

République Algérienne Démocratique Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Sciences, Technologies et Sciences de la Matière

Département Informatique

Thème

Utilisation de Moodle sous Android à travers des applications natives invoquant les web services de Moodle – cas de l'application « Affichage des notes des étudiants »

Pour l'obtention du grade de Master

Spécialité : Informatique

Option : Génie logiciel

Réalisé par : Mr. NOUACER NOUREDDINE

Mr. KASSIDI ABDERRAHMANE

Dirigé par : Mr. BEKKI KHADIR

Mr. MOSTFAOUI SID AHMED

Encadré par : Mr. TALBI OMAR

Année scolaire 2015-2016

Dédicaces

La part de KASSIDI Abderrahmane

A ma très chère mère

Honorable, aimable : Tu représentes pour moi l'exemple qui n'a pas cessé de m'encourager et de prier pour moi. Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu ; le tout puissant, te préserver et T'accorder santé, longue vie et bonheur.

A mon très chère Père

Aucune dédicace ne saurait exprimer l'amour, le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as Consentis pour mon éducation et ma formation.

Je dédis aussi ce travail a :

A mes chers frères et sœurs.

A tous mes copains et copines avec qui j'ai passé de très bons moments.

A tous mes enseignants qui m'encourage à faire des efforts et encore à tous mes profs de lycée grâce à leur intérêt et leur respect.

Pour ceux qui je ne cité pas, bien sur ne croyaient pas que je vous oublie, je vous porte toujours de mon cœur.

Dédicaces

La part de NOUCER Nouredine

A ma très chère mère

Honorable, aimable : Tu représentes pour moi l'exemple qui n'a pas cessé de m'encourager et de prier pour moi. Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu ; le tout puissant, te préserver et T'accorder santé, longue vie et bonheur.

A mon très chère Père

Aucune dédicace ne saurait exprimer l'amour, le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as Consentis pour mon éducation et ma formation.

Je dédis aussi ce travail a :

A ma famille et mes chère fils et filles.

A tous mes copains et copines avec qui j'ai passé de très bons moments.

A tous mes enseignants qui m'encourage à faire des efforts et encore à tous mes profs de lycée grâce à leur intérêt et leur respect.

Pour ceux qui je ne cité pas, bien sur ne croyaient pas que je vous oublie, je vous porte toujours de mon cœur.



Remerciement

En préambule à ce mémoire nous remercions ALLAH qui nous a aidé et Nous donné la patience et le courage durant ces longues années d'études.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce travail ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenons à remercier notre encadreur, Mer : TALBI OMAR pour sa disponibilité et ses conseils, son orientation et Surtout son aide, et tous les enseignants qui de notre spécialité et le département de l'informatique.

Il nous toujours aidés dans les moments difficiles et nous a guide dans la bonne direction dans notre travail.

On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Résumé

L'objectif de cette mémoire est de présenter quelques aspects pour l'utilisation des Téléphones « Smartphones » dans notre vie quotidienne qui à connus une grande demande et succès dans le marché des mobiles avec sa simplicité d'utilisation et sa portabilité et sa disponibilité de connexion internet via la génération 3G, où tous les étudiants de l'université possèdent un Smartphone et sont inscrits dans la plateforme LMS(*Learning Management System*) Moodle, mais a chaque affichage des notes sur papier après les examens, les étudiants rencontrent beaucoup problèmes au niveau de scolarité notamment le problème de déplacement, les couts,...., pour cela nous avons crée un bloc dans Moodle qui sert a afficher ces notes a travers un navigateur web approprié des téléphones, mais cette solution reste insuffisante pour les téléphones mobile de petit écran, mémoire RAM limité ce qui à crée un nouveau problème, et nous à poussé de trouver autre solution, où on à crée une application native sous le système Android invoquant les web services de Moodle.

Sommaire

Introduction Générale.....	4
Chapitre 1 : Le e-learning et MOODLE.	5
Introduction.....	5
1. Le <i>e-learning</i> :.....	5
1.1 Pourquoi développer des cours E-learning?	5
1.2 Qualité du e-learning :.....	6
1.3 Avantages de e-learning :.....	7
1.4 Plate-forme LMS:.....	7
1.4.1 Utilisateurs de la plate-forme LMS :.....	8
1.4.2 Objectifs de la plate-forme LMS.....	8
1.4.3 Quelques Systèmes de Gestion des Cours	9
1.5 Définition du m-Learning:	10
2. MOODLE :	10
2.1 État de développement :	11
2.2 Architecture de Moodle :	11
2.3 Moodle code :	13
2.4 Server Web de Moodle :.....	14
Conclusion	15
Chapitre 2 : Les systèmes d'exploitation mobiles	16
Introduction.....	16
1. Système d'Exploitation(SE) :.....	16
1.1. BlackBerry :.....	16
1.2. PalmOS :	17
1.3. Symbian :	17
1.4. Apple I OS :	17
1.5. Windows Mobile :	18

2. Android :.....	18
2.1. Architecture d'android :.....	19
2.2. Les composants d'une application :.....	21
Conclusion	22
Chapitre 3 : Conception et modélisation	23
Introduction :.....	23
1. Méthodes agiles :.....	23
2. Le Processus (UP) - Unified Process:	23
3. La méthode 2TUP	23
4. Conception du logiciel :.....	25
4.1 Etude Préliminaire :.....	25
4.1.1 Présentation du projet à réaliser :.....	25
4.1.2 Recueil des besoins fonctionnels :	25
4.1.3 Recueil des besoins techniques :.....	26
4.1.4 Choix techniques :	27
4.1.5 Identification des acteurs.....	27
4.1.6 Identification des messages	28
4.1.7 Modélisation du contexte	28
5. CAPTURE DES BESOINS FONCTIONNELS.....	29
5.1 Déterminer les cas d'utilisations	29
6. CONCEPTION	29
6.1 Diagramme de classes :.....	30
6.2 Diagramme de séquence :.....	30
6.3 Architecture de l'application	31
7. CODAGE :.....	32
7.1 Structure générale de l'application :.....	32
Conclusion	34
Chapitre 4 : Mis en œuvre.....	35

Introduction.....	35
1. Partie MOODLE :.....	35
1.1 Installation du plugin sur MOODLE :	35
1.2 Web service dans MODDLE:	36
2. Partie Android :	37
2.1 Exporter un fichier (*.apk) :.....	37
2.2 Installation sur téléphone :	39
2.3 Premier lancement :	39
Conclusion :	40
Chapitre 5 : Application native et Application Web.....	41
1. Application native :	41
2. Application web:	42
3. L'installation et l mise à jour :	42
4. Le coût de développement :	42
5. L'expérience utilisateur :	43
6. Conclusion :	43
Conclusion Générale	45
Bibliographie	46
Table de figures :	47
Webographie.....	48

Introduction Générale

L'utilisation du e-learning dans le domaine de l'enseignement et de la formation se généralise, particulièrement dans l'enseignement supérieur. Les étudiants n'hésitent plus à utiliser des plateformes de formation à distance telles que Moodle pour accéder à des cours mis à leur disposition par le corps enseignant. Compte tenu de la possibilité d'intégrer des plugins à cette même plateforme, des applications répondant à divers besoins émis par les enseignants, les étudiants et parfois même par administration ont été développées dans ce sens. Ces applications gagnent en pertinence et en puissance en profitant des aspects sécuritaires, de l'accès partagé, multiutilisateurs et à distance qu'offre Moodle. Nous proposons dans le cadre de ce projet de fin d'études, l'utilisation de Moodle sur les Smartphones, notamment ceux fonctionnant avec le système android à travers des applications natives invoquant les services web de Moodle. Pour ce faire nous avons axé notre travail sur une méthode de génie logiciel, à savoir, *Two Track Unified Process (2TUP)*. Cette méthode répond à la fois aux besoins fonctionnelles et aux besoins techniques.

Nous structurons notre mémoire en 05 chapitres. Dans le premier chapitre nous présentons le e-learning et Moodle comme une plateforme de formation à distance. Dans le second chapitre nous abordons les systèmes d'exploitation des téléphones mobiles notamment Android le plus utilisé. Dans le chapitre trois, nous faisons une étude détaillée concernant le projet à réaliser en appliquant le processus 2TUP et en suivant les phases suivantes : capture des besoins fonctionnels, capture des besoins techniques, conception du logiciel et codage. Dans le chapitre quatre, nous présentons la mise en œuvre de notre application. Dans le chapitre cinq nous montrons la différence entre les applications web et les applications natives développées sous android. Finalement nous clorons avec une conclusion générale.

Chapitre 1 : Le e-learning et MOODLE.

Introduction

Aujourd'hui, la matière grise représente la principale richesse d'un pays. C'est aussi l'atout compétitif majeur d'une entreprise, ainsi que pour des parents où la formation de leurs enfants constitue aujourd'hui la meilleure dot.

Dans ce chapitre, On va commencer par donner une présentation du E-learning, ses avantages et pourquoi le E-learning. Comme champ d'application du e-learning, on a choisi de présenter la plate-forme de formation MOODLE.

1. Le e-learning :

Plusieurs définitions du e-learning ont été proposées et sont utilisées, nous vous proposons ci-dessous la définition retenue par la Communauté Européenne :

« Le e-learning est l'utilisation des nouvelles technologies multimédia et de l'Internet, pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que des échanges et la collaboration à distance ». (Prat, 2008)

1.1 Pourquoi développer des cours E-learning?

De nombreux organismes et institutions utilisent le e-learning, parce qu'il peut se révéler aussi efficace que la formation traditionnelle pour un coût inférieur.

Élaborer une formation numérique coûte plus cher que préparer du matériel pour une classe et former des formateurs, en particulier si les méthodes utilisées sont multimédias ou hautement interactives. Toutefois, les coûts de diffusion d'un cours e-learning (y compris les frais pour les serveurs Web et l'assistance technique) sont considérablement plus faibles que ceux qu'impliquent l'utilisation de salles de classe, la rémunération de formateurs, le financement des voyages des participants et le temps de travail perdu pour assister aux sessions de formation. (Ghirardini, 2011)

En outre, le e-learning permet d'atteindre un public plus vaste, en intégrant des apprenants qui ont des difficultés à participer à des formations conventionnelles parce que :

- Ils sont géographiquement dispersés avec peu de temps et/ou de ressources pour voyager ;
- Ils sont occupés par leur travail ou des engagements familiaux qui ne leur permettent pas de suivre des cours à des dates précises avec un calendrier établi à l'avance ;

- Ils sont situés dans des zones de conflit ou post-conflit et sont limités dans leur mobilité pour des questions de sécurité ;
- En raison de croyances culturelles ou religieuses, ils ne peuvent pas participer à toutes les sessions en salle de classe;
- Ils font face à des difficultés de communication en temps réel (p. ex. des apprenants d'une langue étrangère ou très timides).

Les méthodes pédagogiques utilisées dans le cadre du e-learning sont efficaces car elles permettent d'allier pratique et feedback, de combiner des activités de collaboration avec un apprentissage individualisé, de personnaliser les parcours d'apprentissage en fonction des besoins des apprenants et de leur proposer des jeux didactiques et des simulations. En outre, tous les apprenants reçoivent la même qualité d'enseignement, puisque celle-ci ne dépend en aucun cas d'un formateur particulier. (Ghirardini, 2011)

1.2 Qualité du e-learning :

La qualité d'un cours e-learning est renforcée par :

- **Contenu centré sur l'apprenant** : le curriculum du cours numérique doit être précis et adapté aux besoins de l'apprenant ainsi qu'à ses fonctions et responsabilités professionnelles. Les compétences, les connaissances et les informations contenues dans le cours doivent viser cet objectif.
- **Granularité** : le contenu de la formation électronique doit être segmenté afin de faciliter l'assimilation des nouvelles connaissances et permettre de moduler le temps consacré à l'apprentissage.
- **Contenu stimulant** : les techniques et les méthodes d'enseignement doivent être utilisées de manière créative afin d'élaborer un cours stimulant et motivant pour l'apprenant.
- **Interactivité** : de fréquentes interactions avec l'apprenant sont nécessaires pour maintenir l'attention et encourager l'apprentissage.
- **Personnalisation** : les cours en auto-apprentissage doivent être personnalisables afin de refléter les besoins et les intérêts des apprenants ; dans les cours dirigés, le formateur, le tuteur et/ou le facilitateur doivent être capables de suivre les progrès et les performances des apprenants de manière individuelle. (Ghirardini, 2011)

1.3 Avantages de e-learning :

Quels sont les avantages et les inconvénients d'un dispositif de formation intégrant le e-learning ?

Que permet le e-learning par rapport à la formation traditionnelle ?

Une solution e-learning :

- Permet la formation • Juste à temps • : en effet, l'entreprise peut concevoir et réaliser des programmes de formation pour les utiliser à un moment bien précis, le moment où les enseignements sont réduits.
- Offre une formation souple en termes de lieu et de temps d'apprentissage : l'apprenant peut choisir le lieu et le moment de l'apprentissage.
- Le e-learning propose un large éventail de possibilités : l'apprenant peut se former dans une classe, un bureau, un centre de ressources ou à la maison. Il peut également choisir le moment de sa formation : en dehors du travail, sur le temps de travail. sur une période planifiée ou non. Juste avant d'utiliser les compétences couvertes par le module.
- Réduit les coûts de formation. L'entreprise qui choisit le e-learning réduit ses coûts de formation, en particulier les frais d'hébergement, de déplacement, de salaire des formateurs. Il faut bien sûr relativiser cette réduction des coûts en tenant compte de l'importance des frais de développement et de déploiement des dispositifs e-learning. (Prat, 2008)

1.4 Plate-forme LMS:

Un LMS *Learning Management System* ou une plate-forme d'enseignement à distance est un logiciel intégré basé sur deux points suivantes :

1. reprenant une large étendue de fonctionnalités typiques d'un enseignement basé sur le Web:

- outils de communication synchrone (chat, forum de discussion) et asynchrone (envoi d'e-mail, groupes de discussion),
- outils d'évaluation,
- possibilité de travail collaboratif,
- réalisation de contenu,

- analyse de traces de l'utilisateur,...

2. et permettant la gestion des espaces de cours, des utilisateurs et le contrôle des accès.

Cela implique l'intervention de différents types ou profils d'acteurs, incluant non seulement les enseignants, leurs assistants et les étudiants, mais aussi les administrateurs - administrateur du système et administrateur de l'outil. L'administrateur système est responsable du hardware et de l'installation du produit et de ses mises à jour tandis que l'administrateur de la plate-forme est responsable des différents paramétrages, de la gestion de la base de données des cours et des utilisateurs, et de la gestion des accès.

(lecompte, 2007)

1.4.1 Utilisateurs de la plate-forme LMS :

Il y a trois principaux types d'utilisateurs accédant à la plate-forme : l'administrateur, les apprenants et les enseignants.

Chaque enseignant est responsable d'un ou plusieurs modules d'enseignement. La plate-forme permet à l'enseignant de structurer les modules en différentes briques de contenus de cours. Il peut ajouter un contenu de cours en téléchargeant sur le serveur de la plate-forme n'importe quel type de fichier. Il peut mettre à jour et supprimer chaque contenu.

Un apprenant peut consulter ces différentes briques de cours et éventuellement les télécharger pour une lecture ultérieure hors connexion.

L'administrateur, de son côté, assure l'installation et la maintenance du système, gère les droits d'accès, crée des utilisateurs, les intitulés de module etc...

Il y a 4 types de contenus de cours : exercice, ressource, test et concept. (Kouninef, Djelti, & Rerbal)

1.4.2 Objectifs de la plate-forme LMS

La plateforme LMS permet la mise à disposition de ressources pédagogiques (Textes, Exercices, Tests, Evaluations, etc.) et permet la communication entre les formateurs et les apprenants (forum, chat, dépôts de fichiers). Elle permet de faire la gestion administrative des formations (inscriptions, annuaires, bulletins de notes, etc.).

C'est une plate-forme simple d'utilisation tant du point de vue « enseignant créateur de cours » que du point de vue « apprenant » ou du point de vue « administrateur » de la formation.

L'enseignant doit pouvoir créer son cours sur sa machine et le basculer facilement sur LMS peu importe le format de son cours. Le seul effort demandé est le découpage des modules en différentes briques (concepts, ressources, exercices, tests) afin de permettre aux étudiants d'aborder par morceaux les travaux à effectuer pour chaque module. La plateforme LMS permet de mettre à disposition tout type de contenus de cours. Il y a une grande liberté dans les formats des fichiers de cours. Cette liberté est appréciée des enseignants qui veulent chacun avoir leur propre type de support (Slides PowerPoint, documents Word, html, pdf, etc.) (Kouninef, Djelti, & Rerbal)

1.4.3 Quelques Systèmes de Gestion des Cours

Nous donnons quelques exemples de Système de gestion des cours :

Claroline (Plateforme d'origine belge, initialement créée par l'Université de Louvain sous licence Open Source, puis étendue grâce à l'implication de nombreux réseaux de développeurs et de formateurs d'origine internationale).

Fonctionnalités de la plateforme : (Gestionnaire de cours, Agendas, Annonces, Gestion des documents et des liens, Création d'exercices (QCM, Correspondance, texte à trous), Tracking, Gestion des parcours pédagogiques, Gère les contenus SCORM*, Forum, Gestion des groupes (partage de fichiers, forum). Personnalisation de la plateforme (HomePage, style CSS).

(Kouninef, Djelti, & Rerbal)

Ganesh (Plate-forme développée par la société Anéma Formation sous licence Open Source).

Fonctionnalités de la plate-forme (Forum, Messagerie intégrée, Chat, Zone de dépôt de documents, Gère les contenus standardisés SCORM, Tracking et évaluation, Suivi pédagogique, Administration Web, Multilingue). (Kouninef, Djelti, & Rerbal)

Moodle (*Modular Object-Oriented Dynamic Learning Environment*) Plate-forme, initiée en 1990 à Perth en Australie (Curtin University) par Martin Dougiamas qui continue à conduire le projet. Moodle est une plate-forme d'apprentissage en ligne (e-learning en anglais) sous

licence open source servant à créer des communautés d'apprenants autour de contenus et d'activités pédagogiques.

Fonctionnalités de la plateforme (Zone de dépôt de documents, Multilingue, Administration Web, Suivi pédagogique, Tracking et évaluation, Messagerie, Générateur HTML Wysiwyg, Gère les contenus standardisés SCORM, Sondage, Forum, Chat, Modules additionnels (Plugins). (Kouninef, Djelti, & Rerbal)

1.5 Définition du m-Learning:

M-Learning est un terme utilisé pour désigner toute manière de délivrer des cours en utilisant des outils mobiles tels que les Tablet PC1, les Pocket PC, les téléphones portables ou les PDA.

Dans la définition du m-Learning, il faut privilégier la notion de mobilité aux dépens de la notion de fonctionnalités : la mobilité signifie le transport facile, la petite taille, etc,

(Kouninef, Djelti, & Rerbal)

Le tableau suivant montre les appareils utilisés dans le e-learning et M-learning :

Fonctionnalités		Mobilité		
Ordinateurs de bureau	Ordinateurs portables	PDA Palms Handhelds	Smartphones	Téléphones portables
← E-learning →		← M-Learning →		

Tableau 0.1 : Mobilité et fonctionnalités dans la définition du m-Learning

2. MOODLE :

Moodle est une plateforme d'apprentissage en ligne (en anglais : *Learning Management System*) sous licence libre servant à créer des communautés s'instruisant autour de contenus et d'activités pédagogiques. Le mot « Moodle » est l'abréviation de *Modular Object-Oriented Dynamic Learning Environment* : « Environnement orienté objet d'apprentissage dynamique modulaire ». À un système de gestion de contenu (SGC), Moodle ajoute quelques fonctions pédagogiques et communicatives créant un environnement d'apprentissage en ligne : cette application permet de créer, via le réseau, des interactions entre pédagogues, apprenants et ressources pédagogiques.

De tels systèmes de e-formation sont aussi nommés « formation ouverte et à distance ». Moodle permet aux enseignants d'organiser leurs ressources pédagogiques et aux étudiants d'y accéder à travers un simple navigateur web.

Moodle est un logiciel libre répandu, utilisé dans de nombreuses universités et organismes de formation, qui dispose d'une communauté active sur internet, dans le monde entier.

(Jonathan & Michael, 2010)

2.1 État de développement :

Moodle est en constante évolution. La première version est sortie au mois d'août 2002. La version 1.9 de l'application est lancée en 2008, alors que l'actuelle version (2.6) est disponible depuis novembre 2013.

Au mois d'août 2013, plus de 86 000 sites dans 237 pays ont enregistré leur implantation. La plateforme a été traduite dans près de 120 langues. À part Moodle.org lui-même, l'implantation la plus importante comporte exactement 19 356 cours avec 5 000 000 d'utilisateurs.

La plateforme elle-même jouit d'une bonne réputation, mais son usage demande une attitude active qui doit s'instaurer avec le temps : « Depuis quelques années, les plates-formes d'apprentissage en ligne sous licence « open source » telles que Moodle semblaient très prometteuses à cause de leur capacité de favoriser la collaboration entre les apprenants. Toutefois, nous constatons que Moodle est le plus souvent utilisé pour annexer des documents .pdf, .doc ou .ppt ». (Jonathan & Michael, 2010)

2.2 Architecture de Moodle :

Comme illustré par la figure 1.1 le premier rôle est celui de **l'enseignant** (professeur sur la figure) qui est chargé de concevoir le matériel pédagogique des cours. Le **concepteur informatique**, ou **l'enseignant** lui-même, réalise alors les médias (texte, image, vidéo, etc.).

L'enseignant crée par ailleurs des **parcours pédagogiques** types et individualisés de son enseignement. Le **tuteur** effectue un suivi du travail des **apprenants** grâce aux retours (évaluation, temps de parcours, etc.), ainsi qu'une assistance dans l'apprentissage de ces derniers. Il peut éventuellement réguler leurs parcours pédagogiques.

L'apprenant consulte en ligne ou télécharge les contenus pédagogiques qui lui sont recommandés, organise son travail, effectue des exercices, s'auto-évalue et transmet des travaux au tuteur qui les évalue. (Wikipedia)

La figure 1.1 présente l'architecture de Moodle.

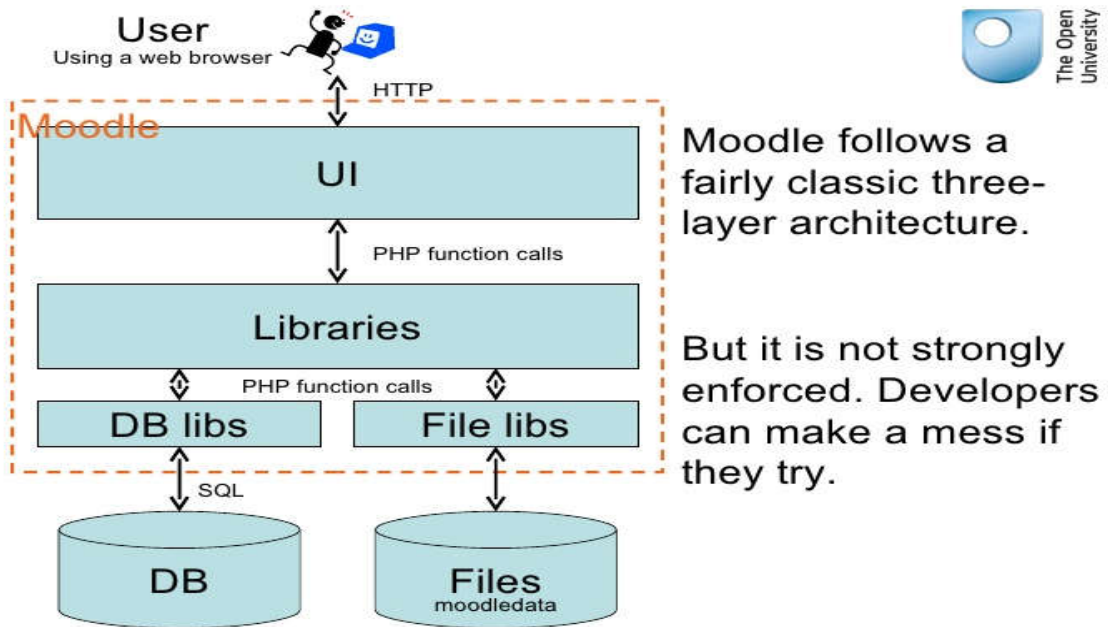


Figure 0.1 : Exemple d'architecture de Moodle

La figure 1.2 est une autre présentation L'interaction du client avec Moodle :

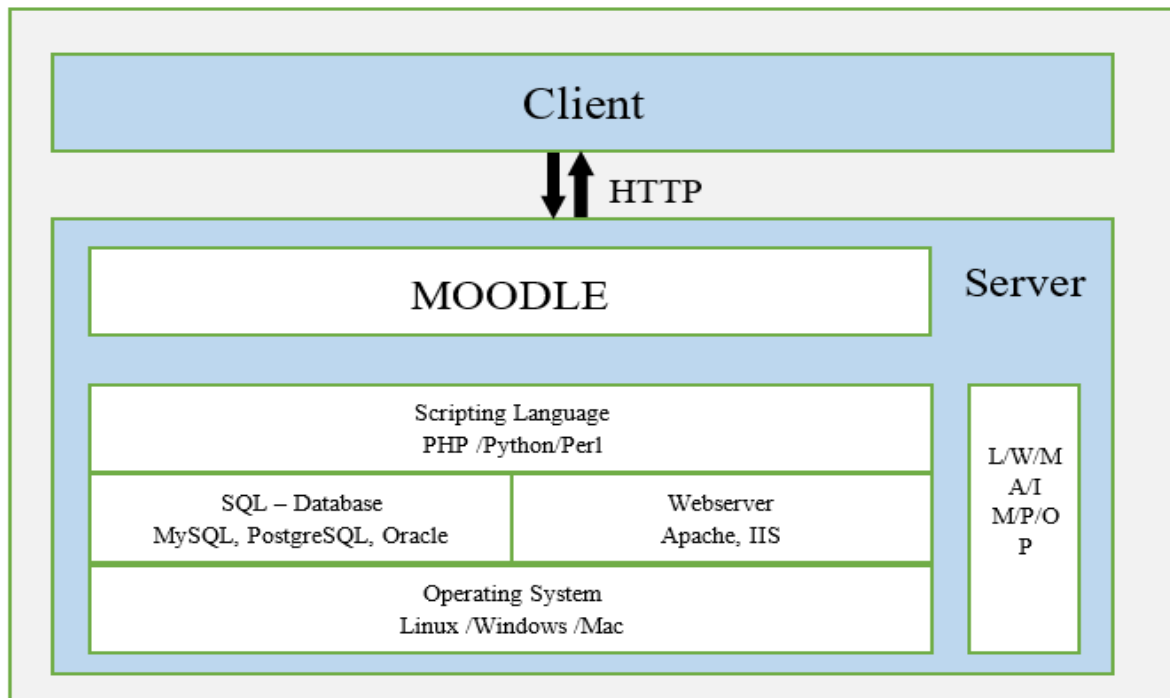


Figure 0.2 : Exemple d'architecture de Moodle

2.3 Moodle code :

Puisque Moodle est écrit en langage PHP , Moodle est stocké sous forme de fichiers code source sur le serveur Web. Quand un fichier particulier est demandé sur le serveur, le PHP interpréteur analyse le code à la volée, et la sortie résultante est envoyée via un logiciel de serveur web. Comme mentionné précédemment, le «M» dans Moodle est synonyme de "modulaire", et la structure de répertoire qui reflète.

Chaque dossier de haut niveau représente une importante composante de Moodle. La plupart des principaux composants prennent en charge les modules plug-in. Chaque plugin a son propre dossier dans le dossier du composant.

Dans certains cas, les modules auront également un soutien pour les plugins supplémentaires. Un exemple de ceci est l'activité quiz module, qui prend en charge les types de questions modulaires. Du point de vue de l'utilisateur final, les modules sont installés en copiant le module dans l'emplacement du dossier approprié sur le serveur. Moodle détecte le nouveau module la prochaine fois qu'un administrateur se connecte au système, localise le code SQL du module, court, et finalement affiche Les résultats. Google Traduction pour les entreprises (Jonathan & Michael, 2010)

2.4 Server Web de Moodle :

Moodle fonctionne généralement avec tout serveur Web qui prend en charge l'exécution des versions appropriées de PHP.

Dans la pratique, le serveur web le plus utilisé est Apache, qui est disponible pour la plupart des systèmes d'exploitation. Internet Information Services (IIS) est un autre serveur Web populaire pour l'hébergement Moodle. Il y a aussi un nombre croissant plaide pour des *lighthttpd* comme un serveur web, et cela est rapporté dans le Moodle forums communautaires (<http://moodle.org/forums/>) que le travail bien avec Moodle.

Lighthttpd a un nombre croissant de défenseurs en raison de sa conception de la mémoire faible. C'est extrêmement populaire pour son utilisation avec des environnements virtuels (*VMware, Xen, et Amazon Elastic Compute Cloud*), où l'empreinte mémoire est plus préoccupant Moodle comme une application web a soutien et de développement des contraintes qui sont différente de la norme.

Ceci est dû à la façon dont Moodle est utilisé. Un utilisateur interagissant avec Moodle aura un montant plus élevé que la normale de clics, et Moodle génère de nombreuses requêtes SQL comme il construit une page. Moodle est très effiient au ce qu'il fait. Cependant, ce qu'il fait est assez complexe. Cela signifie, en tant que développeurs, nous devons être conscients du type d'architectures que nos modifications seront probablement utilisées à l'intérieur. Cela signifie aussi que nous devons être conscients des implications de performance de notre codage. (Wikipedia)

La figure suivant illustre plusieurs configurations couramment utilisés pour Moodle dans un environnement de production. Moodle est déployé dans un large éventail de settings-partout à partir d'un seul enseignant en cours d'exécution sur son bureau tous le chemin jusqu'à multi-machines grappes de serveurs de haute performance.



Figure 0.3 : WAMP Server

Conclusion

Dans ce chapitre, nous avons vu la définition de e-learning et sa position dans le domaine de formation et l'enseignement à distance. Nous avons également présenté comment s'est développé le e-learning jusqu'à arriver à une nouvelle technologie qui est le M-learning. Dans le chapitre suivant on va voir les systèmes intégrés dans le monde de la nouvelle génération des téléphones portables.

Chapitre 2 : Les systèmes d'exploitation mobiles

Introduction

Dans cette section, nous aborderons quelques systèmes d'exploitation existants, du BlackBerry à Android, en passant par Symbian, Windows Mobile. Cela nous permettra d'avoir une idée assez générale sur les systèmes d'exploitation tournants sur mobiles, et d'essayer de donner le positionnement du système Android dans son environnement.

1. Système d'Exploitation(SE) :

Un système d'exploitation est un logiciel destiné à faciliter l'utilisation d'un ordinateur (je sais que cette assertion sera pour certains lecteurs une véritable provocation). L'ordinateur constitue le matériel composé de fils, circuits, etc., inutilisable sans le logiciel constitué de programmes.

Les ordinateurs actuels sont suffisamment complexes pour qu'il soit inconcevable de les utiliser sans la médiation d'un système d'exploitation, mais ce n'a pas toujours été le cas, et d'ailleurs certains petits ordinateurs qui vivent cachés à bord des fours à micro-ondes, des ascenseurs, des baladeurs ou des avions en sont parfois encore dépourvus. (Bloch, 2003)

Ces quelques systèmes d'exploitation mobiles :

1.1. BlackBerry :

Système d'exploitation propriétaire, il a été créé par la société Canadienne Research In Motion (RIM) pour le businessman initialement.

Marchent exclusivement sur les appareils du BlackBerry, il offre les applications tels que la messagerie électronique, les messages PIN, les SMS, Messenger et bien plus.

C'est un système multitâche, car son protocole Mobile Information Device Profile (MIDP 2.0) permet une synchronisation complète avec les messageries d'entreprise. D'où initialement conçu pour le businessman.

Il devient de plus en plus populaire dans le monde, car il offre en plus les fonctionnalités pour le multimédia, la carte Media, etc.

Malgré tout cela, ces caractéristiques de base font à ce qu'il soit optimal pour les grands déploiements (c'est-à-dire adapté pour les entreprises). Donc important pour des choses simples. (NARCISSE, 2011)

1.2. PalmOS :

Palm OS a été développé par la société Palm. En général, Palm OS est logique, intuitif et simple, plus en détail, Palm OS est facile d'utilisation et simple à apprendre. Il optimise les étapes pour naviguer entre les écrans et pour choisir les applications. Par exemple, pour lancer un programme, appuyez sur son icône. Quand on passe à une autre application, cette application se termine.

1.3. Symbian :

C'est un SE pour smartphone développé initialement par la société Symbian, jusqu'au rachat des actions de la société à 100% par la société Nokia. Aujourd'hui il est fédéré autour d'un consortium entre les différents constructeurs Psion, Nokia, Motorola, Ericson, Siemens, Panasonic. (NARCISSE, 2011)

C'est une plate forme extrêmement puissante, offrant les fonctionnalités essentielles du système, ainsi que les API communes. Il est aussi très flexible, car chaque constructeur peut ajouter ses technologies ou enlever des fonctionnalités à la plateforme.

Depuis février 2010, le code source est officiellement téléchargeable, concrétisant ainsi l'annonce faite en octobre 2009 par Nokia de mettre sous Symbian sous licence Open Source. Cela afin de devenir un compétiteur sérieux des autres plateformes.

Malgré tous ces avantages, elle est actuellement en cours d'abandon par Nokia (action majoritaire du projet), au profit de Windows phone 7. Ceci depuis février 2011.

(NARCISSE, 2011)

1.4. Apple I OS :

C'est un système développé par Apple pour les produits Iphone, Ipod et Ipad. Ce système est dérivé de Mac OS X, dont il partage les fondations. C'est pour cela que le Kit de

développement (SDK) n'est disponible que pour Mac OS X. SDK publié depuis le 6 mars 2008, permettant de créer des applications pouvant tourner sur iOS.

Comme le BlackBerry, iOS depuis sa version 4.0 présentée en avril 2010 supporte le multitâche. En plus des fonctionnalités telles que le copier-coller, le système « Push », le dictaphone, le spotlight, l'interface Multi-touch et toutes autres fonctionnalités. Cette plateforme possède aussi une boîte mail unifiée, et une interface utilisateur personnalisable depuis sa version 4.0.

C'est une plateforme très robuste, mais possède de nombreuses limites imposées par le constructeur. On peut citer entre autre :

- Comme BlackBerry, c'est un système propriétaire ;
- Système fermé (pour bénéficier de certaines fonctionnalités, il est nécessaire de Hacker le système) ;
- La dernière version possède des fonctionnalités qui sont disponibles depuis sur d'autres plateformes. (NARCISSE, 2011)

1.5. Windows Mobile :

Comme iOS et BlackBerry, c'est un système propriétaire conçu par la société Microsoft pour les appareils mobiles tels que les Smartphones ou Pocket PC.

L'un des avantages principaux est qu'il possède la capacité de faire fonctionner des logiciels Microsoft sur son téléphone tels que Microsoft Office ou Windows Live.

C'est un concurrent sérieux de BlackBerry, car il permet de recevoir des courriels en temps réel. Mais comme les autres plateformes, il souffre des mêmes inconvénients dus au fait que celui-ci soit propriétaire. (NARCISSE, 2011)

2. Android :

Android a été développé par une startup de même nom, racheté par la suite par la société Google. Mais depuis sa première sortie en 2007, il est développé au sein de l'Open Handset Alliance, qui regroupe une trentaine de partenaires. Depuis 2008, c'est une plateforme en code source ouvert.

Android est basé sur un noyau linux (ce qui lui confère un très grand avantage sur la sécurité et la puissance des systèmes linux), mais comporte une plateforme développée en Java, dans lequel les applications sont exécutées via un interpréteur JIT. C'est un système gratuit et complètement ouvert, car le code source et les APIs sont ouverts. Ainsi les développeurs ont la permission d'intégrer, d'agrandir et de remplacer les composants existants. C'est aussi un système très flexible. C'est-à-dire qu'un constructeur peut adapter les applications aux besoins du client ou remplacer entièrement. C'est l'un des avantages qui a permis aux constructeurs de maîtriser leur produit, ainsi d'apporter une valeur ajoutée à leur téléphone, pour ne pas être un simple assembleur comme pour les ordinateurs.

En termes d'application, Android a été conçu pour intégrer plusieurs services Google comme Gmail, Google Maps, Google Agenda, Google Talk, YouTube. Avec un accent particulier sur la géo localisation avec Google Latitude et la météo.

Pour développer une application sur Android, Google a mis à la disposition de la communauté Open Source un SDK, qui offre en complément les APIs.

En fin des années 2010, Android était devenu un concurrent sérieux face aux autres plateformes, car le nombre de mobile tournant sur celui-ci était en grand augmentation.

(NARCISSE, 2011)

2.1. Architecture d'android :

Les applications Android sont développées en JAVA. C'est ainsi qu'Android dispose d'un ensemble de bibliothèques du JDK¹, en plus des fonctionnalités supplémentaires qui lui sont propres (c'est le cas par exemple de la gestion de l'interface graphique). Etant en JAVA, on aurait pu penser que ces applications s'exécutent sur la JVM (machine virtuelle java de Sun). Ce qui n'est pas le cas, car ils s'exécutent sur la "Dalvik VM".

Il s'agit de la machine virtuelle développée par Google pour Android et adaptée aux systèmes restreints en mémoire et puissance de processeur. (NARCISSE, 2011)

La figure 2.1 présente l'architecture ou différentes couches du système android :

¹ *Java development kit* : un outil de développement sous java.

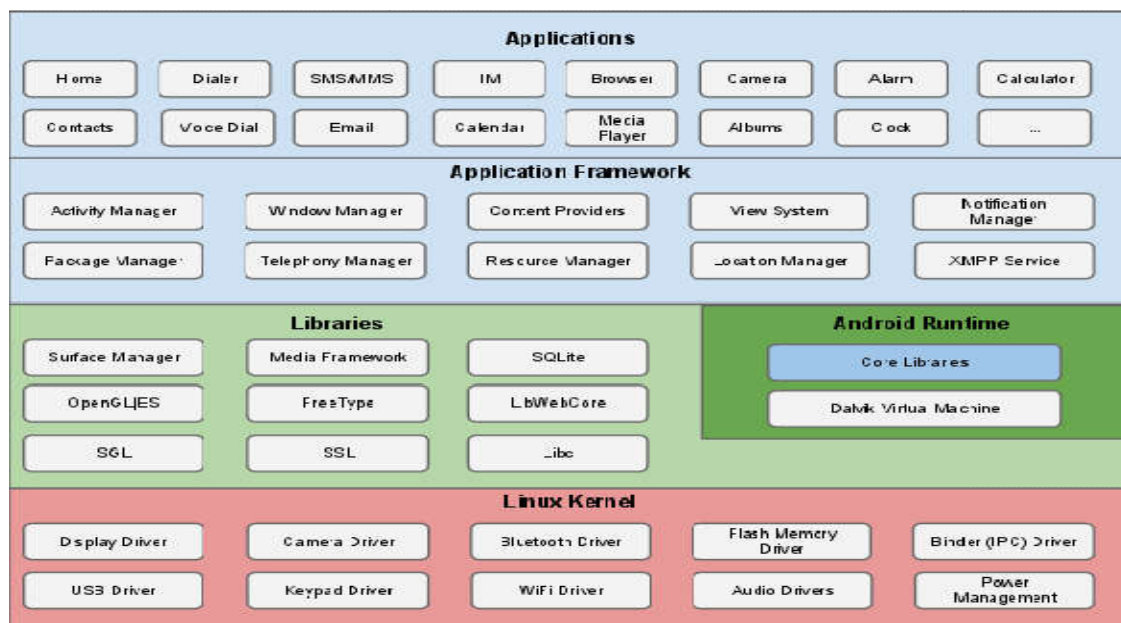


Figure 0.1 : Différentes couches du système.

En observant la figure ci-dessus, on constate qu'Android est basé sur le noyau linux 2.6.XX. Alors il bénéficie de plusieurs avantages dont dispose le système linux, tels qu'une bonne gestion de la mémoire et des processus, modèle de sécurité, soutien de bibliothèque partagée, etc.

Au dessus, on trouve une collection de plusieurs bibliothèques C/C++, utilisées par un certain nombre de composant du système Android.

Au dessus de ceux-ci, on trouve l'Android Runtime, siège de la Dalvik VM et les bibliothèques cœurs du Framework.

Plus au dessus, on a la couche "Application Framework", couche permettant au développeur de créer des applications.

Enfin on trouve les applications au dessus du Framework. Ce sont les applications qui marchent sous la plateforme Android. Toutes développées en Java.

2.2. Les composants d'une application :

Les applications Android sont composées de 4 principaux :

- **Activity** : Gestion de la vie des applications, car une application se compose de plusieurs écrans. Et chaque écran peut être réalisé par une activité, implémentée sous la forme d'une classe qui hérite de la classe *Activity*.
- **Services** : Les services n'ont pas d'interface graphique et tournent en tâche de fond (en arrière-plan). C'est-à-dire qu'une application s'exécute quand une autre application est en train de s'exécuter comme les services de lecture de musique.
- **Broadcast receives** : Il écoute et réagit aux annonces *broadcast* (réactions sur les événements extérieurs). Par exemple appel entrant, changement de fuseau horaire.
- **Content providers** : Gestion du partage de données entre applications.

En résumé Android permet le partage de composant entre applications ainsi que de gérer leur cycle de vie. Cycle de vie des activités plus précisément. Ce cycle de vie est exprimé par la figure ci-dessous.

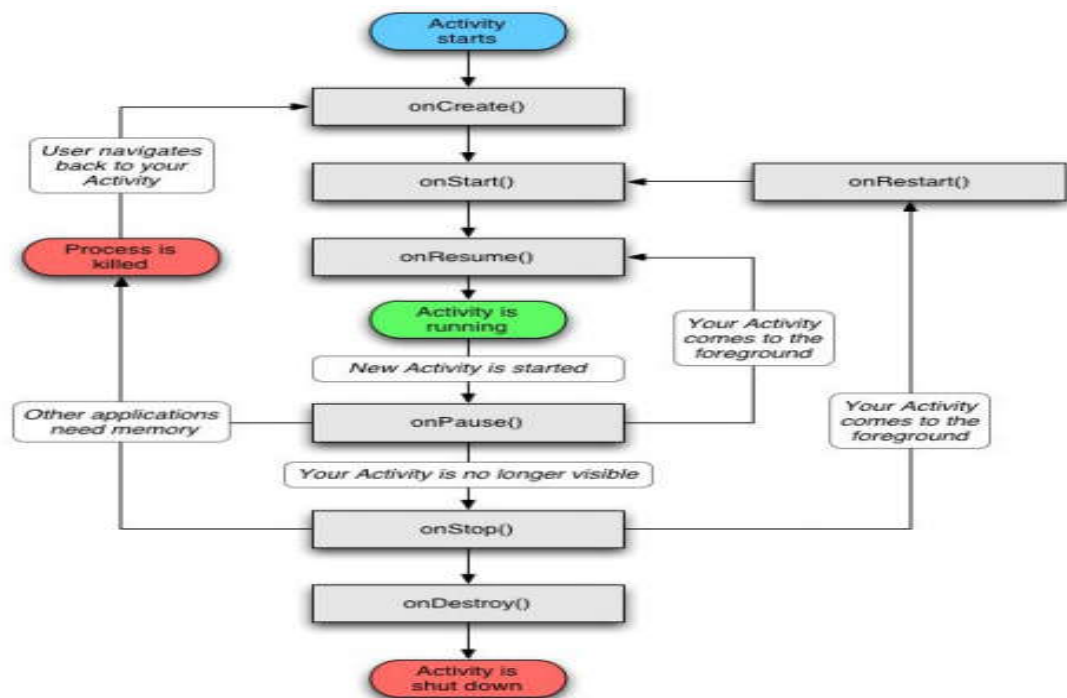


Figure 0.2 : Cycle de vie d'une Activité sous Android

Conclusion

Dans ce chapitre, nous avons présenté le système Android et d'autres systèmes d'exploitation. Nous avons fait ressortir leurs avantages et leurs inconvénients. Le chapitre suivant traitera du processus du développement de notre projet.

Chapitre 3 : Conception et modélisation

Introduction :

Dans ce chapitre nous présentons une étude scientifique concernant le projet à réaliser. Pour ce faire, nous appliquerons une démarche d'ingénierie logicielle qui illustre les différentes étapes du processus adopté appelé 2TUP dérivé de UP qui s'appuie sur une conception UML. Nous donnerons les besoins fonctionnels et les besoins techniques et comment les fusionner pour donner une conception détaillée du système attendu pour le Bloc ajouté dans la plateforme Moodle et l'application Java.

1. Méthodes agiles :

Les méthodes de développement dites « **méthodes agiles** » (en anglais *Agile Modeling*, noté *AG*) visent à réduire le cycle de vie du logiciel (donc accélérer son développement) en développant une version minimale, puis en intégrant les fonctionnalités par un processus itératif basé sur une écoute client et des tests tout au long du cycle de développement.

2. Le Processus (UP) - Unified Process:

La méthode du Processus Unifié (UP pour Unified Process) est un processus de développement itératif et incrémental, ce qui signifie que le projet est découpé en phases très courtes à l'issue de chacune desquelles une nouvelle version incrémentée est livrée.

Il s'agit d'une démarche s'appuyant sur la modélisation UML pour la description de l'architecture du logiciel (fonctionnelle, logicielle et physique) et la mise au point de cas d'utilisation permettant de décrire les besoins et exigences des utilisateurs.

3. La méthode 2TUP

Notre projet est basé sur un processus de développement bien défini qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final.

Ce processus se base lui-même sur le **Processus Unifié (Unified Process)** qui est devenu un standard général réunissant les meilleures pratiques de développement. Cette méthode ne se

base aucunement sur un processus *linéaire* mais bien, sur un développement *itératif et incrémental*. Nous allons d'abord définir les différents concepts qui vont être utilisés dans ce document.

2TUP signifie « 2 Track Unified Process ». C'est un processus qui répond aux caractéristiques du **Processus Unifié**. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées au système d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des « chemins fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.

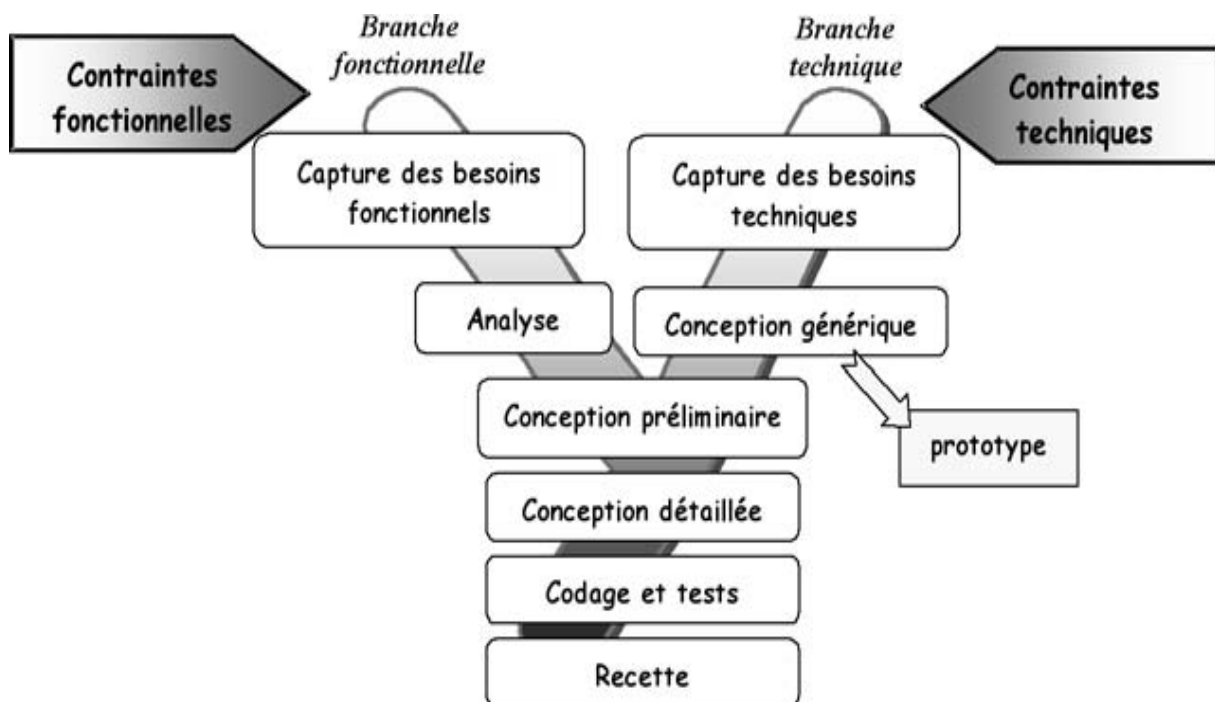


Figure 0.1 : le processus 2TUP

« *Unified Modeling Language* » : UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue.

4. Conception du logiciel :

4.1 Etude Préliminaire :

4.1.1 Présentation du projet à réaliser :

Dans cette partie on va parler de notre projet qui consiste à construire un bloc dans Moodle qui n'existe pas sur ces versions. Ce bloc permet aux étudiants de voir leurs notes et leurs moyennes sans utiliser l'affichage classique sur papier et au lieu de déplacer les étudiants à l'université. Pour ce faire, on a proposé deux solutions :

1. Utiliser ce bloc sous Moodle à partir de :
 - a) Un micro-ordinateur, micro-portable à l'aide d'un navigateur comme mozilla firefox, Google chrome ...
 - b) Un téléphone portable intelligent doté pas le système Android à l'aide d'un navigateur approprié des téléphones.
2. Créer une application sous Java installée sur les téléphones portables intelligents qui accède et affiche seulement les notes, moyens, on utilisant les web services.

4.1.2 Recueil des besoins fonctionnels :

Nous avons effectué des recherches pour identifier au mieux les besoins de l'application, et ceci afin de répondre aux attentes des utilisateurs « étudiants ». Nous sommes allés chercher les informations au sein de l'administration de la Faculté. Cette phase correspond à une recherche sur le terrain pour bien définir le cadre de notre système. Ou ont à obtenir les informations concernant le mode de fonctionnement du système LMD :

Une université se compose de plusieurs départements (informatique, physique, maths, chimie ...), dont chacun est dirigé par un chef de département.

Chaque département voit son parcours de formation structuré en paliers :

- Le premier palier doit être composé au plus de 2 semestres.
- Le deuxième est une étape de spécialisation.

Les parcours sont organisés en unités d'enseignements(UE) articulées entre elles. Une UE est constituée d'une ou de plusieurs « matières » dispensés par toute forme d'enseignements (Cours, Travaux Dirigés, Travaux Pratiques, projets).

L'enseignant peut donner des cours à la classe comme il peut les charger dans l'environnement Moodle.

Après avoir fait passer les examens aux étudiants, les enseignants peuvent remettre les notes à l'administration « scolarité ».

Les étudiants peuvent voir leurs notes sur un affichage ~~sur~~ papier.

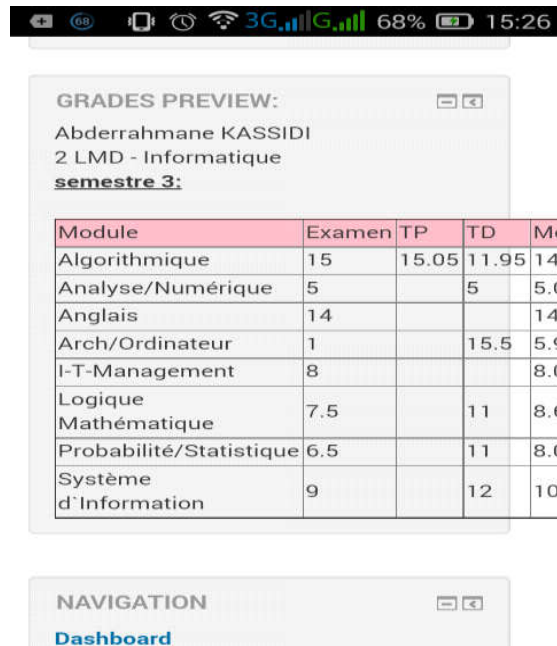
Les étudiants utilisent Moodle comme un moyen de télécharger leurs cours.

4.1.3 Recueil des besoins techniques :

Nous avons listé avant les besoins de l'application, et nous sommes allés chercher les informations au sein de l'administration de la Faculté pour déterminer les besoins techniques.

Le point de vue matériel met en évidence les contraintes d'exploitation technique suivantes :

- Il existe une base de données au niveau de la scolarité comprene 36 tables créées en Access. On a choisi 5 tables qui correspondent à notre projet, à savoir :
 - Etudiant
 - Résultats
 - Options
 - Année_cycle
 - Cours
- Pour que le « *block* » Moodle fonctionne avec la BDD, il doit nécessairement convertir cette BDD en MySQL.
- La mise à jour de la base de données doit être synchronisée au niveau de la BDD de Moodle.
- L'export des informations sur Internet impose d'isoler le serveur Web du réseau.
- Comme le style d'architecture 3-tiers définit un moyen logiciel intermédiaire entre les applications clientes et les serveurs de base de données, il doit fournir au système les moyens techniques qui lui permettent de garantir des temps de réponse constants, quelque soit le nombre d'utilisateurs connectés.
- Les Smartphones possèdent des ressources matérielles limitées (taille de l'écran, mémoire RAM).



GRADES PREVIEW: Abderrahmane KASSIDI
2 LMD - Informatique
semestre 3:

Module	Examen	TP	TD	Mo
Algorithmique	15	15.05	11.95	14
Analyse/Numérique	5		5	5.0
Anglais	14			14
Arch/Ordinateur	1		15.5	5.9
I-T-Management	8			8.0
Logique				
Mathématique	7.5		11	8.6
Probabilité/Statistique	6.5		11	8.0
Système d'Information	9		12	10

NAVIGATION
Dashboard

Figure 0.2 : capture de navigation a partir un browser sur android

4.1.4 Choix techniques :

Voici les choix techniques qui ont été adoptés pour le projet :

- La modélisation avec UML.
- Adoption d'une architecture à 3 tiers.
- Utilisation du langage PHP.
- Utilisation du langage Java.
- Utilisation d'une base de données qui existe déjà dans le secrétariat .
- Utilisation de StarUML Comme un langage de modélisation.

4.1.5 Identification des acteurs

Les acteurs du système identifiés dans un premier temps sont :

Etudiant : Un étudiant peut consulter ses relevés de notes.

Scolarité : la scolarité introduit les notes des étudiants.

Enseignant : l'enseignant affecte les notes des étudiants de son module.

Administrateur : crée les profils utilisateurs et attribue les droits d'accès de Moodle.

4.1.6 Identification des messages

Définition : un *message* représente la spécification d'une communication unidirectionnelle entre les objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur.

Le système reçoit les messages suivants :

La liste des étudiants inscrits dans une année de cycle.

La liste des options « code filiere»

Les résultats des étudiants « examen, TP, TD, et Moyenne module »

Liste des cours

Le système émet les messages suivant :

Les notes des étudiants.

4.1.7 Modélisation du contexte

A partir des informations obtenues lors des deux précédentes étapes, nous allons modéliser le contexte de notre application mentionné dans le tableau suivant :

Utilisateurs finaux	Description des besoins fonctionnels
Etudiant	L'application doit permettre à l'étudiant de : <ul style="list-style-type: none">• S'authentifier• Consulter son relevé de notes
Administrateur (programmeur)	L'application doit permettre à l'étudiant de : <ul style="list-style-type: none">• S'authentifier• Créer les profils utilisateurs• Donner des droits d'accès• Mise à jour de la Base de données• Installe et assure la maintenance du système

Tableau 0.1 : Utilisateurs du système.

5. CAPTURE DES BESOINS FONCTIONNELS

5.1 Déterminer les cas d'utilisations

Utilisation d'outils de génération de diagrammes UML : Tout au long du projet, nous sommes passés par **StarUML**, un outil qui génère les diagrammes UML. Nous allons faire une présentation rapide de cet outil.

StarUML version 5.02.1570 : est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL².

StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0. (wikipédia)

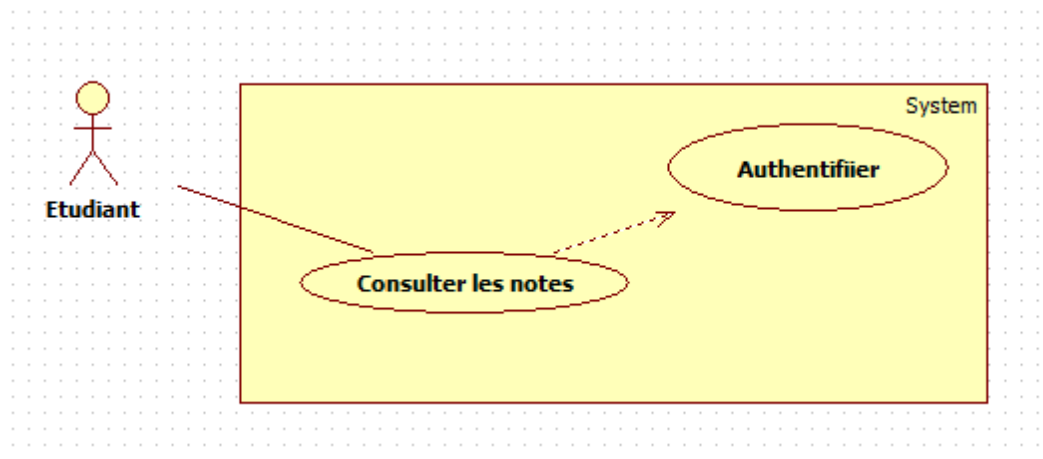


Figure 0.3 : diagramme de cas d'utilisation de la consultation des notes

6. CONCEPTION

La *conception préliminaire* est l'étape où s'effectue la fusion des études fonctionnelles et techniques.

La *conception détaillée* qui vient juste après est une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. Le modèle logique y est particulièrement important dans la mesure où c'est dans cette étape qu'on génère le plus grand nombre d'informations. Il est ainsi possible de confier les catégories à des

² La licence publique générale GNU, ou *GNU General Public License*

personnes différentes, qui pourront travailler indépendamment les unes des autres. Les concepteurs dans cette phase construisent les *classes*, les *vue d'IHM*, les *interfaces*, les *tables* et les *méthodes* qui vont donner une image « prête à coder » de la solution.

6.1 Diagramme de classes :

Cette phase va préparer la *modélisation orientée objet* en aidant à trouver les classes principales du futur modèle statique d'analyse. Donc nous avons obtenu le diagramme suivant :

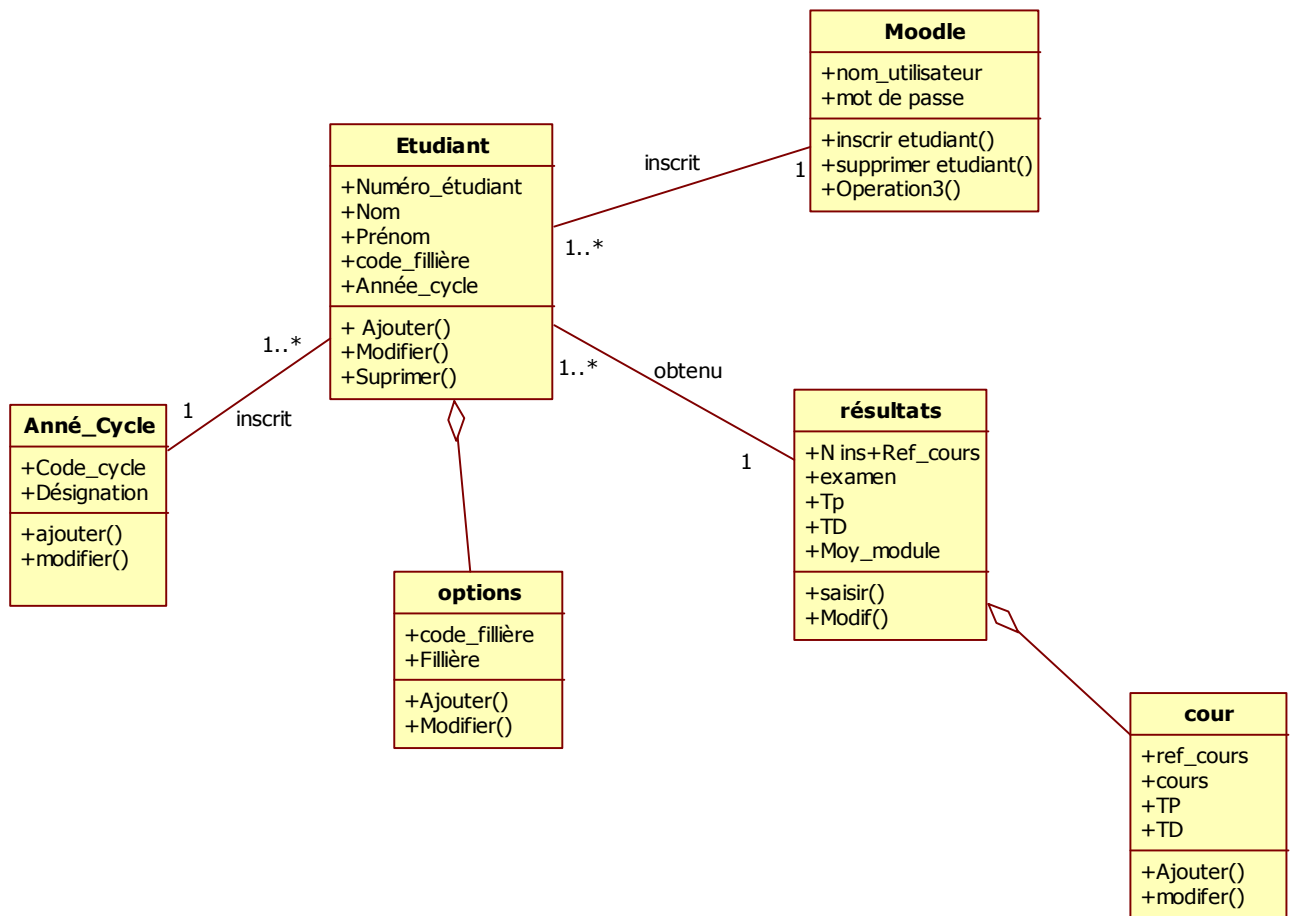


Figure 0.4 : Diagramme de classes (Moodle)

6.2 Diagramme de séquence :

L'objectif du **diagramme de séquence** est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.

Etudiant :

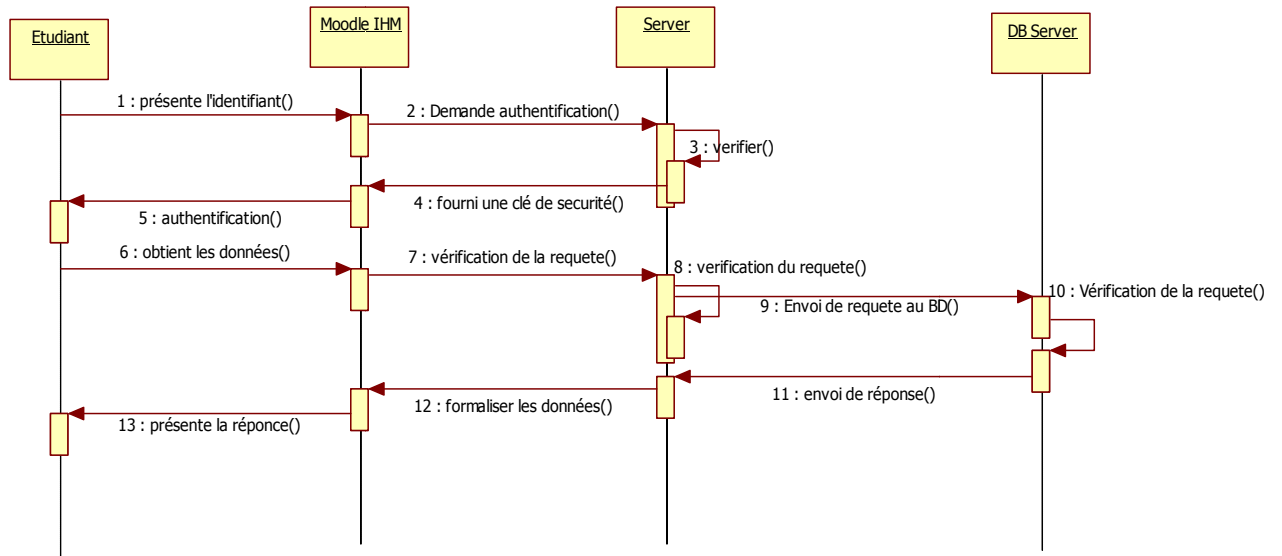


Figure 0.5 : Diagramme de séquence (obtenir de données).

6.3 Architecture de l'application

Dans les deux applications nous avons utilisé l'architecture 3 tiers montrée dans la figure 3.4 et 3.5 suivante :

- Architecture 03 tiers pour L'utilisation de Moodle par un micro-portable via l'internet :

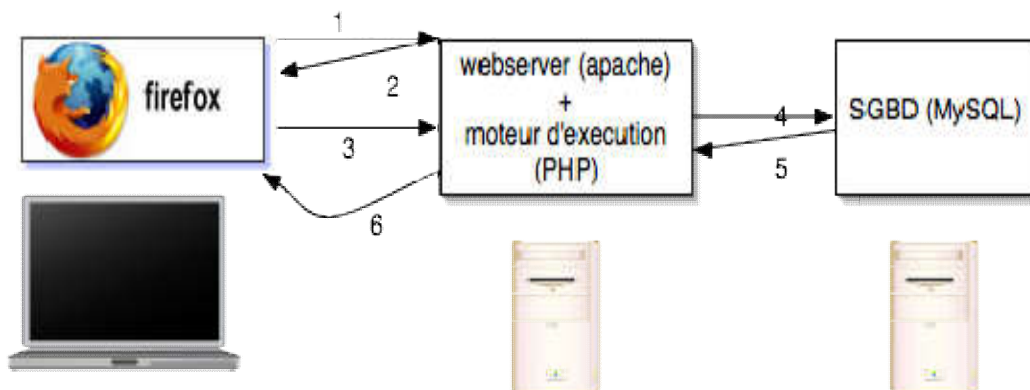


Figure 0.6: architecture 03 tiers pour Moodle

- Architecture 03 tiers pour L'utilisation de l'application sous Android par un Smartphone via l'internet.

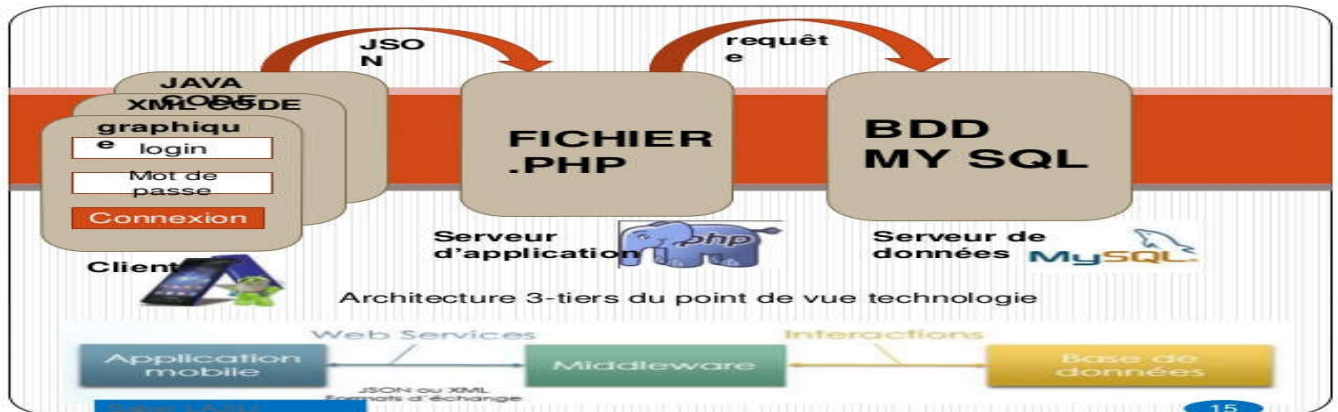


Figure 0.7: architecture 03 tiers pour application Android

7. CODAGE :

Le choix du langage s'est porté vers **Java**, pour l'application Adroid et le langage PHP et MySQL pour le Block sur Moodle.

7.1 Structure générale de l'application :

L'application est découpée en 3 couches distinctes, *Présentation*, *Métier* et *DAO*.

- La couche « *Métier* » est la logique métier de l'application, elle est le cœur et c'est elle qui définit toutes les règles régissant au fonctionnement de l'application.
- La couche « *Présentation* » est chargée de tout ce qui est affichage.
- La couche d'accès aux données « *DAO* » *Data Access Object* est l'intermédiaire entre les autres couches et la Base de données.

La couche *Métier* :

Voici quelques figures représentant un échantillon du code source de cette couche :

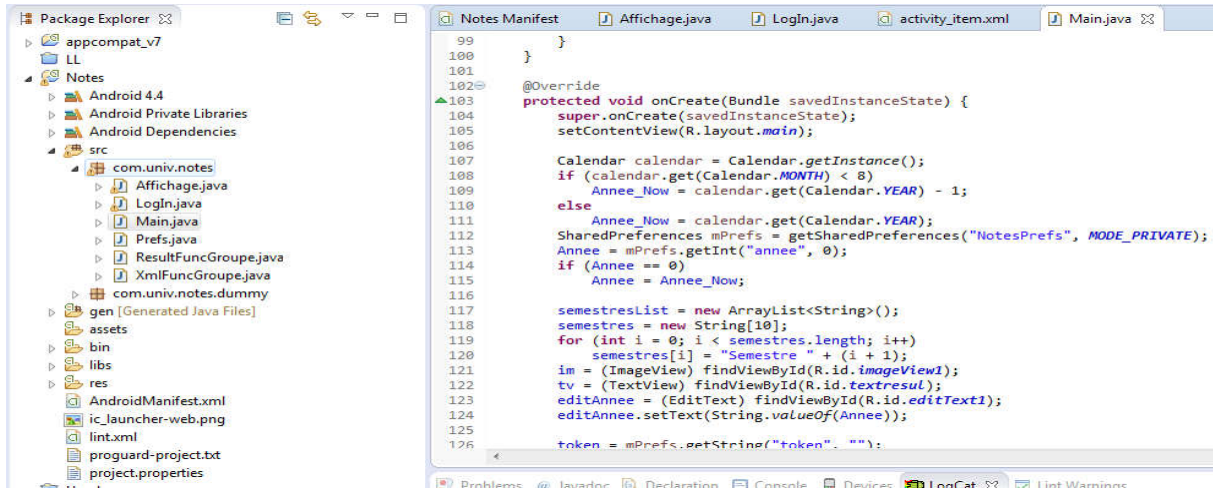


Figure 0.8 : Extrait de code la classe Main de l'application



Figure 0.9 : Extrait de code PHP du plugin moodle

La couche *Présentation* :

Voici quelques figures représentant l'interface du logiciel :

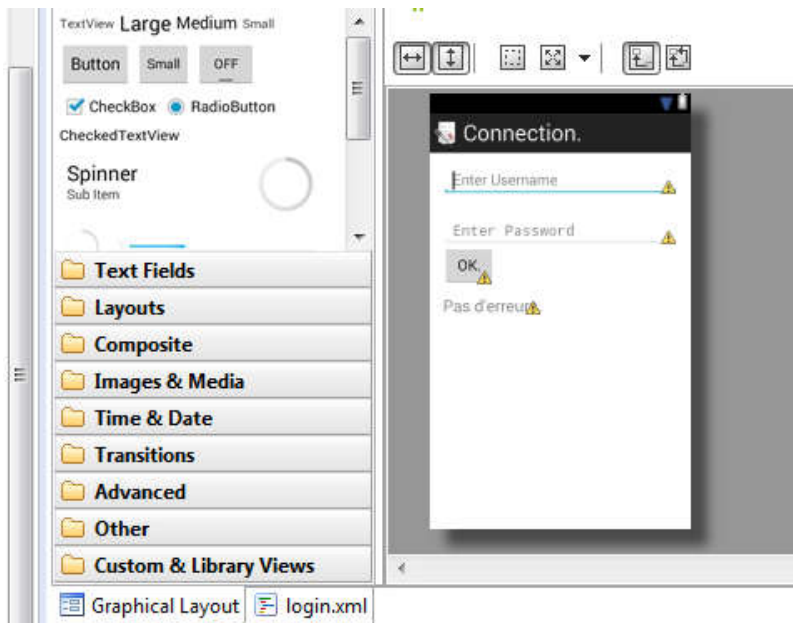


Figure 0.10 : L'interface de l'activité Connexion de l'application

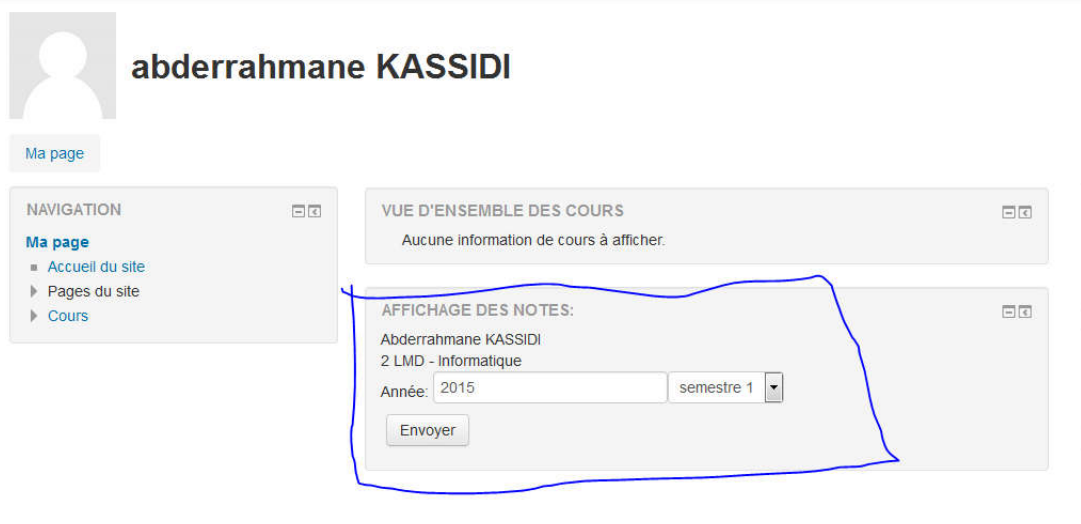


Figure 0.11 : L'affichage du plugin dans moodle

Conclusion

Nous avons vu dans ce chapitre une étude abordant les contraintes techniques du système et le choix du processus 2Tracks UP adopté qui répond aux besoins fonctionnels et techniques dès la première étape jusqu'à l'écriture du code de l'application. Dans le chapitre suivant nous présentons la mise en œuvre de l'application.

Chapitre 4 : Mis en œuvre

Introduction

Dans ce chapitre on présente la dernière étape de développement qui se traduit par la mise en œuvre ou l'exécution de l'application. On présente les outils requis, la configuration nécessaire et la démarche pour mettre l'application en état « prête à être utilisée ».

1. Partie MOODLE :

Concernant la partie de MOODLE, on a développé un plugin (block) qui sera installé sur la plate-forme.

1.1 Installation du plugin sur MOODLE :

On peut procéder de deux façons différentes pour installer un plugin dans Moodle, à savoir : à partir (A.) d'un fichier compressé (.zip) ou bien en (B.) accédant directement aux dossiers de Moodle.

A. Un fichier (notes.zip) : suivre les étapes :



Figure 0.1 : page de l'installation du plugin

B. Accès directement aux dossiers de MOODLE :

Copier le dossier du Plugin (Block) dans le dossier Blocks de MOODLE :

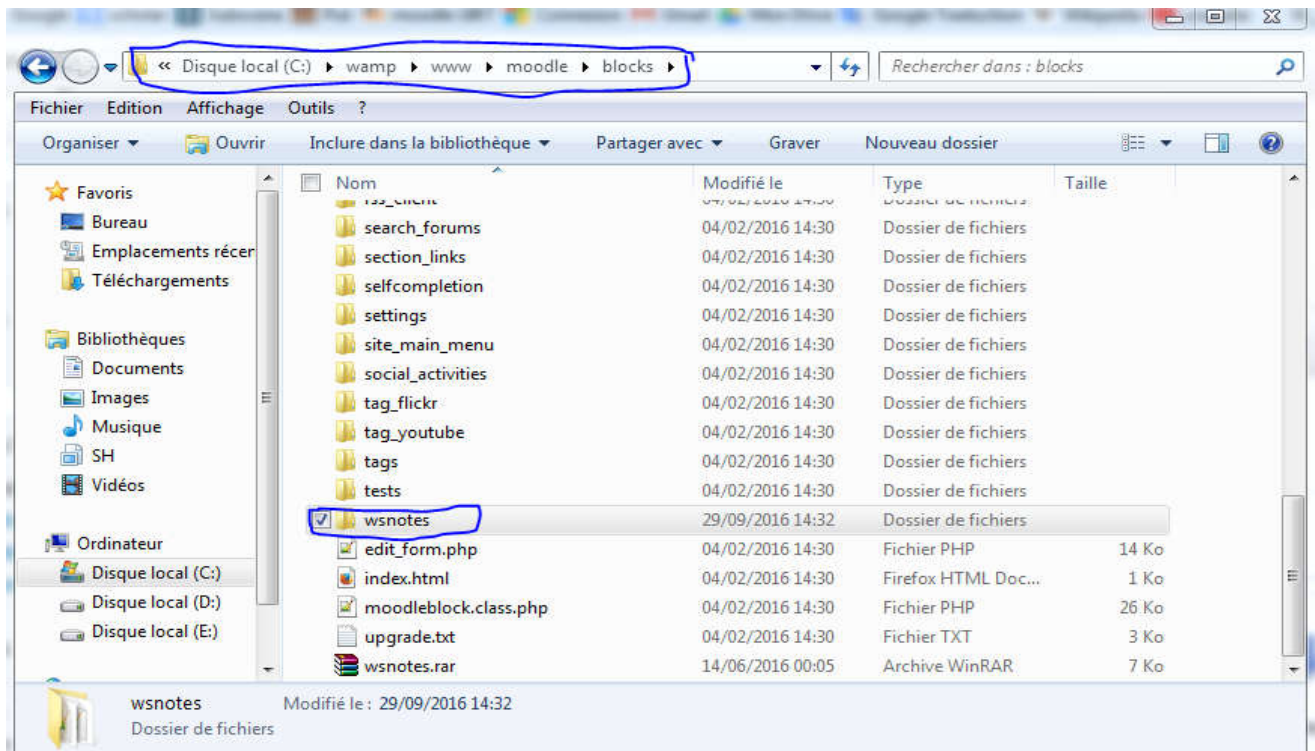


Figure 0.2 : placer nos fichiers code directement dans répertoire moodle

1.2 Web service dans MODDLE:

Gérer les Protocoles web service :



Figure 0.3 : Les protocoles supportés par moodle.

Gérer les jetons (clé de sécurité):



Figure 0.4 : gérer les clés de sécurité pour la communication.

2. Partie Android :

2.1 Exporter un fichier (*.apk) :

(*.apk) c'est l'extension des fichiers d'installation sur le système android.

- Sélectionner **export** du menu de notre projet puis **Export Android Application** :

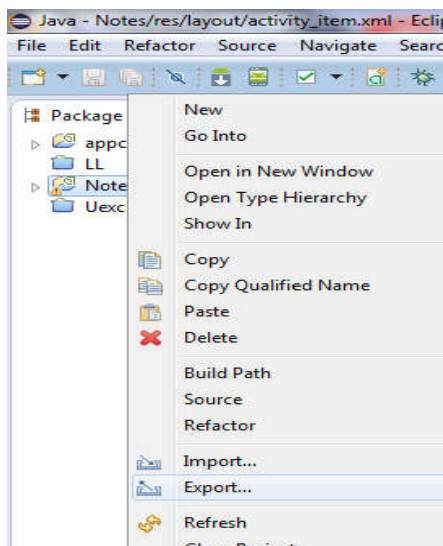


Figure 0.5 : Exporter le projet.

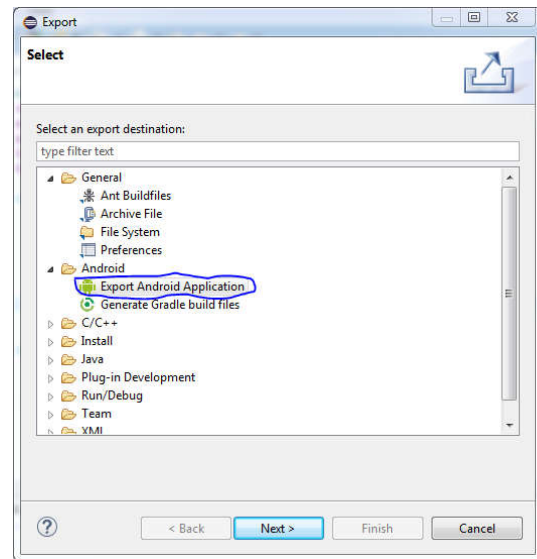


Figure 0.6 : Menu Export.

- Nous avons besoin d'un **KeyStore** pour certifier notre application :

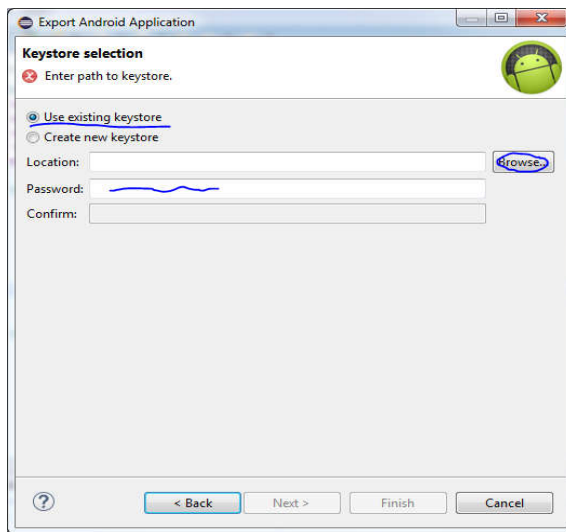


Figure 0.7 : Utiliser une KeyStore qui existe

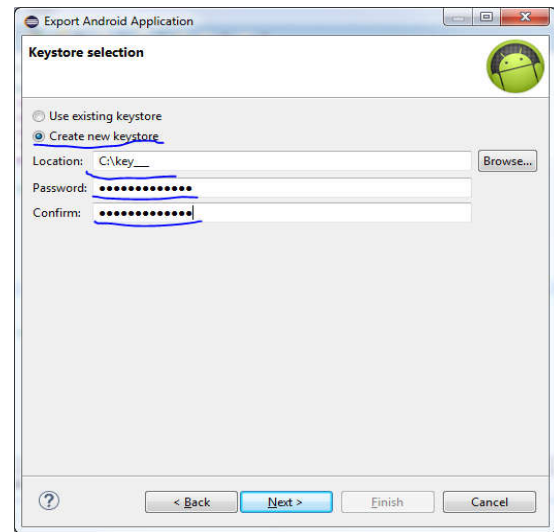


Figure 0.8 : Créer une KeyStore

Suivre les étapes ensuite de sa c'est simple....

- Dernier étape on a terminé :

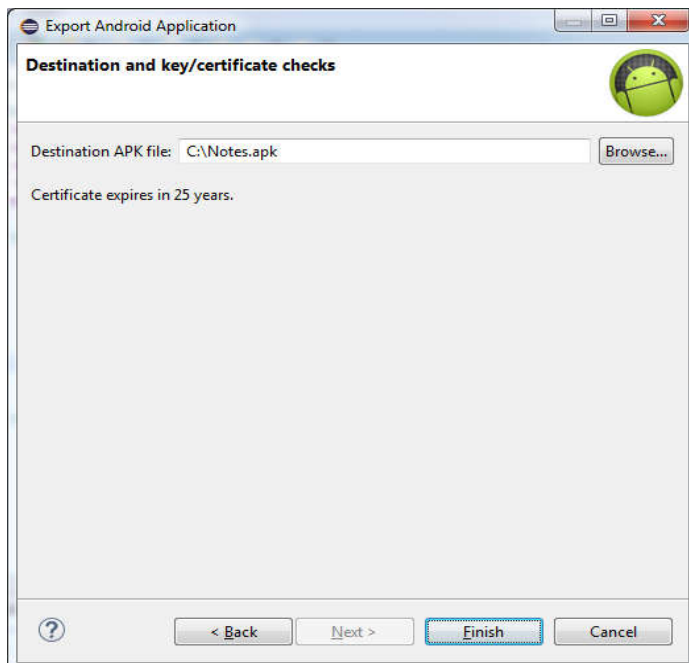


Figure 0.9 : Dernier étape pour un fichier d'installation, chemin de sauvegarde.

2.2 Installation sur téléphone :

Le fichier obtenu par les étapes précédentes est une application native à installer directement sur le téléphone comme tous les applications régulières.

2.3 Premier lancement :

La première Activité (fenêtre) que va être affiché est **Paramètre**:



Figure 0.10 : Saisir l'adresse de la plate-forme MOODLE

Après sa l'activité de **Connexion** va être affiché :

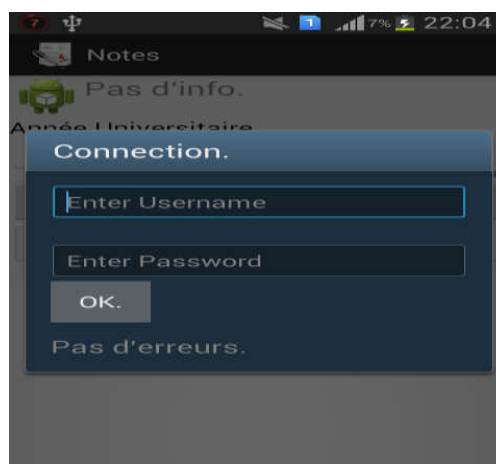


Figure 0.11 : Connexion

A ce point l'application est prête à l'utilisation.

Conclusion :

Dans ce chapitre nous avons présenté la phase de la mise en œuvre ou l'exécution de l'application. Dans le chapitre suivant on va présenter une comparaison entre les applications natives et les Web App (Application web) leurs avantages et inconvénients.

Chapitre 5 : Application native et Application Web

1. Application native :

L'application native est une application mobile que l'on télécharge (par exemple sur l'App Store ou le Play Store) et que l'on installe ensuite sur notre Smartphone ou sur notre tablette, comme un logiciel sur un ordinateur. C'est-à-dire que chaque fabricant a son propre langage de programmation :

- Pour iPhone et iPad, le langage de programmation s'appelle l'Objective-C.
- Pour tous les terminaux Android et BlackBerry, le langage de programmation s'appelle le Java.
- Pour les Windows Phones, le langage de programmation s'appelle le C#.

Le tableau suivant montre la relation entre les sociétés prédictives des systèmes d'exploitation et leurs produits.

Société	Système d'exploitation	Langage de développement	Plateforme
Apple	iOS	Objective-C	AppStore
Microsoft	Windows Phone	C#	MarketPlace
Google	Android	Java	GooglePlay

Tableau 0.1 : les différents systèmes d'exploitation mobiles et les sociétés prédictives

Le développement d'une application native permet d'utiliser la mémoire du Smartphone ainsi que toutes les fonctionnalités liées au système d'exploitation visé (GPS, accéléromètre, appareil photo, etc.). Cela permet également de proposer des applications plus

riches en termes de fonctionnalité, de qualité, de performance et de résolution que les web applications en HTML5 ou les applications hybrides.

2. Application web:

L'application web est pratiquement un site Internet créé exclusivement pour les terminaux mobiles, mais attention de ne pas confondre application web et site Internet **Responsive**³. La Web app est accessible uniquement via les navigateurs mobiles. Les langages de programmation sont le HTML5, le CSS3 et le JS (JavaScript).

3. L'installation et la mise à jour :

Application native : Comme nous l'avons précisé plus haut une application native doit être téléchargée puis installée sur le device. Lorsqu'une mise à jour de l'application est effectuée par le développeur. L'utilisateur doit la faire manuellement pour garder l'application en question à jour. Ce qui peut entraîner une multitude de fichiers sur son mobile. À l'installation d'une application, une icône de raccourci est créée sur le « Bureau » ce qui permet le lancement rapide.

Application web : Il n'y a aucune installation à effectuer et les mises à jour sont automatiques, car à chaque changement de page ou de consultation l'application web est rechargée. Les mises à jour seront donc transparentes pour les utilisateurs.

4. Le coût de développement :

Application native : Une application native ne va pas être la même sur tous les systèmes d'exploitation, autrement dit chaque type de téléphone va avoir une version différente de l'application. Par exemple, une application Android ne va pas être agencée de la même manière qu'une application pour iPhone. Cette différence oblige bien souvent à reprendre toute la partie création de l'application depuis le début (maquette graphique, développement suivant le langage...).

Une application native n'est pas disponible immédiatement, il faut tout d'abord qu'elle soit validée par l'App Store ou le Play Store par exemple. De plus, pour que votre future

³ Des sites qui adaptent la navigation selon l'appareil utilisé (mobile, pc, tablette...).

application soit en téléchargement sur l'App Store (librairie d'application Apple) il faut payer un abonnement auprès d'Apple. Le coût de développement d'une application native reste pour le coup assez **élevé**.

Application web : Le développement d'une application web est beaucoup moins contraignant. En effet, celle-ci fonctionne comme un site Internet, donc il suffit simplement d'avoir un navigateur Web (Chrome, Safari...) sur notre mobile, car celle-ci est hébergée sur un serveur Web. Graphiquement, il n'y aura aucune différence d'affichage sur un iPhone ou sur Galaxy S3 par exemple, ce qui **réduit énormément** le coût de développement de l'application.

5. L'expérience utilisateur :

Il y a une certaine différence sur l'expérience utilisateur entre une application native et une application web.

Application native : Ce type d'application sera plus rapide et fluide et s'exécutera en plein écran contrairement à l'application web. De plus, l'application native pourra elle, utiliser les fonctionnalités du téléphone ou de la tablette comme le GPS, l'accéléromètre, l'appareil photo, ou même le gyroscope (permet de détecter les mouvements de l'utilisateur). Ce type d'application va pouvoir également utiliser le système des notifications « Push » que tout le monde connaît.

Application web : Il faut faire la différence entre un site Internet et une Web app. L'utilisateur verra votre application web avec le même design qu'une application native, avec un menu mobile, des animations, mais avec les éléments de votre navigateur mobile (barre d'adresse, flèches de navigation...). Lors du développement d'une application web il faut optimiser le rendu des images et le poids des pages pour ne pas faire attendre l'utilisateur trop longtemps, car il faut savoir que le débit sur le réseau téléphonique (même s'ils nous promettent des débits incroyables), on est loin du débit ADSL.

6. Conclusion :

Application native :

- Les avantages :

- Accessibilité de l'application hors connexion ;
- Meilleure expérience utilisateur ;
- Meilleur référencement dut aux téléchargements sur les plateformes comme l'App Store ou le Play Store.
- Inconvénients :
 - Pas la même application sur les différentes plateformes (Apple, Android, Windows Phone, BlackBerry) ;
 - Coût de développement important ;
 - Problème de compatibilité après les mises à jour.

Application web :

- Les avantages :
 - Un seul et unique code pour les différentes plateformes.
 - Coût de développement moins important.
 - Compatible avec tous les navigateurs.
- Inconvénient :
 - Non accessible en mode hors connexion (sauf s'il y a une mise en cache du site).
 - Ne peut pas accéder aux applications natives du mobile (GPS, appareil photo...).
 - Manque de fluidité sur les anciens modèles de Smartphone.

Conclusion Générale

Ce projet de fin d'études nous a permis d'aborder des grands chapitres en technologie de l'informatique, à savoir, le e-learning, Moodle, système android et comment exploiter ces derniers pour donner une solution aux étudiants pour accéder à leurs notes et à leurs moyennes de façon administrative via l'Internet en utilisant un navigateur (application web) ou bien programme sous android (application native).

Sachant que la quasi-totalité des étudiants et enseignants possèdent un Smartphone, ils peuvent donc utiliser ces applications. Cette utilisation a l'avantage d'être indépendante du temps et du lieu. Elle élimine de ce fait les contraintes de temps et de déplacement.

Bien que ce domaine soit vaste, nous avons essayé d'apprendre le maximum et de donner des solutions aux problèmes posés pendant la période de l'affichage des notes aux étudiants.

Ce projet nous a donné une expérience intéressante, il nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation. Il nous a permis également de travailler en équipe pendant une durée courte mais enrichissante.

Nous espérons que cette méthode et ces applications soient adoptées par l'administration du département de l'informatique de l'université de Tiaret comme une solution provisoire pendant une période d'essai afin de dégager des conclusions, et pourquoi pas généraliser ce projet dans le futur ?

Bibliographie

Bloch, L. (2003). *Les systèmes d'exploitation des ordinateurs." Histoire, fonctionnement, enjeux.*

Ghirardini, B. (2011). *méthodologies pour le développement de cours e-learning.*

Kouninef, B., Djelti, M., & Rerbal, S. *CONCEPTION ET REALISATION D'UNE PLATE FORME E-LEARNING.*

lecompte, B. (2007). *le choix d'un learning Management System.*

Murphy, M. (2010). *L'art du développement Android. Pearson Education France.*

NARCISSE, K. F. (2011). *DEVELOPPEMENT D'UN SYSTEME DE GEOLOCALISATION EN UTILISANT LA PLATEFORME ANDROID.*

Prat, M. (2008). *E-learning, réussir un projet: pédagogie, méthodes et outils de conception, déploiement, évaluation.* Editions ENI.

Wikipedia. (s.d.). *Moodle.* Consulté le juin 2016, sur <https://fr.wikipedia.org>:
<https://fr.wikipedia.org/wiki/Moodle>

Table de figures :

Figure 1.1 : Exemple d'architecture de Moodle	12
Figure 1.2 : Exemple d'architecture de Moodle	13
Figure 1.3 : WAMP Server	14
Figure 2.1 : Différentes couches du système.....	20
Figure 2.2 : Cycle de vie d'une Activité sous Android.....	21
Figure 3.1 : le processus 2TUP	24
Figure 3.2 : capture de navigation a partir un browser sur android	27
Figure 3.3 : diagramme de cas d'utilisation de la consultation des notes	29
Figure 3.4 : Diagramme de classes (Moodle)	30
Figure 3.5 : Diagramme de séquence (obtenir de données)	31
Figure 3.6: architecture 03 tiers pour Moodle	31
Figure 3.7: architecture 03 tiers pour application Android.....	32
Figure 3.8 : Extrait de code la classe Main de l'application	33
Figure 3.9 : Extrait de code PHP du plugin moodle	33
Figure 3.10 : L'interface de l'activité Connexion de l'application	34
Figure 3.11 : L'affichage du plugin dans moodle.....	34
Figure 4.1 : page de l'installation du plugin	35
Figure 4.2 : placer nos fichiers code directement dans répertoire moodle.....	36
Figure 4.3 : Les protocoles supportés par moodle.....	36
Figure 4.4 : gérer les clés de sécurité pour la communication.....	37
Figure 4.5 : Exporter le projet. Figure 4.6 : Menu Export.....	37
Figure 4.7 : Utiliser une KeyStore qui existe Figure 4.8 : Créer une KeyStore	38
.....	38
Figure 4.9 : Dernier étape pour un fichier d'installation, chemin de sauvegarde.	38
Figure 4.10 : Saisir l'adresse de la plate-forme MOODLE.....	39
Figure 4.11 : Connexion	39

Webographie

1. <http://docs.moodle.org>
2. <http://www.linda.com>
3. <http://www.developer.android.com>
4. <http://www.Mkyong.com>
5. <http://www.theopentutorials.com>
6. <http://fr.wikipedia.org>
7. <http://stackoverflow.com>
8. <http://fr.openclassrooms.com>