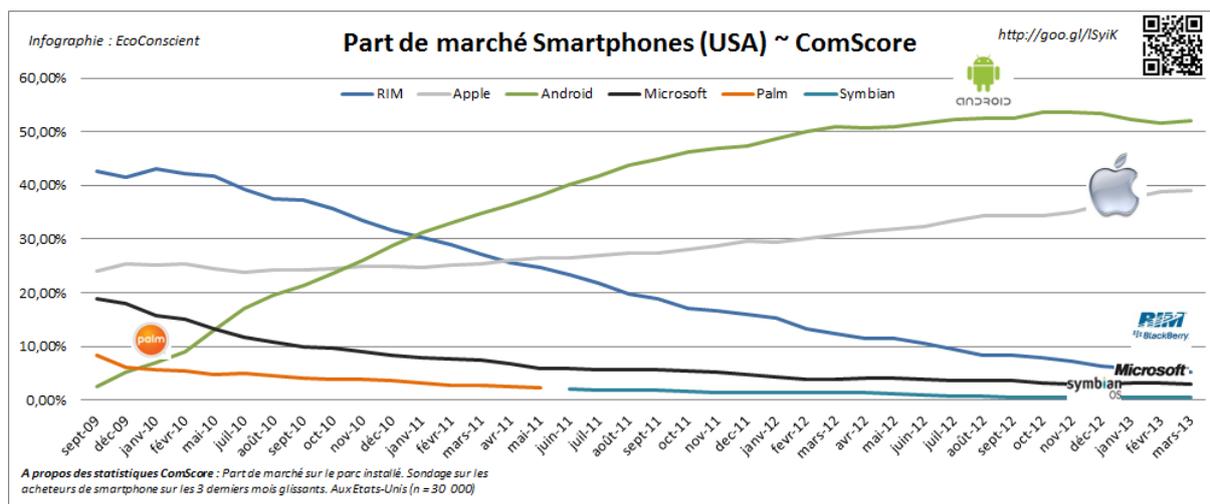
Clavier Android par défaut

À partir de la version 1.5, Android offre à l'utilisateur le choix de son clavier tactile. C'est une très bonne chose puisque cela pousse les éditeurs à trouver des solutions innovantes de saisie. Alors que certains tentent d'améliorer le clavier virtuel en mettant l'accent sur la correction à l'aide du T9 ou sur la réactivité et la prise en charge de la multitouche ; d'autres créent de nouveaux concepts de saisie allant d'une simple modification de la disposition des touches à de nouveaux concepts de saisis. C'est ainsi que la société Swype Inc. a créé une méthode basée sur le glissement du doigt sur chaque lettre composant un mot. Tout comme le reste des applications, l'installation d'un clavier virtuel se fera via un market et il sera possible de choisir quel clavier utiliser.

II. Le marché

1. Évolution du marché des smartphones

Bien qu'Android soit utilisé pour une multitude de types d'appareil, du smartphone au netbook, nous allons nous concentrer sur le marché des smartphones puisque c'est sa principale cible.



D'autre part, plutôt que de différencier les smartphones par leur constructeur, nous allons le faire en fonction de leur système d'exploitation. En effet, une répartition par constructeur n'est pas intéressante dans notre cas puisqu'Android est un système d'exploitation destiné à une multitude de constructeurs.

2. Un même système d'exploitation pour une multitude d'appareils

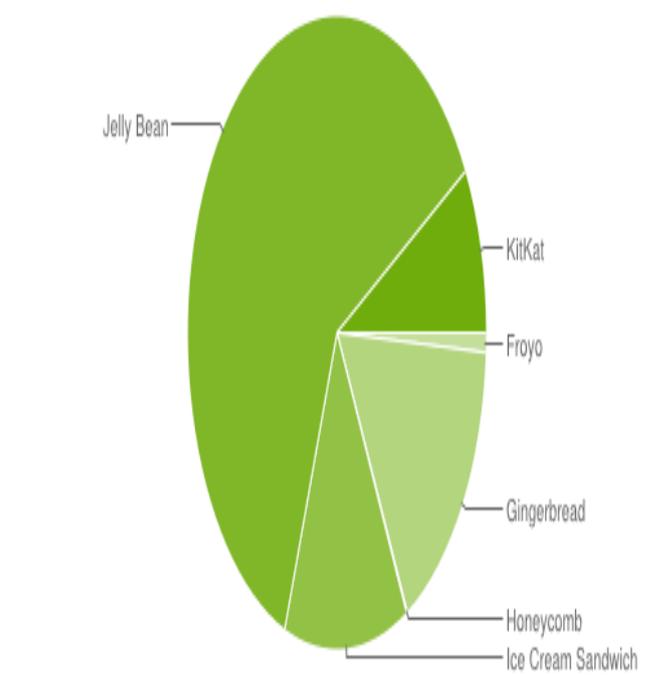
Android étant gratuit et open source, il offre une personnalisation accrue, tout en profitant de la fiabilité d'un système commun et d'une interopérabilité des applications.

De part ces particularités, les applications développées pour Android fonctionnent sur tous les appareils embarquant Android. Cependant, pour un bon fonctionnement, il est nécessaire de respecter certaines règles de programmation comme, par exemple, prendre en compte la résolution des écrans qui peut différer d'un appareil à l'autre. Il sera donc judicieux d'utiliser des unités de mesure adaptatives telles que des pourcentages plutôt que des pixels.

Une autre différence d'un appareil à l'autre peut être la version d'Android, bien que les évolutions des versions d'Android apportent essentiellement des améliorations du SDK, il est judicieux de les développer dans la mesure du possible pour la plus petite version d'Android.

Répartition actuelle des versions d'Android :

Version	Codename	API	Distribution
2.2	Froyo	8	1.0%
2.3.3 - 2.3.7	Gingerbread	10	16.2%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	IceCream Sandwich	15	13.4%
4.1.x	Jelly Bean	16	33.5%
4.2.x		17	18.8%
4.3		18	8.5%
4.4		19	8.5%



<http://developer.android.com/resources/dashboard/platform-versions.html>

Nous pourrions penser que tous les appareils vendus sous Android sont mis à jour avec les nouvelles versions du système, mais ce n'est pas le cas, les constructeurs doivent adapter leurs personnalisations car leurs drivers ne proposent pas la mise à niveau de toutes leurs gammes.

Grâce à la mise sur le marché de nouveaux appareils et à la mise à jour de certains existants, cette répartition est amenée à évoluer. Il sera donc judicieux de consulter ce graphique avant de commencer tout projet de développement.

3. Possibilités offertes par le SDK

Les possibilités offertes par le SDK regroupent, entre autres, l'accès à toutes les fonctionnalités de l'appareil comme l'accès à internet, à la partie téléphonie (appel, gestion des contacts et des SMS), à l'appareil photo, au GPS, à l'accéléromètre ou encore à la boussole numériques.

Lors des chapitres traitant du développement d'applications nous reviendrons sur les différentes sécurités mises en œuvre pour contrôler l'accès aux fonctionnalités.

Les opportunités de marché du développement sur Android, sont extrêmement nombreuses et de nouveaux concepts sont découverts au fur et à mesure que les appareils sont enrichis en divers capteurs et que le SDK est mis à jour.

Les applications mobiles sont à leurs débuts et ont toutes les chances de devenir incontournables dans un futur proche.

Qui pourrait trouver inutile les fameux annuaires téléphoniques par exemple ?

N'est-il pas pourtant beaucoup plus pratique d'avoir le contenu de ses centaines de numéros dans votre poche ?

Ajoutons à la simple fonctionnalité de recherche une géo-localisation et la possibilité aux utilisateurs de laisser des commentaires, la prochaine fois que vous serez à la recherche d'un restaurant en vacances vous ne vous contenterez pas d'un fastfood..., beaucoup d'experts s'accordent sur le fait que l'informatique mobile est l'avenir de l'informatique grand public. Certes les supports seront multiples, il sera difficile de se contenter d'un écran de quelques pouces pour une utilisation intensive.

Cependant, la plupart des utilisateurs d'ordinateur et d'internet font des tâches ne nécessitant pas un poste de travail comme nous le connaissons. Une tablette tactile d'une dizaine de pouces de diagonale conviendrait très bien à leurs utilisations si les applications existaient.

4. Rémunération

Afin de rentabiliser les applications, trois possibilités s'offrent à l'éditeur, nous allons donc les énumérer et citer leurs avantages et inconvénients.

a) Rémunération fixe

La rémunération fixe est la plus évidente, elle consiste à fixer un prix pour l'obtention de l'application, cependant plusieurs méthodes s'offrent à l'éditeur pour vendre son application.

La méthode la plus simple consiste simplement à diffuser l'application sur les différents market en y spécifiant son prix, cependant en plus de réduire la cible aux différents pays auxquels le market en question donne accès aux applications payantes, un pourcentage sera prélevé sur les ventes par la société fournissant le market. À titre d'exemple, l'Android

Market fourni par Google conserve 30% du prix payé par l'acheteur, dans notre pays par exemple l'Ostore de Ooredoo conserve 50% du prix.

Une autre solution serait alors de fournir une version de démonstration sur les différents market que l'utilisateur pourrait débloquent à l'aide d'un numéro de série qu'il aurait préalablement acheté sur un site de e-commerce. Cette solution, bien que plus contraignante pour l'utilisateur final à l'avantage d'éliminer la diffusion de l'application.

D'autant que pour éviter l'utilisation de générateur de clé, cette clé peut être générée au moment de l'achat, stocker dans une base de données et lier à un compte client créé pour l'achat au niveau de l'application il suffit alors de demander en plus du numéro de série les identifiants de l'utilisateur.

b) Rémunération par la publicité

Une solution qui à l'avantage d'élargir sensiblement sa cible et de rémunérer l'application par la publicité plutôt que par un coût d'achat. Bien que la plupart des gens aient horreurs de la publicité, nous l'acceptons tous les jours par tous les moyens. Un bandeau de publicité a certainement moins de chance de repousser un utilisateur qu'un prix à payer, aussi minime soit-il, parmi les régies publicitaires proposant actuellement de la publicité pour les applications Android, il existe AdMob ou encore AdSense de Google.

c) Rémunération par abonnement

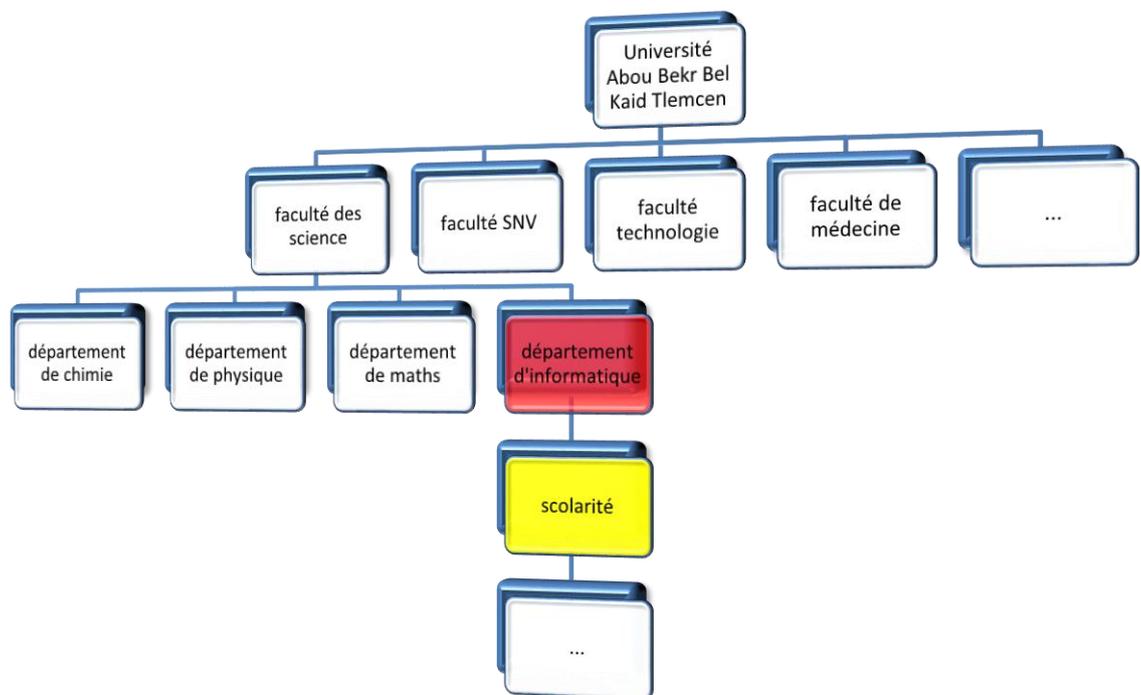
Pour certains services, il peut être souhaitable de distribuer l'application gratuitement, mais de demander un abonnement à l'utilisateur pour utiliser le service. Il est clair que ceci ne peut avoir d'intérêt que pour le cas d'application interfaçant un service web. Parmi les applications fonctionnant sur ce principe il y a Deezer et Spotify qui propose tous deux l'accès à leurs musiques en streaming contre un abonnement. Pour les personnes utilisant beaucoup leur smartphone comme lecteur de musique, cette fonctionnalité peut leur être très avantageuse.

B. ANALYSE ET CONCEPTION

En premier lieu, nous allons analyser la situation actuelle en faisant un inventaire des difficultés rencontrées pour les quelles nous tentons d'apporter des améliorations.

I. Présentation de l'organisme d'accueil

Organigramme du département d'informatique :



II. Etude des documents

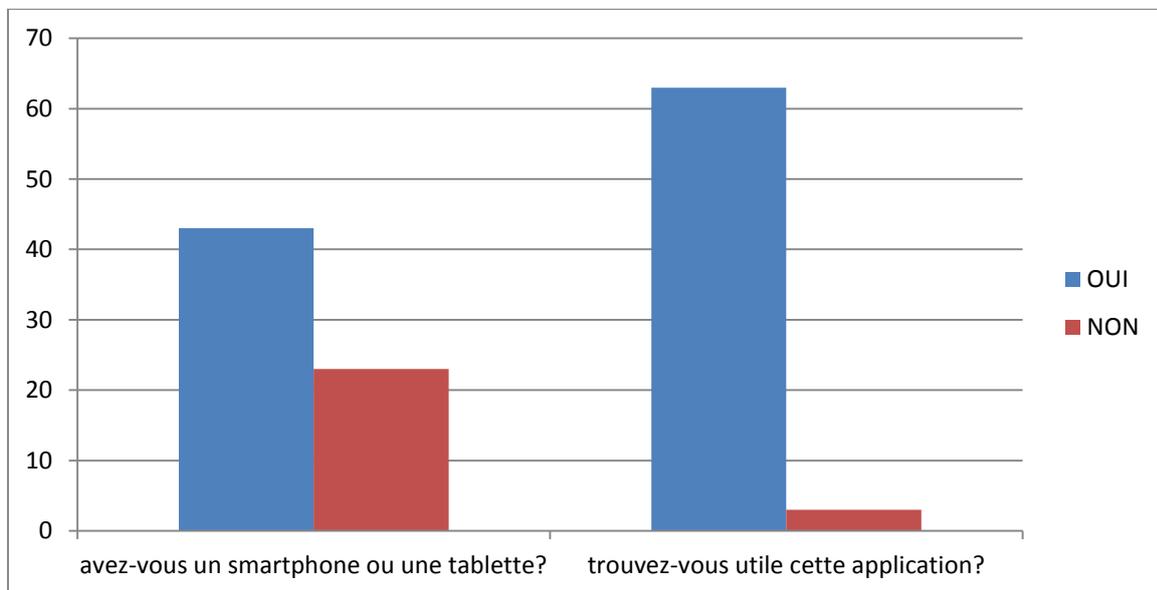
Cette étude permet d'avoir une idée plus claire sur les documents utilisés.

- Emplois du temps :
Horaires, modules, salles.
- Liste des matières et unités d'enseignements :
ID UE, intitulés des UE, crédits

- Listes des enseignants :
Nom et prénoms.
- Listes des étudiants :
Nom et prénoms, Promotion, Groupe.
- Thèmes projet de fin d'étude :
Thèmes, nom de l'encadreur.
- Listes des notes :
Nom et prénoms des étudiants, nom UE, notes.

III. Sondage

Dans le cadre de notre étude, nous avons mené un sondage sur 66 étudiants de la faculté des sciences qui a donné les résultats suivants :



Après le sondage effectué, nous pouvons en conclure les points suivants :

- Plus de 65 % des étudiants possèdent des smartphones ou tablettes
- Tous les appareils mobiles possédés par ces étudiants peuvent héberger l'application qui va être développé.
- Plus de 95% de ces étudiants sont très favorables pour ce projet.

IV. Critiques et suggestions

Critiques :

L'affichage ne se fait pas correctement.

L'affichage n'est pas toujours mis au bon endroit.

L'arrachement des affichages par certain étudiants.

Bousculade aux moments de l'affichage des résultats.

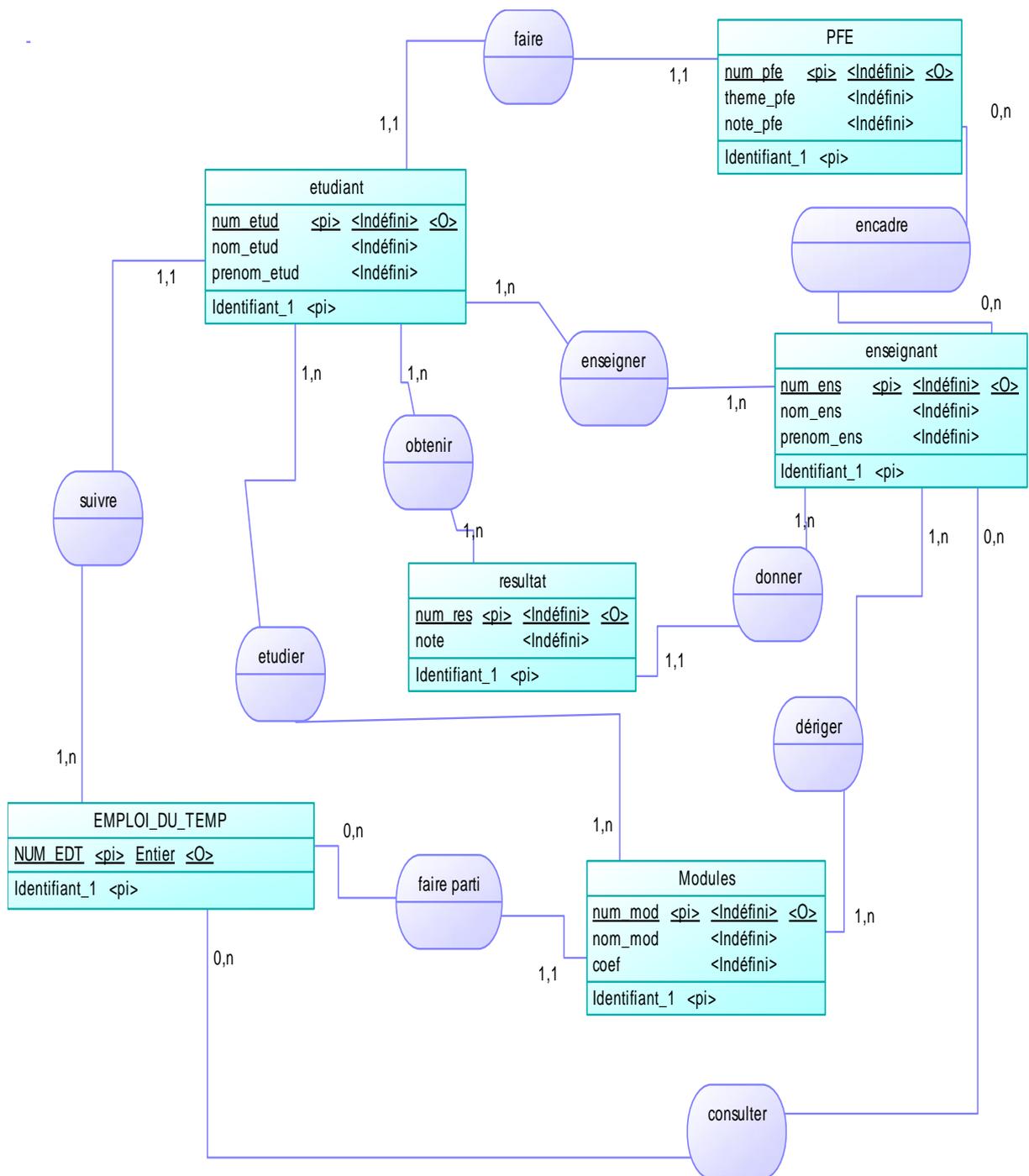
Solution :

Concevoir une application qui permet d'avoir toutes les informations en un temps réel.

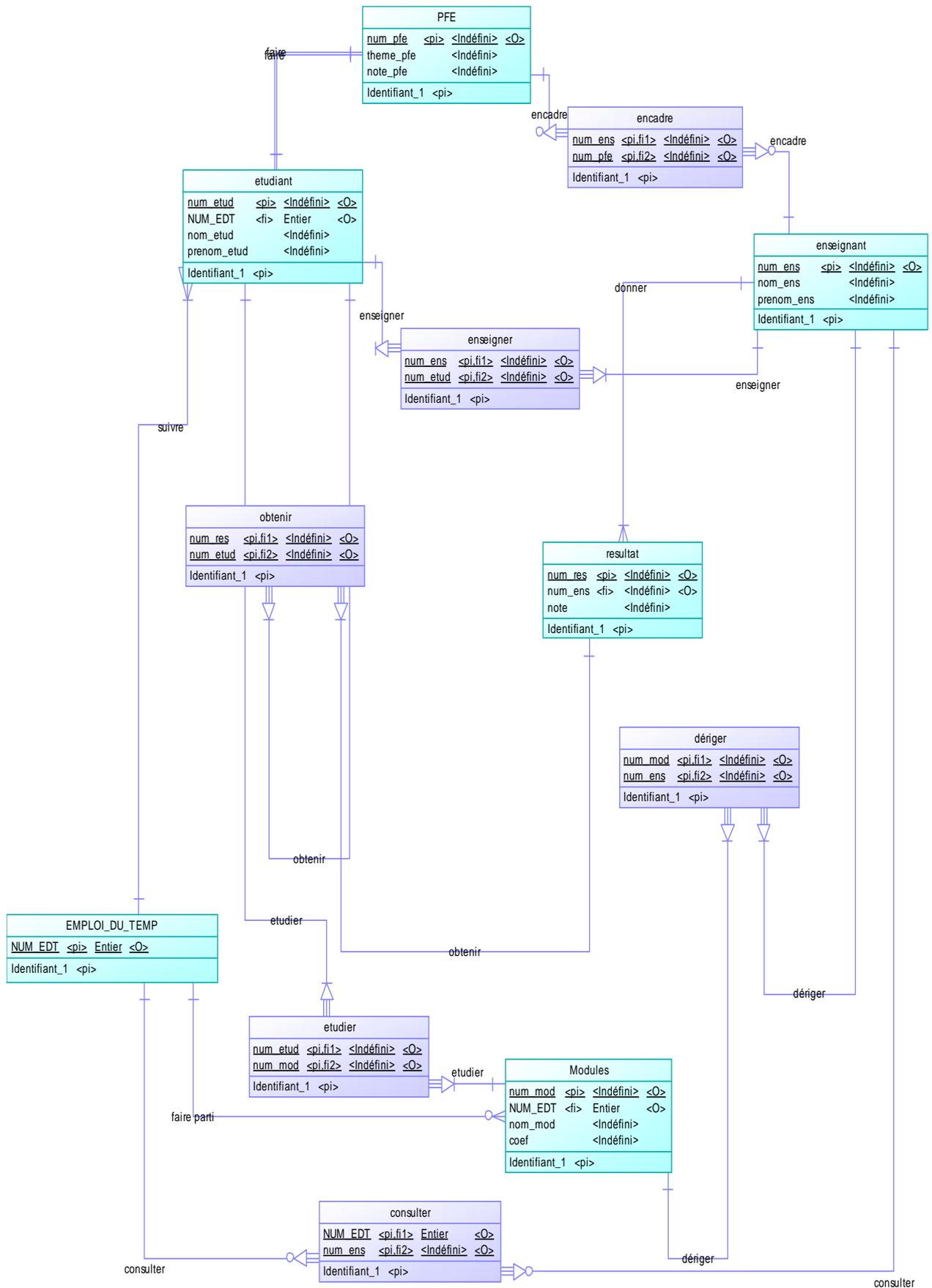
V. Dictionnaires des données :

Abréviation	Désignation	Type	Taille
nom_mod	Nom du module	A	30
num_mod	Numéro du module	N	5
Coef_mod	Coefficient du module	N	3
num_etud	Numéro d'étudiant	N	10
Nom_etud	Nom de l'étudiant	A	30
Prenom_etud	Prénom de l'étudiant	A	30
note_etud	Note de l'étudiant	N	5
nom_ens	Nom d'un enseignant	A	30
Prenom_ens	prénom d'un enseignant	A	30
grd_ens	Grade d'un enseignant	A	4
theme_pfe	Thème projet de fin d'étude	A	50
note_pfe	Note projet de fin d'étude	N	5
num_emp	Numéro de l'emploi du temps	N	10

VI. MCD



VII. MLD



C. REALISATION

I. Présentation de l'environnement de développement

1. Langage

a) Java

Pour développer sur Android, Google a mis à disposition un SDK 1 permettant d'accéder à toutes les fonctionnalités des appareils, par exemple l'accéléromètre ou l'écran tactile et ceci de manière simple en utilisant des bibliothèques existantes, l'ensemble de ces bibliothèques constitue le Framework Android.

Ce SDK, basé sur du Java, demande quelques mises à niveau, mais est à la portée de tout développeur ayant des connaissances en développement orienté objet.

Pour des jeux ou des bibliothèques demandant un maximum de ressources ou ayant besoin d'une optimisation accrue, un NDK3 est mis à disposition. Ce NDK permet de compiler du code C++ offrant une exécution beaucoup plus rapide qu'avec le Java.

b) Interface XML

Dans la lignée du Flex d'Adobe ou du XAML de Microsoft, les interfaces utilisateurs sous Android sont définies dans un langage balisé de type XML1. Cette approche donne la possibilité au développeur d'écrire son interface, d'utiliser un outil de création d'interface ou encore de le développer lui-même. Ce format étant ouvert et humainement compréhensible, il est facilement traitable dans un script ou une application pour, par exemple, remplacer en masse un élément présent dans plusieurs interfaces.

2. Mise en place de l'espace de travail

La mise en place d'un espace de travail pour développer sur Android est relativement aisée. Contrairement au développement pour l'iPhone qui exige l'utilisation d'un ordinateur Mac fonctionnant sous MacOS X, il est possible d'avoir un environnement de développement pour Android sous Linux, Microsoft Windows et Mac OS.

Pour développer sur Android, il faut commencer par télécharger l'EDI Eclipse gratuitement disponible sur le site officiel (<http://www.eclipse.org/downloads/>). Ensuite, il faut télécharger le SDK sur le site officiel de développement sous Android (<http://developer.android.com>) et suivre les instructions d'installation.

3. Machine virtuelle

Comme toute application développée en Java, une machine virtuelle permet son exécution. Cette machine virtuelle traduit en temps réel le code précompilé en code interprétable par la machine. Bien que cette solution ralentisse sensiblement la vitesse d'exécution du programme, elle résout, à l'aide de bibliothèques de classes extensibles, beaucoup de problèmes de programmation rencontrés avec un langage comme le C ou le C++. En effet, elle met à disposition des fonctionnalités permettant d'augmenter la productivité du développeur et fournit le GarbageCollector.

Le GarbageCollector s'occupe de gérer la mémoire de manière optimale en supprimant automatiquement les ressources qui ne sont plus utilisées. Lors de la programmation dans des langages compilés comme le C ou le C++, le développeur a l'importante tâche de libérer la mémoire qu'il n'utilise plus.

4. Les outils mis à disposition par le SDK

a) ADB (Android Debug Bridge)

ADB est un outil disponible en ligne de commande. Comme son nom l'indique, il permet de se connecter à l'émulateur ou au téléphone à l'aide d'un câble USB standard.

En outre, il donne la possibilité d'installer ou de désinstaller des applications, d'envoyer ou de récupérer des fichiers et de se connecter en ligne de commande sur l'appareil. Il est aussi utilisé par le SDK pour publier et déboguer les applications en développement et peut-être aussi sollicité par certaines applications, par exemple, « Remote SMS » qui permet d'envoyer des SMS depuis le navigateur d'un ordinateur.

b) L'émulateur

Le SDK d'Android met aussi à disposition des développeurs un outil en ligne de commande appelé « emulator ». Cet outil permet de configurer puis de lancer un émulateur de smartphone sous Android. L'avantage de cet outil est qu'il donne la possibilité de développer une application pour Android sans avoir de matériel particulier, comme un smartphone. De plus, il est entièrement intégré graphiquement sous Eclipse.

5. Les différents types d'applications sous Android

Une application sur Android hérite toujours d'un des quatre types définis ci-dessous.

a) Activity

L'activity est l'application de base sous Android. Comme une application avec une interface pour l'utilisateur dépend toujours d'une activity, la plupart des applications en ont donc au moins une. L'activity va permettre d'afficher une vue contenant des contrôles. Ces contrôles sont simplement les listes, images, boutons, champs de saisie, etc. Dans le modèle MVC (Model, View, Controller), l'activity est le controller. C'est donc en quelque sorte le moteur de l'application. Une activity peut en ouvrir une autre, lui fournir des données et en récupérer à sa fermeture.

b) Service

Un service est une application sans interface pour l'utilisateur puisque fonctionnant en arrière-plan. C'est justement avec ces services que le multitâche d'Android a un sens. Le service permettra par exemple de lire un morceau de musique tout en faisant autre chose. Ou encore continuer le téléchargement d'un fichier malgré la fermeture de l'application.

c) BroadcastReceiver

Les broadcastreceiver sont une alternative aux services. C'est une sorte de service endormi qui se réveille suite à une demande spécifiée. Ils ont été créés pour répondre au besoin d'économie d'énergie que demandent les appareils comme les smartphones. Il est possible de placer des broadcastreceiver sur des alarmes pour réaliser des actions différées ou périodiques comme la mise à jour d'un widget météo. Ils permettent aussi d'écouter des messages du système ou d'autres applications comme la réception d'un SMS. Il est bien sûr possible d'envoyer ses propres messages broadcast.

d) ContentProvider

Les ContentProvider, comme leur nom l'indique, sont des gestionnaires de données. Ils permettent la mise à disposition de données filtrables aux autres applications. Ils sont utilisés pour récupérer la liste des SMS ou encore la liste des contacts. Il est aussi possible de créer ses propres ContentProvider afin de rendre disponibles les données de l'application en question. Par exemple, pour une application de carte de visite, il peut être judicieux de proposer un accès contrôlé aux données constituant les cartes de visite.

6. Quelques caractéristiques du SDK d'Android

a) Création du projet simplifiée

La création de projet Android est très simple grâce à l'assistant fourni par le SDK. Pour ce faire, il suffit de sélectionner le menu File > New > Android Project.

Il faut commencer par saisir le « Project name ». Cela correspond au nom du projet sous Eclipse et pas au nom de l'application. Il est donc possible d'y mettre un nom correspondant à une nomenclature particulière.

La case à cocher « Createprojectfromexistingsample » permet de charger les sources d'un des nombreux exemples fournis par le SDK.

Parmi ces exemples, il y a « ApiDemos » qui est une application regroupant le code source permettant d'utiliser la plupart des fonctionnalités du SDK. Cela démontre comment utiliser la boussole, le GPS ou encore utiliser le moteur de rendu 3D.

Il y a aussi « GestureBuilder », un code source pour s'initier à la gestion des détections de formes dessinées aux doigts sur l'écran tactile. Cette fonctionnalité, apparue avec le SDK 1.6 Donut, peut-être très intéressante pour proposer une navigation rapide dans l'application.

Sont aussi présents deux exemples, « Home » et « SoftKeyboard », pour s'initier au développement de home et de clavier alternatif. Ces projets de démonstration sont une bonne source d'information pour découvrir par l'exemple le SDK Android.

La sélection du « buildtarget » n'est pas à prendre à la légère. Comme nous l'avons déjà vu, il faudra choisir un compromis entre les nouvelles fonctionnalités offertes par chaque SDK et la cible d'utilisateurs souhaitée. En effet, choisir d'utiliser le SDK 2.0.1 peut-être très intéressant voir indispensable pour certain projet, mais seulement un utilisateur sur cinq aura la possibilité d'utiliser l'application. Il faut donc toujours se référer au graphique de répartition des versions disponibles sur le site officiel « Android Developers » <http://developer.android.com/intl/zh-TW/resources/dashboard/platform-versions.html>.

Enfin, il ne reste plus qu'à spécifier les propriétés du projet.

« Application name » correspond au nom réel de l'application C'est celui qui sera affiché sur le terminal de l'utilisateur après l'installation. Il est toute fois possible de le changer ultérieurement.

« Package name » est un élément important de l'application. Il permet de différencier une application d'une autre donc il doit être impérativement unique. Afin de s'en assurer, il est convenu d'utiliser le nom de domaine de l'éditeur suivi du nom de l'application. Ainsi, pour un éditeur dont le nom de domaine serait domain.tld, le « package name » devrait être tld.domain.applicationname.

L'option « CreateActivity » permet de créer automatiquement « l'activity » principale de l'application. Ça sera donc « l'activity » lancer à son démarrage

Enfin, « Min SDK Version » est renseigné automatiquement lors de la sélection du « Build Target ». Il représente le numéro de version d'Android à avoir ou dépasser pour accéder à l'application.

b) Architecture de base d'un projet

Afin de voir les différents éléments composant un projet, nous allons détailler l'architecture d'un nouveau projet Android.

La zone surlignée en bleu correspond aux sources Java du projet. C'est dans ce dossier src que sont disposées les « Activity », les classes d'accès aux données et toute la partie moteur du projet.

La zone surlignée en rouge comprend les fichiers générés par le SDK et les librairies

qu'utilise le projet. Les fichiers générés ne doivent pas être modifiés puisqu'ils sont souvent mis à jour automatiquement par le SDK. Le fichier « R.java » permet au SDK de faire le lien entre les identifiants utilisés dans les sources et les ressources auquel ils correspondent.

La zone surlignée en vert contient les différentes ressources du programme. Il peut s'agir d'image, de son, de vue ou de n'importe quels fichiers devant être utilisés par l'application.

Plus précisément, le dossier « assets » permet la copie de fichiers de tous types alors que le dossier « res » correspond aux ressources identifiables par le SDK. Les ressources



contrairement aux « assets » sont prises en compte par le SDK. Elles doivent être classées dans des sous-dossiers spécifiques et seront accessibles dans le code de manière simplifiée.

C'est dans le fichier « R.java » que le SDK se chargera de renseigner l'ajout ou la modification de ressources.

La dernière zone, surlignée en jaune, correspond au fichier définissant les propriétés de l'application et du projet. Le fichier « AndroidManifest.xml » comprend toutes les propriétés de l'application comme son nom, son « package name », ou encore sa version. C'est aussi dans ce fichier que l'on va spécifier les permissions demandées par l'application. Le fichier « default.properties » est un fichier généré automatiquement par Eclipse et contient les propriétés du projet. Ce fichier n'est donc pas à prendre en considération.

c) Le fichier Android Manifest.xml

Le fichier « manifest » contient toutes les propriétés de l'application que l'on a spécifiées lors de la création du projet. De plus, il contient une référence de chaque « activity » composant l'application. Pour chacune d'elles, il est possible de spécifier des propriétés.

Ainsi, l'activité principale appelée « .Main » aura une propriété « action » spécifiant que c'est l'activité principale du projet et une propriété « category » pour préciser que l'application sera une application présente dans le « launcher », soit la liste des applications installées sur l'appareil de l'utilisateur.

Il est indispensable de spécifier toutes les « activity » de l'application dans ce fichier.

C'est aussi dans ce fichier que seront spécifiées les permissions que demande l'application.

Le SDK propose une interface graphique pour remplir ce fichier. Bien qu'avec le temps il est plus productif d'écrire directement dans le fichier, les assistants sont toujours utiles pour découvrir les possibilités offertes par le système.

d) Les ressources

Les ressources regroupent tout ce qui n'est pas du code source. Elles permettent de charger facilement des images, créer des interfaces utilisateurs, des animations, des styles. Leur architecture propose aussi une solution très simple pour l'internationalisation.



Ci-contre, les différentes ressources d'un petit projet.

- Les Drawables

Les « drawables » sont des images au format PNG, JPG ou encore BMP. On peut ensuite les appeler par le code à l'aide de « R.drawable. ressource_name ». Ainsi, pour afficher l'image « note.png » nous utiliserions l'identifiant « R.drawable. Note » au sein du code ou « @drawable/note » dans une vue.

- Les Layouts

Les « layouts » sont les vues et les contrôles de notre application. Il est possible de construire ces layouts à l'aide d'une interface graphique ou les écrire au format XML.

D'autant que l'écriture du code XML est aidée par une « auto-complétion¹ » fournie par Eclipse.

La création d'interface utilisateur sous Android est à l'apport de tous. En effet il est possible, après une courte période d'apprentissage, de dessiner ses "layout" à l'aide de glissé-déplacé. Bien que ce « designer » ne soit pas aussi efficace que ceux proposés par Microsoft avec Visual Studio ou Expression Blend, il permet l'essentiel. Ainsi, il donne la possibilité de placer les contrôles sur le layout et de spécifier leurs propriétés comme : leur identifiant, leur label ou leur taille.

- L'internationalisation

Grâce à l'architecture des ressources sous Android l'internationalisation est extrêmement simplifiée.

Une solution vraiment efficace a été mise en place afin d'inciter les développeurs à le faire.

Pour gérer l'internationalisation, il suffit de créer les dossiers de ressources terminant par le code ISO de la langue. Ainsi, pour un multilingage anglais-français, nous aurons « values-fr » et « values-en ». Il est aussi possible de créer un dossier commun « values ». La grande force de ce système réside dans son application sur tous les dossiers de ressources. Ainsi, avec les

¹ Le complètement ou complètement automatique, ou encore par anglicisme complétion ou auto-complétion, est une fonctionnalité informatique permettant à l'utilisateur de limiter la quantité d'informations qu'il saisit avec son clavier, en se voyant proposer un complément qui pourrait convenir à la chaîne de caractères qu'il a commencé à taper. (Source: Wikipedia)

deux dossiers « drawable », nous pourrions avoir une icône d'application correspondant à la langue de l'appareil. Il en est de même pour les « layout », les styles, ou toute autre ressource.

Contenu du fichier « values » commun situé dans /res/values/strings.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">applicationName</string>
</resources>
```

Contenu du fichier « values » en anglais situé dans /res/values-en/strings.xml :

```
<? xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World !</string>
</resources>
```

Contenu du fichier « values » en français situé dans /res/values-fr/strings.xml :

```
<? xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Bonjour le monde !</string>
</resources>
```

Ensuite, il ne reste plus qu'à utiliser les identifiants des strings dans le code source avec « R.strings.name » et dans les layouts avec @string/name pour afficher le texte en fonction de la langue de l'appareil.

e) Les bases de données SQLite

Android permet l'utilisation de base de données SQLite² et le SDK fournit tout le nécessaire pour créer et utiliser une ou plusieurs bases de données par application.

Pour se faire, il faut simplement créer une classe héritant la classe «SQLiteOpenHelper» intégrée au «framework».

²SQLite est une bibliothèque écrite en C qui propose un moteur de bases de données relationnelles accessibles par le langage SQL. SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID. Contrairement aux serveurs de bases de données traditionnels, comme MySQL ou PostgreSQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur, mais d'être directement intégrée aux programmes. L'entièreté de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme. (Source : Wikipedia)

Lorsque l'on étend la classe « SQLiteOpenHelper », il est indispensable d' « override » les méthodes « onCreate » et « onUpgrade » comme le montre l'exemple ci-dessous.

```
@Override
public void onCreate (SQLiteDatabase db) {
    String sql = "";
    sql = "CREATE TABLE tableName ("
        + "_id INTEGER PRIMARY KEY AUTOINCREMENT, "
        + "date INTEGER NOT NULL, "
        + "title TEXT NOT NULL, "
        + "body TEXT NOT NULL)";
    db.execSQL(sql);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    int currentVersion = oldVersion + 1;
    while (currentVersion < oldVersion){
        switch (currentVersion) {
            case 2:
                //Update from version 1 to version 2
                break;
            case 3:
                //Update from version 2 to version 3
                break;
            default:
                break;
        }
        currentVersion++;
    }
}
```

La méthode « onCreate » est exécutée à la première exécution de l'application et permet de créer les différentes tables constituant la base de données.

La méthode « onUpdate » est appelée pour mise à jour de la base de données suite à une mise à jour de l'application. Elle permettrait par exemple de réaliser des « ALTER TABLE ».

Après ouverture de la base de données, il est possible d'exécuter des requêtes à l'aide de diverses méthodes de l'objet « SQLiteDatabase ». Par exemple, pour récupérer tous les éléments de la table « tableName » il suffira d'écrire la ligne suivante.

```
Cursor cursor =
    db.query("tableName", null, null, null, null, null, "date DESC");
```

Le « cursor » que renvoie cette fonction est un objet pointant sur des données de la base de données. Il pourra être utilisé avec un « CursorAdapter » similaire à l' « ArrayAdapter »

f) Les Listview

Les « Listview » sont des listes d'éléments. Sous Android, ces dernières sont extrêmement personnalisables. Chaque « Listview » peut avoir comme élément n'importe quel contrôle du « Framework Android ». Il sera donc très simple de faire une liste de boutons ou même une liste de « LinearLayouts » contenant plusieurs contrôles.

II. Publication sur l'Android Market

Afin de démontrer la simplicité et la force de l'Android Market, nous allons montrer les différentes étapes pour publier une application.

- **Exportation du projet**

Pour exporter un projet sous Android, il suffit de faire un clic droit sur le projet dans Eclipse, puis de sélectionner « Export Signed Application Package » dans « Android Tools ».

Ceci ouvre un assistant.

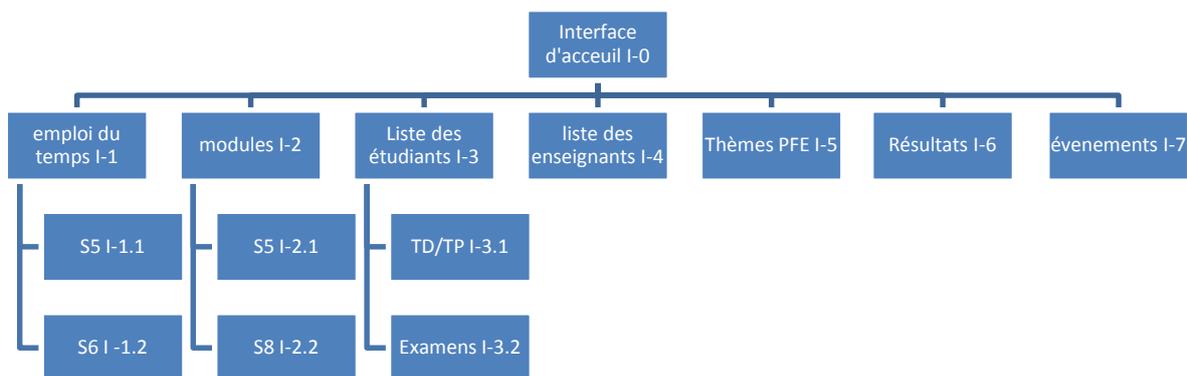
On sélectionne le projet que l'on souhaite exporter

Pour éviter qu'une personne mal intentionnée ayant récupéré les identifiants GMail du développeur fournisse une fausse mise à jour vérolée d'une application, il est nécessaire de signer une application pour sa mise en ligne sur le Market.

III. Application « UABBT App »

1. Structure générale de l'application

L'architecture générale de notre application est illustrée dans le schéma ci-dessous :



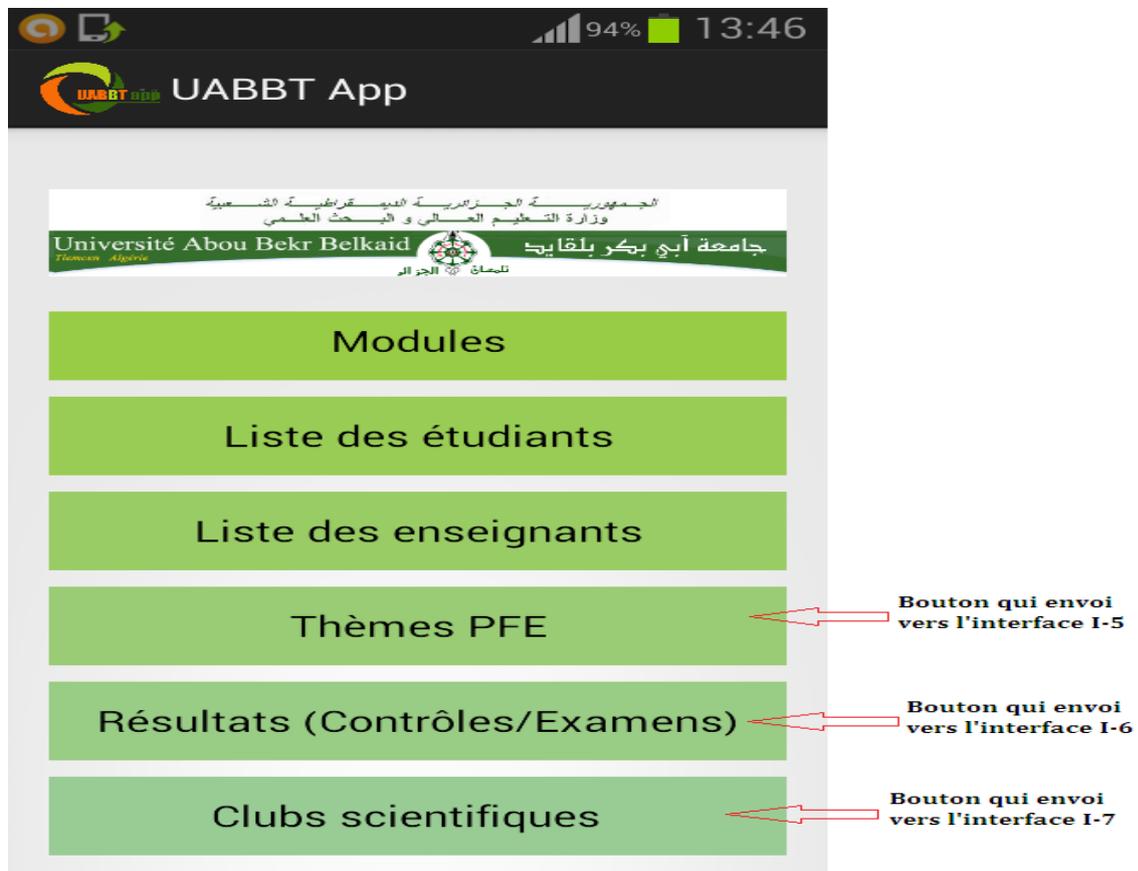
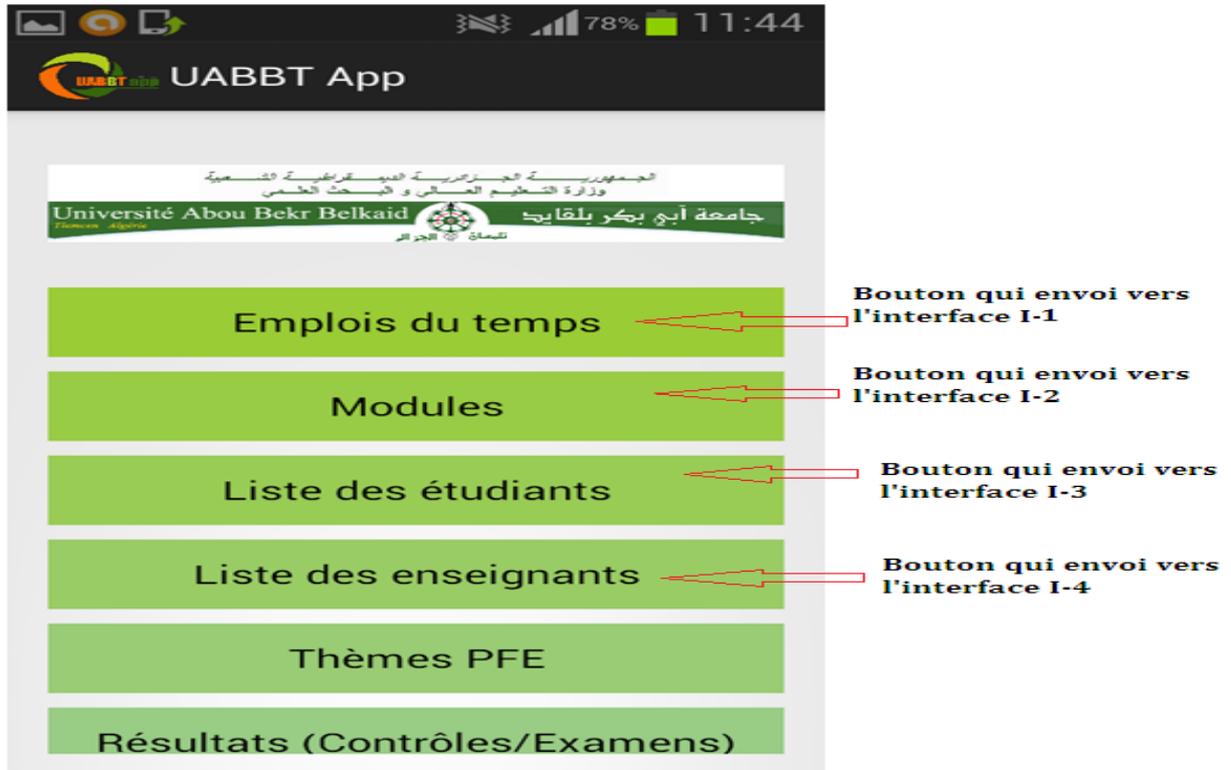
2. Page de chargement et bouton pour accéder à l'interface d'accueil

Après un splash de quelques secondes, le bouton qui permet d'accéder à l'interface d'accueil s'affiche :



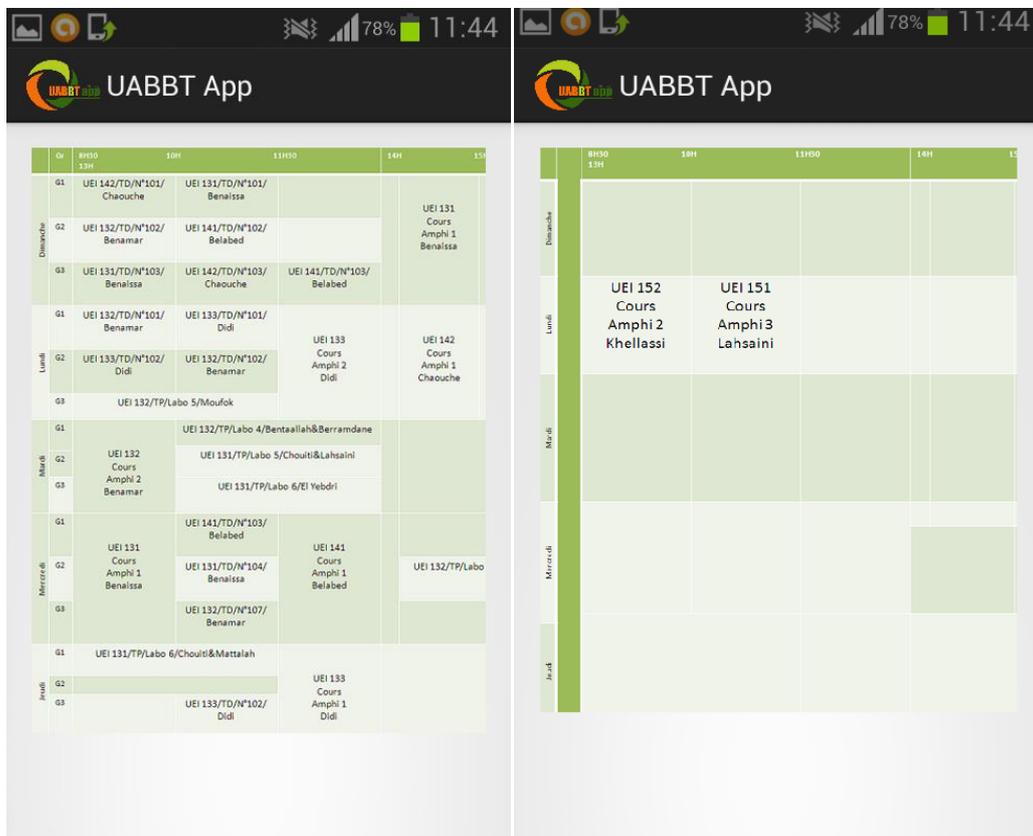
3. Menu principal

Interface I-0

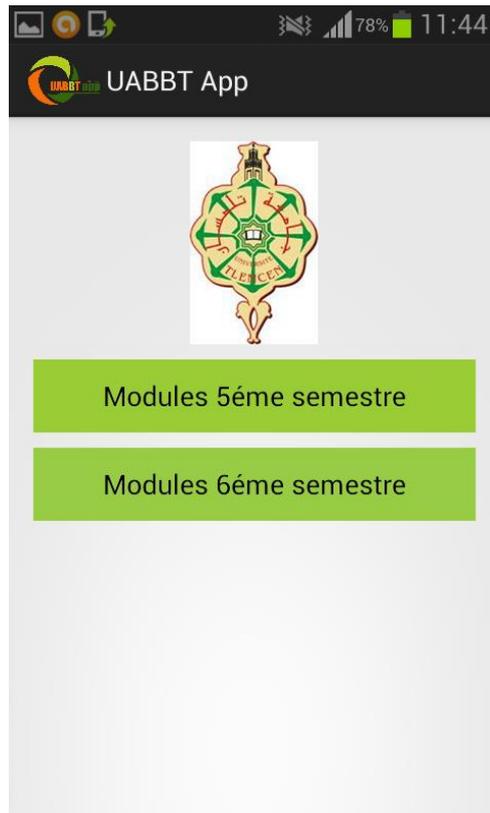


4. Les différentes interfaces de l'application

Emplois du temps I-1 :



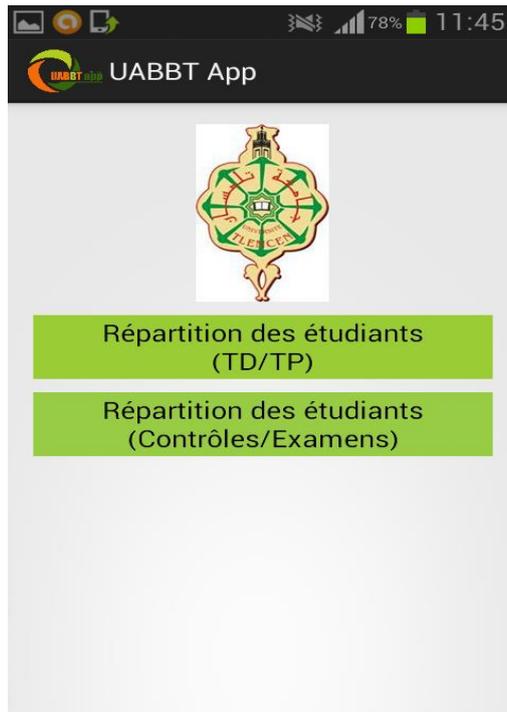
I-2 : Liste des modules



Modules	Nombre de cours	Nombre de TD	Nombre de TP	Coefficient
UEI 143: Infographie	1	0	0	4
UEI 142: Théorie des graphes	1	1	0	4
UEI 141: PROLOG	1	1	0	4
UEI 139: Réseau	2	1	0	6
UEI 132: Système d'exploitation	2	1	1	6
UEI 131: Compilation	2	1	1	6

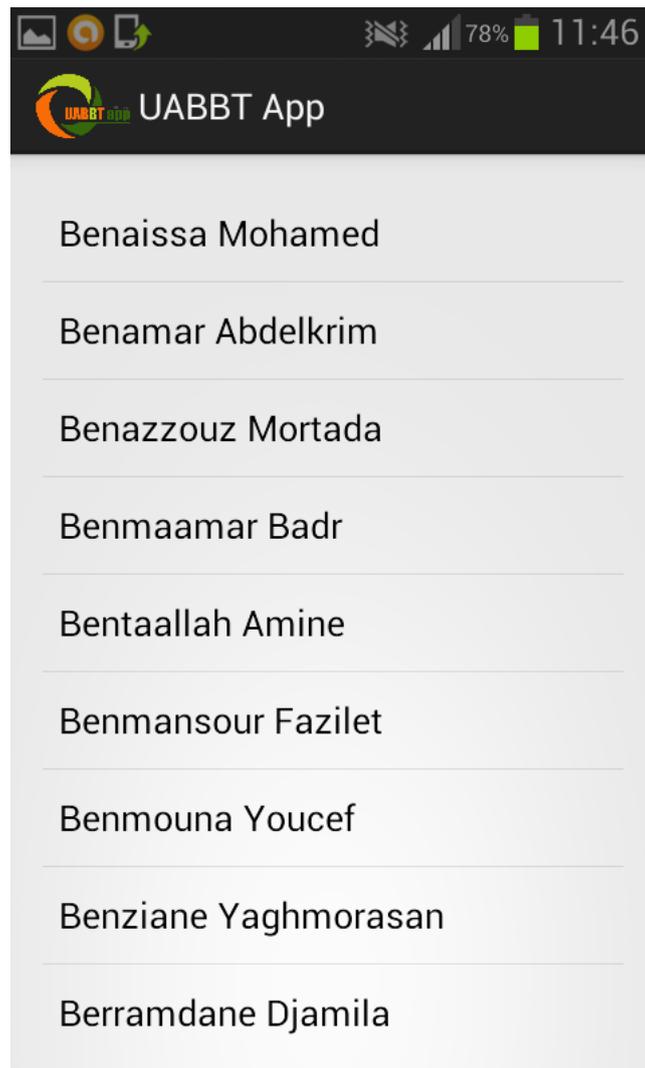
Modules	Nombre de cours	Nombre de TD	Nombre de TP	Coefficient
UEI 151: Cryptographie	1	0	0	5
UEI 152: Systèmes distribués	1	0	0	5
UEI 16: Projet	0	0	0	20

I-3 : Liste des étudiants



UABBT App		UABBT App	
BENFEDEL AHMED		BELLAHCENE MOHAMED AMINE	
BENGHELIMA MOHAMED AMINE		BEN BACHIR FATIMA	
BENGUEDIH ANWAR		BENABADJI ADIL	
BENKHELOUF IKRAM		BENACHENHOU WALID	
BENMAAMAR SOUAD		BENADDA AHMED	
BENMANSOUR FATIMA ZOHRA		BENALI (EP.SAIDI) NADJAT	
*****G2*****		BENALLAL SOUAD	
BENOSMAN IMANE		BENALLAL ZEYNEB	
BENSAHLA TALET WAFAE SARA		DIB ABDELKRIM	
BENSAHLA TANI HIDAYET		BENAMARA MOHAMMED EL AMIN	
BENSIDHOUM AYYOUB		*****SALLE 102*****	
BENSOUNA Karim		BENAMMAR NASSIMA	
BERRABAH RAHMA		BENDADA MERIEM	
BOUABANE SAMIYA		BENDAHMANE ZAHIRA	
BOUAFIA NOURIA		BENEDDOUCHE RIDA	
BOUAYAD AGHA MEHDI YOUSOUF		BENFEDEL AHMED	
BOUAZZA MOHAMMED REDA		BENGHELIMA MOHAMED AMINE	
BOUDAOUZ CHAHRAZED		BENGUEDIH ANWAR	
BOUGUENAYA (EP.GHEMB) YAMINA		BENKHALFA SIDI MOHAMMED	
BOUKLI HACENE ZOULEYKHA NOUR		BENKHELOUF IKRAM	
BRAHIMI SABIHA		BENMAAMAR SOUAD	
BRAHMI MOHAMMED		BENMANSOUR ABDERRAHMANE	
BRAHMI SARA		BENMANSOUR FATIMA ZOHRA	

I-4 : Liste des enseignants



I-5 : Les thèmes de projet de fin d'étude

UABBT App

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة أبي بكر بلقايد
Tlemcen, Algérie

تلمسان الجزائر

N°	Thème	Encadreur
1	Construction d'une application de gestion d'emploi du temps pour le département d'informatique	Mr BENAMAR Abdelkrim
2	Développement d'un site web sur Smartphone pour le département d'informatique	Mr BENAMAR Abdelkrim
3	Construction d'une application de gestion des plannings d'exams pour le département d'informatique	Mr BENAMAR Abdelkrim
4	Un système interactif de support de décision médicale.	Mr KHELASSI Abdeljalil
5	Une application de classification dans des données extraites des réseaux sociaux	Mr KHELASSI Abdeljalil
6	Classification des documents médicale selon le niveau d'abstraction des utilisateurs (articles wiki et pubmed)	Mr KHELASSI Abdeljalil
7	Un système distribué (client-serveur) pour la gestion d'une cabine médicale.	Mr KHELASSI Abdeljalil
8	Une application E-Commerce pour la recommandation et les critiques des produits divers.	Mr KHELASSI Abdeljalil
9	Sécuriser 1 réseau Wifi par la méthode EAP.	Mme DIDI Fedoua
10	Conception d'un site Web dynamique par l'outil Joomla (avec application pour entreprise)	Mr Benzian Yaghmoracen
11	Modélisation d'une scène 3D par les mailles avec OpenGL	Mr Benzian Yaghmoracen
12	Gestion et Administration Réseaux.	Mme DIDI Fedoua
13	Conception d'un site de e-commerce.	Mme DIDI Fedoua
14	Gestion de stock en réseau LAN.	Mme DIDI Fedoua
15	Application de télé-surveillance par un réseau Wifi	Mme DIDI Fedoua
16	Application de téléphonie sur un réseau IP.	Mme DIDI Fedoua
17	Sécurité Informatique.	Mme DIDI Fedoua
18	Installation et Configuration d'un firewall pour un LAN.	Mme DIDI Fedoua
19	Gestion de stock sous réseau (avec INTERBASE)	Mr Benzian Yaghmoracen
20	L'implémentation d'une interface de requête sur les données RDF	Mr Be lebed Amine
21	L'implémentation d'un système à base de logique floue pour l'évaluation d'une soutenance	Mr Be lebed Amine
22	Développement d'une application sur smart phone	Mr BENAZZOUI Mortada

I-6 : Résultats contrôles, tp et examens

fs.univ-tlemcen.dz

FACULTE DES SCIENCES

ACTUALITÉS

Échéancier de fin d'année (Département d'Informatique)
Échéancier de fin d'année (Dates d'exams, remise de notes et soutenances de PFE)

Planning des Epreuves Finales du 2eme Semestre (Département d'Informatique)
L3 et M1 SIC/MID/RSD/GL: du 25 au 29 Mai 2014.

Licence 3 (Semestre 5)
Master 1 GL (Semestre 2)
Master 1 MID (Semestre 2)
Master 1 SIC (Semestre 2)
Master 1 RSD (Semestre 2)

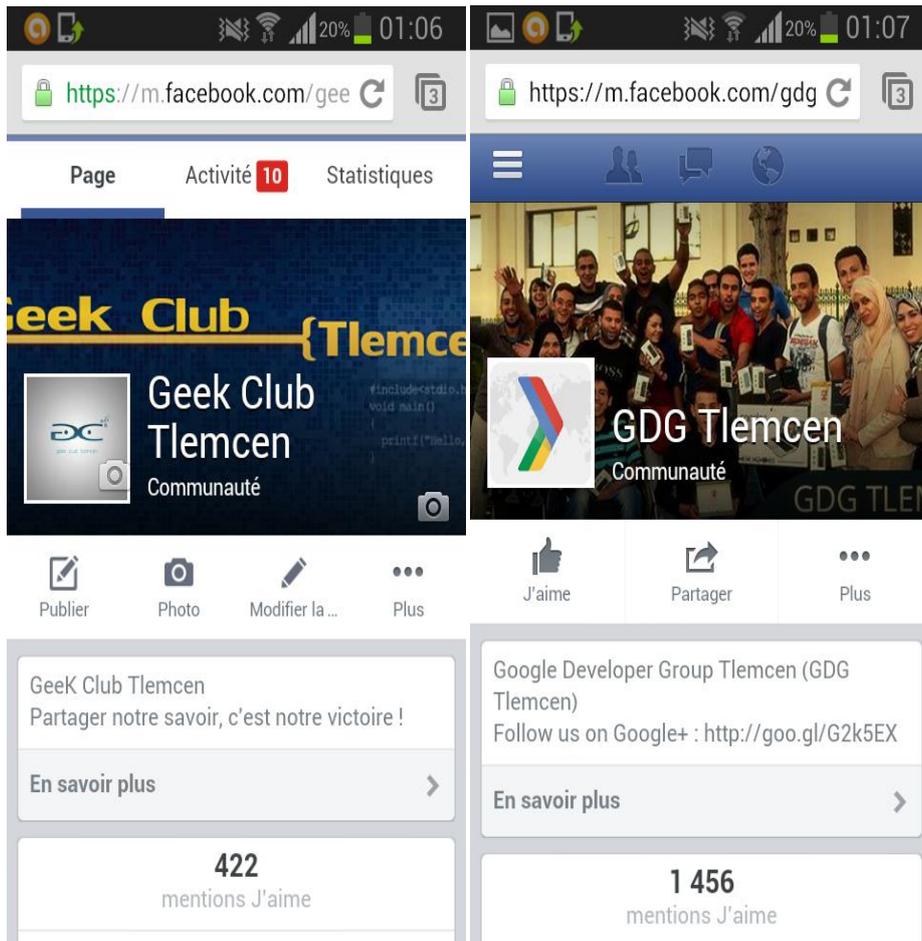
L2: du 1er juin au 5 juin 2014.
Licence 2 (Semestre 4)

Résultat contrôle et examen "Système d'exploitation"

Salle 101				
N°	Nom et prénom	EP	TP	CC
1	ABDELLOU ABDELRAHMAN ALAMA	10,5		14
2	ABDELLOU NAOUA BOUSSENE	16,5	11	14,5
3	ABDELRAHMAN BOUSSENE	11		12
4	ABDELRAHMAN BOUSSENE	3,5		
5	ABDELRAHMAN BOUSSENE	3		
6	ABDELRAHMAN BOUSSENE	10		12
7	ABDELRAHMAN BOUSSENE	13		15
8	ABDELRAHMAN BOUSSENE	11		13,5
9	ABDELRAHMAN BOUSSENE	3,5		13
10	ABDELRAHMAN BOUSSENE	4		10
11	ABDELRAHMAN BOUSSENE	10		10
12	ABDELRAHMAN BOUSSENE	8,5		14,5
13	ABDELRAHMAN BOUSSENE	8,5		10
14	ABDELRAHMAN BOUSSENE	9		14,5
15	ABDELRAHMAN BOUSSENE	13,5		16
16	ABDELRAHMAN BOUSSENE	8,5		10
17	ABDELRAHMAN BOUSSENE	11,5		15
18	ABDELRAHMAN BOUSSENE	2		
19	ABDELRAHMAN BOUSSENE	2		
20	ABDELRAHMAN BOUSSENE	10		10
21	ABDELRAHMAN BOUSSENE	12		10
22	ABDELRAHMAN BOUSSENE	16,5		19,5
23	ABDELRAHMAN BOUSSENE	11		14
24	ABDELRAHMAN BOUSSENE	8,5		
25	ABDELRAHMAN BOUSSENE	12,5		14,5
26	ABDELRAHMAN BOUSSENE	12,5		14,5
27	ABDELRAHMAN BOUSSENE	13,5		17
28	ABDELRAHMAN BOUSSENE	7,5		10
29	ABDELRAHMAN BOUSSENE	3,5		10
30	ABDELRAHMAN BOUSSENE	3,5		12
31	ABDELRAHMAN BOUSSENE	2		11
32	ABDELRAHMAN BOUSSENE	14		18,5
33	ABDELRAHMAN BOUSSENE	13		16,5
34	ABDELRAHMAN BOUSSENE	11,5		10
35	ABDELRAHMAN BOUSSENE	4,5		

Salle 102				
N°	Nom et prénom	EP	TP	CC
1	ABDELRAHMAN BOUSSENE	8,5		11
2	ABDELRAHMAN BOUSSENE	12,5		19
3	ABDELRAHMAN BOUSSENE	10,5		15,5
4	ABDELRAHMAN BOUSSENE	12,5		15,5
5	ABDELRAHMAN BOUSSENE	14,5		17
6	ABDELRAHMAN BOUSSENE	11		13,5
7	ABDELRAHMAN BOUSSENE	8		12,5
8	ABDELRAHMAN BOUSSENE	2,5		10

I-7 : Clubs actifs au sein de la faculté d'informatique



Conclusion

Concevoir et réaliser une application informatique mobile pour notre département informatique s'est fixé comme objectif principal de notre projet de fin d'études.

Notre travail s'est basé sur le développement d'un programme sur les technologies mobiles (Smartphone), ceci nous a amené à découvrir une nouvelle plateforme de développement et à enrichir notre savoir et notre expérience.

L'application que nous avons réalisée permet à un étudiant d'avoir toutes les informations utiles durant sa formation (note de contrôles/tp/examens, délibérations, emploi du temps ou changement d'emploi du temps, séances de rattrapage, changement de salles, soutenances, dates de dépôt de mémoires,...) directement sur leurs smartphones ou tablettes, aussi ça va les tenir au courant de toute manifestation scientifique ou n'importe quel autre événement.

Les objectifs fixés de notre projet de fin d'études ont été atteints à un degré très avancé. D'un côté nous avons présenté une application bénéfique aussi pour les étudiants que pour l'administration. D'un autre côté, ce projet nous a permis de nous familiariser avec Android et d'approfondir nos connaissances dans le domaine des applications mobiles.

Rappelons que cette application a été soumise pour le concours national **istart** « **oobarmijoo** » organisé par l'opérateur Ooredoo en février 2014, et a été acceptée par le jury parmi les 50 meilleures applications. En conséquence nous avons bénéficié d'une formation gratuite à Alger avec prise en charge totale.

Au niveau des perspectives de notre travail, nous envisageons d'améliorer notre application sur plusieurs aspects, par exemple introduire une base de données qui peut être mise à jour en permanence pour une meilleure efficacité.

Nous pourrions par la suite étendre notre application aux autres promotions et pourquoi ne pas élargir ce projet aux autres départements et aux autres facultés.

Nous sommes à la naissance du système Android et malgré toutes les opportunités que nous apporte ce produit, nous avons que peu d'idées sur ses multiples possibilités.

Seule notre imagination freine le développement du système Android.

Liste des abréviations

SMS Short Message Service

MMS Multimedia Messaging Service

EDI Electronic Data Interchange

ADB Android Debug Bridge

SDK Software Development Kit

NDK Native Development Kit

GPS Global Positioning System

XML eXtensibleMarkupLanguage

XAML eXtensible Application MarkupLanguage

PNG Portable Network Graphics

OS operating system

MCD modèle conceptuel des données

MLD modèle logique des données

Bibliographie

<http://www.android.com/>

<http://www.wikipedia.fr>

<http://developer.android.com/>

<http://android.smartphonefrance.info/>

<http://sosandroid.fr/>

<http://www.openhandsetalliance.com/>

<http://www.pointgphone.com/>

<https://www.eclipse.org/>

<http://sqlitebrowser.org/>