

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou

Faculté de Génie Electrique et d'informatique
Département d'informatique



Mémoire

de fin d'études

En vue de l'obtention du diplôme de Master en informatique

Options : Conduite de projets informatiques
Systèmes informatiques

Thème

*Conception et réalisation d'une application
mobile-learning multiplateforme
orientée Services Web*

Dirigé par :

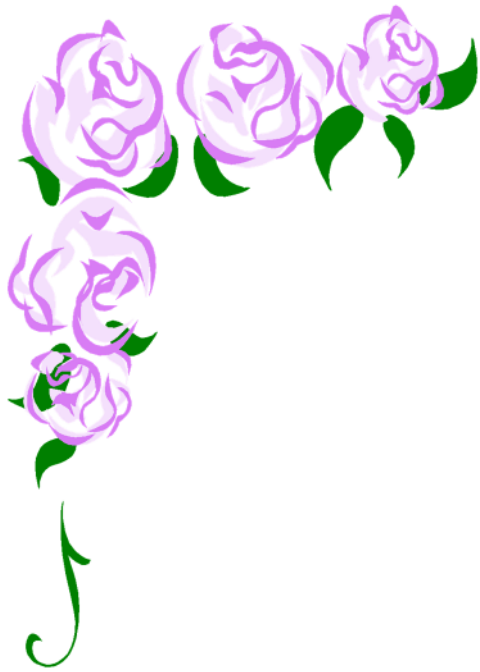
M^{me} BOUARAB Farida

Réalisé par :

M^{elle} : OUAKED Salima

M^{elle} : KRIM Safia

Promotion 2014-2015



Remerciements

Tout d'abord, Nous remercions ALLAH le tout puissant et miséricordieux qui nous a offert la vie, et seul capable de m'offrir la foi, le bonheur, la prospérité et la santé. Dieu merci de nous avoir accordé la force, la volonté et surtout le courage pour mener à terme le présent mémoire.

En second lieu, nous tenons à exprimer notre profonde gratitude à notre chère promotrice M^{me}F. BOUARAB pour sa confiance, patience et sa bonne orientation. Nous lui sommes très reconnaissantes pour ses remarques, conseils qui nous ont été si utiles.

Nous espérons avoir été à la hauteur de la confiance qu'elle nous a prêtée. C'était un honneur pour nous de travailler avec elle.

nous aimons également adresser nos vifs remerciements aux membres du jury « Mr Ramdane », « Melle Ait Adda » et « Mr Habet » devant qui nous avons l'honneur de présenter notre.

Nos remerciements vont à toute personne ayant contribué à la réussite de ce projet.

Merci



Liste des figures

Chapitre I :

| | |
|--|----|
| Figure 1: Ressources d'apprentissage | 10 |
| FIGURE 2: Capture d'écran d'une e-leçon..... | 11 |

Chapitre II :

| | |
|--|----|
| Figure 3: Classification des réseaux sans fil suivant leur taille..... | 36 |
| Figure 4: <i>Le mode Ad Hoc.</i> | 38 |
| Figure 5: <i>Le mode infrastructure.</i> | 39 |

Chapitre III :

| | |
|--|----|
| FIGURE 6: Scénario d'utilisation des services Web par les acteurs Client/Fournisseur | 52 |
| FIGURE 7: Format d'un message SOAP..... | 59 |
| FIGURE 8: Structure d'un message SOAP..... | 60 |
| FIGURE 9: Structure générale d'un fichier WSDL 1.1 | 67 |
| FIGURE 10: Mécanisme d'accès aux services fournis par UDDI Register | 71 |
| FIGURE 11: schema représentant l'architecture REST..... | 74 |
| FIGURE12 : Comparaison entre JSON et XML..... | 76 |
| FIGURE 13: La Requête GET. | 78 |
| FIGURE 14: La Requête POST. | 78 |
| FIGURE 15: La Requête PUT | 78 |
| FIGURE 16 : La Requête DELETE. | 79 |
| FIGURE 17: SOAP VS REST | 81 |

Chapitre IV :

| | |
|--|----|
| FIGURE 18: Architecture générale du système | 86 |
| FIGURE 19 : Diagramme de contexte | 88 |
| FIGURE 20: Structuration des cas d'utilisation..... | 89 |
| FIGURE 21 : Diagramme de cas d'utilisation « consulter module »..... | 90 |
| FIGURE 22 : Diagramme de cas d'utilisation « Gérer module » | 90 |
| FIGURE 23 : Diagramme de cas d'utilisation « Gestion filière» | 91 |

| | |
|---|-----|
| FIGURE 24 : Diagramme de cas d'utilisation « Authentification » | 92 |
| FIGURE 25 : Diagramme de cas d'utilisation « Inscription » | 92 |
| FIGURE 26 : Diagramme de séquence « Authentification » | 99 |
| FIGURE 27 : Diagramme de séquence « Inscription étudiant » | 100 |
| FIGURE 28 : Diagramme de séquence « Consulter cours » | 101 |
| FIGURE 29 : Diagramme de séquence « Publier PV de notes » | 102 |
| FIGURE 30 : Diagramme de séquence « Supprimer cours » | 103 |
| FIGURE31 : Diagramme de classe général | 105 |
| FIGURE 32: Diagramme de déploiement de l'application | 110 |
| Chapitre V : | |
| FIGURE 33: Fonctionnalités du service web EtudiantFacadeREST | 113 |
| FIGURE 34: Test de la web méthode getAllEtds()..... | 113 |
| FIGURE 35: Fonctionnalités du service web EnseignantFacadeREST | 114 |
| FIGURE 36: Test de la web méthode returnEns() | 114 |
| FIGURE 37: Fonctionnalités du service web AdminFacadeREST | 114 |
| FIGURE 38: Fonctionnalités du service web CoursFacadeREST | 115 |
| FIGURE 39: Fonctionnalités du service web NoteFacadeREST..... | 115 |
| FIGURE 40: Fonctionnalités du service web ModuleFacadeREST | 115 |
| FIGURE 41: Fonctionnalités du service web SpécialitéFacadeREST | 116 |
| FIGURE 42: Capture des Smartphones Sony(Android) et Nokia (Windows phone)..... | 122 |
| FIGURE 43: capture des packages générés après compilation de l'application..... | 123 |
| FIGURE 44: la page d'accueil | 124 |
| FIGURE 45: Contact de l'administrateur..... | 125 |
| FIGURE 46: la page d'authentification | 126 |
| FIGURE 47: message d'erreur dans l'authentification | 126 |
| FIGURE 48: page d'inscription | 127 |
| FIGURE 49: page espace étudiant | 127 |
| FIGURE 50: page des spécialités..... | 128 |

| | |
|---|-----|
| FIGURE 51: page des modules | 128 |
| FIGURE 52: liste des cours..... | 129 |
| FIGURE 53: page du module de l'enseignant | 129 |
| FIGURE 54 : page de création d'un cours | 130 |
| FIGURE 55: page espace admin | 130 |
| FIGURE 56: page de suppression d'un étudiant..... | 131 |

Introduction générale..... 1

Chapitre I: Technologies pour le mobile-learning

I. Introduction..... 4

II. Présentation 4

II.1. L'avènement de e_learning 4

II.2. Définition de e_learning 5

III. Critères de choix de formation 6

IV. La force du e_learning 6

IV.1. Pourquoi e_learning ? 6

IV.2. La solution e_learning 7

IV.3. Le développement de compétences avec e_learning 8

V. Les différentes approches de l'e_learning 8

VI. Les formes de communication en e_learning 9

VII. Interactivité dans e-learning 10

VII.1. Eléments de base 10

VII.2. Qualité du e-learning..... 12

VIII. Evaluation 13

VIII.1. Avantages de l'e_learning 13

VIII.2. Inconvénients de l'e-learning..... 13

IX. Les plates-formes d'e-learning 14

IX.1. Types de plates-formes 14

IX.2. Les acteurs de plates-formes e-learning..... 15

IX.3. Fonctionnalités d'une plate-forme d'e-learning..... 16

X. E-learning sur support mobile 16

XI. Conclusion 17

Chapitre II : Technologies pour le mobile-learning

I. Introduction..... 18

II. Les technologies multiplateformes 18

II.1. Les approches multiplateformes 18

II.2. Les solutions de développement multiplateforme mobile 20

| | |
|--|----|
| III. Les réseaux | 33 |
| III.1. Définition..... | 33 |
| III.2. Evaluation : | 33 |
| III.3. Eléments d'un réseau sans fil : | 34 |
| III.4. Techniques de transmission sans fil..... | 34 |
| III.5. Classification des réseaux sans fil | 35 |
| III.5.1. Classification des réseaux en fonction de la taille..... | 35 |
| III.5.2. Classification des réseaux suivant le mode de fonctionnement..... | 38 |
| III.6. Sécurité dans les réseaux sans fil..... | 39 |
| III.6.1. Les attaques contre les réseaux sans fil | 39 |
| III.6.2. Sécurisation | 41 |
| III.7. Caractéristiques logiques des réseaux | 42 |
| III.7.1. Modèle de référence OSI | 42 |
| III.7.2. Modèle de référence TCP/IP | 44 |
| IV. Systèmes d'exploitations mobiles | 44 |
| IV.1. Les principaux systèmes d'exploitation mobiles | 44 |
| V. Conclusion | 45 |

Chapitre III : Services Web

| | |
|---|----|
| I. Introduction..... | 46 |
| II. Historique | 46 |
| III. Présentation des services Web..... | 47 |
| III.1. Définition d'un service Web..... | 47 |
| III.2. Motivations..... | 49 |
| IV. Types des services web..... | 50 |
| IV.1. Services Web SOAP | 50 |
| 1. Présentation | 50 |
| 2. Architecture..... | 50 |
| 3. Technologies des services Web..... | 52 |
| 4. Le langage XML (<i>eXtensible Markup Language</i>) | 53 |
| 4.1. Définition | 53 |
| 4.2. XML et les services Web | 54 |
| 5. Le protocole SOAP (<i>Simple Object Access Protocol</i>) | 55 |

| | |
|---|----|
| 5.1. Définition | 55 |
| 5.2. Origines | 56 |
| 5.3. Principes de SOAP | 56 |
| 5.4. Modèle d'échange de message SOAP | 57 |
| 5.5. Relation avec le XML | 57 |
| 5.6. Format d'un message SOAP | 57 |
| 5.7. Modèles de données | 62 |
| 5.8. Protocoles utilisés par SOAP | 64 |
| 5.9. SOAP et XML | 64 |
| 5.10. Le transport | 64 |
| 5.10.1. SOAP via HTTP | 64 |
| 6. Le langage WSDL (<i>Web Service Description Language</i>) | 67 |
| 6.1. Présentation de WSDL | 67 |
| 6.2. Description fonctionnelle et structurelle | 67 |
| 7. Le protocole UDDI | 70 |
| 7.1. Définition | 70 |
| 7.2. Données du registre UDDI | 71 |
| 8. Fonctionnement global des services Web SOAP | 72 |
| IV.2. Services Web REST | 73 |
| 1. Présentation | 73 |
| 2. Architecture | 73 |
| 2.1 Ressource | 74 |
| 2.2 Représentation | 75 |
| 2.2.1 JSON | 75 |
| 2.3 Stateless | 76 |
| 2.4 Méthodes HTTP | 77 |
| 3. JAX_RS | 79 |
| 3.1. Les annotations JAX_RS | 79 |
| V.SOAP VS REST | 80 |
| V.1. Evaluation | 80 |
| V.2. Argumenter le choix du service web | 81 |
| VI. E_learning et les services web | 82 |
| VII. L'approche services web | 82 |

VIII. Conclusion..... 83

Chapitre IV : Analyse & conception

I. Introduction 84

II. La démarche d’élaboration du projet 84

III. Présentation d’UML 85

III.1. Introduction à l’UML 85

IV. Objectif du projet 85

V. Architecture du système 85

VI. Spécification des besoins 87

VI.1. Identification des acteurs et des besoins 87

VI.2. Le diagramme de contexte 87

VII. Structuration des cas d’utilisation 88

VII.1. Détermination des cas d’utilisation 89

VII.2. Description textuelle des cas d’utilisation..... 92

VII.2.1. Cas d’utilisation « Consulter module » 93

VII.2.1.1.1. Identification du cas d’utilisation 93

VII.2.1.1.2. Description des scénarios 93

VII.2.1.2.1. Identification du cas d’utilisation 93

VII.2.1.2.2. Description des scénarios 93

VII.2.2. Cas d’utilisation « Gestion de filière » 94

VII.2.2.1.1. Identification du cas d’utilisation 94

VII.2.2.1.2. Description des scénarios 94

VII.2.2.2.1. Identification du cas d’utilisation 94

VII.2.2.2.2. Description des scénarios 95

VII.2.2.3.1. Identification du cas d’utilisation 95

VII.2.2.3.2. Description des scénarios 95

VII.2.3. Cas d’utilisation « Gérer module » 96

VII.2.3.1.1. Identification du cas d’utilisation 96

VII.2.3.1.2. Description des scénarios 96

VII.2.3.2.1. Identification du cas d’utilisation 96

VII.2.3.2.2. Description des scénarios 96

| | |
|---|---------|
| VII.2.4. Cas d'utilisation « Authentification » | 97 |
| VII.2.4.1. Identification du cas d'utilisation | 97 |
| VII.2.4.2. Description des scénarios | 97 |
| VII.2.5. Cas d'utilisation « Inscription » | 97 |
| VII.2.5.1. Identification du cas d'utilisation | 97 |
| VII.2.5.2. Description des scénarios | 98 |
| VII.3. Description graphique des cas d'utilisation | 98 |
| VIII. La conception objet : | 104 |
| VIII.1. Les bases de données | 104 |
| IX. Le diagramme de déploiement | 110 |
| X. Conclusion | 111 |
| Chapitre V : Implémentation & réalisation | |
| I. Introduction | 112 |
| II. Description des différents services web | 113 |
| III. L'environnement technique de développement | 116 |
| III.1. Outils de développement | 116 |
| III.2. Les langages utilisés | 119 |
| IV. Présentation des interfaces | 121 |
| V. Conclusion | 131 |
| Conclusion générale | 132 |
| Bibliographie | 133 |
| ANNEXE | 134 |

Sommaire

| | |
|-----------------------------------|----------|
| Introduction générale..... | 1 |
|-----------------------------------|----------|

Chapitre I : Avènement de e-learning

| | |
|--|-----------|
| I. Introduction | 3 |
| II. Présentation..... | 3 |
| II.1. L'avènement de e-learning..... | 3 |
| II.2. Définition de e-learning..... | 6 |
| III. Critères de choix de formation | 6 |
| IV. La force de e-learning..... | 6 |
| IV.1. Pourquoi e-learning ? | 6 |
| IV.2. La solution e-learning | 7 |
| IV.3. Le développement de compétences avec e-learning. | 8 |
| V. Les différentes approches de e-learning | 8 |
| VI. Les formes de communication en e-learning | 9 |
| VII. Interactivité dans e-learning | 10 |
| VII.1. Éléments de base..... | 10 |
| VII.2. Qualité de e-learning..... | 12 |
| VIII. Evaluation | 13 |
| VIII.1. Avantages de e-learning | 13 |
| VIII.2. Inconvénients de e-learning | 13 |
| IX. Les plateformes d'e-learning | 14 |
| IX.1 Types de plateformes..... | 14 |
| IX.2 Les acteurs de plateformes e-learning..... | 15 |
| IX. Fonctionnalités d'une plateforme e-learning..... | 16 |
| X. E-learning sur support mobile | 16 |
| XI. Conclusion | 17 |

Chapitre II : Technologies pour le mobile learning

| | |
|--|-----------|
| I. Introduction | 13 |
| II. Les technologies multiplateforme..... | 13 |
| 1. Les approches multiplateforme..... | 13 |
| I.1.1. Définition d'un poste de travail | 13 |

| | |
|--|----|
| I.1.2. Liste des postes de travail..... | 13 |
| I.2. Etude des documents | 16 |
| I.2.1. Introduction..... | 16 |
| I.2.2. Liste des documents..... | 16 |
| I.3. Etude des registres | 25 |
| I.4. Etude des procédures | 29 |
| I.5. Diagramme des flux..... | 29 |
| II. Critiques de l'existant et suggestions | 31 |
| Conclusion | 33 |

Chapitre III : Analyse et conception

| | |
|--|----|
| Introduction | 34 |
| I. Description des procédures utilisées dans le domaine d'étude..... | 34 |
| II. Les objectifs de notre application | 35 |
| III. Analyse | 35 |
| 1. Spécification des besoins | 35 |
| 2. Identification des acteurs | 36 |
| 3. Identification des cas d'utilisations..... | 37 |
| 4. Diagramme de cas d'utilisation..... | 39 |
| IV. Conception..... | 53 |
| 1. Diagramme de séquence | 53 |
| 2. Diagramme de classes | 57 |
| 3. Le modèle relationnel | 58 |
| 4. Le modèle physique | 59 |
| Conclusion | 61 |

Chapitre V : Réalisation

| | |
|---|----|
| Introduction | 62 |
| 1. Critique de l'existant et suggestion..... | 62 |
| 1.1. Environnement technique de développement..... | 62 |
| 1.2. Présentation du langage de programmation | 62 |

| | |
|--|----|
| 1.3. Présentation du l'outil de développement..... | 63 |
| 2. Présentation de quelques interfaces de notre application..... | 67 |
| o Conclusion générale. | |
| o Annexe. | |
| o Bibliographie. | |

La salle de classe, qu'elle soit réservée à des étudiants ou à des employés d'entreprises désirant se perfectionner dans une discipline, connaît depuis peu un concurrent sérieux au travers d'une nouvelle approche d'apprentissage apparu sous le nom d'e-Learning.

Le e-learning commence à se généraliser dans plusieurs domaines (entreprises, universités,...) et cela grâce au développement de l'informatique et des TIC (technologies de l'informations et de la communication). En particulier le réseau mondial Internet est devenu facile d'accès pour la plupart et offre des débits de plus en plus importants permettant l'exploitation de divers outils qui augmentent l'interactivité entre les différents participants à une séance d'apprentissage. L'internet a accentué cette orientation en offrant aux apprenants de plus grandes facilités dans les communications et les constitutions de savoirs communs et facilite l'accès à l'information.

En effet Le web a permis à l'enseignement électronique une expansion sans précédent, ce qui a fait que plusieurs organismes de formation de nos jours développent et offrent des plateformes e_learning sur internet. De nouveaux types d'applications, des technologies et des normes internationales, sont mise en oeuvre et mise à jour pour prendre en charge les différentes progression sconnues dans le domaine de l'informatique en général. Parmi ces technologies nous trouvons les web services. Ils permettent la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués, il s'agit d'un ensemble de fonctionnalités exposées sur Internet ou Intranet, que l'on peut atteindre via les protocoles Internet constituant cette technologie prometteuse.

D'autre part, dans le but d'atteindre un large public et étendre l'accessibilité, les entreprises modernes se dirigent tout naturellement vers le développement mobile.

Cependant, La diversité qui existe dans le domaine mobile, notamment le nombre important de systèmes d'exploitation qui utilisent des technologies différentes, engendre cependant une fragmentation importante : « environnement IOS/Objective-C pour l'iPhone et l'iPad, SDK Java spécifique pour Android, J2ME pour Symbian, etc. ». Cette diversité système a poussé les développeurs a cherché à mettre en place plusieurs approches afin de parvenir à développer pour plusieurs plateformes à la fois : il s'agit du développement multiplateforme.

Objectif du mémoire:

L'objectif de ce travail est d'apporter un éclairage sur le développement multiplateforme d'applications mobile-learning orientées services web, par un survol des technologies utilisées tout en essayant de montrer l'intérêt d'un tel développement.

Structuration du mémoire :

Notre mémoire comporte cinq chapitres :

- ✓ Le premier chapitre intitulé « L'avènement de e_learning » explique le domaine sur lequel porte notre application qui est l'enseignement à distance.
- ✓ Le second chapitre intitulé « Technologies pour le mobile-learning », quant à lui, il est consacré à la présentation des différentes solutions multiplateforme et technologies réseaux intervenant dans l'élaboration de notre application.
- ✓ Le troisième chapitre intitulé « Services Web », nous traitons en détails les web services, leur émergence, apport, architecture et les technologies qui les composent.
- ✓ Dans le quatrième chapitre, nous abordons la démarche suivie dans la conception de notre application.
- ✓ Enfin, le cinquième chapitre décrit l'environnement de développement et l'implémentation de notre application.

I. Introduction

Lorsqu'il s'agit d'entreprendre des études, la majorité des gens pensent tout de suite à l'école, apparemment incontournable. Ils s'y voient assis dans une classe en face d'un enseignant ou d'une enseignante. Mais si, pour mille et une raisons, une personne ne peut pas ou ne souhaite pas fréquenter un établissement scolaire, doit-elle renoncer à tout projet d'études? Pas nécessairement! Il existe en effet une solution de rechange à l'enseignement traditionnel. Cette forme d'enseignement relativement jeune, c'est la formation à distance. Or, parce qu'elle n'est pas le mode d'enseignement prédominant, les gens la connaissent mal. Et parce qu'ils la connaissent mal, ils ont parfois tendance à s'en méfier, à la sous-estimer. Il suffit pourtant de s'y intéresser un peu pour en découvrir toute la pertinence et l'efficacité. La formation à distance permet d'acquérir des connaissances et de développer des habiletés sans avoir à fréquenter un établissement d'enseignement et sans la présence physique d'une personne qui enseigne. Le domicile ou tout autre lieu devient l'école, le collègue ou l'université et le matériel didactique devient la ressource pour apprendre.

Certaines personnes mettent en doute la validité et l'efficacité de cette forme d'enseignement et d'apprentissage. Pour ces personnes, l'équation est simple : « professeur manquant, étudiant manqué ». Elles ne conçoivent pas qu'on puisse apprendre à distance, chez soi, par soi même. Pourtant, chaque jour et durant la vie, chacun apprend déjà diverses choses à distance. En regardant des documentaires ou des reportages à la télévision, par exemple, on apprend forcément une foule de choses.

II. Présentation

II.1. L'avènement de e_learning

Depuis 1998, avec la généralisation d'Internet, l'impact des TIC sur les activités de développement des compétences et des connaissances s'amplifie. Ce mouvement est moins rapide en Algérie que dans les autres pays maghrébins et européens. Les usages sont peu intégrés dans les organisations : les offres ou pratiques tirant partie d'Internet sont placés à part du système principal de formation et la vision industrielle est trop souvent absente des ouvrages et supports méthodologiques qui n'envisagent que des projets isolés.

Dans la plupart des pays avancés, la e-formation ou le e-learning est plutôt considéré comme une évolution d'un système global. On évoque en général la « e-transformation » des activités de formation ou de manière plus technique, la formation développée ou améliorée grâce aux technologies.

La mise en œuvre de la e-formation présente pourtant des avantages décisifs : au niveau interne de la formation (rapport coût-efficacité) comme au niveau externe (rapport coût-effet sur le travail). Cela laisse présager une croissance importante.

L'avance observée aux Etats-Unis a conduit à présenter l'université d'entreprise et l'Université virtuelle comme des modèles qui s'imposeraient. Ces modèles d'organisation sont probablement peu applicables en Europe, moindre en Algérie, nous observons cependant que la ou elle se développe de manière productive, la e-formation dépasse la simple modalité pédagogique et technique. La réussite des projets (durée et déploiement) implique des évolutions fortes qui relèvent d'une certaine industrialisation.

Le « monde Internet » entre dans le monde de la formation et le bouleverse, mais les nouvelles technologies étaient déjà présentes bien avant Internet depuis une vingtaine d'années. En effet, dans les années quatre-vingt – quatre-vingt-dix, l'EAO était à l'ordre du jour, des milliers d'heures de contenu de formation ont été « médiatisés ». On a cru alors que l'apprenant pouvait apprendre seul devant son ordinateur, plus particulièrement dans les années quatre-vingt dix lorsque la technique du multimédia l'a rendu possible.

Le e-learning, quant à lui, a vraiment pris naissance dans un laps de temps s'étalant entre les années quatre-vingt-dix et le début des années 2000. Il est associé à l'utilisation d'Internet dans la formation.

Tentons de définir tout d'abord ce que signifie ce néologisme anglais « e-learning », le « e » exprimant explicitement la référence aux technologies, et voyons quelles réalités il recouvre concrètement en formation.

II.2. Définition de e_learning :

Le terme e_learning est souvent sans connaissance précise de sa sémantique. Etant donné qu'il s'agit de l'un des mots clés de ce travail, nous allons essayer de dissiper le flou autour de ce terme à travers différentes définitions.

« L'e_learning est l'utilisation des nouvelles technologies multimédias et de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance. »

« L'e_learning est méthode d'apprentissage spéciale basée sur l'utilisation de l'ordinateur. Ce mode d'apprentissage virtuel se distingue par des systèmes et du matériel pédagogiques spéciaux : il est dispensé sous forme numérisée ; sous forme médicale ; et/ou hyper médiale (information réticulaire), il permet l'interactivité entre l'utilisateur apprenant, le système, le coach et le co_apprenant sur place ou sur le réseau ; il est directement à la disposition de l'utilisatrice et l'utilisateur en ligne. » Définition fournie par le modèle pilote ANUBA de Basse-Saxe Rhénanie-Du-Nord-Westphalie.

« L'e_learning désigne tout dispositif de formation utilisant Internet comme canal de diffusion. C'est l'acte pédagogique qui se vit pour tout ou pour une partie en ligne. De l'autoformation tutorée à la classe virtuelle synchrone, ses formes sont très variées. De manière plus large, ce terme désigne tout système de formation reposant globalement sur l'usage des technologies issues de l'Internet. C'est le processus de formation dans son ensemble (et pas simplement l'action de former ou de se former) qui est repensé par l'usage des technologies Internet. » Définition tirée du site « e_learning Agency ».

- **Préfixe e- Quelle signification ?**

"e" Abréviation de "électronique" et maintenant de "en ligne" :

Préfixe indiquant qu'un dispositif électronique est utilisé avec le terme qui suit. Avec le développement de l'Internet et des réseaux, l'aspect électronique correspond surtout à celui de "en ligne". Se prononce "i".

III. Critères de choix de formation

Au moment de choisir entre l'apprentissage numérique (e-learning), les formations traditionnelles ou d'autres types d'apprentissage non formel ou en milieu de travail, il est nécessaire de se poser les questions suivantes :

- Quel est le coût relatif de chaque type de formation ?
- La formation sera-t-elle plus efficace si elle est dispensée en une seule unité ou répartie dans le temps ?
- La formation répond-elle à des besoins d'apprentissage à court-terme ou à long-terme ?
- Les participants ont-ils accès au matériel informatique et aux outils de communication nécessaires ?
- Les participants sont-ils suffisamment motivés pour des modes d'apprentissage numérique ou en autonomie ?
- Est-ce que les horaires et les situations géographiques des participants ciblés leur permettent de participer à des formations de type salle de classe ou à d'autres types d'apprentissage synchrone ?

IV. La force du e_learning

IV.1. Pourquoi 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.

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.

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.

IV.2. La solution e_learning

Le e-learning est une bonne solution lorsque :

- Il existe une quantité significative de contenu à délivrer à un grand nombre d'apprenants ;
- les apprenants proviennent d'endroits géographiquement dispersés ;
- les apprenants ont une mobilité réduite ;
- le temps que les apprenants peuvent consacrer chaque jour à l'apprentissage est limité;
- les apprenants ne possèdent pas de compétences de communication suffisantes ;
- les apprenants ont au moins des compétences de base en matière d'informatique et d'Internet ;
- les apprenants sont fortement motivés et apprécient d'apprendre à leur propre rythme ;
- le contenu doit être réutilisé pour d'autres groupes d'apprenants dans le futur ;
- la formation vise à renforcer des compétences cognitives, plutôt que des capacités psychomotrices ;
- le cours répond à des besoins de formation à long terme plutôt qu'à court terme ;
- il est nécessaire de recueillir et d'assurer le suivi des données.

L'apprentissage numérique n'est pas idéal dans toutes les situations, il est ainsi peu probable qu'il remplacera complètement la formation en classe au sein d'une organisation. Le e-learning peut représenter une solution plus économique pour compléter la formation conventionnelle, et atteindre le plus d'apprenants possibles.

IV.3. Le développement de compétences avec e_learning

Le e-learning peut-il servir à développer n'importe quel type de compétence ?

Les programmes de formation visent à développer différents types de compétences :

- les compétences cognitives : qui impliquent d'accroître les connaissances et la compréhension (p. ex. des concepts scientifiques), de suivre des instructions (c.-à-d. des compétences en matière de procédures) et d'appliquer des méthodes dans des situations nouvelles pour résoudre des problèmes (c.-à-d. des compétences intellectuelles ou de réflexion) ;
- les compétences interpersonnelles : comme celles nécessaires dans les activités d'écoute active, de présentation ou de négociation ;
- les compétences psychomotrices : qui impliquent l'acquisition de perceptions physiques et de mouvements (p. ex., pratiquer un sport ou conduire une voiture).

La plupart des cours e-learning sont développés pour renforcer des compétences cognitives ; le domaine cognitif est le plus approprié à l'apprentissage numérique. Dans le domaine cognitif, le développement des capacités de réflexion nécessite des activités d'apprentissage interactives, car c'est « en faisant » que ces compétences s'acquièrent le mieux. Le renforcement des compétences du domaine des relations interpersonnelles peut également être envisagé à travers l'apprentissage numérique à l'aide de méthodes spécifiques. Par exemple, des jeux de rôle interactifs suivi d'un feedback (évaluation) peuvent servir à modifier des attitudes et des comportements.

V. Les différentes approches de l'e_learning :

Il existe deux approches générales de l'apprentissage numérique : l'auto-apprentissage et la formation facilitée/dirigée par un formateur.

Les apprenants en auto-apprentissage sont seuls et complètement indépendants, tandis que les cours de formation facilités et dirigés offrent différents niveaux de soutien de la part de tuteurs et de formateurs et permettent une collaboration entre les apprenants.

Les cours e-learning combinent souvent les deux approches, mais par souci de simplicité, nous nous pencherons sur chacune de ces approches séparément :

- **Cours e_learning en auto-apprentissage :** Les apprenants sont seuls devant leurs écrans (isolés), libres d'apprendre à leur propre rythme et de définir des parcours d'apprentissage personnels en fonction de leurs intérêts et besoins individuels. Ceux qui fournissent le cours numérique n'ont pas besoin de planifier, de gérer ou d'assurer le suivi des apprenants à travers un processus spécifique. Le contenu e-learning est développé selon un ensemble d'objectifs d'apprentissage et assuré à l'aide de diverses composantes multimédia, comme des textes, des illustrations, de l'audio et de la vidéo. Le cours doit fournir autant que possible des solutions qui faciliteront l'apprentissage (par le biais d'explications, d'exemples, de commentaires, de glossaires, d'interactivité etc.), afin de rendre les apprenants autonomes. Cependant, les apprenants reçoivent en général également du soutien, sous forme de courriers électroniques ou de « e-tutorat ». Lorsque le cours e-learning en auto-apprentissage est proposé par le biais d'une connexion Internet, il est possible d'assurer le suivi des actions des apprenants dans une base de données centralisée.

- **Cours e-learning dirigé par un formateur :** Dans ce modèle, un programme linéaire est développé, qui intègre plusieurs activités et éléments dans un cours ou un syllabus chronologique.

Le cours est prévu et dirigé par un formateur à travers une plateforme d'apprentissage en ligne. Le cours e-learning peut être complété par des exposés présentés par les formateurs, des travaux individuels et des activités de collaboration entre les apprenants. Les apprenants et les formateurs peuvent utiliser des outils de communication tels que les courriels, les forums de discussion, les chats, les sondages, les tableaux blancs interactifs, les fonctionnalités de partage d'application et de conférence audio et vidéo pour communiquer et collaborer.

Le cours se termine généralement par un exercice ou un test permettant d'évaluer les résultats de l'apprentissage.

VI. Les formes de communication en e_learning :

Les communications dans le cadre d'apprentissage en ligne peuvent être synchrones ou asynchrones.

Communication synchrone : Les communications synchrones se déroulent en temps réel. Pour qu'une communication entre deux personnes soit synchrone, elles doivent être toutes les

deux présentes à un moment donné. Les chats et les conférences audio/vidéo sont des exemples d'activités synchrones.

Communication Asynchrone : Les événements asynchrones sont indépendants des questions de temps. Un cours en auto-apprentissage est un exemple d'apprentissage en ligne asynchrone puisque l'apprentissage peut se dérouler à n'importe quel moment. Les forums de discussion ou les courriels sont des exemples d'outils de communication asynchrones.

VII. Interactivité dans e-learning

VII.1. Eléments de base :

Comme nous l'avons vu, les approches en matière d'apprentissage numérique peuvent combiner différents éléments, notamment :

❖ **le contenu d'apprentissage** : Le contenu de la formation numérique peut inclure des :

Ressources d'apprentissage simple : Les ressources d'apprentissage simples sont des ressources non interactives telles que des documents, des présentations PowerPoint, des fichiers audio et vidéo. Ces ressources ne sont pas interactives au sens où les apprenants peuvent seulement lire ou regarder le contenu.

Ces ressources peuvent être développées rapidement et, lorsqu'elles correspondent à des objectifs définis et sont conçues de façon structurée, elles peuvent représenter une ressource d'apprentissage précieuse, même si elles ne permettent aucune interactivité.



Figure 1: Ressources d'apprentissage

Leçons en ligne interactives : L'approche la plus courante en matière d'e-learning en auto-apprentissage, consiste en un ensemble de leçons interactives ou e-leçons en ligne. Une e-leçon est une séquence linéaire d'écrans qui peut inclure des textes, des images, des animations, de l'audio, de la vidéo et des modules interactifs sous forme de questions et de

commentaires. Une e-leçon peut également inclure une liste d'ouvrages à lire et des liens vers des ressources en ligne, ainsi que des informations supplémentaires sur des sujets spécifiques.



FIGURE 2: Capture d'écran d'une e-leçon

- ❖ **E-tutorat** : Il est possible de proposer aux apprenants des services qui ajoutent des dimensions humaines et sociales afin de les aider durant le processus d'apprentissage. Le e-tutorat permet de fournir un soutien et des commentaires personnalisés aux apprenants grâce à des outils en ligne et des techniques de facilitation.
- ❖ **l'apprentissage collaboratif** : Discuter, partager des connaissances et travailler ensemble sur un projet commun sont des activités de collaboration. Les logiciels sociaux, tels que les chats, les forums de discussion et les blogs, sont utilisés pour permettre une collaboration en ligne entre les apprenants :

Discussions en ligne : Les discussions en ligne synchrones et asynchrones sont conçues pour faciliter la communication et le partage de connaissances entre les apprenants. Les apprenants peuvent commenter et échanger des idées sur les activités du cours ou contribuer à l'apprentissage collectif en partageant leurs connaissances.

Collaboration : Les projets collaboratifs impliquent que les apprenant collaborent pour effectuer une tâche. Les activités en collaboration peuvent inclure la réalisation de projets et de travaux sur la base de scénarios.

- ❖ **La classe virtuelle** : Une classe virtuelle est une méthode d'enseignement plus semblable à la formation en salle de classe traditionnelle, car elle est entièrement dirigée par un formateur. Une classe virtuelle est un événement d'apprentissage en ligne durant lequel un instructeur enseigne à distance et en temps réel à un groupe d'apprenants en utilisant une combinaison de divers matériels (diapositives PowerPoint, matériel audio ou vidéo, etc.). On parle aussi d'apprentissage synchrone.

Les apprenants et les formateurs doivent avoir à leur disposition les technologies appropriées (par exemple, le logiciel pour la salle de classe virtuelle et une bonne connexion à Internet).

VII.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, doit être capable de suivre les progrès et les performances des apprenants de manière individuelle.

VIII. Evaluation

VIII.1. Avantages de l'e_learning

On peut constater que l'e_learning possède plusieurs avantages :

- Gain de temps, pas de déplacement pour le tuteur et l'apprenant.
- L'apprenant peut obtenir un diplôme ou une qualification, se perfectionner ou actualiser ses connaissances dans un domaine qu'il connaît déjà.
- Souplesse du lieu ; l'apprenant peut ne pas être dans le même pays mais accéder aux cours.
- Souplesse des horaires ; l'apprenant étudie lorsqu'il a envie, il n'a pas d'heures spécifiques.
- Un suivi personnalisé.
- Personnalisation des parcours ; l'apprentissage se déroule dans un environnement familier de chaque apprenant.
- Une évaluation plus précise grâce à des tests en continu tout le long de l'apprentissage.
- Possibilité d'avoir un nombre très important d'apprenants.
- La réduction des coûts de la formation pour l'apprenant, en économisant les frais de déplacement, de logement.
- L'élasticité et la rapidité de la transmission du savoir.
- La possibilité d'échanges avec les autres apprenants à une grande échelle à l'aide de forums et chats.
- Le tuteur dispose de cours facilement réactualisables grâce des incorporés dans la plate-forme.
- Grande liberté en organisation des actions d'apprentissage (lieu, moment, rythme, contenu, méthodes).

VIII.2. Inconvénients de l'e-learning

Malheureusement e-learning n'est pas sans inconvénients, ce sont ces difficultés que la recherche essaie d'éliminer :

- Les investissements sont importants.
- Le marché est mouvant.
- L'évaluation du travail de l'apprenant est difficile.

- Les tuteurs doivent acquérir des notions d'informatique.
- Certains domaines ne peuvent être enseignés à distance (ex : cours d'art).
- L'étudiant risque de souffrir de solitude (d'isolement), ce qui peut conduire à sa démotivation.
- L'apprenant doit faire preuve de :
 - * d'initiative et d'implication.
 - * d'autonomie.
 - * d'organisation.
 - * de formulation de but.
 - * de communication y compris par écrit.

IX. Les plates-formes d'e-learning

De quoi s'agit-il ?

Selon Bouthry [Bouth, 03] « les plates-formes d'e-learning sont avant tout conçues comme des outils pour la gestion des cours par correspondance électronique ».

D'une manière plus générale, les plates d'e-learning sont un ensemble de moyens techniques et de solutions logicielles adaptées, implantées sur des réseaux de télécommunications dans les fonctions principales permettent de produire puis d'intégrer, gérer et d'administrer des contenus de formation, de diffuser une formation en ligne comprenant des ressources pédagogiques (supports multimédias), de présenter des programmes de formation, permettre un positionnement puis un suivi de la progression de l'apprenant, de construire des parcours de formation individualisé ou des parcours communs, d'encadrer des individus et d'animer des groupes.

IX.1. Types de plates-formes : [Vanw, 04]

a) CMS (Course Management System) :

Ces plates-formes ont été créées dans le but d'aider les enseignants. Elles sont peu modulables mais assez faciles d'utilisation. Les CMS sont « orientés cours » et contiennent le plus souvent des outils d'évaluation de progrès, des outils de collaboration asynchrone ou encore, des outils de création de contenus. Les CMS sont quelques fois aussi appelés bibliothèques de cours.

Le but premier de ce genre d'outils est d'épauler les enseignants et donc de fournir des suppléments au cours magistral.

Etant donné que cet outil cible un cours particulier, les normes d'interopérabilité ne sont pas une priorité.

b) LMS (Learning Management System) :

Le LMS est plus centré sur l'utilisateur. Il permet de suivre celui-ci tout au long de sa formation. Le LMS est conçu pour pouvoir s'adapter aux différentes normes vu qu'il peut traiter plusieurs cours différents.

Les LMS sont destinés principalement aux entreprises. Ce sont souvent des plates-formes ouvertes permettant d'intégrer plusieurs sortes de contenus. En plus des possibilités offertes par une plate-forme de CMS, le LMS apportera principalement une gestion de suivi des apprenants, il sera plus défini comme un véritable portail.

Ce type de plates-formes doit nécessairement regrouper les outils nécessaires afin de faciliter les rôles et les fonctions des acteurs de ces plates-formes.

IX.2. Les acteurs de plates-formes e-learning

Nous pouvons avoir un découpage de plates-formes d'e-learning en acteurs et rôles comme suit :

L'auteur (concepteur): le tuteur qui fait des parcours pédagogiques types et individualisés, et incorpore des ressources multimédias.

L'enseignant tuteur (formateur): effectue un suivi de l'apprenant, anime son groupe.

L'apprenant (étudiant) : l'étudiant consulte en ligne le scénario d'apprentissage, télécharge ou explore des ressources documentaires, résout des problèmes et transmet des devoirs à corriger et échanger des idées et des informations au cours de débats en télé discussions.

L'administrateur : gère les inscriptions, s'occupe de l'installation de la plate-forme et de sa maintenance, ainsi que de la procédure de la sécurisation des accès.

IX.3. Fonctionnalités d'une plate-forme d'e-learning

Une plate-forme d'e-learning doit aussi offrir des possibilités de définir des dispositifs capable d'assurer les fonctionnalités suivantes :

1. Création de cours et de plans de formation (création de programmes de cours, édition de tests) ;
2. Gestion des documents pédagogiques : classification, indexation, administration des matériaux pédagogiques ;
3. Gestion administrative de la formation (scolarité) : inscription dans la plate-forme, dans une formation; gestion de données administratives de la formation, scolarité ;
4. Organisation du tutorat : constitution des groupes d'apprentissage; choix des enseignants, des groupes d'apprenants et de leurs droits d'accès, de types de contenus, de modes de communication, de types de tests.
5. Evaluation ;
6. Apprentissage : consultation à distance de contenus pédagogiques, communication entre formateurs et apprenants et entre apprenants, individualisation des apprentissages, télé tutorat, possibilité de remise des travaux.
7. Outils et services : forums, système de messagerie, blogs, groupes de discussions.

X. E-learning sur support mobile

Des sondages montrent que les apprenants plébiscitent à 80% le papier comme vecteur de l'information. Les recherches effectuées montrent que c'est en premier lieu la mobilité du support qui les intéresse. Un livre peut être emporté avec soi, il est manipulable n'importe où (dans les transports, devant la télévision, pendant les moments creux, etc.). C'est de là qu'est venue l'idée du papier virtuel qui n'est rien d'autre que le transfert de l'e-learning sur support de faible encombrement. L'émergence et la généralisation des réseaux sans fils combinés à la multiplication des terminaux ultralégers a permis sa concrétisation.

En effet, la mobilité offre un avantage concurrentiel car elle permet d'accéder à l'information à tout instant et partout où l'utilisateur en a besoin. Ceci permet l'apprentissage ou la formation permanente tout au long de sa vie, appelé « mobile-learning ».

Dans l'optique de rendre les cours plus mobiles, donc de mieux les intégrer dans le quotidien, le mobile-learning devient un axe de recherche et de développement essentiel et promoteur.

XI. Conclusion

Après avoir mis l'accent sur l'e-learning, nous concluons que l'avènement de cette méthode d'apprentissage s'explique par le fait de vouloir abolir les notions de temps et d'espace. Nous devons mettre en œuvre des méthodes pédagogiques exploitant de nouvelles possibilités que nous offrent les nouvelles technologies.

Nous présenterons dans le chapitre suivant les nouvelles technologies qui permettront la concrétisation de notre travail.

I. Introduction

A l'aube de **2015**, le marché des mobiles est toujours en pleine expansion: Smartphones et tablettes de toutes tailles, livres électroniques, consoles de jeux... notre monde devient toujours plus connecté, mais aussi plus fragmenté.

En téléphonie mobile la fragmentation technologique désigne la multiplication des constructeurs ainsi que la disparité des caractéristiques des téléphones portables : tailles d'écran, types de clavier, fonctionnalités, systèmes d'exploitation et plus généralement des systèmes mobiles.

Cette fragmentation implique l'adaptation nécessaire de toute application ou logiciel pour mobile aux différents systèmes d'exploitation, que ce soit **iOS** d'Apple (sur iPhone, iPod, iPad), **Android** de Google (sur mobiles HTC et Samsung, Galaxy Tab et autres), ou bien encore **BlackBerry OS** de RIM.

On parle bien de systèmes d'exploitation et pas d'appareils. C'est en effet lui qui exécute les applications plus que le matériel qui n'est que le support et fournit les interfaces et la puissance permettant aux applications de s'exécuter. On parle de plateforme quand on veut parler de l'ensemble des appareils qui partagent un même système d'exploitation et sont donc très compatibles entre eux en ce qui concerne les applications.

Quant à la technologie de développement d'applications mobiles, Chaque plateforme nécessite des outils de développement différents. Si on veut déployer une application sur différentes plateformes, il semble donc nécessaire de développer autant de fois ces applications qu'on veut adresser de plateformes. Mais il existe des solutions pour permettre de ne développer qu'une fois l'application et la rendre multiplateforme.

L'objectif de ce chapitre est donc de présenter ces solutions et de les comparer entre elles, chacune ayant des avantages et des inconvénients.

II. Les technologies multiplateformes

1. Les approches multiplateformes

Pour réaliser l'ambitieux projet que l'on peut résumer par le motto de Sun Microsystems « write once, run anywhere », les solutions multiplateformes ont mis en place trois approches se mélangeant parfois les unes aux autres.

1. Le natif : C'est l'approche la plus naturelle. On développe séparément pour chacune des plateformes dans le langage prévu pour elles. Cette technique permet d'obtenir le résultat le plus proche du système ciblé. On utilise les outils dédiés à chacun des environnements et on dispose des meilleures conditions pour la réalisation et les tests de notre application.

Dans le cas de Windows Phone par exemple, on utilisera la barre d'application alors que cette fonctionnalité est représentée différemment sur Android. Sur Windows Phone toujours, on utilisera le principe du pivot alors que la tendance sur iOS est de présenter des onglets plus classiques.

Faire du natif, c'est faire 3 développements distincts. Très peu de choses sont mutualisées et chaque mise à jour ou correctif devra être réalisé 3 fois.

Cependant, cela offre :

- La souplesse nécessaire à la réalisation d'une application qui colle parfaitement à l'écosystème de chaque téléphone
- Des performances qui *peuvent* être excellentes (relatif à la qualité du développement, évidemment).

C'est l'approche la plus chère et qui nécessite le plus de compétences et de connaissances mais c'est celle qui donne souvent la meilleure garantie de qualité.

2. Le quasi-natif : Si on écarte la solution native pour des raisons de coût et de réactivité lors des mises à jour et des correctifs, il faut se diriger vers une approche qui permet de partager le maximum de code possible.

Une des manières d'arriver à ce résultat est ce qu'on va appeler le "quasi-natif". Le principe est simple : on utilise un seul langage de programmation et on développe une seule fois mais on utilise un ensemble d'outils pour compiler et packager une version de l'application adaptée à chaque cible. Pour cela, soit le code "générique" est traduit dans le langage compris par la plateforme puis compilé nativement, soit il est compris et exécuté par une machine virtuelle lors de l'exécution.

Deux frameworks/outils sortent du lot pour atteindre cet objectif : Xamarin et Titanium.

3. Le web : La 3ème approche est une extension naturelle d'une des fonctions premières des Smartphones depuis leur apparition : l'accès au world wide web. Accéder au web à partir d'un mobile signifie notamment une chose : il y a un navigateur web sur le téléphone et on peut y exécuter de l'HTML et du Javascript.

Depuis toujours, on essaye de rendre nos sites web accessibles sur les mobiles. On a commencé en développant une version Mobile distincte puis en appliquant tous les principes qu'on regroupe derrière le terme responsive.

En partant de ces deux constats (le web marche sur tous les mobiles et on sait faire des sites qui s'affichent correctement sur un mobile) on peut se demander : pourquoi ne pas packager nos sites dans une application ? Cela aurait plein d'avantages : elle serait disponible sur les stores, en jouant un peu avec le CSS on peut donner l'impression que c'est une application classique et comme c'est de l'HTML c'est multiplateforme par définition.

Là où l'idée se corse, c'est quand on veut utiliser des fonctionnalités du téléphone. La sauvegarde de fichier en local par exemple, l'utilisation de l'appareil photo ou encore l'accès aux contacts. C'est possible, mais complexe à réaliser. C'est là qu'entrent en jeu les frameworks comme **PhoneGap**.

2. Les solutions de développement multiplateforme mobile

Heureusement, les solutions existent pour résoudre ce problème. C'est les frameworks de développements mobiles. La liste des solutions est assez longue avec une dizaine de solutions du marché retenues permettant en un seul développement de générer des applications sur tous les supports mobiles.

Mais toutes ces solutions ne partagent pas une même diffusion ou une même reconnaissance du public, c'est pourquoi nous les avons classé en deux catégories : les dominantes et les challengers. Pour chaque solution nous avons indiqué :

- ✓ Un descriptif de la solution.
- ✓ Ses avantages et ses inconvénients.
- ✓ Un tableau décrivant les caractéristiques de la solution.

Les différentes solutions ayant des portabilités très variables et des couvertures fonctionnelles très différentes les unes des autres, nous avons mis en place deux tableaux de comparaison :

- Un premier tableau indiquant la disponibilité de la solution pour chaque plateforme,
- Un second tableau indiquant les fonctionnalités de la solution.

Enfin nous concluons en donnant notre analyse sur les solutions présentées.

2.1. Les solutions dominantes

- **Phonogap** : est une solution hybride. C'est une bibliothèque permettant d'accéder depuis le JavaScript d'une page web à de nombreuses ressources matérielles de l'appareil. Le système de callback sur lequel il fonctionne pose des problèmes lors de son intégration avec des bibliothèques JavaScript tierces. Phonogap ne gère pas l'interface utilisateur qui est entièrement laissée à la charge de l'utilisateur et du navigateur embarqué. La présentation se fait donc à l'aide de CSS, de bibliothèques JavaScript tierces, ou des langages spécialisés tels que Mobl.

Evaluation ?

- Une petite bibliothèque, des accès à de nombreuses ressources matérielles, une solution extensible, une communauté active, et la gratuité.

✘ Windows phone 7 n'est pas supporté pour le moment.

Caractéristiques de la solution

| CRITÈRE | DESCRIPTION | COMMENTAIRE |
|--------------------------|----------------------------|--|
| Type de solution | Bibliothèque | |
| Type de déploiement | Hybride | |
| Gestion de L'IHM | Non | L'ergonomie est laissée à la charge de l'utilisateur. En général on choisira une bibliothèque JavaScript pour construire l'interface utilisateur |
| Type de mise en page | Coordonnées et pourcentage | On peut utiliser les CSS pour construire l'interface. Cependant on utilisera généralement la mise en page proposée par la solution que l'on a retenue pour créer l'interface graphique |
| Langage | HTML 5, CSS 3, JavaScript | |
| Vitesse de développement | Rapide / Moyenne | Les fonctions sont souvent basées sur un système de callback pouvant produire des problèmes de compatibilité avec certaines bibliothèques tierces. Par exemple créer un simple objet Proxy de base de données avec Sencha Touch et le faire communiquer avec l'API de Phonogap afin d'accéder à une base de données n'est pas trivial à réaliser |

| | | |
|----------------------------|--|--|
| Courbe d'apprentissage | Rapide | Phonegap est une petite bibliothèque qui utilise toujours les mêmes mécanismes |
| EDI | N'importe quel EDI supportant les technologies du développement web (HTML, CSS et JavaScript) | |
| Outil graphique pour l'IHM | Aucun | |
| Débogueur | Débogueurs JavaScript (weinre, safari, chrome, bugzilla, ...) | |
| Documentation | Référence de l'API, wiki | |
| Outils de support | IRC, Google group et blog | |
| Popularité | Très grande | |
| Outils de formation | Webinars , forum privé (avec les plans Basic et Starter) et support par email (Starter) | |
| Prix | La bibliothèque est gratuite (community). Les plans de support sont : Basic 249\$/y (181€), Starter 999\$/y (729€) | |
| Licence | Licence BSD modifiée ou Licence MIT | |
| Open source | Oui | |
| Support des tablettes | Oui | |
| Outils de déploiement | « Phonegap Build » | |
| Extensibilité | Oui à travers des plugins | Il est possible d'étendre indéfiniment Phonegap à travers l'écriture de plugins. Cependant il faut écrire un plugin par plateforme à supporter |

- **Titanium** : est une API et une plateforme JavaScript éditées par «Appcelerator ». Il permet de créer des applications natives et offre une API supportant une grande quantité de ressources matérielles. Appcelerator fournit également un IDE basé sur Eclipse, « Titanium studio », qui permet de compiler et tester son code à la volée. Le codage est rapide et il suffit de peu de code pour développer une application. Cependant Titanium souffre de deux inconvénients majeurs. Le premier est qu'il faut une connexion Internet permanente pour pouvoir utiliser Titanium Studio, cela signifie que si on perd sa connexion ou si le site de Titanium est en maintenance, alors on ne peut plus travailler. Le second inconvénient est la documentation : il y a beaucoup d'outils mais l'information y est divisée et aucun d'eux n'est vraiment exhaustif, on peut passer de précieuses minutes

(voire heures) à chercher une information. A noter également que quelques incohérences graphiques se produisent parfois, l'équipe d'Appcelerator travaille actuellement dessus. Dans l'ensemble le plus gros problème reste la documentation, car on gagne vraiment du temps à développer avec Titanium, malheureusement on perd ce temps à chercher comment résoudre tel ou tel problème.

Evaluation ?

- ☑ L'application native, l'aspect natif et performances, l'accès aux ressources matérielles, la vitesse de développement, et l'extensibilité.
- ✘ La mauvaise documentation, le manque de ressources d'apprentissage, l'IDE réclamant une connexion Internet permanente.

Caractéristiques de la solution

| CRITÈRE | DESCRIPTION | COMMENTAIRE |
|----------------------------|------------------------------------|--|
| Type de solution | Runtime, Framework | |
| Type de déploiement | Natif | |
| Gestion de L'IHM | Oui | Oui |
| Type de mise en page | Coordonnées | |
| Langage | JavaScript | |
| Vitesse de développement | Rapide / Moyenne | |
| Courbe d'apprentissage | Moyenne / Longue | Malgré l'utilisation du langage JavaScript très répandu, la courbe d'apprentissage s'allonge considérablement à cause d'une mauvaise documentation. L'application de référence "Kitchen Sink" par exemple est construite sur un design non recommandé. Alors que Tweetanium, également écrit par l'équipe de Titanium, propose de bien meilleures pratiques mais n'est pas mis en avant. |
| EDI | Titanium Studio (basé sur Eclipse) | C'est une bonne intégration qui contient les simulateurs, la coloration des éléments de l'API et un débogueur. Cependant l'IDE a besoin d'une connexion permanente avec le serveur d'Appcelerator pour fonctionner. |
| Outil graphique pour l'IHM | Aucun | |
| Débogueur | Titanium Studio | La couche d'abstraction entre l'API JavaScript et le code natif final étant très grande, il est difficile à la vue des informations délivrées par le débogueur de corriger de manière productive les exceptions |

| | | |
|-----------------------|--|---|
| | | générees par ce code (Objective C par exemple). Le JavaScript par contre se débogue très bien. |
| Documentation | Q&A, référence de l'API, wiki, blog et screencasts | Trop d'outils incomplets. Certaines fonctionnalités sont définies dans certains outils et pas d'autres ce qui rend difficiles les recherches : il faut chercher à travers chaque outil pour avoir une information, quand elle existe. |
| Outils de support | Q&A, devlinks et blog | Le système de question/réponse basé sur un modèle similaire à « Stackoverflow » est intéressant et performant. La mise en relation avec d'autres développeurs Titanium est une excellente chose. |
| Popularité | Très grande | |
| Outils de formation | Classes en anglais, par exemple : « Building Native Mobile Apps » dure deux jours et coute £1,250.00 (1424€). En Europe il y a deux endroits où les classes sont données en anglais : Londres et Malmö (Suède) | |
| Prix | La bibliothèque est gratuite (community). Les plans de support sont : Basic 249\$/y (181€), Starter 999\$/y (729€) | Les différences concernent la disponibilité des modules (Paypal, SMS, Gamekit, etc.) et le degré de support disponible |
| Licence | Apache public license v2 | |
| Open source | Oui | |
| Support des tablettes | Oui | |
| Outils de déploiement | Aucun | |
| Extensibilité | Oui, on peut écrire des modules pour iOS et Android pour étendre Titanium | |

- **Rhomobile** : La société Rhomobile propose une série d'outils et de solutions de développement pour les parties clientes et serveur des applications. « Rhodes » est le Framework que propose Rhomobile pour le développement d'applications mobiles. Le développement se fait dans Eclipse à l'aide du plugin « Rho Studio » et du debugger « Rho Debugger » à installer sur l'appareil de test. Une application « Rhodes » est une application hybride de type « Ruby On Rails » pouvant accéder à certains « widgets » et aux ressources matérielles des appareils. On peut le combiner avec un Framework HTML tel que Sencha pour compléter les éléments graphiques qu'il propose. Rhomobile propose d'autres produits en synergie avec Rhodes. Le serveur RhoConnect (ancien RhoSync) permet d'interfacier et de synchroniser très simplement les données de l'application avec

celles hébergées par un site web. NB : RhoConnect n'a pas besoin que l'application cliente soit écrite avec Rhodes. RhoHub permet de développer son application en ligne, de la déployer et d'héberger un serveur de synchronisation des données. RhoConnect et RhoHub sont payants.

Evaluation ?

Quelques « widgets » natifs, la similarité au très populaire « Ruby On Rails », l'environnement de test et de développement efficace et puissant, la forte structuration du code de type MVC, et la gratuité.

✘ La courbe d'apprentissage au début et peu d'accès aux « widgets » natifs.

Caractéristiques de la solution

| CRITÈRE | DESCRIPTION | COMMENTAIRE |
|----------------------------|---|--|
| Type de solution | Plateforme | |
| Type de déploiement | Hybride, Web | |
| Gestion de L'IHM | Oui | La solution fournit trois composants natifs (Toolbar, tabBar, splitview). Le reste de l'interface graphique est laissé à la charge du développeur. En général on choisira une bibliothèque JavaScript pour construire l'interface utilisateur. |
| Type de mise en page | Coordonnées et pourcentage | On peut utiliser les CSS et les éléments natifs fournis. Cependant généralement on utilisera la mise en page proposée par la solution qu'on a retenue pour créer l'interface graphique. |
| Langage | Ruby, HTML, CSS, JavaScript | |
| Vitesse de développement | Rapide / Moyenne | Développement similaire à une application Ruby On Rails (RoR), avec le même genre de génération par ligne de commande et l'accès aux « gems » ruby |
| Courbe d'apprentissage | Rapide / Moyenne | Cela demande de connaître Ruby et les conventions utilisées dans « RoR ». Pour quelqu'un ayant déjà pratiqué « Rails » ça sera très rapide |
| EDI | Le plugin Eclipse « RhoStudio » ou n'importe quel éditeur supportant le développement web et Ruby | RhoStudio donne accès aux lignes de commandes élémentaires et permet de construire les applications sur les plateformes supportées |
| Outil graphique pour l'IHM | Aucun | |
| Débogueur | Rhodes debugger | Pour déboguer on pourra également utiliser une « gem » tel que « Rspecs » |
| Documentation | Wiki référençant l'API, screencasts | La Documentation est bonne, elle contient des guides et une référence de l'API détaillés avec des |

| | | |
|-----------------------|--|--|
| | | exemples expliqués et illustrés. Les « screencasts » sont bien faits et un nouveau est créé chaque semaine. Il manque un tutoriel étape par étape permettant de créer une mini application afin de mieux comprendre le fonctionnement général d'une application Rhodes. Car ce point est très peu traité dans la documentation |
| Outils de support | Google group, email | Le système de question/réponse basé sur un modèle similaire à « Stackoverflow » est intéressant et performant. La mise en relation avec d'autres développeurs Titanium est une excellente chose |
| Popularité | Moyenne | |
| Outils de formation | Webinar gratuit tous les vendredis | |
| Prix | Rhodes est gratuit, Rhohub à partir de 500MB de données (\$5K, 3671€), à partir de 5GB de données (\$10K, 7342€) | |
| Licence | Licence MIT | |
| Open source | Oui | |
| Support des tablettes | Oui | |
| Outils de déploiement | RhoHub permet de créer des applications, d'héberger un serveur et de déployer ses applications (Premium \$5 K/y (3652€), Enterprise \$10 K/y (7304€) | |
| Extensibilité | Oui | |
| Produits associés | Serveur de synchronisation (RhoSync) | |

- **Adobe Flex** : Le développement mobile « Flex » profite de l'intégration des différents outils Adobe. L'environnement de développement « Adobe Flash Builder » est très complet, depuis l'intégration au workflow des outils graphiques comme Photoshop ou Fireworks jusqu'au monitoring des communications entre le serveur et l'application. « Flash Builder » permet de construire graphiquement son application à la manière d'un « Interface Builder ». Les points négatifs se retrouvent au niveau de l'ergonomie : il y a des problèmes de fluidité et de performance lors de la navigation dans de grosses applications et il n'existe pas pour le moment de « Template » complet pour imiter complètement un aspect natif.

Evaluation ?

L'environnement de développement très complet et performant, la bonne documentation et nombreux tutoriels, la communauté active, l'accès à de nombreuses ressources natives, la gestion vidéo poussée (enregistrement et lecture), et la gratuité.

Les performances, aucun aspect natif par défaut, et l'environnement de développement payant.

Caractéristiques de la solution

| CRITÈRE | DESCRIPTION | COMMENTAIRE |
|----------------------------|---|---|
| Type de solution | Framework | |
| Type de déploiement | Framework | |
| Gestion de L'IHM | Oui | Proche d'un aspect natif avec des thèmes, cependant ces derniers sont incomplets pour iOS et Android entre autres |
| Type de mise en page | Pourcentage | |
| Langage | ActionScript 3, MXML | |
| Vitesse de développement | Rapide / Moyenne | |
| Courbe d'apprentissage | Moyenne | Bonne documentation, la communauté est très active et aide beaucoup |
| EDI | Adobe Flash Builder | |
| Outil graphique pour l'IHM | Adobe Flash Builder | |
| Débogueur | Emulateurs et débogueur AIR d'Adobe Flash Builder | |
| Documentation | Documentation, cookbooks | |
| Outils de support | Forum, blogs, support | |
| Popularité | Grande | |
| Outils de formation | Tutoriels, vidéos, cours privés (exemple : 5 jours 1995€) | |
| Prix | Gratuit | |
| Licence | MPL 1.1 (Mozilla Public License) | |
| Open source | Oui | |
| Support des tablettes | Oui | |

| | | |
|-----------------------|--|---|
| Outils de déploiement | Non | |
| Extensibilité | Oui | |
| Produits associés | Le runtime AIR sur lequel tourne l'application et l'IDE Adobe Flash Builder qui permet de facilement construire, coder et tester son application | Tous les produits Adobe sont intégrables dans le workflow. Il est très facile par exemple de faire une image sous Photoshop et de la transférer dans Flash Builder. |

2.2. Les challengers

- ❖ **Mobl**: est un langage récent déclaratif et impératif, il permet de construire une application avec une simplicité déconcertante. Le langage est compilé via un plugin dans Eclipse ou en ligne de commande qui génèrent des fichiers HTML/CSS/JavaScript. Il faut noter cependant que de nombreuses ressources matérielles ne sont pas prises en charge (caméra et GPS par exemple) et que l'aspect d'une application Mobl reste éloigné d'une application native.

Evaluation ?

- La rapidité du développement, la simplicité du code, « design » du langage orienté mobile, l'environnement de test et développement, et la gratuité
- La documentation pauvre, le langage jeune et pauvre, pas de changement automatique de style par plateforme, la petite communauté, et le peu d'accès aux ressources matérielles.
- ❖ **OpenPlug**: est un environnement de développement. Il utilise une API nommée « Elips » qui permet de coder facilement une application pour mobile en Flex et de la déployer sur les différents smartphones. Les fonctionnalités présentées ainsi que les plateformes supportées sont nombreuses. L'IDE est complet mais la documentation est inexistante et les tutoriels très peu nombreux. Cependant l'IDE intègre quelques applications exemples. Il est très difficile de trouver des informations concernant OpenPlug mais il semble plus performant que Flex seul.

Evaluation ?

- De nombreuses plateformes supportées, et de nombreuses ressources matérielles accessibles.

- ✘ Une petite communauté, la documentation inexistante, et l'aspect final très éloigné des applications natives.
- ❖ **Corona Labs** : permet de créer des applications très performantes en un minimum de code. Il permet d'avoir accès à une bonne partie des ressources matérielles. L'interface graphique est composée de deux couches : une couche est native (alertes et champs de saisie par exemple) et une autre est dessinée en OpenGL (listes et barres de navigation par exemple). La politique actuelle de Corona est axée sur la création de jeux et par exemple l'affichage de « widgets » est encore en bêta. Le SDK est complet (simulateur, débogueur) et les tutoriels ne manquent pas. La communauté est vaste et active.

Evaluation ?

- Le langage de script Lua très simple et puissant, la grosse communauté.
- ✘ Orienté jeux, et peu de « widgets » natifs.
- ❖ **MoSync** : passe par une programmation en C/C++, donc plus bas niveau que ses concurrents. Cela permet un niveau de performances permettant de traiter des domaines comme le jeu vidéo. Cependant ses utilisateurs l'utilisent surtout pour faire des applications plus classiques mais sur une quantité de terminaux considérable. Il y a d'ailleurs nettement plus de tutoriaux et exemples pour des applications faites sur des « non Smartphones ». La communauté est dynamique, la documentation est complète. L'IDE basé sur Eclipse est très bien conçu.

Evaluation ?

- Énormément de plateformes supportées en plus des smartphones (MoRE émulateur), et la bonne documentation.
- ✘ La programmation bas niveau (C/C++).
- ❖ **AppMobi** : c'est une petite librairie JavaScript similaire à Titanium. Son principal intérêt vient de l'environnement de simulation qui donne accès à de nombreuses ressources matérielles habituellement exclues des simulateurs comme l'accéléromètre et le GPS. Appmobi permet de simuler les principaux appareils du marché : iPhone, iPad, HTC, etc.

Evaluation ?

☑ Un environnement complet et très intégré, une bonne documentation et nombreux tutoriels, et une communauté active.

✗ Les performances, et l'aspect éloigné de l'aspect natif.

❖ **QuickConnectFamily** : QCFamily est une API JavaScript permettant d'accéder à un nombre impressionnant de ressources matérielles. Cependant sa documentation est pauvre et la communauté très réduite.

Evaluation ?

☑ De très nombreuses fonctionnalités, et un accès à de nombreuses ressources matérielles.

✗ Une communauté inexistante, et une documentation très pauvre.

❖ **Worklight** : intègre Phonegap, on retrouve donc toutes les fonctionnalités de Phonegap dans Worklight. Ce dernier ajoute des fonctionnalités pour gérer les sessions, récupérer des données depuis des webservices, internationaliser l'application, debugger, utiliser des ressources matérielles propres à certaines plateformes. Worklight ajoute de nombreux outils dont un IDE complet basé sur Eclipse intégrant les principaux simulateurs et un serveur web. Plusieurs fonctionnalités sont disponibles comme :

- Le push.
- La connexion à un large éventail de bases de données et web services.
- Un mécanisme de mise à jour directe évitant d'avoir à faire valider chaque nouvelle mise à jour par les stores.
- Un mécanisme de gestion de version.
- Un mécanisme de « skinning » permettant d'adapter l'application à différents appareils d'une même plateforme.
- Une intégration flexible des processus d'identification et de sécurité de type « SSO ».
- Une console de gestion centralisée pour toutes les applications déployées et leurs versions, adaptateurs d'intégration et règles de push.

Evaluation ?

- Un accès aux ressources matérielles, une synchronisation client/serveur, une gestion de version, et des prix à la carte.

✘ Cher.

2.3. Tableaux comparatifs

2.3.1. Présentation des tableaux

Les tableaux de ce chapitre comparent les solutions entre elles. Le premier tableau indique la disponibilité de ces dernières pour les principales plateformes du marché, le second indique les fonctionnalités supportées par les différentes solutions.

Légende :

| | |
|--------------------------|---|
| Disponible | ✓ |
| Non disponible | ✘ |
| Prochainement disponible | ➤ |

3.1.1.1. Systèmes d’exploitation supportés par les solutions

| | EDITEUR | PHONEGAP | TITANIUM | RHOMOBILE | FLEX | MOBL | OPENPLUG | CORONA | MOSYNC | APPMOBL | QUICKCONNECT | WORKLIGHT |
|----------------|-----------|----------|----------|-----------|------|------|----------|--------|--------|---------|--------------|-----------|
| iOs | Apple | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Android | Google | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Blackberry Os | RIM | ✓ | ✓ | ✓ | ✓ | ✓ | ✘ | ✘ | ➤ | ✘ | ✓ | ✓ |
| Symbian | Nokia | ✓ | ✘ | ➤ | ✘ | ✓ | ✓ | ✘ | ✓ | ✘ | ✓ | ✘ |
| Windows mobile | Microsoft | ✘ | ✘ | ✓ | ✘ | ✘ | ✓ | ✘ | ✘ | ✘ | ✘ | ✓ |

| | | | | | | | | | | | | |
|-----------------|-----------|---|---|---|---|---|---|---|---|---|---|---|
| Windows Phone 7 | Microsoft | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ➤ | ✗ | ✗ | ✗ |
| Bada | Microsoft | ✓ | ✗ | ➤ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |

3.1.1.2. Fonctionnalités supportées par les solutions

| | PHONEGAP | TITANIUM | RHOMOBILE | FLEX | MOBL | OPENPLUG | CORONA | MOSYNC | APPMOBL | QUICKCONNECT | WORKLIGHT |
|----------------------------|----------|----------|-----------|------|------|----------|--------|--------|---------|--------------|-----------|
| Application multi-threadée | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ |
| Modèle de données | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| SQLite | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| LocalStorage | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ |
| Système de fichiers | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Disponibilité réseau | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| Connexion wifi adhoc | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ➤ | ✗ | ✗ | ✗ | ✓ |
| Bluetooth | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ➤ | ✓ | ✗ | ✗ | ✗ |
| SMS | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| Email | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Téléphone | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ |
| API des contacts | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ➤ | ➤ | ✗ | ✓ | ✓ |
| Cartes natives | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Géolocalisation (GPS) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Changement d'orientation | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Clipboard | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Gesture / Multi-touch | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| Audio (Lecture) | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| Audio (Enregistrement) | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ➤ | ✗ | ✓ | ✓ |
| Caméra (Prise de photo) | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| | | | | | | | | | | | |
|-------------------------------------|---|---|---|---|---|---|---|---|---|---|---|
| Caméra (Enregistrement vidéo) | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ➤ | ✗ | ✗ | ✗ |
| Vibration | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Accéléromètre | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Magnétomètre, boussole | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Push Notification | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ |
| XMPP | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| SOAP | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| YQL | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Gestion de la mémoire | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |

III. Les réseaux :

III.1. Définition

Un réseau sans fil (en anglais Wireless Network) est un réseau dans lequel au moins deux périphériques connectés entre eux et peuvent s'envoyer et recevoir des données sans aucune connexion filaire reliant ces différents composants, mais en utilisant des ondes hertziennes ou radio pour établir des liaisons.

Les réseaux sans fil permettent de relier très facilement, des équipements distants d'une dizaine de mètre à quelques kilomètres.

Il existe plusieurs technologies se distinguant de la fréquence d'émission utilisée ainsi que le débit et la portée des transmissions.

Intérêt principal : assurer une connexion au réseau tout en permettant la mobilité de l'utilisateur (déplacement de l'utilisateur tout en restant connecté).

III.2. Evaluation :

Avantages

L'utilisation des réseaux sans fil procure plusieurs avantages, notamment :

- Mobilité ;
- Topologie dynamique ;

- Facilité et réduction du temps d'installation (déploiement) ;
- Gain en coût pour la mise en place d'un réseau sans préalablement la mise en place d'un câblage.
- La réduction des coûts d'entretien ;
- L'augmentation de la connectivité ; etc.

Inconvénients :

Or ce type de réseaux peut présenter quelques désavantages, comme :

- Problèmes liés aux ondes radios : taux d'erreur plus important et interférences (provenant d'autres réseaux).
- Effets sur la santé.
- La sécurité.

III.3. Eléments d'un réseau sans fil :

Hôtes sans fils : PC portable, Tablette, Smartphones..

Station de base : - Connectée typiquement à un réseau câblé.

- Relais : responsable de l'envoi de paquets entre le réseau câblé et les hôtes sans fil dans sa zone.

Liaison sans fil : Typiquement utilisée pour relier des équipements mobiles à la station de base.

III.4. Techniques de transmission sans fil

1) Infrarouge

Les cartes réseaux émettent un signal lumineux très puissant pour émettre vers les récepteurs. Plusieurs techniques infrarouges existent. Selon la technique, le signal peut voyager directement d'une station à une autre ou passer par une antenne commune dont la distance limite est de 100 pieds entre deux postes.

2) Laser

Contrairement à l'infrarouge, le signal ne rebondit pas, il est coupé. Notons qu'il doit être en ligne direct entre l'émetteur et le récepteur. La distance limite est d'environ 5 Km, avec une vitesse de 100Mbps.

3) Fréquence Radio (bande étroite)

L'émetteur et le Récepteur sont ajustés sur une même fréquence radio et le signal émis est de haute fréquence. Cette méthode est lente (5Mbps) avec une distance maximum de 1Km.

4) Fréquence Radio (large bande)

L'émetteur et le Récepteur sont ajustés sur une large gamme de fréquence radio et le signal émis est aussi de haute fréquence. Etant donné la gamme de fréquence disponible, si un problème survient pour une fréquence, le système change automatiquement de fréquence et continue la transmission. Cette méthode est lente et la distance maximum est de 2,5 Km.

III.5. Classification des réseaux sans fil**III.5.1. Classification des réseaux en fonction de la taille**

Les réseaux sans fil sont habituellement répartis en plusieurs catégories, selon le périmètre géographique offrant la connectivité (zone de couverture), on distingue : WPAN, WLAN, WMAN et WWAN.

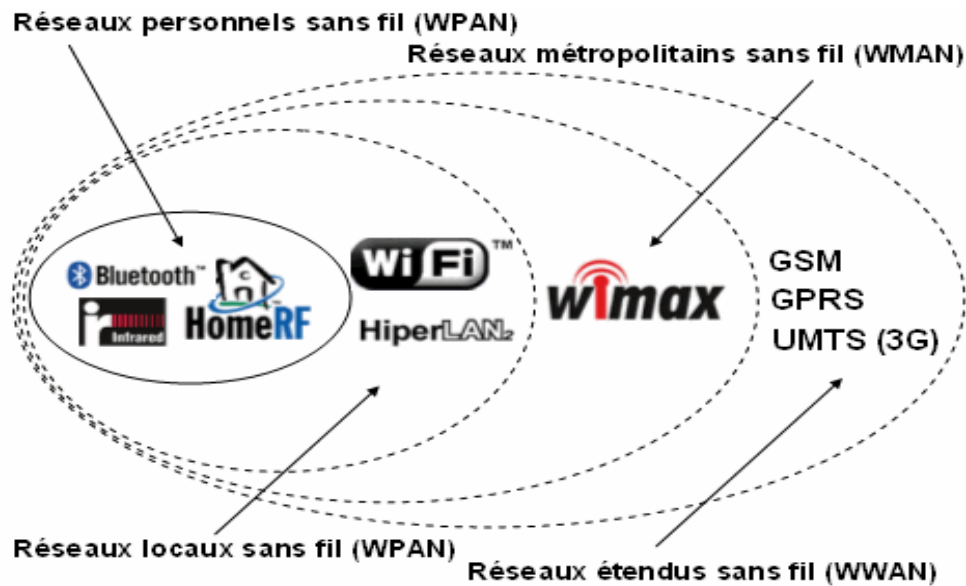


Figure 3: Classification des réseaux sans fil suivant leur taille.

➤ WPAN (Wireless Personal Area Networks)

Dans cette catégorie, on retrouve les réseaux sans fil à l'échelle humaine dont la portée maximale est limitée à une très faible portée autour de l'utilisateur (bureaux, salles de conférence...).

Technologies utilisées:

Bluetooth : standard développé par Ericsson et normalisé par l'IEEE 802.15.1, débit = 1 Mb/s.

Home RF : pour *Home Radio Frequency*, lancée par le HomeRF Working Group (formé notamment par les constructeurs Compaq, HP, Intel, Siemens, Motorola et Microsoft), utilise la bande de fréquence 2,4 GHz, et un débit de 10 Mb/s.

Openair : un standard proche de 802.11 avec un débit de 3 à 10 Mb/s.

Zigbee : une norme sécurisée à faible débit et à faible consommation d'énergie.

Infrarouge : faisceau de lumière qui peut créer des liaisons de quelques mètres pouvant monter jusqu'à quelques Mégabits/s, utilisé pour les télécommandes.

➤ **WLAN (Wireless Local Area Network)**

Réseaux couvrant l'équivalent d'un réseau local d'entreprise (une portée sur une centaine de mètres).

Technologies utilisées:

Wi-Fi: pour *Wireless Fidelity*, un standards issue de la norme IEEE 802.11b/g , débit offert 11 Mb/s pour le IEEE 802.11b et 54 Mb/s pour le IEEE 802.11g , sur une portée de 10 à 200m.

HiperLAN: pour *High Performance Local Area Network*, une technologie développée par l'ETSI (European Telecommunication Standard Institute), utilise une bande de fréquence de 5 GHz, et offre un débit de 54 Mb/s.

➤ **WMAN (Wireless Metropolitan Area Network)**

Appelé aussi BLR (Boucle LocaleRadio); ce type de réseau utilise le même matériel que celui qui est nécessaire pour constituer un WLAN mais peut couvrir une plus grande zone de la taille d'une ville, basé sur le standard IEEE 802.16, il offre un débit de 1 à 10 Mb/s sur 4 à 10 Km.

Technologies:

Wi-Max (World wide Interoperability for Microwave Access) offre un haut débit sur une zone étendue.

➤ **WWAN (Wireless Wide Area Networks)**

C'est la catégorie de réseaux cellulaires mobiles dont la zone de couverture est très large, à l'échelle mondiale.

Technologies:

GSM : *Global System for Mobile communication*, standard de téléphonie mobile très utilisé en Europe.

GPRS: *General Packet Radio Service*, une variante de GSM permettant le transfert de données par paquets comme Internet.

UMTS : *Universal Mobile Telecommunication System*, système de téléphonie mobile (3G) avec un débit de 2Mb/s, permet la vidéo conférence sur téléphone mobile avec une bonne qualité.

III.5.2. Classification des réseaux suivant le mode de fonctionnement

Les réseaux Wifi sont prévus pour deux modes de fonctionnement :

-Le mode sans infrastructure (Ad hoc)

En mode ad hoc, il n'y a pas de point d'accès fixe, l'infrastructure n'est composée que des stations elles-mêmes, ces dernières jouant à la fois le rôle de terminaux et de routeurs pour permettre le passage de l'information d'un terminal vers un autre sans que ces terminaux soient reliés directement. Le mode Ad Hoc est idéal pour échanger des informations dans une salle de réunion qui ne dispose pas d'une infrastructure réseau. Les réseaux Ad Hoc sont donc généralement éphémères : Ils existent le temps de transférer des fichiers.

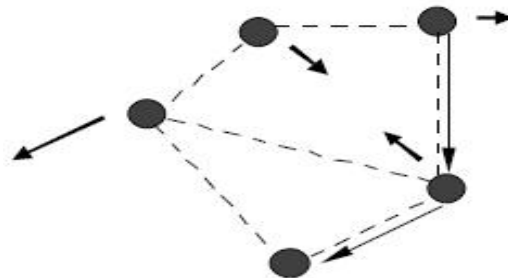


Figure 4: *Le mode Ad Hoc.*

-Le mode Infrastructure

En mode infrastructure, le réseau est composé de plusieurs cellules et chacune d'elles comprend une station de base (ou un point d'accès) par laquelle toutes les autres stations de la cellule accèdent au réseau intra et intercellulaire. Les différents points d'accès sont reliés entre eux et/ou au réseau Internet à l'aide d'une technologie supplémentaire qui peut être filaire ou hertzienne. Dans cette catégorie, on trouve les réseaux WLAN (Wi-Fi), WMAN (WiMAX) et WWAN (GSM).

Chaque réseau sans fil possède son propre nom. Ce nom est ce que l'on appelle le ``SSID: Service Set Identifier " du réseau.

Les périphériques se connectent à ces points d'accès sans fils. La norme IEEE 802.11 définit le protocole que les clients sans fils utilisent pour les connexions. Cet ensemble définit une cellule (BSS: Basic Service Set). Les BSS connectés en sous réseaux constituent l'ESS (Extended Service Set).

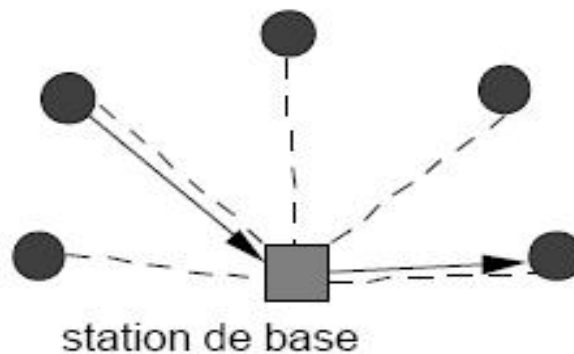


Figure 5: *Le mode infrastructure.*

III.6. Sécurité dans les réseaux sans fil

La sécurité informatique est considérée comme l'un des critères les plus importants dans le jugement de la fiabilité d'un système informatique. Cependant, les réseaux sans fil ne satisfaits pas cette contrainte, ce qui fait d'eux une cible intéressante pour les pirates. Les organisations déploient aujourd'hui la technologie sans fil à un rythme soutenu, souvent sans tenir compte de la fiabilité et leur niveau de sécurité.

Bien que le succès des réseaux sans fil s'explique facilement par les avantages qu'ils procurent : rapidité et simplicité d'installation, une mobilité qui simplifie le déplacement de l'utilisateur (entre les bureaux, les salles de réunion...), accès partagé à des services de haut débit Internet, un certains nombre de problèmes apparaissent, ce qui implique une nécessité d'une mise en place d'une politique de sécurité spécifique et efficace.

III.6.1. Les attaques contre les réseaux sans fil

Les principales attaques contre les réseaux sans fil sont :

Le déni de service :

Le déni de service consiste à rendre inopérant un système afin d'en empêcher l'accès à des utilisateurs réguliers. Cela consiste, par exemple, à saturer le point d'accès en multipliant artificiellement le nombre de demandes. Le point d'accès considère alors que de nombreuses machines veulent se connecter. Par conséquent, la surconsommation d'énergie due à l'obligation de répondre aux sollicitations de l'attaquant provoque un affaiblissement rapide des batteries, c'est ce que l'on appelle un ***déni de service sur batterie***.

Ce type d'attaque est particulièrement difficile à déceler. Les impacts ne sont pas francs et la localisation géographique de la source nécessite des équipements d'analyse radio sophistiqués.

Le sniffing :

C'est l'attaque la plus classique. Par définition, un réseau sans fil est ouvert, c'est-à-dire non sécurisé. Cette attaque consiste à écouter les transmissions des différents utilisateurs du réseau sans fil, et de récupérer n'importe qu'elles données transitant sur le réseau si celles-ci ne sont pas cryptées. Il s'agit d'une attaque sur la confidentialité.

Pour un particulier la menace est faible car les données sont rarement confidentielles. En revanche, dans le cas d'un réseau d'entreprise, l'enjeu stratégique peut être très important.

Le war driving :

Elle consiste à circuler dans des zones urbaines avec un équipement d'analyse à la recherche des réseaux sans fils « ouverts ». Il existe des logiciels spécialisés permettant de détecter des réseaux Wi-Fi et de les localiser géographiquement en exploitant un GPS (Global Positioning System). L'ensemble des informations, relative au réseau découvert, est mis en commun sur des sites Internet dédiés au recensement. On y trouve généralement une cartographie des réseaux à laquelle sont associées les informations techniques nécessaires à la connexion, y compris le nom du réseau SSID et éventuellement la clé WEP de cryptage.

Le spoofing :

Le spoofing consiste à usurper soit l'adresse IP, soit l'adresse MAC d'une autre machine. En modifiant l'adresse I P source dans l'entête du paquet, le récepteur croira avoir reçu un paquet

de cette machine. Si le serveur considérait cette machine comme une machine de confiance, beaucoup de données sensibles pourront être consultées, modifiées, voir même supprimées.

III.6.2. Sécurisation

La sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés.

Ainsi, la sécurité informatique doit être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance.

C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, dont la mise en œuvre se fait selon différentes solutions, telles que :

- L'authentification : le procédé qui consiste à vérifier que seul l'utilisateur autorisé accède au service auquel il a souscrit, et aussi qui permet à ce dernier de reconnaître son réseau sans fil.

L'authentification est un élément important dans la sécurité d'un système d'information. Elle permet d'authentifier toute station voulant s'associer à un réseau. C'est donc une étape nécessaire et très sensible. Si l'authentification n'est pas assurée, l'accès aux données sera plus facile pour les attaquants, ainsi que leurs modifications éventuelles.

- Le chiffrement : Le procédé de chiffrement des données (la cryptographie) est l'outil fondamental de la sécurité informatique. En effet, la mise en oeuvre de la cryptographie permet de réaliser des services de confidentialité des données transmises ou stockées, des services de contrôle d'intégrité des données et d'authentification d'une entité, d'une transaction ou opération. Le chiffrement est l'opération par laquelle on chiffre un message, c'est une opération de codage. Chiffrer ou crypter une information permet de la rendre incompréhensible à ceux ne possédant pas un décodeur particulier.

- La vérification de l'intégrité de données : chaque nœud du réseau envoie les données accompagnées d'une information supplémentaire permettant au récepteur de savoir si les données envoyées n'ont pas été altérées pendant la transmission.

- **Le filtrage des adresses MAC** : Chaque équipement informatique possède une adresse physique qui lui est propre, appelée **adresse MAC** (Media Access Control). C'est un identifiant matériel unique inscrit dans chaque carte réseau, définie une fois pour toute en usine par le fabricant de la carte.

Le filtrage par adresse MAC est une fonctionnalité de sécurité que l'on trouve dans certains points d'accès, elle est basée sur la technique **ACL** (Access Control List), elle consiste à utiliser des listes d'accès. En effet, chaque point d'accès dispose d'une liste où sont inscrites toutes les adresses MAC des stations mobiles autorisées à l'accès. Le point d'accès procède alors à un filtrage sur la base des adresses MAC répertoriées. Chaque liste doit être continuellement mise à jour, manuellement ou par un logiciel spécialisé, afin d'ajouter ou de supprimer des utilisateurs.

III.7. Caractéristiques logiques des réseaux

La conception des premiers ordinateurs ont connu le problème d'hétérogénéité où les concepteurs n'avaient tenu compte de l'aspect matériel au détriment de l'aspect logiciel en oubliant que les données dans les réseaux devaient provenir des différentes applications qui pouvant être différent d'un ordinateur à un autre.

Les réseaux sans fil se réfèrent aux modèles de références OSI et TCP/IP pour l'échange des données. Mais comme ils appartiennent aux réseaux des télécommunications servant d'accès aux réseaux informatiques, leurs équipements correspondent aux équipements du sous réseau. Ces équipements (cartes réseau, points d'accès, stations de bases, etc.) renferment essentiellement les deux couches inférieures du modèle OSI qui correspondent à la couche d'accès réseau du modèle TCP/IP.

III.7.1. Modèle de référence OSI

Avant 1918 il était difficile de connecter les réseaux et les systèmes hétérogènes.

L'interconnexion des réseaux privés s'étend avéré impossible en raison de leur incompatibilité, il est apparu nécessaire de mettre en place un mode standard. Ainsi donc naîtra en 1980 sous l'inspiration de l'**ISO (International Standardization Organization)** l'**architecture OSI**. Dès lors, il est possible de faire communiquer des systèmes hétérogènes.

Ce modèle de référence a été défini en 7 couches pour communiquer entre elles. Il décrit le fonctionnement d'un réseau à communication des paquets.

1. **Couche physique**: Elle assure l'établissement, le maintien de la liaison physique et le transfert de bits sur le canal physique (support). Elle comprend donc les spécifications mécaniques (connecteurs) et les spécifications électriques (niveaux de tension);

2. **Couche liaison de données** : Elle assure le maintien de la connexion logique, le transfert des blocs de données (les trames et les paquets), le contrôle, l'établissement, le maintien et la libération du lien logique entre entités ;

3. **Couche réseau** : Assure des blocs de données entre les deux systèmes d'extrémités, le routage (choix du chemin à parcourir à partir des adresses), lors d'un transfert à travers un système relais, l'acheminement des données (paquets) à travers les différents nœuds d'un sous réseau (routage) et elle définit la taille de ses blocs ;

4. **Couche de transport** est le pivot du modèle OSI. Elle assure le contrôle du transfert de bout en bout des informations (message) entre les deux systèmes d'extrémité. Elle assure le découpage des messages en paquets pour le compte de la couche réseau et les reconstitue pour les couches supérieures. Elle utilise les protocoles TCP et UDP ;

5. **Couche session** : Elle assure l'échange des données, et la transaction entre deux applications distantes. C'est une interface entre les couches qui assurent l'échange de données (transaction) entre les applications distantes. La fonction essentielle de la couche session est la synchronisation et le séquençement de l'échange par la détection et la reprise de celui-ci en cas d'erreur ;

6. **Couche présentation** : Elle assure la mise en forme des données; elle est une interface entre les couches qui assurent l'échange de données et celle qui manipule, celle couche assure la mise en forme de données, les conversions de code nécessaires pour délivrer à la couche supérieure un message dans une syntaxe compréhensible par celle-ci ;

7. **Couche application** : Est la couche située au sommet des couches de protocoles TCP/IP. Elle ne contient pas les applications utilisateurs, mais elle assure la communication, à l'aide de processus, un ensemble de fonctions (entités d'application) permettant le déroulement correct des programmes communicants (transferts des fichiers, etc.).

Ce modèle a pour objectifs, décomposer, structurer et assurer l'indépendance vis-à-vis du matériel et du logiciel.

III.7.2. Modèle de référence TCP/IP

Le but du modèle TCP/IP (Transfer Control Protocol / Internet Protocol) était de permettre une interconnexion des réseaux, en offrant aux utilisateurs un mode commun d'adressage et des protocoles de communication indépendants des technologies utilisées, du nombre et de la position d'équipements d'interconnexion. Il est structuré en quatre couches.

1. La couche d'accès au réseau : Aussi appelée couche hôte réseau, cette couche renferme tous les détails sur la couche physique et la couche liaison de données du modèle OSI.

2. La couche Internet : Le but du protocole IP (Internet Protocol) est de masquer aux utilisateurs (et aux protocoles des couches supérieures) la topologie et la multiplicité du réseau, en adoptant pour ce dernier un adressage universel (au moyen d'adresses IP), et en assurant le routage.

3. La couche transport : Elle renferme deux types de protocoles : le protocole TCP (Transmission Control Protocol) et le protocole UDP (User Datagram Protocol). Tous les deux utilisent IP comme couche réseau.

4. La couche application : Elle gère les protocoles de haut niveau, elle combine les fonctions des couches session, présentation et application du modèle OSI.

IV. Systèmes d'exploitations mobiles :

un système d'exploitation « est un ensemble de programmes qui dirige l'utilisation des capacités d'un ordinateur par les logiciels applicatifs ». Cette définition s'applique aussi aux téléphones intelligents (Smartphones) que nous retrouvons sur le marché car ils possèdent les mêmes caractéristiques qu'un ordinateur à savoir : une mémoire interne de stockage, un système d'exploitation, un processeur etc.

IV.1. Les principaux systèmes d'exploitation mobiles:

Android : est un système d'exploitation open source utilisant le noyau Linux, pour smartphones, tablettes tactiles, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 5 novembre 2007.

Le noyau Linux modifié fournit des pilotes de périphériques, gestion de la mémoire, gestion des processus, ainsi que des fonctionnalités de mise en réseau.

IOS : anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch, et l'iPad. Il est dérivé de Mac OS X dont il partage les fondations (le kernel hybride XNU basé sur le micronoyau Mach, les services Unix et Cocoa, etc.). IOS comporte quatre couches d'abstraction, similaires à celles de Mac OS X : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ». Le système d'exploitation occupe moins d'un demi-gigaoctet (Go) de la capacité mémoire totale de l'appareil.

BlackBerry OS : est un système d'exploitation propriétaire pour téléphone mobile de la gamme BlackBerry, conçu par la société canadienne Research In Motion (RIM).

Il s'agit d'un système multitâche. Le système est surtout connu pour son support natif des courriels à travers le protocole Mobile information device **profile** (MIDP 1.0), et plus récemment un sous-ensemble de MIDP 2.0 qui permet une synchronisation complète avec les messageries d'entreprise telles que Microsoft Exchange ou IBM Lotus Domino.

Windows Phone : est un système d'exploitation mobile développé par Microsoft pour succéder à Windows Mobile, sa précédente plateforme logicielle qui a été renommée pour l'occasion en Windows Phone Classic. Contrairement au système qu'il remplace, Windows Phone 7 est principalement destiné au grand public plutôt qu'au marché des entreprises.

IV. Conclusion :

Dans ce chapitre nous avons apporté un éclairage sur les différentes solutions multiplateforme et technologies réseaux intervenant dans l'élaboration de notre application.

I. Introduction :

Le nombre de plateformes e-learning à base de services web est de plus en plus croissant. Ces plateformes sont de différents fournisseurs et de différentes caractéristiques et fonctionnalités. Par conséquent, leur découverte devient un défi très important. Les critères de choix et de sélection d'un service d'une plateforme e-learning dépendent, généralement, des contraintes pédagogiques, financières, ergonomiques et technologiques.

La fonction principale des plates-formes *e-Learning* (e-formation) est de fournir à l'apprenant les bonnes activités avec les bons outils au bon moment en fonction de ses besoins. Cela nécessite l'application de mécanismes d'animation et de coordination des modules et des activités pédagogiques. Si un système e-Learning est une collection d'activités ou de processus, nous pouvons découper ses fonctionnalités en un certain nombre de fonctions autonomes qui peuvent alors être réalisées séparément sous la forme d'applications autonomes ou de e-services, en utilisant la technologie des services Web.

En effet, une plateforme e-learning peut être vue comme un ensemble de services web qui coopèrent entre eux pour fournir certaines fonctionnalités aux acteurs de la plateforme. Ainsi, il est possible d'utiliser / réutiliser des services externes qui appartiennent à d'autres plateformes e-learning. Mais avant d'utiliser / réutiliser ces services, il est nécessaire de les localiser. Cette localisation (découverte) est une opération importante qui doit être automatique et efficace.

Mais tous d'abord que veut dire un service Web ? En quoi consiste-t-il ? Quel est son apport ? Quels sont ses protocoles de bases ?..

II. Historique : [Cauld, 01]

Les services Web doivent leur origine à l'informatique distribuée et au développement du Web. Au cours des années 1980, les composants ont vu le jour marquant ainsi la fin de l'ère des applications monolithiques hébergeant l'ensemble des fonctionnalités nécessaires, et une nouvelle génération d'applications a vu le jour. Celle-ci est caractérisée par le fait que le code n'est plus compilé en une seule et unique entité, ce qui caractérisait les applications monolithiques, mais scindé en plusieurs composants compilés indépendamment avec des interfaces et une sémantique bien définies. Ces composants sont construits pour être réutilisables et pour communiquer via une infrastructure fournie par le système d'exploitation, ouvrant ainsi la porte à leur exploitation de façon externe. Pour cela, il suffit que l'application

appelante connait les appels de l'infrastructure par composants et l'interface des composants qu'elle veut appeler et réutiliser.

Au fil des années, trois modèles de programmation distribuée à base de composants se sont imposés. Il s'agit de DCOM/COM (Distributed Component Object Model/Component Object Model) de Microsoft, JAVA/RMI (Remote Method Invocation) de Sun Microsystems et CORBA (Common Object Request Broker Architecture) de l'OMG (Object Management Group).

Néanmoins, les modèles de composants distribués sont généralement confinés dans un fonctionnement au sein de réseaux maîtrisés, comme par exemple un intranet d'entreprise. Ils sont complexes à mettre en œuvre, péniblement interopérables et peu compatibles avec les par-feus. Ajoutons à cela qu'ils s'appuient sur des solutions propres à un éditeur.

L'insuffisance des technologies de programmation distribuée existantes, l'évolution d'internet et le succès du Web, ainsi que l'émergence du langage XML comme norme de structuration de données ont tous contribué à favoriser l'essor des services Web.

III. Présentation des services Web

III.1. Définition d'un service Web :

La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes. Les services Web sont basés sur le modèle SOA.

D'autres technologies telles que RMI, DCOM et CORBA ont précédemment adopté ce style architectural mais ont généralement échoué en raison de la diversité des plates-formes utilisées dans les organisations et aussi parce que leur usage n'était pas adapté à Internet (problème de passage à travers des Firewalls, etc.) d'où la lenteur, voire l'absence de réponses sur ce réseau. Les applications réparties fondées sur ces technologies offrent des solutions caractérisées par un couplage fort entre les objets. Les solutions proposées par les services Web, permettent néanmoins un couplage moins fort. De plus, l'utilisation des technologies standards du Web telles HTTP et XML par les services Web facilite le développement d'applications réparties sur Internet, et permet d'avoir des applications très faiblement couplées. L'intégration est sans doute le facteur essentiel qui favorise l'utilisation des services Web.

On retrouve plusieurs définitions des services Web :

Citation : W3C

Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet.

Citation : Dico du Net

Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent.

Citation : Wikipédia

Un service Web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.

En d'autres termes, Un service Web est un composant logiciel qui interagit avec d'autres composants logiciels autonomes au moyen de protocoles universels, via le réseau internet ou n'importe quel réseau local basé sur les protocoles standards du Web, qui utilise un système de messagerie standard XML, et n'est lié à aucun système d'exploitation ou langage de programmation.

En reprenant la définition du consortium W3C, voici les principaux avantages d'un service Web, à savoir:

- son interface décrite d'une manière interprétable par les machines, qui permet aux applications clientes d'accéder aux services de manière automatique;
- son utilisation de langages et protocoles indépendants des plates-formes d'implantation, qui renforcent l'interopérabilité entre services ;
- son utilisation des normes actuelles du Web, qui permettent la réalisation des interactions faiblement couplées et favorisent aussi l'interopérabilité.

III.2. Motivations

Un Web service est un mécanisme qui tend à donner plus d'interactions pour permettre à deux entités hétérogènes (entreprises, clients, applications, etc. ...) de dialoguer au travers du réseau Internet. Les logiciels écrits dans divers langages de programmation (C#, Visual Basic, Java, etc.), sur diverses plateformes (Linux, Windows, etc. ...) et avec diverses architectures peuvent employer des services Web pour échanger des données à travers des réseaux informatique. Chaque Web service doit pouvoir être découvert et invoqué dynamiquement par les applications.

Les aspects purement pratiques n'ont eux rien de fondamentalement novateurs. Au contraire, l'architecture des services Web s'est imposée (tout comme le langage XML) grâce à sa simplicité, à sa lisibilité et à ses fondations normalisées. L'objectif principal des services Web est de faciliter le plus possible l'accès aux applications entre entités et ainsi de simplifier les échanges de données. Cette interopérabilité est due à l'utilisation de normes ouvertes. L'OSI et le W3C sont les comités de coordination responsables de l'architecture et de standardisation des services Web. Pour améliorer l'interopérabilité entre les réalisations de service Web, l'organisation WS-I a développé une série de profils pour faire évoluer les futures normes impliquées. L'aspect le plus important des Web Services est qu'ils reposent sur plusieurs standards qui permettent la structuration des architectures. Cette collection de normes et de protocoles est appelée Web Services Protocol Stack.

Une des raisons principales pour lesquelles les services Web sont employés semble être qu'ils se fondent sur Internet et HTTP pour fonctionner. Pour comprendre ceci, il faut garder à l'esprit que beaucoup d'organisations se sont protégées en employant des firewalls qui filtrent et bloquent beaucoup de trafic d'Internet pour des raisons de sécurité. Dans ce milieu, beaucoup de ports (voire quasiment tous) sont fermés au trafic entrant et sortant, et les administrateurs des réseaux n'ont pas l'envie de les ouvrir. Il en est un qui par contre est toujours ouvert, celui des navigateurs par lequel transite le protocole HTTP. De cette façon les applications peuvent continuer à interagir entre elles et ce sans modifier la configuration de sécurité des organisations.

Si l'on devait résumer les raisons de la création des services Web, les qualificatifs tels que la simplicité des échanges, l'amélioration de la communication entre les applications en seraient les points principaux. En ajoutant à cela l'interopérabilité des programmes indifféremment de leur langage et de leur plateforme, les services Web nous prouvent une nouvelle fois que leur

technologie est très attrayante. Le véritable point fort du concept c'est la normalisation des données au travers de standards connus et acceptés par tous.

IV. Types des services web :

On distingue deux types de services web : SOAP et REST.

IV.1. Services Web SOAP :

1. Présentation :

SOAP (acronyme de *Simple Object Access Protocol*) est un protocole de RPC orienté objet bâti sur XML.

Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP.

Le protocole SOAP est composé de deux parties :

- une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement,
- un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre.

SOAP a été initialement défini par Microsoft et IBM, mais est devenu une référence depuis une recommandation du W3C, utilisée notamment dans le cadre d'architectures de type **SOA** (*Service Oriented Architecture*) pour les Services Web WS-*.

2. Architecture :

Pour comprendre le fonctionnement d'une architecture de services Web SOAP, il faut commencer par revoir certains principes. Si l'on reprend la définition de Mark Colan, Web Service and XML Chief Advocate chez IBM, les Web Services SOAP sont des "applications modulaires basées sur Internet qui exécutent des tâches précises et qui respectent un format spécifique". Les services Web donc ne sont que des applications modulaires qui peuvent être présentées, publiées, situées et invoquées dans un réseau et ce automatiquement. Ainsi, les

applications peuvent faire appel à des fonctionnalités situées sur d'autres machines dans d'autres applications. Au final, on peut affirmer que le but initial d'un service Web est de rendre possible l'utilisation d'un composant applicatif de façon distribuée.

Il s'agit maintenant d'identifier chaque acteur de ses Web services et de comprendre comment ils interagissent les uns avec les autres. Les trois éléments les plus importants des services Web sont les fournisseurs de service, les annuaires de services et les consommateurs de service.

- **Le fournisseur de service (ou serveur) :** correspond au propriétaire du service, qui crée le service Web et publie toutes ces caractéristiques dans l'annuaire. D'un point de vue technique, il est constitué de la plate-forme d'accueil du service.
- **L'annuaire de services:** correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.
- **Le client (ou consommateur) :** correspond au demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service. L'application cliente peut être elle-même un service Web.

Cette architecture fonctionne de la manière suivante :

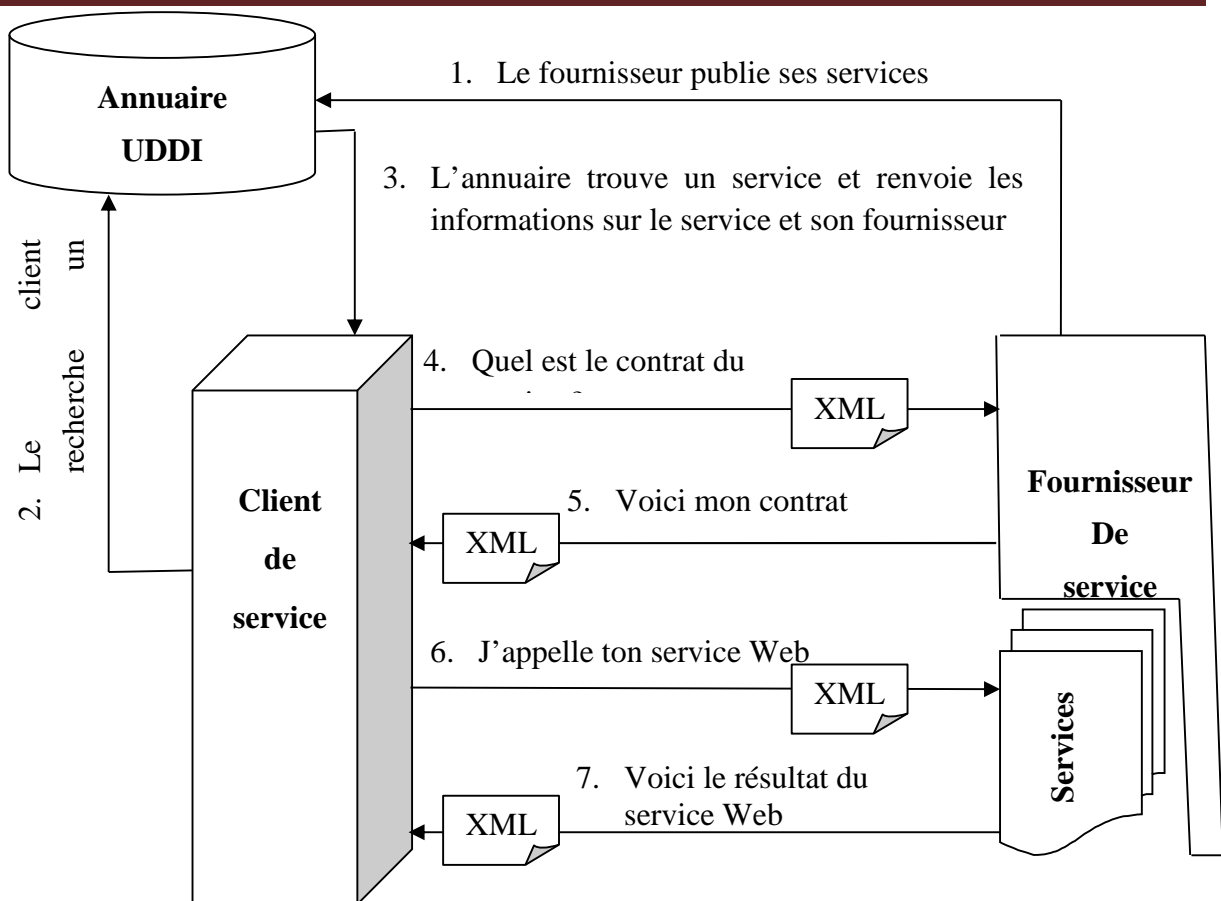


FIGURE 6: Scénario d'utilisation des services Web par les acteurs Client/Fournisseur

1. Le fournisseur de service publie ses Web services sur l'annuaire en utilisant UDDI.
2. Le client envoie une requête à l'annuaire de Service pour trouver le service Web dont il a besoin.
3. L'annuaire cherche pour le client, trouve le service Web approprié et renvoie une réponse au client en lui indiquant quel fournisseur (serveur) détient ce qu'il recherche.
4. Le client envoie une deuxième requête au serveur pour obtenir le contrat du service (quel est le format d'appel du service).
5. Le fournisseur envoie sa réponse sous la forme établie par WSDL en langage XML.
6. Le client appelle le service Web selon son contrat en envoyant un document XML représentant sa requête.
7. Le service Web retourne le résultat de l'appel.

3. Technologies des services Web

L'originalité de l'infrastructure des services Web consiste à les mettre en place en se basant exclusivement sur les protocoles d'Internet tels que HTTP et les formats standards d'échange tels que XML. L'infrastructure des services Web s'est concrétisée autour de trois spécifications considérées comme des standards, à savoir SOAP, UDDI et WSDL [Cerami, 2002].

❖ XML «eXtensible Markup Language»: [W3C-XML]

XML est un standard de représentation de données semi-structurées. Dans un document XML, la structure des données est fournie par le biais de l'utilisation de balises (comme en SGML Standard Generalized Markup Language [Goldfard 1990], mais en s'affranchissant des aspects liés à la présentation des données). Cette structure n'est pas aussi rigide que celle d'une base de données relationnelle par exemple. Dans un document XML, les données peuvent être définies selon un schéma, mais cela n'est pas obligatoire et le schéma peut laisser quelques parties partiellement spécifiées.

❖ UDDI «Universal Description, Discovery and Integration»:

C'est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients.

❖ SOAP « Simple Object Access Protocol »:

C'est le protocole qui assure l'échange de message dans les architectures orientés services SOA. Du fait qu'il est basé sur XML, il permet l'échange de données structurées indépendamment des langages de programmation ou des systèmes d'exploitation. SOAP permet l'échange d'informations dans un environnement décentralisé et distribué, comme Internet, indépendamment du contenu du message. SOAP utilise principalement les deux standards HTTP et XML.

❖ WSDL « Web Service Description Language » :

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

4. Le langage XML (*eXtensible Markup Language*) :**4.1. Définition :**

XML (Langage Extensible de Balisage) est le langage destiné à succéder à HTML. Comme HTML (*Hypertext Markup Language*) c'est un langage de balisage (*markup*) : il représente de

l'information encadrée par des balises. XML est un métalangage, ce qui veut dire que contrairement à HTML qui possède un ensemble de balises de présentation prédéfinies, qui ne sont relatives qu'à la présentation du contenu c'est-à-dire rien ne permet à un logiciel de connaître le sens (la *sémantique*) du texte, XML va permettre d'inventer de nouvelles balises d'isolement d'informations ou d'agrégats élémentaires que peut contenir une page Web. XML a avec HTML un ancêtre commun : le SGML (Standard Generalized Markup Language). L'une de ses caractéristiques était la séparation du formatage et du contenu (le fond et la forme). En effet, le format décrit indépendamment du contenu du document permettait d'obtenir un rendu identique pour une feuille de n'importe quel format. Ce principe est appliqué dans le sens où les données et le schéma du document sont séparés, le schéma représentant la structure et les types de données, incluant la sémantique (importante pour que le document puisse interagir avec différents langages de programmation et systèmes).

4.2. XML et les services Web :

Le but de l'informatique distribuée était de permettre aux systèmes d'information de faire apparaître leurs rouages (bases de données, services distribués, applications, etc.) comme des composants fonctionnant avec une API commune. Cependant, les systèmes distribués à base de composants CORBA, JAVA/RMI et COM/DCOM ne répondaient que partiellement à ce but, du fait que leur implémentation utilisait des interfaces spécifiques, ce qui conduisait à des solutions propriétaires qui réduisaient la production coopérative des documents. L'arrivée des services Web XML qui proposaient un modèle objet distribué proche de celui de CORBA et DCOM sur le plan technique, a fait des systèmes d'information de simples assemblages de composants de services XML indépendants capables de produire et de consommer des documents XML et qui communiquent entre eux à travers un canal de communication. Les services Web XML reposent sur une architecture objets distribués XML/http, et ne nécessitent pas de disposer d'une API commune, ils étendent les modèles JAVA, COM, CORBA à internet et assurent l'extension de l'interopérabilité des composants existants. Le déploiement des services Web (SOAP, WSDL, UDDI) est assuré par le langage XML, qui est utilisé comme un format d'importation et d'exportation d'informations et dans les appels RPC pour l'invocation des procédures à distance. L'apport du langage XML et son impact sur l'informatique distribuée en particulier les services Web, sont résumés dans les points suivants :

- XML est un langage simple, définit un standard de description de données. Les messages XML sont autodescriptifs, ils permettent le traitement des données sans en

connaître le format. De tels messages sont indépendants des systèmes d'exploitation, des plates-formes, des langages de programmation et des formats d'affichage. Ceci facilite l'échange de données entre les différents partenaires.

- XML et les technologies qu'y sont liées (SOAP, WSDL, UDDI) forment un Framework des services Web. Une association qui a révolutionné la manière dont les applications communiquent entre elles. Ce langage a fait des services Web des applications à accès universel à partir de n'importe quel ordinateur ou appareil, indépendamment de sa technologie propriétaire de base.
- Les services Web XML ont fait d'internet un réseau orienté services, sur lequel à peu près tous les services ou applications dont nous pourrions avoir besoin s'y trouvent.
- XML suivi par tous les éditeurs (Microsoft, Sun, Oracle, etc.) ce qui rend disponible les outils nécessaires pour le développement des services Web, tel est le cas pour les plates-formes J2EE de Sun Microsystems et .NET de Microsoft.

5. Le protocole SOAP (*Simple Object Access Protocol*) :

5.1. Définition :

C'est un protocole de dialogue par appels de procédures à distance entre objets logiciels. Sa syntaxe d'utilisation est fondée sur XML et ses commandes sont envoyées sur Internet par l'intermédiaire du protocole HTTP mais aussi SMTP et POP sous forme de texte structuré.

Il permet aux systèmes objets distribués de solliciter et d'obtenir des services rendus par d'autres objets, il est moins lourd à mettre en œuvre que d'autres protocoles et c'est pour cela qu'il est de plus en plus adopté. Le protocole SOAP est une note du Consortium W3C dont Microsoft fait partie, mais qui n'est pas spécifique à Microsoft et Windows. IBM a également participé à l'élaboration de ce protocole.

De plus il existe des implémentations Java, et Borland vient déjà d'implémenter SOAP sous Windows dans Delphi 6 et sous Linux avec Kylix. Bien qu'il soit utilisable avec d'autres protocoles de transport, HTTP est le plus couramment utilisé. Le deuxième standard, XML, utilisé pour la structuration des données sous forme de messages est quand à lui le seul utilisé. Du fait qu'il est basé sur XML, il permet l'échange de données structurées indépendamment des langages de programmation et des systèmes d'exploitation.

5.2. Origines :

- En 1998 : proposition d'un groupe de sociétés :
 - Microsoft, DevelopMentor et Userland Software.

- En 1999 : soumission à l'IETF et émission d'une spécification officielle :
 - Arrivée d'autres acteurs importants : IBM, Lotus, Compac, IONA, Intel...

- En 2001 : Proposition de SOAP 1.1 :
 - 8 mai 2001 : Diffusion des spécifications de SOAP 1.1 par le W3C
 - 17 décembre 2001 : SOAP 1.2 Working draft du W3C.

5.3. Principes de SOAP

SOAP codifie simplement une pratique existante, à savoir l'utilisation conjointe de XML et HTTP. SOAP est un protocole minimal pour appeler des méthodes sur des serveurs, services, composants, objets.

Il a pour avantages de ne pas imposer une API ou un runtime, ni l'utilisation d'un ORB (CORBA, DCOM, ...) ou d'un serveur web particulier (Apache, IIS...) et non plus de ne pas imposer un modèle de programmation.

Une des volontés du W3C vis à vis de SOAP est de "ne pas réinventer une nouvelle technologie". SOAP a été construit pour pouvoir être aisément porté sur toutes les plateformes et les technologies SOAP peut être vu selon deux aspects :

❖ SOAP peut être vu comme un protocole d'échange de "message" :

- La requête contient un seul message (appel sérialisé d'une méthode sur un objet) ;
- La réponse contient un seul message (retour sérialisé d'un appel de méthode sur un objet) ;

❖ SOAP peut être vu comme un format d'échange de documents :

- La requête contient un document XML.
- Le serveur retourne une version transformée.

5.4. Modèle d'échange de message SOAP

En dehors du protocole auquel SOAP est attaché, les messages sont routés sur un chemin appelé un 'message path', qui permet de transmettre un message à un destinataire en passant par un ou plusieurs noeuds intermédiaires.

Une application SOAP recevant un message SOAP doit traiter ce message en faisant les actions suivantes :

- Identifier toutes les parties du message SOAP ;
- Vérifier que toutes les parties mandataires sont supportées par l'application pour ce message et les traiter en conséquence. Si ce n'est pas le cas le message est supprimé ;
- Si l'application SOAP n'est pas la destination finale du message alors le message est forwardé.

5.5. Relation avec le XML

Tous les messages SOAP sont encodés en XML. Une application SOAP doit inclure le namespace SOAP adéquat pour tous les éléments et les attributs définis par SOAP. Elle doit pouvoir supprimer les messages qui ont un namespace incorrect.

5.6. Format d'un message SOAP

SOAP se sert le plus souvent de deux protocoles : HTTP et XML.

Un message SOAP est écrit en XML. HTTP est utilisé comme protocole de transport.

Les messages SOAP vont donc être encapsulés dans HTTP, ce qui permet une utilisation et une compatibilité très importante avec les réseaux et équipements existants. HTTP est le protocole de transport le plus utilisé mais il n'est pas impossible de trouver des implémentations de SOAP sur d'autres protocoles (avec SMTP par exemple).

Un message SOAP est un document XML qui possède une enveloppe SOAP et éventuellement une déclaration XML. L'enveloppe SOAP est composée d'un corps SOAP et éventuellement d'un en-tête.

Les règles de syntaxe sont les suivantes :

- Un message SOAP MUST être codé en XML

- Un message SOAP MUST avoir une enveloppe SOAP
- Un message SOAP CAN avoir un entête SOAP (header)
- Un message SOAP MUST avoir un corps SOAP (body)
- Un message SOAP MUST utiliser l'espace de désignation de l'enveloppe SOAP.
- Un message SOAP MUST utiliser l'espace de désignation d'encodage SOAP.
- Un message SOAP MUST NOT contenir une référence à une DTD.
- Un message SOAP MUST NOT contenir des instructions de type XML Processing.

Un message SOAP va être constitué d'une enveloppe et d'un corps. C'est à l'intérieur du corps (body) que l'on trouve le contenu :

```
//enveloppe du message
<soap:Envelope

//espace de nommage pour l'enveloppe SOAP
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"

//on définit le type d'encodage du message
SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

//espace de nommage pour les types de variables
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

//espace de nommage pour l'appel de méthodes
xmlns:tns="http://soapinterop.org/"

//corps
<soap:Body>

    //espace de nommage pour la fonction getStateName
    <m:getStateName xmlns:m="http://soapware.org/">

//on spécifie un argument à passer à getStatName
    <statenum xsi:type="xsd:int">41</statenum>

    </m:getStateName>

    //gestion d'erreurs
    <soap:fault>
    </soap:fault>

</soap:Body>
</soap:Envelope>
```

FIGURE 7: Format d'un message SOAP

<envelope> est la racine ; <header>, <body> et <fault> sont les enfants.

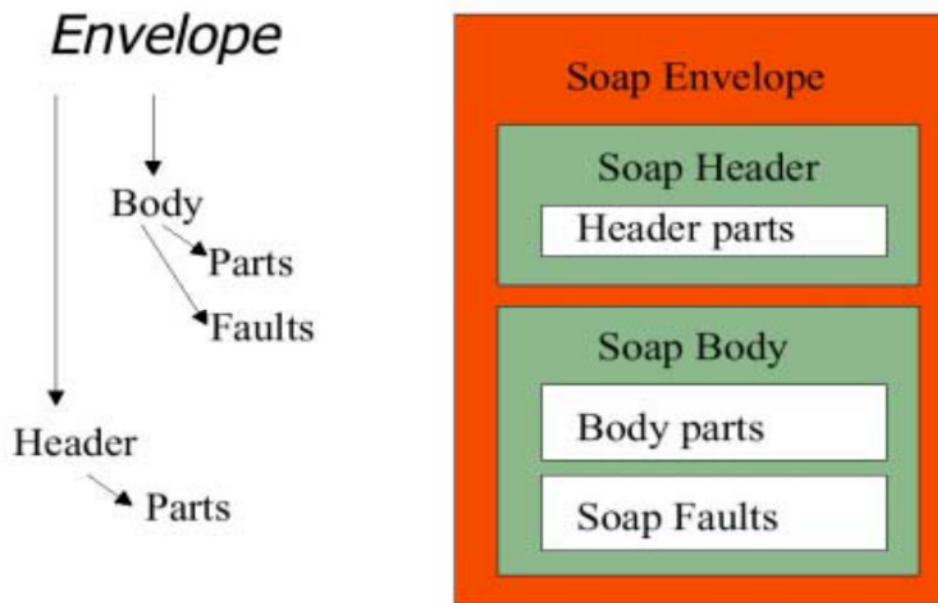


FIGURE 8: Structure d'un message SOAP

❖ L'enveloppe SOAP

Un message SOAP est un document XML qui consiste en une enveloppe mandataire SOAP, un header optionnel et un body. Un message SOAP contient :

- L'enveloppe : est l'élément de base du document XML, qui représente le message ;
- L'entête (header) : est un mécanisme générique qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisé sans agrément entre les parties qui communiquent. C'est ici qu'il est indiqué si le message est mandataire ou optionnel ;
- Le corps (body) : Offre un mécanisme simple d'échange des informations mandataires destinées au receveur du message SOAP. Cette partie contient les paramètres fonctionnels tels que le nom de l'opération à invoquer, les paramètres d'entrée et de sortie des rapports d'erreurs.

Les règles de grammaires sont les suivantes :

☑ **Enveloppe :**

- i. Le nom de l'élément est 'envelope' ;
- ii. L'élément doit être présent dans un message SOAP ;
- iii. L'élément peut contenir une déclaration de namespace et des attributs additionnels.

☑ En-tête (header) :

- i. Le nom de l'élément est 'Header' ;
- ii. L'élément peut être présent dans un message SOAP. S'il est présent, il doit être le premier enfant d'un élément SOAP envelope ;
- iii. L'élément peut contenir une collection d'en-têtes qui doivent être ses enfants immédiats.

☑ Corps (body) :

- i. Le nom de l'élément est 'Body' ;
- ii. L'élément doit être présent dans un message SOAP et doit être un enfant immédiat d'une enveloppe SOAP. Il doit suivre le header s'il est présent. Sinon il doit être le fils immédiat de l'enveloppe ;
- iii. L'élément peut contenir une collection de body qui sont tous les fils immédiats d'un body.

❖ L'en-tête SOAP

SOAP fournit un mécanisme permettant d'étendre un message d'une manière modulaire et décentralisée sans mécanisme d'apprentissage entre les deux parties communicantes. Des exemples d'extension qui peuvent être implémentées comme des en-têtes sont des authentifications, des transactions, des paiements...

L'élément Header est encodé en tant que premier enfant de l'enveloppe SOAP. Tous les enfants immédiats du header sont appelés des entrées du header.

Les règles d'encodage pour les entrées du header sont les suivantes :

- Une entrée d'header est identifiée par son nom d'élément, qui consiste en le namespace URI et le nom local ;
- L'attribut SOAP : 'encodingStyle' peut être utilisé pour indiquer le style de codage de l'entrée ;
- L'attribut SOAP : 'mustUnderstand' peut être utilisé pour indiquer comment traiter l'entrée.

❖ Le corps du message SOAP (Body)

Le corps d'un message SOAP fournit un mécanisme pour échanger les informations mandataires à l'attention du récepteur final du message. L'utilisation typique de l'élément body inclut les appels RPC et les rapports d'erreurs.

L'élément body est encodé en tant que fils direct de l'élément SOAP envelope. Si un header est présent alors le body doit être le fils immédiat de celui-ci.

Tous les enfants immédiats de l'élément body sont appelés des entrées du corps et chaque entrée est encodée comme un élément indépendant dans l'élément body.

Les règles d'encodage des entrées du body sont les suivantes :

- Une entrée body est identifiée par son nom d'élément, qui correspond au namespace URI et au nom local ;
- L'attribut SOAP 'encodingStyle' peut être utilisé pour indiquer le style d'encodage pour l'entrée.

SOAP définit une entrée body, qui est l'entrée 'Fault' utilisée pour le rapport d'erreurs.

Message 'Fault' :

L'élément Fault de SOAP est utilisé pour acheminer des erreurs ou des informations sur le statut à travers un message SOAP. S'il est présent, l'élément Fault de SOAP doit impérativement apparaître comme une entrée du 'body' et ne doit en aucun cas apparaître plus d'une fois à l'intérieur de l'élément 'body'.

5.7. Modèles de données

Les éléments ainsi véhiculés peuvent contenir deux types de donnée :

- les types simples :

| Type | Exemple |
|-----------------|----------------------------|
| Int | 58502 |
| Float | 314159265358979E+1 |
| negativeInteger | -32768 |
| String | Louis 'Satchmo ' Armstrong |

La déclaration d'un type simple s'effectue en deux temps :

1. Il est nécessaire de fixer le type ainsi que le nom de la variable que l'on souhaite manipulée.
2. Comme pour tout langage, nous pouvons ensuite affecter la variable à une valeur.

Exemple:

```
<element name="age" type="int"/>
<element name="height" type="float"/>
<element name="displacement" type="negativeInteger"/>
  <age>45</age>
  <height>5.9</height>
  <displacement>-450</displacement>
```

- Les types composés :

Exemple : Les énumérations

Contrairement aux éléments simples, il faut trois étapes pour utiliser une énumération :

1. Il faut déclarer un nouvel élément qui possède le type que l'on s'apprête à créer
2. Ensuite, on définit les valeurs de l'énumération
3. On utilise l'énumération en affectant la valeur souhaitée.

Exemple :

```
<element name="EyeColor" type="tns:EyeColor"/>
<simpleType name="EyeColor" base="xsd:string">
  <enumeration value="Green"/>
  <enumeration value="Blue"/>
  <enumeration value="Brown"/>
</simpleType>
<Person>
  <Name>Henry Ford</Name>
  <Age>32</Age>
  <EyeColor>Brown</EyeColor>
</Person>
```

5.8. Protocoles utilisés par SOAP

SOAP se sert le plus souvent de deux protocoles : HTTP et XML.

5.9. SOAP et XML :

Les interactions entre services Web s'effectuent par le biais d'envois de messages structurés au format XML. Le protocole SOAP (Simple Object Access Protocol) [W3C-XMLP-Group] fournit le cadre permettant ces échanges. SOAP est originellement issu de tentatives précédentes visant à standardiser l'appel de procédures à distance, et en particulier de XML-RPC [Winer 1999]. Mais à la différence des technologies RPC, SOAP n'est pas fondamentalement lié à la notion d'appel de procédure. En effet, SOAP vise à faciliter l'échange de messages XML, sans se limiter à des messages dont le contenu encode des paramètres d'appel de procédure et sans favoriser des échanges bidirectionnels de type requête-réponse comme c'est le cas des protocoles RPC.

5.10. Le transport :

Le transport est la méthode qui permet d'acheminer un message SOAP à un récepteur via des protocoles de transport comme HTTP.

Les messages SOAP vont donc être encapsulés dans HTTP, ce qui permet une utilisation et une compatibilité très importante avec les réseaux et équipements existants. HTTP est le protocole de transport le plus utilisé mais il n'est pas impossible de trouver des implémentations de SOAP sur d'autres protocoles tels que le protocole d'échange de messages électroniques SMTP, et les protocoles de transport orientés-message de Microsoft et IBM, à savoir MSMQ et MQSeries respectivement.

5.10.1. SOAP via HTTP :

HTTP constitue un excellent transport pour SOAP en raison de sa popularité. La combinaison de HTTP, protocole de transport pour le Web, et de SOAP, meilleur candidat au titre format de messagerie standard, permet d'obtenir un outil très puissant.

L'une des caractéristiques de SOAP est de faire abstraction de la couche de transport sous-jacente. Bien que la pratique la plus répandue soit d'utiliser SOAP au-dessus de HTTP, il existe aussi des implantations de SOAP au-dessus d'autres protocoles. La manière d'implanter SOAP au-dessus d'un protocole de transport donné est appelée une liaison SOAP (« SOAP binding » en anglais).

Une liaison SOAP définit, en particulier, l'encodage des messages (nécessaire en particulier lorsque le protocole sous-jacent utilise un format binaire), la méthode pour l'échange de messages, l'encodage des noms d'opérations (appelés « SOAP Actions »), et la façon dont différents messages (y compris les messages d'erreur) appartenant à la même interaction sont corrélés. Par exemple, la liaison SOAP au-dessus de HTTP définit que les messages sont encodés dans un « type de média » appelé « application/soap+xml » (c'est-à-dire en XML avec quelques extensions), que le nom de l'opération correspondant à une requête est donné dans un en-tête HTTP appelé « SOAPAction », et que les messages dans une interaction sont échangés au travers des méthodes POST et GET fournies par HTTP. D'autres règles définissent la manière dont les messages appartenant à un même échange (y compris les messages d'erreur) sont corrélés en exploitant les caractéristiques des méthodes POST et GET.

Exemple: Message SOAP encapsulé dans une requête HTTP

Voici une requête SOAP typique (avec les en-têtes HTTP) pour un appel à une méthode RPC nommée EchoString, qui prend une chaîne comme paramètre

```
POST /test/simple.asmx HTTP/1.1
Host: 131.107.72.13
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://soapinterop.org/echoString"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://soapinterop.org/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:echoString>
      <inputString>string</inputString>
    </tns:echoString>
  </soap:Body>
</soap:Envelope>
```

Le nom de la méthode est appelé par la balise `<tns:echoString>` et le paramètre de type chaîne est défini par `<inputString>`. La méthode qui est invoqué est donc de cette forme :

```
public String echoString(String inputString);
```

Le serveur répond alors:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://soapinterop.org/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:echoStringResponse>
      <Return>string</Return>
    </tns:echoStringResponse>
  </soap:Body>
</soap:Envelope>
```

6. Le langage WSDL (*Web Service Description Language*)

6.1. Présentation de WSDL :

Afin de décrire les services disponibles sur un serveur, un langage de description de service a été mis en place : le WSDL (*Web Service Description Language*). Ce langage est basé sur XML. Les fichiers WSDL contiennent donc la description de l'accès à des Web services ainsi que des messages qui seront échangés avec les services en question. (Arguments, types de donnée...), et définit les opérations exposées par ce service et le format qu'un client doit employer pour le solliciter. Une fois muni de cette description, appelée aussi parfois « contrat », le client va pouvoir dialoguer avec les services de manière adéquate.

Pour comprendre les différentes zones d'un descripteur WSDL, le schéma suivant où se situent les différents éléments décrits dans un fichier WSDL.

6.2. Description fonctionnelle et structurelle

Le schéma suivant nous montre les différents éléments décrits dans un fichier WSDL.

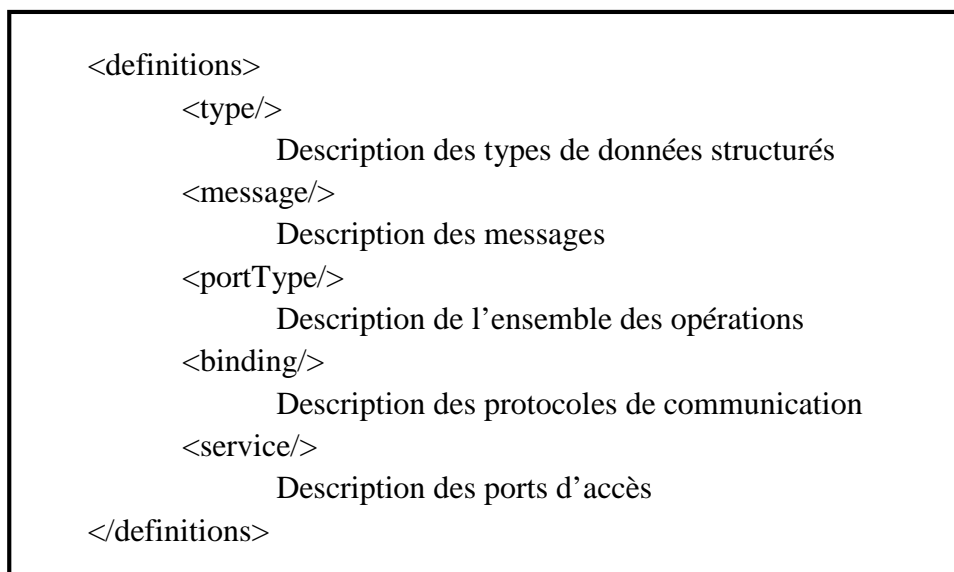


FIGURE 9: Structure générale d'un fichier WSDL 1.1

Un fichier WSDL commence par `<definition>` et finit par `</definition>`. C'est à l'intérieur de cette espace que l'on va déclarer tous les éléments constituant la description.

Nous allons prendre l'exemple du Web Service de Google pour montrer comment se construit un fichier WSDL.

Début de la description des services avec déclaration des différents schémas utilisés

Nom unique identifiant cet ensemble de service au niveau du serveur

```
<definitions name="urn:GoogleSearch"
  targetNamespace="urn:GoogleSearch"
  xmlns:typens="urn:GoogleSearch"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  ...
</definition>
```

a) Services

Un service est la mise à disposition d'une ou plusieurs méthodes. On peut imaginer par exemple une classe avec plusieurs méthodes invocables à distance. La classe sera le service, et chaque méthode sera une opération sur ce service.

La définition d'un service se fait par l'utilisation de `<service></service>`.

b) Port

Pour chaque service on va définir des ports par lesquels ce service sera disponible. En effet, il est possible de rendre disponible un service sur plusieurs supports différents : HTTP GET, SOAP, SMTP...

Définition d'un service

Définition d'un port avec l'adresse du service (ici une URL)

```
<service name="GoogleSearchService">
  <port name="GoogleSearchPort" binding="typens:GoogleSearchBinding">
    <soap:address location="http://api.google.com/search/beta2"/>
  </port>
</service>
```

c) Message

Les messages sont les éléments échangés entre le client et le serveur lors d'une opération sur le port d'un service. Ces messages sont constitués de plusieurs parties (part) qui représentent chacune une donnée avec un type associé.

```

<message name="doGoogleSearch">
  <part name="key" type="xsd:string"/>
  <part name="q" type="xsd:string"/>
  <part name="start" type="xsd:int"/>
  <part name="maxResults" type="xsd:int"/>
  <part name="filter" type="xsd:boolean"/>
  <part name="restrict" type="xsd:string"/>
  <part name="safeSearch" type="xsd:boolean"/>
  <part name="lr" type="xsd:string"/>
  <part name="ie" type="xsd:string"/>
  <part name="oe" type="xsd:string"/>
</message>

```

Définition d'un message

Chaque « part » du message a un nom et un type. Ici un type simple

Utilisation d'un type complexe

```

<message name="doGoogleSearchResponse">
  <part name="return" type="typens:GoogleSearchResult"/>
</message>

```

d) Port Type et Opérations

Une méthode peut être représentée par une opération qui prend un message en entrée et renvoie un message en sortie. Ainsi chaque opération représentée par `<operation></operation>` indique les flux de messages en entrée et en sortie correspondants en définissant les éléments `<input/>` et `<output/>`. Enfin, la collection de toutes les opérations d'un service est rassemblée dans un port Type.

Définition d'une opération (méthode)

```

<portType name="GoogleSearchPort">
  <operation name="doGetCachedPage">
    <input message="typens:doGetCachedPage"/>
    <output message="typens:doGetCachedPageResponse"/>
  </operation>

  <operation name="doSpellingSuggestion">
    <input message="typens:doSpellingSuggestion"/>
    <output message="typens:doSpellingSuggestionResponse"/>
  </operation>

  <operation name="doGoogleSearch">
    <input message="typens:doGoogleSearch"/>
    <output message="typens:doGoogleSearchResponse"/>
  </operation>
</portType>

```

On indique le nom des messages en entrée et en sortie

e) Bindings

Enfin, pour compléter la description, il faut relier certains éléments entre eux. C'est le rôle des *bindings* représentés par les balises `<binding></binding>`. On va spécifier notamment tout ce qui concerne l'encodage des données dans les messages en indiquant les règles que l'on utilise.

Définition de la couche transport utilisée

Pour chaque opération, on définit les règles d'encodage des données

```
<binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="doGoogleSearch">
    <soap:operation soapAction="urn:GoogleSearchAction" />
    <input>
      <soap:body use="encoded"
        namespace="urn:GoogleSearch"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="encoded"
        namespace="urn:GoogleSearch"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
```

Dans la pratique, les développeurs utilisent des outils de développement générant le document WSDL plutôt de rédiger à la main une description de service, ceci parce que dans la plupart des cas, un service Web publie plusieurs méthodes et plus le nombre de méthodes est important, plus la description WSDL devient importante, donc la rédiger manuellement sera une opération très complexe.

7. Le protocole UDDI (*Universal Description Discovery and Integration*)

7.1. Définition:

UDDI a été conçu en 2000 à l'initiative d'un ensemble s'industriels (Ariba, IBM, Microsoft), en vue de devenir le registre standard de la technologie des services Web. Pour convenir à la technologie des services Web, les services référencés dans UDDI sont accessibles par l'intermédiaire du protocole de communication SOAP, et la publication des informations concernant les fournisseurs et les services doit être spécifiée en XML afin que la recherche et l'utilisation soient faites de manière dynamique et automatique. UDDI constitue un méta-service possédant des fonctions de publication et de recherche en fournissant comment interagir avec un service et permet une immatriculation mondiale. Il peut être implanté au niveau mondial, sur Internet, ou au niveau d'une entreprise, sur intranet.

L'annuaire UDDI propose un cadre technique indépendant des plates-formes et ouvert pour permettre aux entreprises d'enregistrer et de découvrir leurs services Web.

UDDI repose sur le protocole SOAP, et de ce fait, les requêtes et réponses sur l'annuaire sont des messages SOAP.

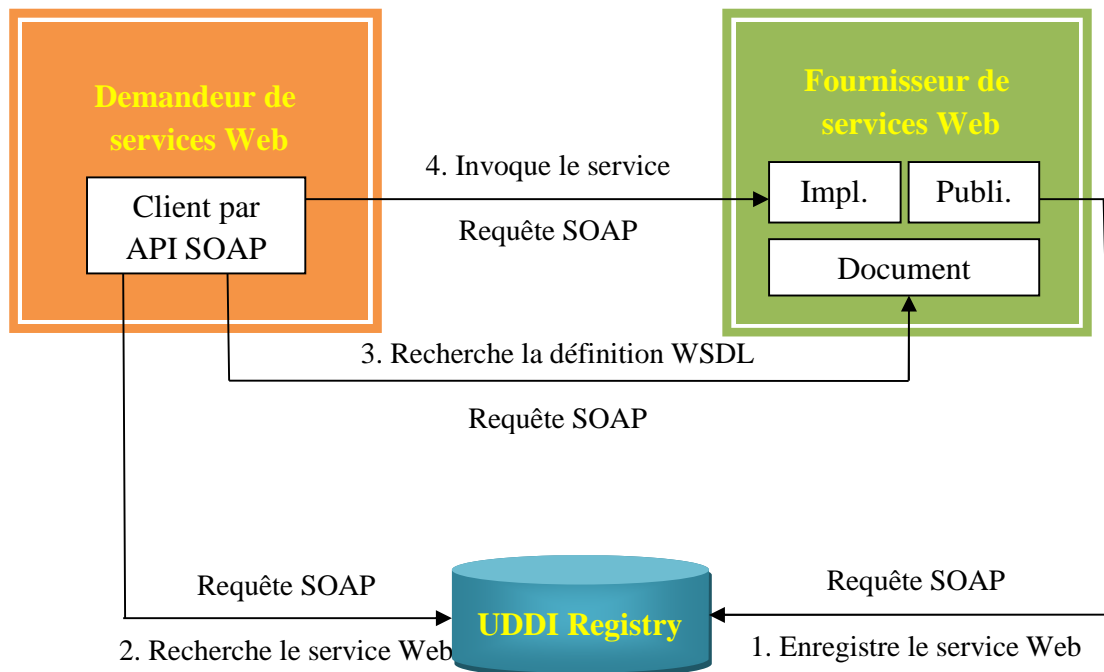


FIGURE 10: Mécanisme d'accès aux services fournis par UDDI Register

7.2. Données du registre UDDI :

La recherche et la sélection dans UDDI reposent sur la publication préalablement décrite du service et de son fournisseur. Le futur client peut connaître par l'intermédiaire de l'UDDI : les fournisseurs de service, les services proposés par un fournisseur donné, les moyens d'invoquer un service. Pour apporter aux clients la réponse à ces questions, UDDI organise l'ensemble des informations qu'il contient en trois catégories, spécifiées en XML. Chacune d'elles peut être utilisée pour faire une recherche via UDDI.

- **Les pages blanches :** ce composant permet de connaître des informations à propos du fournisseur proposant le service. Cette description contient toutes les informations jugées pertinentes pour identifier fournisseur (organisation) telles que : son nom, son adresse physique, numéros de téléphone, fax, sites Web, etc. Le futur client du service retrouve dans les pages blanches les informations que le fournisseur a renseignées dans l'élément *Business Entity* lors de la publication.
- **Les pages jaunes :** les pages jaunes de UDDI contiennent, au format WSDL, la description (non technique) des services renseignée par les fournisseurs eux mêmes. Les services proposés sont répertoriés en catégories.

- **Les pages vertes** : elles comportent les informations techniques liées aux services Web et basée sur leur description WSDL (Processus marqués, descriptions des services rendus, programmation, Plateforme, Implémentation..).

8. Fonctionnement global des services Web SOAP:

Dans le scénario de fonctionnement des services Web, un fournisseur de services (Service Provider) héberge un module logiciel (plate-forme d'exploitation) implémentant un ou plusieurs services Web accessibles via le réseau. Ce module implique des interfaces de connexion permettant : d'être à l'écoute d'une application cliente (généralement un navigateur Web), interrogation dans le cas de la récursivité (un service Web peut appeler un autre service Web) et la publication des fichiers WSDL (enregistrement de la description du service Web sur l'annuaire UDDI). L'annuaire des services est consultable sous plusieurs facettes (pages jaunes, pages vertes, pages blanches), cela dépend des critères transmis dans la requête de recherche (Find) émise du côté de l'application cliente. Ceux-ci peuvent englober les caractéristiques fonctionnelles, caractéristiques techniques des services Web, taxinomie utilisée pour classer l'entreprise qui l'a publié selon son activité, nom ou même la description de l'entreprise qui l'a publié. Suite à cette requête un fichier WSDL (réponse) sera renvoyée à l'application cliente (l'annuaire UDDI suit le modèle requête/réponse), qui le reçoit via son interface de connexion WSDL, ce fichier contient des informations sur le service Web (interfaces des services, emplacement sur le Web) et l'entreprise qui l'a publié. A ce niveau, l'application cliente pourra se connecter physiquement à un emplacement bien précis sur le Web pour interroger le service Web qu'elle vient de découvrir via un protocole de transport (HTTP). Pour ce faire, l'application cliente fait intervenir un processus de sérialisation/désérialisation (Encodeur SOAP) pour générer les messages SOAP, encapsulés dans une requête HTTP. Le service Web réagi à son tour, en envoyant une réponse SOAP via le protocole HTTP.

IV.2. Services Web REST :

1. Présentation

REST est l'Acronyme de **RE**presentational **S**tate **T**ransfert défini par Roy Fielding en 2000 dans la thèse de doctorat à l'Université de Californie ; REST n'est pas un protocole ou un format, contrairement à SOAP, HTTP ou RCP, mais un style d'architecture inspiré de l'architecture du web basé sur le protocole HTTP, ne fait qu'utiliser les principes fondamentaux du Web.

Utilisant le protocole HTTP, REST permet l'envoi de messages sans enveloppe SOAP et dans un encodage libre (XML, JSON, binaire, simple texte). Il est actuellement très utilisé par les sites communautaires (réseaux sociaux) leur permettant de proposer à leurs clients une API facile à utiliser. Des sites comme Facebook, Twitter proposent ainsi de telles API évitant à leurs clients de devoir passer par la case SOAP.

Les Web Services RESTful sont basés sur un type d'architectures fondées sur les concepts de ressources. Le principe est qu'un composant logiciel puisse lire ou modifier une ressource en utilisant une représentation (XML ou JSON par exemple) de cette ressource.

Les services basés sur REST utilisent le protocole HTTP, défini par la norme RFC2616 [voir [Fielding et al. \(99\)](#)] pour permettre au serveur et au client de communiquer ensemble. Ce protocole fournit les opérations nécessaires à la manipulation des ressources : GET, POST, PUT, DELETE.

On parle de RESTful pour tout système suivant l'architecture REST.

2. Architecture

REST est une architecture orientée ressource : une ressource est l'information de base dans une architecture REST.

Les architectures RESTful sont construites à partir de ressources uniquement identifiées : chaque ressource est identifiée unitairement et accessible par son URI unique pour procéder à diverses opérations GET, POST, PUT et DELETE, qui sont des opérations supportées nativement par HTTP.

La manipulation des ressources se fait à travers des représentations : les ressources ont des représentations définies.

Les services REST sont sans états (Stateless) Chaque requête envoyée au serveur doit contenir toutes les informations relatives à son état et est traitée indépendamment de toutes autres requêtes.

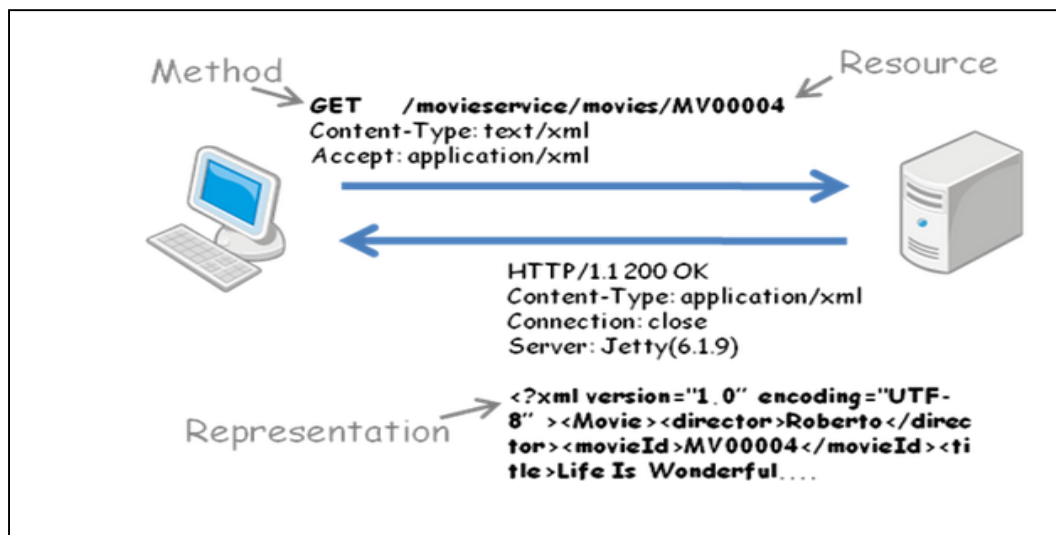


FIGURE 11: schema représentant l'architecture REST.

2.1 Ressource:

L'abstraction principale de l'information dans REST est la ressource. Toute information pouvant être nommée peut être une ressource : un document ou une image, un service temporel (par exemple «le temps d'aujourd'hui à Marseille»), une collection d'autres ressources, un objet non-virtuel (par exemple une personne), ainsi de suite. En d'autres termes, tout concept pouvant être la cible d'une référence hypertexte d'un auteur doit entrer dans la définition d'une ressource. C'est une correspondance conceptuelle à un ensemble d'entités et ce n'est pas l'entité correspondant à cette association à un moment particulier dans le temps. [Thèse de Roy T. Fielding - Traduction du Chapitre 5: REST]

Cette définition abstraite d'une ressource fournit les concepts clés de l'architecture du Web. Elle permet de généraliser en enveloppant de nombreuses sources d'information sans les distinguer artificiellement par leur type ou leur mise en œuvre. Ensuite, elle permet de faire une liaison entre une référence et une représentation, permettant ainsi une négociation de contenu se basant sur les caractéristiques de la demande. Enfin, elle permet à un auteur de mettre en exergue un concept plutôt qu'une représentation donnée de ce concept. Ainsi, la nécessité de changer tous les liens existants à chaque fois que la représentation change n'existe plus (en partant du principe que l'auteur a utilisé le bon identifiant). [Thèse de Roy T. Fielding - Traduction du Chapitre 5: REST]

REST utilise un identifiant de ressource URI pour identifier la ressource particulière impliquée dans une interaction entre les composants (Client et Serveur).

Une ressource est un objet identifiable sur le système (Cours, Etudiant, Note) par une URI unique qui a un rôle important et doit suffire pour identifier une ressource sur le Service web, exemple :(http://localhost:8080/Univers_elearning/webapi/cours/22).

Une ressource peut subir quatre opérations de bases CRUD correspondant aux quatre principaux types de requêtes HTTP (GET, PUT, POST, DELETE).

2.2 Représentation:

Une représentation désigne les données échangées entre le client et le serveur pour une ressource, ce format peut être sous différents types : JSON XML XHTML Text/plain.

Les composants REST effectuent des actions sur une ressource en utilisant une représentation pour capturer l'état courant de cette ressource et en transférant cette représentation entre les composants. Une représentation est une séquence d'octets. Des noms comme document, fichier, entité de message HTTP, instance ou variante sont utilisés pour désigner une représentation de façon générale.

La conception d'un type de média (xml,json,html) peut avoir un impact direct sur la perception de performance d'un système réparti d'hypermédia que peut avoir l'utilisateur. Un format de données qui place l'information de la plus importante au tout début, a pour résultat une meilleure perception de performance par l'utilisateur par rapport à un autre système qui ne le fait pas.

2.2.1 JSON :

JSON est l'acronyme de « JavaScript Object Notation » qui est un format d'échange de données.

Basé sur JavaScript, il est indépendant du langage de programmation. Le site officiel présente (en juin 2013) des solutions d'intégration de JSON pour 55 langages de programmation. Il sert à faire communiquer des applications dans un environnement hétérogène.

JSON est constitué de trois types de données :

JSONObject : commence par un « { » et se termine par « } » et composé d'une liste non ordonnée de paire clefs/valeurs. Une clef est suivie de « : » et les paires clef/valeur sont séparés par « , »

JSONArray : liste ordonnée d'objets commençant par « [» et se terminant par «] », les objets sont séparés l'un de l'autre par « , », un JSONArray est une collection de JSONObject .

Value : Un objet JSON peut être soit un string entre « "" » ou un nombre (entier, décimal) ou un boolean (true, false) ou un objet ou nul .

Ces types de données sont suffisamment génériques et abstraits pour pouvoir être représentés dans n'importe quel langage de programmation, aussi pouvoir représenter n'importe quelle donnée concrète.

Le principal avantage de JSON est qu'il est simple à mettre en œuvre par un développeur.

Il est notamment utilisé comme langage de transport de données par AJAX et les services Web : le type MIME *application/json* est utilisé pour le transmettre par le protocole HTTP.



FIGURE12 : Comparaison entre JSON et XML

2.3 Stateless :

Toutes les interactions REST sont sans état. C'est-à-dire que chaque requête contient toutes les informations nécessaires pour qu'un connecteur (client et serveur) puisse comprendre la demande et ce indépendamment de toutes les requêtes qui ont pu l'avoir précédées.

Mais cette contrainte diminue la performance du réseau en répétant les données envoyées dans une série de requêtes.

Cette contrainte accomplit quatre fonctions :

- enlève tout besoin pour les connecteurs de maintenir l'état de l'application entre les requêtes, réduisant ainsi la consommation de ressources physiques;
- permet à des interactions d'être traitées en parallèle sans exiger de compréhension sémantique par le mécanisme de traitement;
- permet à un intermédiaire de regarder et de comprendre une requête de façon isolée, ce qui peut être nécessaire quand des services sont modifiés dynamiquement; et,
- force toutes les informations qui pourraient être factorisées dans la réutilisation d'une réponse en cache à être présente dans chaque requête.

Un autre type de connecteur, le connecteur de cache, peut être situé sur l'interface d'un connecteur client ou serveur afin de conserver des réponses, pouvant être mises en cache, relatives aux interactions actuelles, de sorte qu'elles puissent être réutilisées pour des requêtes ultérieures. Un cache peut être employé par un client pour éviter la répétition de communication réseau, ou par un serveur pour éviter de répéter le processus de production d'une réponse. Dans les deux cas, il sert à réduire la latence des interactions.

2.4 Méthodes HTTP:

Le protocole HTTP décrit les instructions détaillées de la façon dont un navigateur doit envoyer ses requêtes au serveur.

HTTP permet au navigateur de récupérer divers types d'informations à partir du serveur, la récupération d'une page HTML, ou la soumission d'un formulaire. En fait, HTTP permet d'effectuer huit types de requêtes différentes sur le serveur.

REST s'appuie sur le protocole HTTP pour réaliser Les opérations CRUD sur les ressources :

- CREATE → POST.
- READ → GET.
- UPDATE → PUT.
- DELETE → DELETE.

Méthode GET: La méthode GET renvoie une représentation de la ressource telle qu'elle est sur le système.



FIGURE 13: La Requête GET.

Méthode POST : La méthode POST crée une nouvelle ressource sur le système

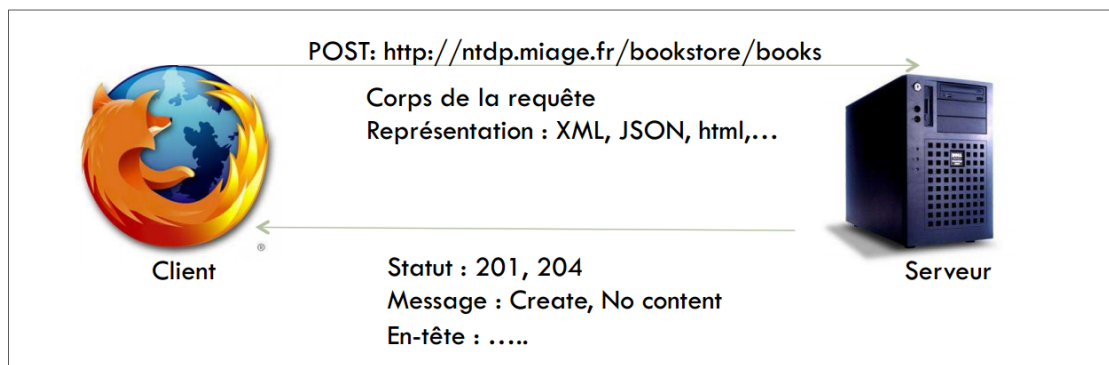


FIGURE 14: La Requête POST.

Méthode PUT : Mise à jour de la ressource sur le système



FIGURE 15: La Requête PUT

Méthode DELETE: Supprime la ressource identifiée par l'URI sur le serveur.



FIGURE 16 : La Requête DELETE.

3. JAX_RS :

Acronyme de **Java API pour RestfulWeb Services** qui est une spécification qui décrit la mise en œuvre des services REST web coté serveur. Son architecture se repose sur l'utilisation des classes et des annotations pour développer les services web.

Plusieurs implémentations sont développées autour de cette spécification :

JERSEY : implémentation de référence fournie par Oracle (<http://jersey.java.net>)

CXF : Fournie par Apache (<http://cfx.apache.org>)

RESTEasy: fournie par JBOSS

RESTLET : L'un des premiers framework implémentant REST pour Java.

3.1 Les annotations JAX_RS :

La spécification JAX-RS dispose d'un ensemble d'annotation permettant d'exposer une classe java dans un service web :

@Produces, @Consumes @PathParam @Path @GET, @POST, @PUT, @DELETE

@Path : définit le chemin de la ressource. Cette annotation se place sur la classe et/ou sur la méthode implémentant le service.

@Produces : spécifie le ou les Types MIME de la réponse du service

@Consumes : spécifie le ou les Types MIME acceptés en entrée du service

@PathParam: permet d'associer un morceau de l'URL de requête à un champ ou un paramètre.

@GET, @POST, @PUT, @DELETE : ne sont utilisables que sur des méthodes, définit le verbe HTTP implémenté par le service permettent de mapper une méthode à un type de requête http.

V. SOAP VS REST :

V.1. Evaluation

REST:

Avantages :

- simplicité de mise en œuvre
- représentations multiples (XML, JSON, TEXT, HTML, ...).
- requête simple sous forme d'URI pour appeler la ressource avec des verbes HTTP.
- REST est Stateless ce qui lui permet d'avoir un découpage client /serveur qui évoluent indépendamment d'ou un faible couplage.

Inconvénients :

- sécurité restreinte par l'emploi des méthodes HTTP.
- utilise uniquement le protocole HTTP comme protocole de transport.

SOAP:

Avantages:

- standardisé s'adapte à différents protocoles de transport.
- sécurité (WS-Security).

Inconvénients :

- enveloppe SOAP supplémentaire qui réduit la performance.
- Un protocole orienté sur du XML ce qui cause une lourdeur dans l'échange d'appel.
- Fort couplage client /Serveur car l'appel se base sur le WSDL qui décrit la manière dont la communication doit se faire, une modification dans le serveur implique un changement cote client.



FIGURE 17: SOAP VS REST

V.2. Argumenter le choix du service web :

Notre projet a pour but de mettre en ligne des services notamment la consultation des cours, PV de note, sur des plateformes mobiles. Pour répondre à ce besoin, nous avons mis en place une architecture distribuée, basée sur un Web Service. En effet, les caractéristiques des Web Services ont orienté notre choix :

- ils permettent de mettre en place l'architecture distribuée pour l'utilisation d'une plateforme mobile
- interaction entre systèmes hétérogènes, distants, avec couplage faible ;
- utilisation des protocoles internet, notamment le protocole standard HTTP ;
- échange de message JSON possible ;
- utilisation de clients légers possibles, type navigateur web.

Il existe une grande variation de plateformes clientes Smartphone, et qui possède des caractéristiques matérielles, des systèmes d'exploitation et des langages de programmation d'application différents, cas iPhone, Windows Phone et Android par exemple. L'architecture de services Web choisie est REST. Ce choix a été fondé de plusieurs critères :

- léger car basé uniquement sur le protocole HTTP, et donc moins de complexité engendrée par l'utilisation des différents intermédiaires des architectures SOAP par exemple ;
- architecture orientée ressources par rapport à une architecture orientée protocole ;

- interface d'accès et de manipulation des ressources uniformes : GET, POST, DELETE du protocole HTTP ;
- facilité d'extension dans un Service type REST.

Pour implémenter cette architecture, nous avons utilisé de le framework Open Source Jersey, car il correspondait bien à nos besoins, qui assure la conversion `JavaObject` vers un `JsonObject` qui est le format de donnée supportée par les mobiles.

VI. E_learning et les services web :

Des auteurs tels que (Abel et al., 2003) rappellent que le *e-Learning* peut désigner des notions aussi variées que la gestion administrative d'une formation sur Internet, la diffusion d'un cours basée sur le web ou la mise à disposition d'outils de conférences virtuelles. Dans la pratique, le terme *e-Learning* désigne souvent de nouveaux services techniques : on l'associe aux cours numériques, à l'enseignement par Internet, ou encore à l'apprentissage en ligne. Nous le voyons plus globalement comme une évolution des systèmes d'information et des dispositifs d'enseignement.

Les efforts de standardisation dans ce domaine se focalisent sur la réutilisation des documents pédagogiques, mais ne concernent pas la réutilisation des fonctionnalités des applications. Les plates-formes, aussi bien commerciales que libres, sont en général centralisées, offrant des cours avec un contenu bien défini.

Un système *e-Learning* est une collection d'activités ou de processus qui ont un effet d'une part sur les étudiants et d'autre part sur le contenu pédagogique convenablement choisi sous forme d'objets pédagogiques (*Learning Objects*). Ainsi, nous pouvons découper les fonctionnalités de base d'un système *e-Learning* en un certain nombre de fonctions, qui peuvent alors être implantées séparément sous la forme d'applications autonomes (*stand-alone*) ou sous la forme de e-Services en utilisant des services web. La mise à disposition de ces services permet la réutilisation du contenu et des fonctionnalités dans une plate-forme *e-Learning*.

VII. L'approche services web :

La publication et la recherche des objets pédagogiques peuvent se faire dans un cadre UDDI faisant partie de l'architecture des services web. Ses caractéristiques assurent le stockage de données concernant la description des objets pédagogiques, c'est-à-dire, des méta-informations, alors que le contenu réel de ces mêmes objets pédagogiques est sauvegardé dans

des sites distribués des auteurs de cours ou des intervenants d'enseignement. A partir de ce schéma de fonctionnement nous identifions certains problèmes surgissant dans la réalisation d'une plate-forme de e-Services : le stockage du contenu pédagogique de manière distribuée, et l'échange dynamique de ce même contenu si nécessaire. En utilisant un tel schéma fonctionnel, le contenu pédagogique serait publié et organisé pour être échangé, et serait accessible dans un environnement fondé sur des services Web.

Dans un système e-Learning, la variété des dispositifs et des composants peut être perçue comme une variété de processus et par conséquent réalisée en tant que services web atomisés ou composés. A partir de différentes sources : (PCBI, 2003) et (Vossen, 2004), nous identifions une liste non exhaustive composants d'un système e-Learning : conception d'un contenu pédagogique ; publication d'un cours à partir d'un contenu choisi ; gestion des objets pédagogiques ; mise à jour d'un contenu pédagogique ; adaptation d'un contenu à la demande ; recherche et présentation d'un contenu pédagogique ; inscription d'un étudiant et gestion de son compte et profil ; évaluation des connaissances acquises, mise en place et gestion d'une classe virtuelle ; gestion d'un système de communication synchronisé de type *chat*. Ainsi, le fonctionnement est décomposé en différentes activités, ou groupes d'activités, qui peuvent être mis en application de façon indépendante sous la forme de services. Le fonctionnement intégral d'un système peut être reconstitué par une composition appropriée des services définis et choisis.

VIII. Conclusion

En conclusion, l'approche services web fournit des avantages pour un grand nombre d'apprentissages et d'apprenants. En particulier, dans l'enseignement supérieur, ces avantages peuvent se ressentir dans le cadre du cursus LMD (Licence, Master, Doctorat) et pour une poursuite de formation à la demande, mieux adaptée aux besoins de chacun. En effet, dans la société contemporaine, des besoins de formation sont nécessaires tout au long de la vie professionnelle, à cause de la mobilité et des mutations technologiques. Le développement des technologies web apporte à l'étudiant une souplesse dans le choix de sa formation.

I. Introduction :

Pour développer des applications informatiques, il est nécessaire d'utiliser une démarche méthodologique et rigoureuse. Alors le choix d'une méthode est très important.

Dans le cadre de notre projet nous avons optés pour le langage de modélisation unifié (UML) qui est comme son nom l'indique est l'unification des méthodes de modélisation par objet à savoir (OMT, BOOCH, OOSE).

Plusieurs raisons nous ont conduits à adopter UML dans notre modélisation. En effet :

- ✓ Il a été normalisé par l'OMG, qui est une organisation internationale se chargeant de la standardisation des technologies objets, c'est donc un langage standard compréhensible par tout le monde.
- ✓ Malgré son évolutivité, il est stable et facilite la compréhension du système grâce à ses représentations simples.
- ✓ Il permet d'utiliser les principes et les concepts objet pour enrichir la phase de la conception des systèmes.

Dans ce quatrième chapitre, nous proposons de modéliser une plate-forme web de documentation ouverte et à distance respectant au mieux les comportements pédagogiques d'un organisme de formation classique.

II. La démarche d'élaboration du projet :

Les étapes de notre projet s'appuient sur la modélisation UML, pour cela nous avons suivi les étapes suivantes :

1-A partir de la définition des besoins, nous allons identifier les acteurs et les activités, desquels nous déduisons les cas d'utilisation.

2-Ceux-ci nous permettent d'établir un ensemble de scénarios d'utilisation standard à l'aide des diagrammes de séquences.

3-Ces diagrammes de séquence nous permettent d'aboutir au diagramme de classe.

Enfin, nous décrirons l'architecture physique des composants matériels qui supportent l'exécution de notre application par un diagramme de déploiement.

III. Présentation d'UML :

III.1. Introduction à l'UML

UML (Unified Modeling Language), que l'on peut traduire par "*langage de modélisation unifié*" est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, il est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

UML n'est pas une méthode dans la mesure où elle ne présente aucune démarche. A ce titre *UML* est un formalisme de modélisation objet. Le mot méthode parfois utilisé par abus de langage ne doit donc pas être entendu comme une démarche.

IV. Objectif du projet :

Notre travail consiste à faire fonctionner notre application e-learning sur différents systèmes d'exploitation rendant ainsi l'accès ouvert à un grand nombre d'utilisateurs utilisant une multitude d'appareils mobiles.

V. Architecture du système :

Le système mis en place est constitué de deux modules complémentaires :

➤ **Module serveur** : est modélisé par l'ensemble des services web mis en œuvre pour la manipulation, gestion et récupération des ressources sollicitées par les clients, depuis une base de données.

➤ **Module client** : C'est l'application mobile facilitant la consultation du compus numérique et gère la manière dont sont affichées les informations sur l'interface.

Notre application mobile est sensée être multiplateforme c'est-à-dire s'adapter aux diverses plateformes mobiles en faisant abstraction des contraintes logicielles et matérielles liés aux terminaux, permettant ainsi d'atteindre un nombre important d'utilisateurs quelque soit leur terminal numérique, quelque soit l'endroit où ils se trouvent : qu'ils utilisent un PC sous Windows au travail, un iPad au café du coin ou un téléphone Android en marchant dans la rue.

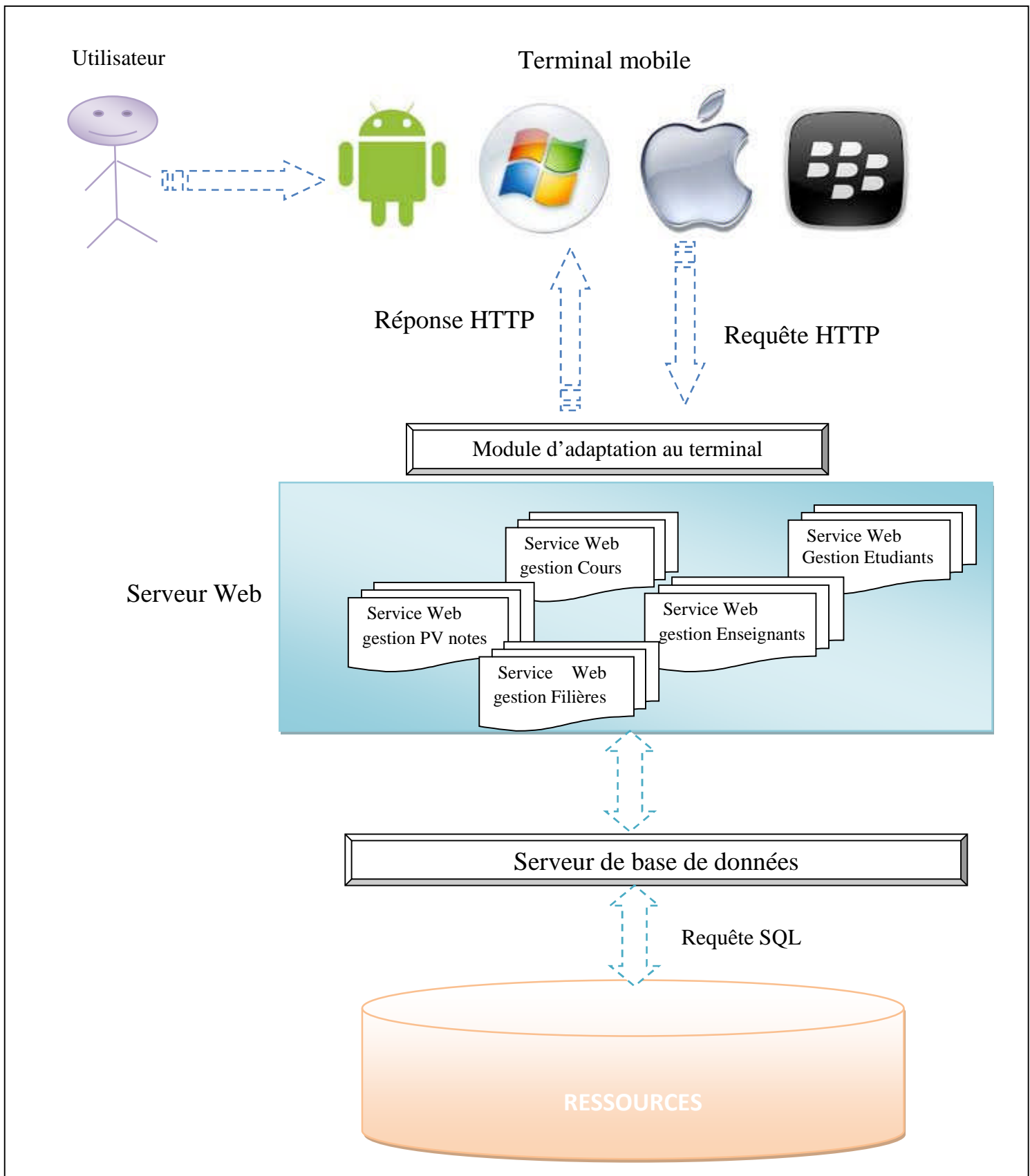


FIGURE 18: Architecture générale du système

VI. Spécification des besoins :

VI.1. Identification des acteurs et des besoins

Dans l'organisation humaine du projet, on distingue principalement les acteurs suivants : l'étudiant, l'enseignant et l'administrateur de l'application.

Etudiant : toute personne inscrite et suivant ses études au sein de l'université. Après une inscription, le système lui réserve un espace propre à lui. L'étudiant peut ensuite après chaque authentification :

- ✓ Consulter et télécharger les cours relatifs aux modules de sa filière d'inscription.
- ✓ Consulter et télécharger les PV de notes obtenus suite aux EMDs des différents modules de sa filière.

Enseignant : toute personne chargée d'enseigner au sein de l'université et d'aider ses étudiants durant leur cursus. L'enseignant peut ensuite après chaque authentification :

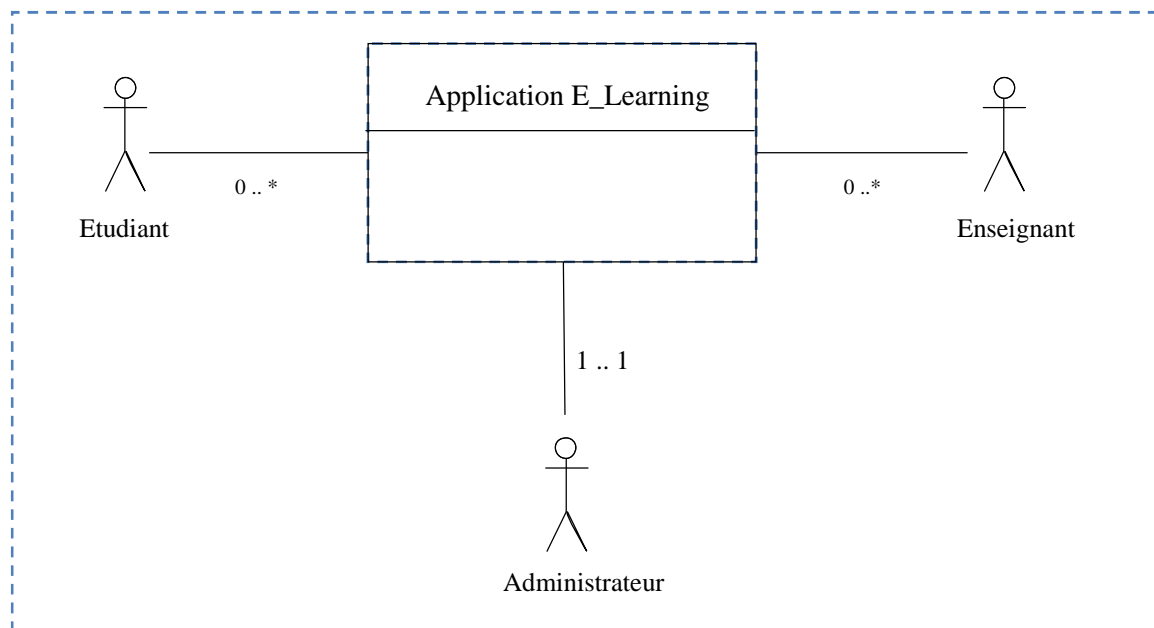
- ✓ Enrichir le campus numérique de l'université par des cours adressés à la communauté estudiantine.
- ✓ Publier les PV après chaque EMD.

Administrateur : une personne ayant pour mission la gestion et la maintenance de la plateforme e_learning à distance.

VI.2. Le diagramme de contexte

Notre application est un système multi-utilisateurs : à tout instant, on peut avoir plusieurs instances de ses différents utilisateurs (étudiant, enseignant).

La définition des besoins présentés ci-dessus nous a permis de dégager le diagramme de contexte suivant :

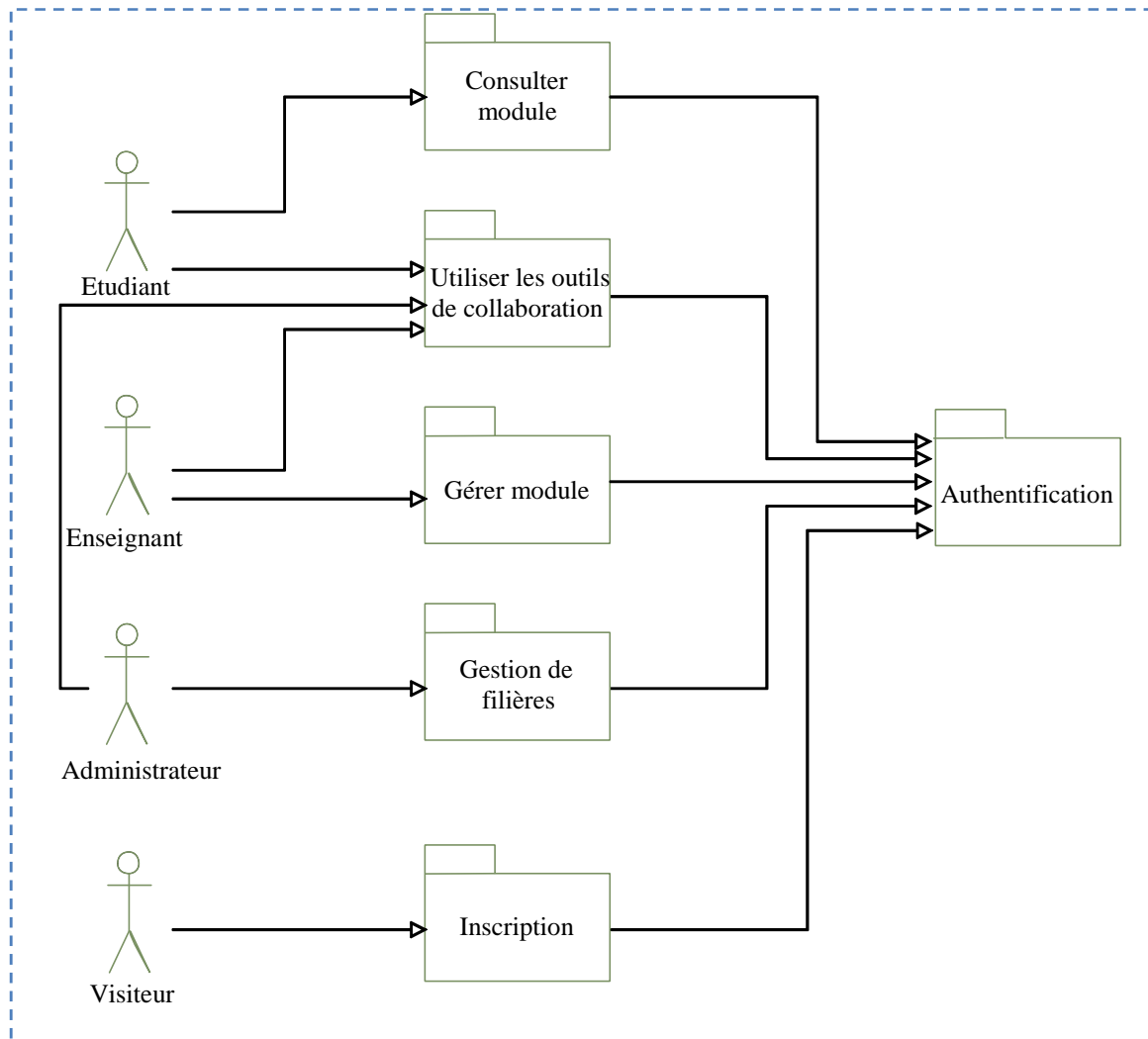
**FIGURE 19 : Diagramme de contexte**

VII. Structuration des cas d'utilisation :

Nous les regroupons par domaine fonctionnel en faisant figurer l'acteur principal devant les packages d'activités auxquels il a accès :

- L'étudiant peut accéder aux packages d'activités suivants : Consulter les modules, utiliser les outils de collaboration, s'authentifier, s'inscrire.
- L'enseignant peut accéder aux packages d'activités suivants : Gérer le module, utiliser les outils de collaboration, s'authentifier, s'inscrire.
- L'administrateur peut accéder aux packages d'activités suivants : Gestion des filières, s'authentifier.
- Le visiteur peut accéder aux packages d'activités suivants : S'inscrire.

La figure suivante modélise les cas d'utilisation en package de ce système :

**FIGURE 20: Structuration des cas d'utilisation**

VII.1. Détermination des cas d'utilisation

Partant de l'identification des acteurs et des packages de leurs activités, nous sommes en mesure de modéliser les cas d'utilisation de ces acteurs, qui sont des détails des packages définis dans la figure précédente.

Voici les différents cas d'utilisation que nous avons modélisés :

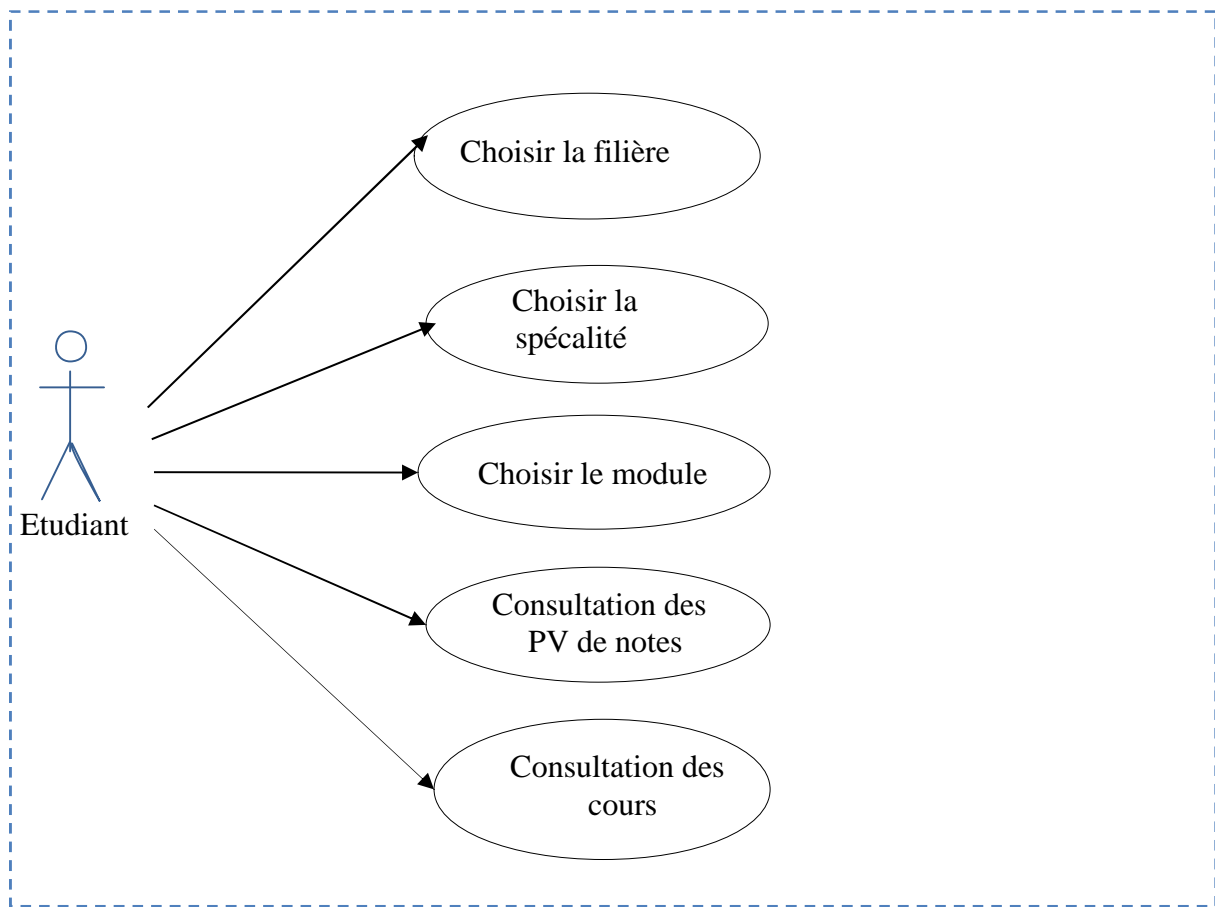


FIGURE 21 : Diagramme de cas d'utilisation « consulter module »

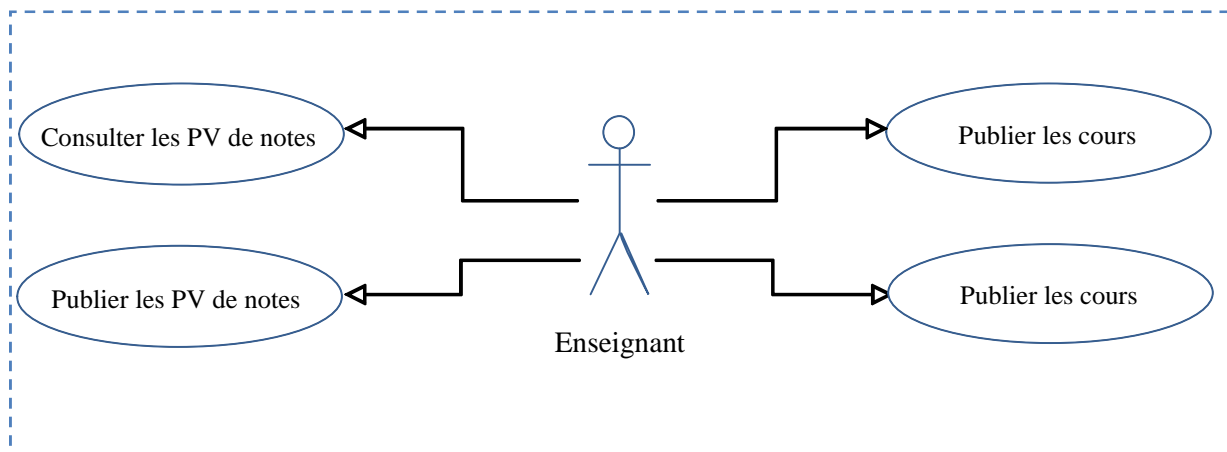


FIGURE 22 : Diagramme de cas d'utilisation « Gérer module »

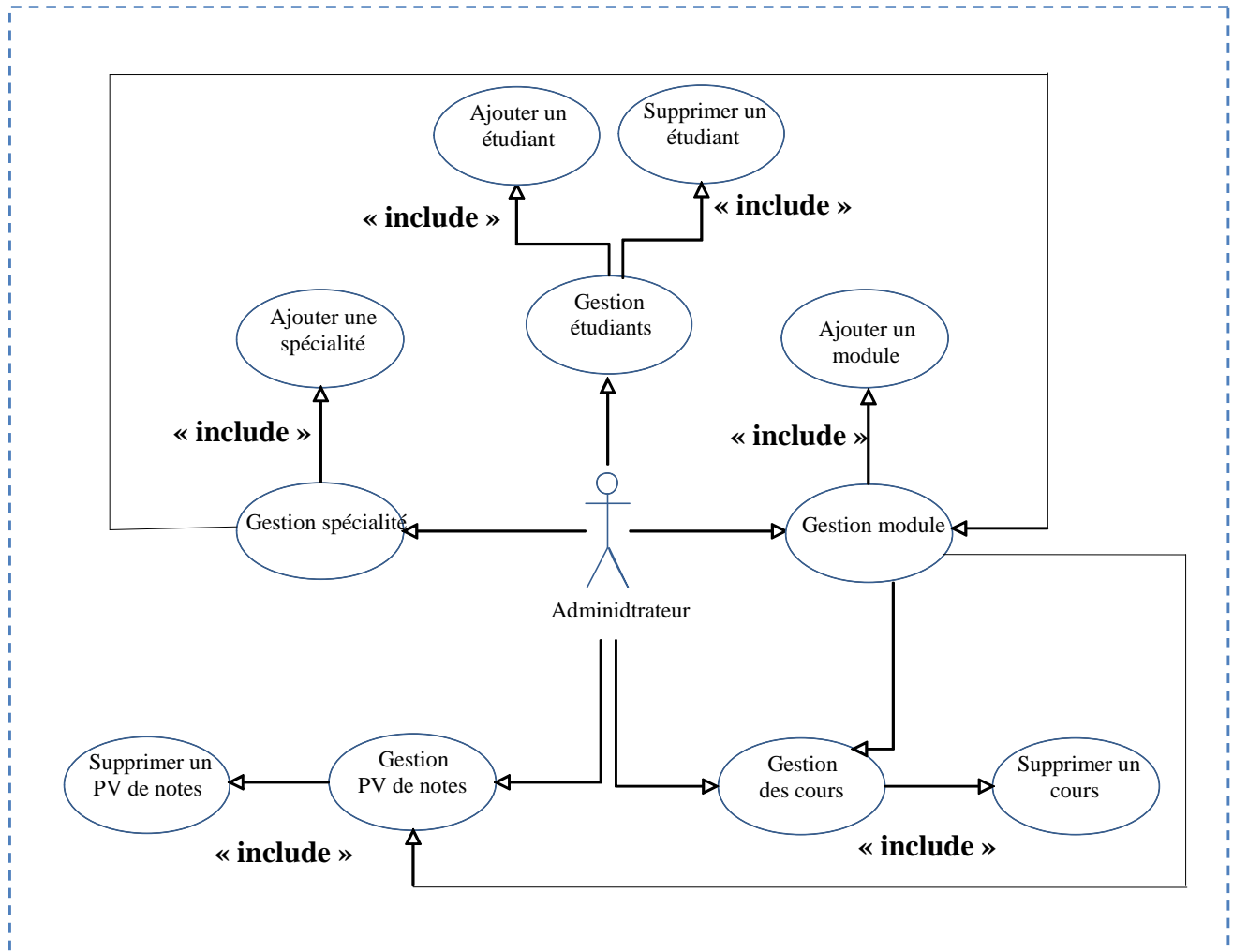


FIGURE 23 : Diagramme de cas d'utilisation « Gestion filière»

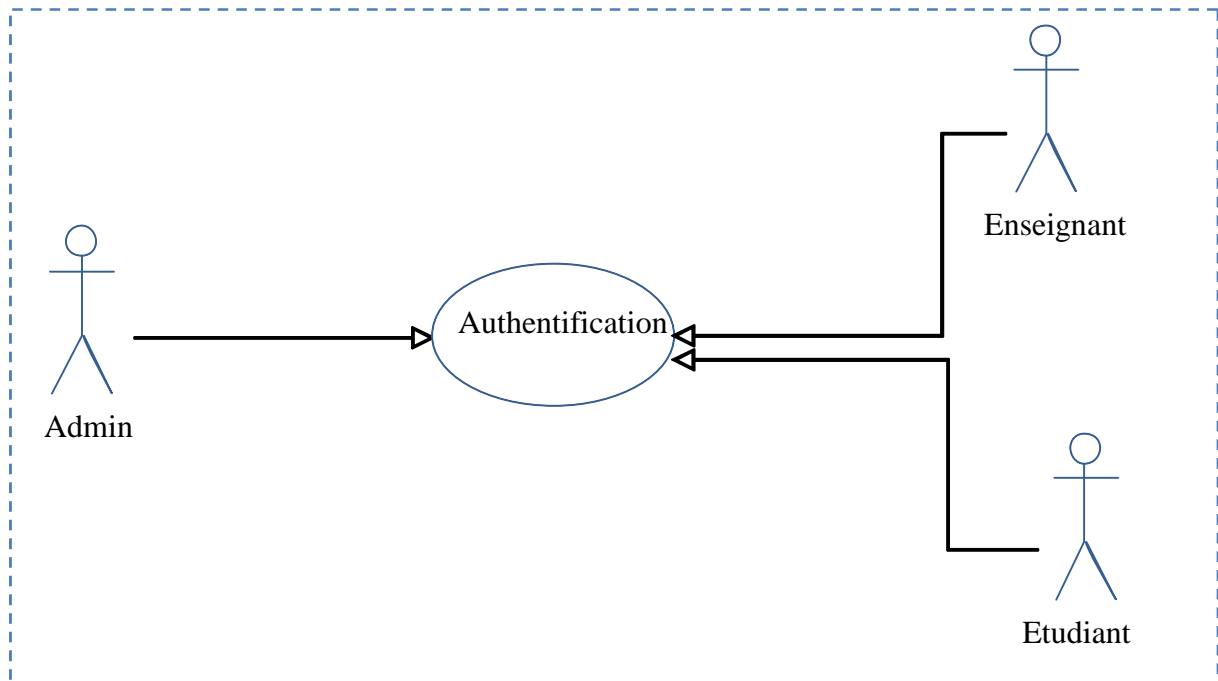


FIGURE 24 : Diagramme de cas d'utilisation « Authentification »

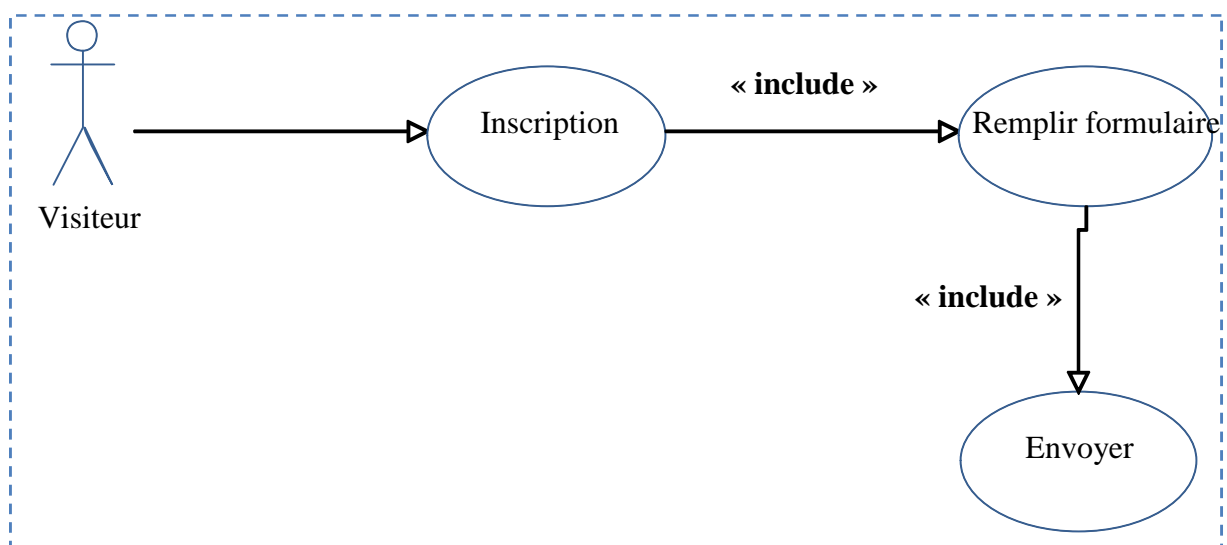


FIGURE 25 : Diagramme de cas d'utilisation « Inscription »

VII.2. Description textuelle des cas d'utilisation

Après avoir déterminé les cas d'utilisation de l'application, on les décrit de manière textuelle avec leurs scénarios argumentés à la fin avec quelques cas de séquences qu'on appellera diagramme de séquence du système.

VII.2.1. Cas d'utilisation « Consulter module »**VII.2.1.1.1. Identification du cas d'utilisation :**

Cas d'utilisation : Consulter cours ;

Acteur : Etudiant ;

Résumé : L'utilisateur accède à son espace personnel dans l'application, et choisit de consulter un cours d'un module donné de la spécialité de sa filière.

VII.2.1.1.2. Description des scénarios :**Scénario nominal**

1. Après authentification,
2. Le système affiche la page de l'espace personnel et l'utilisateur clique sur « suivant »,
3. Le système affiche la filière où l'utilisateur est inscrit, ensuite il clique dessus,
4. Le système affiche la liste des spécialités de la filière choisie auparavant,
5. Le système affiche la liste des spécialités de la filière choisie auparavant,
6. L'utilisateur choisit une spécialité en cliquant dessus,
7. Le système lui affiche la page des modules de sa spécialité, et l'utilisateur en sélectionne un,
8. Le système lui affiche le choix « PV de notes » ou « cours », et le membre décide de consulter les cours de ce module en cliquant sur « cours »,
9. Le système lui affiche l'ensemble de tous les cours du module choisi, et le membre visualise le contenu de l'un de ces cours en cliquant sur son lien pour le télécharger.

VII.2.1.2.1. Identification du cas d'utilisation :

Cas d'utilisation : Consulter PV de notes ;

Acteur : Etudiant ;

Résumé : L'utilisateur accède à son espace personnel dans l'application, et choisit de consulter un PV de notes d'un module donné de la spécialité de sa filière.

VII.2.1.2.2. Description des scénarios :**Scénario nominal**

10. Après authentification,
11. Le système affiche la page de l'espace personnel et l'utilisateur clique sur « suivant »,
12. Le système affiche la filière où l'utilisateur est inscrit, ensuite il clique dessus,
13. Le système affiche la liste des spécialités de la filière choisie auparavant,
14. L'utilisateur choisit une spécialité en cliquant dessus,

15. Le système lui affiche la page des modules de sa spécialité, et l'utilisateur en sélectionne un,
16. Le système lui affiche le choix « PV de notes » ou « cours », et le membre décide de consulter les notes de ce module en cliquant sur « notes »,
17. Le système lui affiche les notes des étudiants du module choisi.

VII.2.2. Cas d'utilisation « Gestion de filière »

VII.2.2.1.1. Identification du cas d'utilisation :

Cas d'utilisation : Créer une nouvelle spécialité;

Acteur : Administrateur

Résumé : Cette fonctionnalité permet à l'administrateur de mettre à jour une filière en lui ajoutant une nouvelle spécialité.

VII.2.2.1.2. Description des scénarios :

Scénario nominal :

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant les choix : « Etudiants », « Enseignants », « Cours », « PV de notes » et « Filières »,
4. L'administrateur décide de gérer les spécialités et clique sur « Filières »,
5. Le système lui retourne une interface contenant l'ensemble de toutes les filières.
6. L'administrateur choisit une filière et clique dessus,
7. Le système affiche la liste des spécialités de la filière choisie auparavant ainsi que le choix « nouvelle spécialité »,
8. L'administrateur crée une nouvelle spécialité en cliquant sur « nouvelle spécialité »,
9. Le système lui affiche la page de création d'une spécialité,
10. L'administrateur remplit les champs nécessaires et clique sur « Valider »,
11. Le système retourne un message de confirmation de création d'une spécialité,

VII.2.2.2.1. Identification du cas d'utilisation :

Cas d'utilisation : Créer un nouveau module;

Acteur : Administrateur

Résumé : Cette fonctionnalité permet à l'administrateur de mettre à jour une spécialité en lui ajoutant un nouveau module.

VII.2.2.2.2. Description des scénarios :**Scénario nominal :**

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant trois choix « Etudiants », « Enseignants », « Cours », « PV de notes » et « Filières »,
4. L'administrateur décide de gérer les spécialités et clique sur « Filières »,
5. Le système lui retourne une interface contenant l'ensemble de toutes les filières.
6. L'administrateur choisit une filière et clique dessus,
7. Le système affiche la liste des spécialités de la filière choisie auparavant ainsi que le choix « nouvelle spécialité »,
8. L'administrateur clique sur une spécialité,
9. Le système lui affiche l'interface des modules de la spécialité ainsi que le choix « nouveau module »,
10. L'administrateur crée un nouveau module en cliquant sur « nouveau module »,
11. Le système lui affiche la page de création d'un module,
12. L'administrateur remplit les champs nécessaires et clique sur « Valider »,
13. Le système retourne un message de confirmation de création d'un module.

VII.2.2.3.1. Identification du cas d'utilisation :

Cas d'utilisation : Supprimer un cours;

Acteur : Administrateur

Résumé : Cette fonctionnalité permet à l'administrateur de supprimer un cours déjà publié par un enseignant.

VII.2.2.3.2. Description des scénarios :**Scénario nominal :**

1. Après authentification,
2. L'administrateur atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche une interface contenant trois choix « Etudiants », « Enseignants », « Cours », « PV de notes » et « Filières »,
4. L'administrateur décide de gérer les cours et clique sur « Cours »,
5. Le système lui retourne une interface contenant l'ensemble de tout les cours publiés.
6. L'administrateur choisit un cours et clique sur supprimer,
7. Le système retourne un message de confirmation de suppression d'un cours.

VII.2.3. Cas d'utilisation « Gérer module »**VII.2.3.1.1. Identification du cas d'utilisation :**

Cas d'utilisation : Créer un nouveau cours;

Acteur : Enseignant

Résumé : Cette fonctionnalité permet à l'enseignant de mettre à jour un module en lui ajoutant un nouveau cours.

VII.2.3.1.2. Description des scénarios :**Scénario nominal :**

1. Après authentification,
2. L'enseignant atteint son interface, et clique sur suivant pour continuer,
3. Le système affiche la page de l'espace personnel avec les choix : « PV de notes », « cours », « publier PV de notes » et « publier cours»
4. L'enseignant décide de publier un nouveau cours en cliquant sur « publier cours»,.
5. Le système lui affiche la page de création d'un cours,
6. L'enseignant remplit les champs nécessaires et clique sur « Valider »,
7. Le système retourne un message de confirmation de création d'un cours.

VII.2.3.2.1. Identification du cas d'utilisation :

Cas d'utilisation : Créer un nouveau PV de notes ;

Acteur : Enseignant

Résumé : Cette fonctionnalité permet à l'enseignant de mettre à jour un module en lui ajoutant un nouveau PV de notes .

VII.2.3.2.2. Description des scénarios :**Scénario nominal :**

1. Après authentification,
2. L'enseignant accède à sa propre interface, et clique sur suivant pour continuer,
3. Le système affiche la page de l'espace personnel avec les choix : « PV de notes », « cours », « publier PV de notes » et « publier cours»
4. L'enseignant décide de mettre en ligne un nouveau PV de notes en cliquant sur « publier PV de notes »,.
5. Le système lui affiche la page de création d'un PV de notes ,
6. L'enseignant remplit les champs nécessaires et clique sur « Valider »,
7. Le système retourne un message de confirmation de création d'un PV de notes.

VII.2.4. Cas d'utilisation « Authentification »**VII.2.4.1. Identification du cas d'utilisation :**

Cas d'utilisation : Authentification.

Acteur : Enseignant/Étudiant/Administrateur.

Résumé : Cette fonctionnalité permet à chaque utilisateur de s'authentifier par un login et mot de passe pour accéder à son espace.

VII.2.4.2. Description des scénarios :**Scénario nominal :**

1. L'utilisateur atteint le site ;
2. Le système lui affiche la page d'accueil;
3. L'utilisateur choisit son profil ;
4. Le système affiche la page d'authentification ;
5. L'utilisateur saisit son login et son mot de passe, puis il clique sur « s'authentifier » ;
6. Le système vérifie les informations saisies avec celles de la base de données;
7. Après validation, l'utilisateur atteint son espace.

Scénario alternatif :

Mot de passe erroné

L'enchaînement nominal démarre au point 6 :

7. le système affiche un message d'erreur indiquant que le login ou le mot de passe est erroné.

L'enchaînement reprend au point 5.

VII.2.5. Cas d'utilisation « Inscription »**VII.2.5.1. Identification du cas d'utilisation :**

Cas d'utilisation : Inscription étudiant.

Acteur : visiteur.

Résumé : Cette fonctionnalité permet au visiteur de s'inscrire en tant qu'étudiant.

VII.2.5.2. Description des scénarios :**Scénarios nominal :**

1. L'utilisateur atteint le site ;
2. Le système lui affiche la page d'accueil;
3. L'utilisateur choisit son profil en cliquant sur le bouton « Etudiant » ;
4. Le système lui affiche la page d'authentification contenant un formulaire d'authentification ainsi qu'un bouton « s'inscrire » ;
5. L'utilisateur clique sur « s'inscrire ».
6. Le système lui affiche le formulaire d'inscription ;
7. Après remplissage du formulaire, l'utilisateur clique sur « Inscription » ;
8. Le système vérifie la validité des informations du formulaire, et retourne un message de confirmation d'inscription.

Scénario alternatif :

Les données du formulaire sont erronées.

L'enchaînement nominal démarre au point 8 ;

9. le système affiche un message.

VII.3. Description graphique des cas d'utilisation

Pour documenter les cas d'utilisation, la description textuelle est indispensable, car elle seule permet de communiquer facilement avec les utilisateurs et de s'entendre sur la terminologie métier employée.

En revanche, avec le texte seul il est difficile de montrer comment les enchaînements se succèdent. Il est alors recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques UML. Pour notre cas nous utilisons les diagrammes de séquences pour décrire les scénarios nominaux des cas d'utilisation. Dans la suite de ce paragraphe nous présentons seulement à titre d'exemple quelques diagrammes.

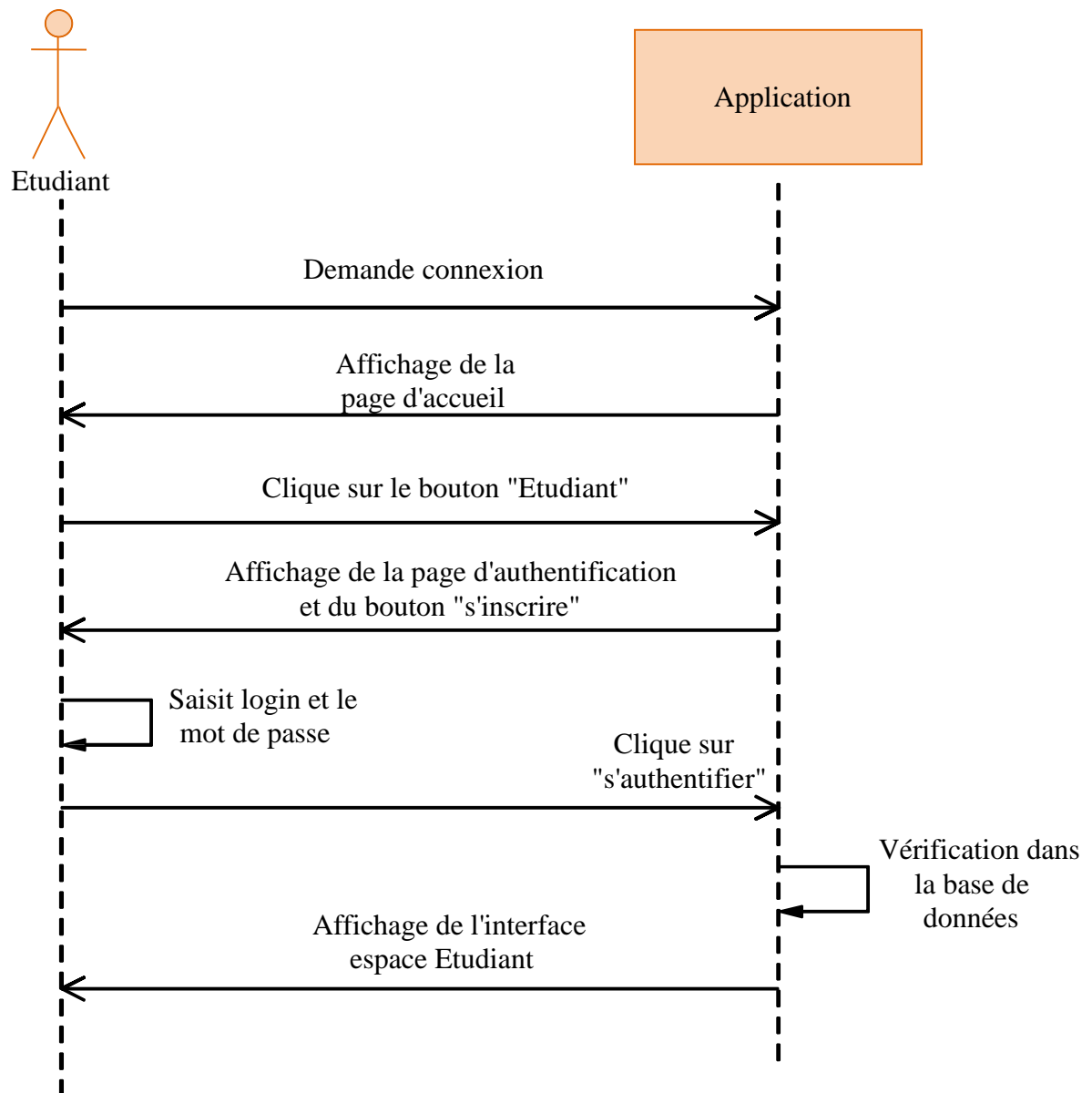
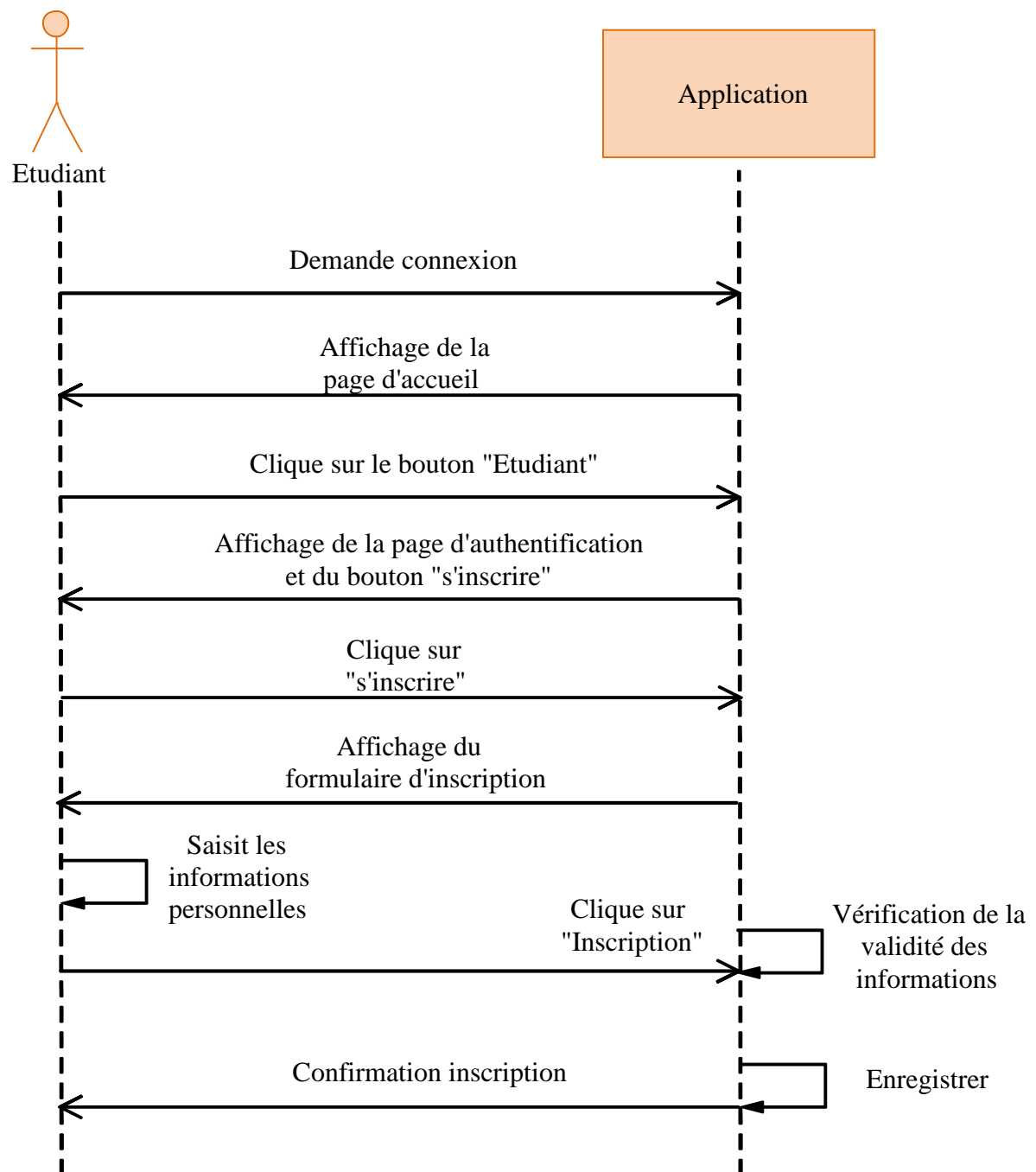


FIGURE 26 : Diagramme de séquence « Authentification »

**FIGURE 27 : Diagramme de séquence « Inscription étudiant »**

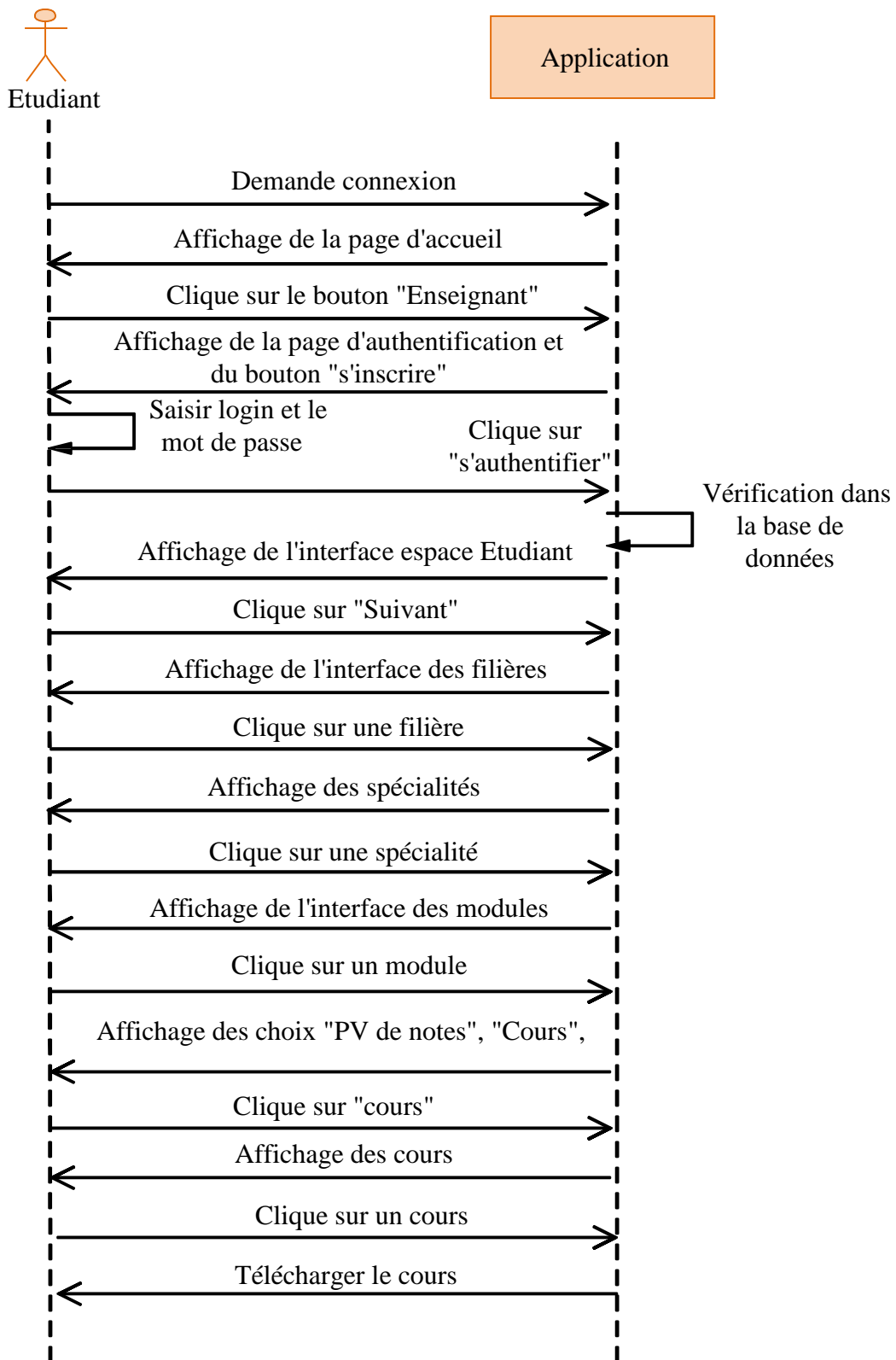


FIGURE 28 : Diagramme de séquence « Consulter cours »

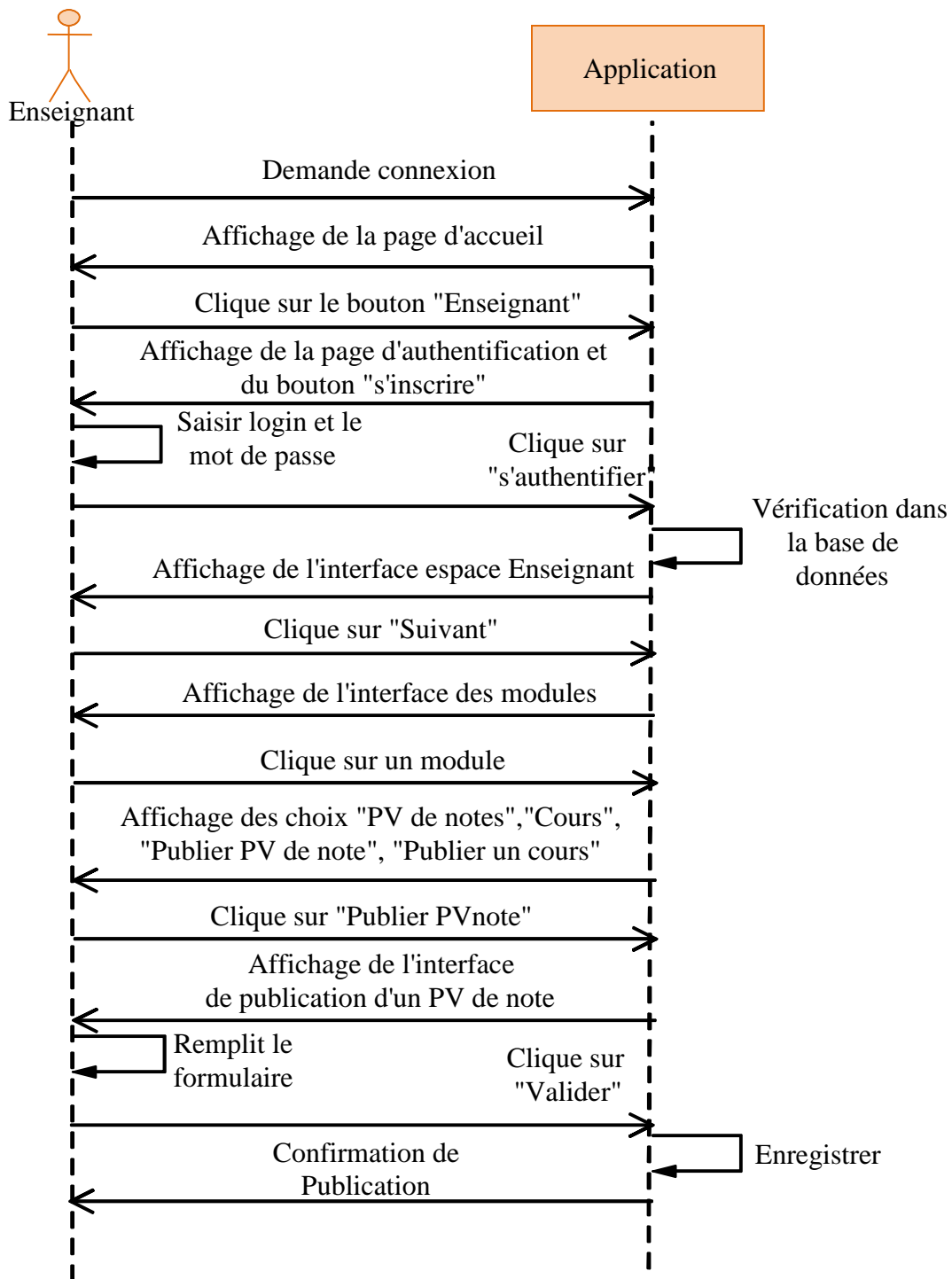
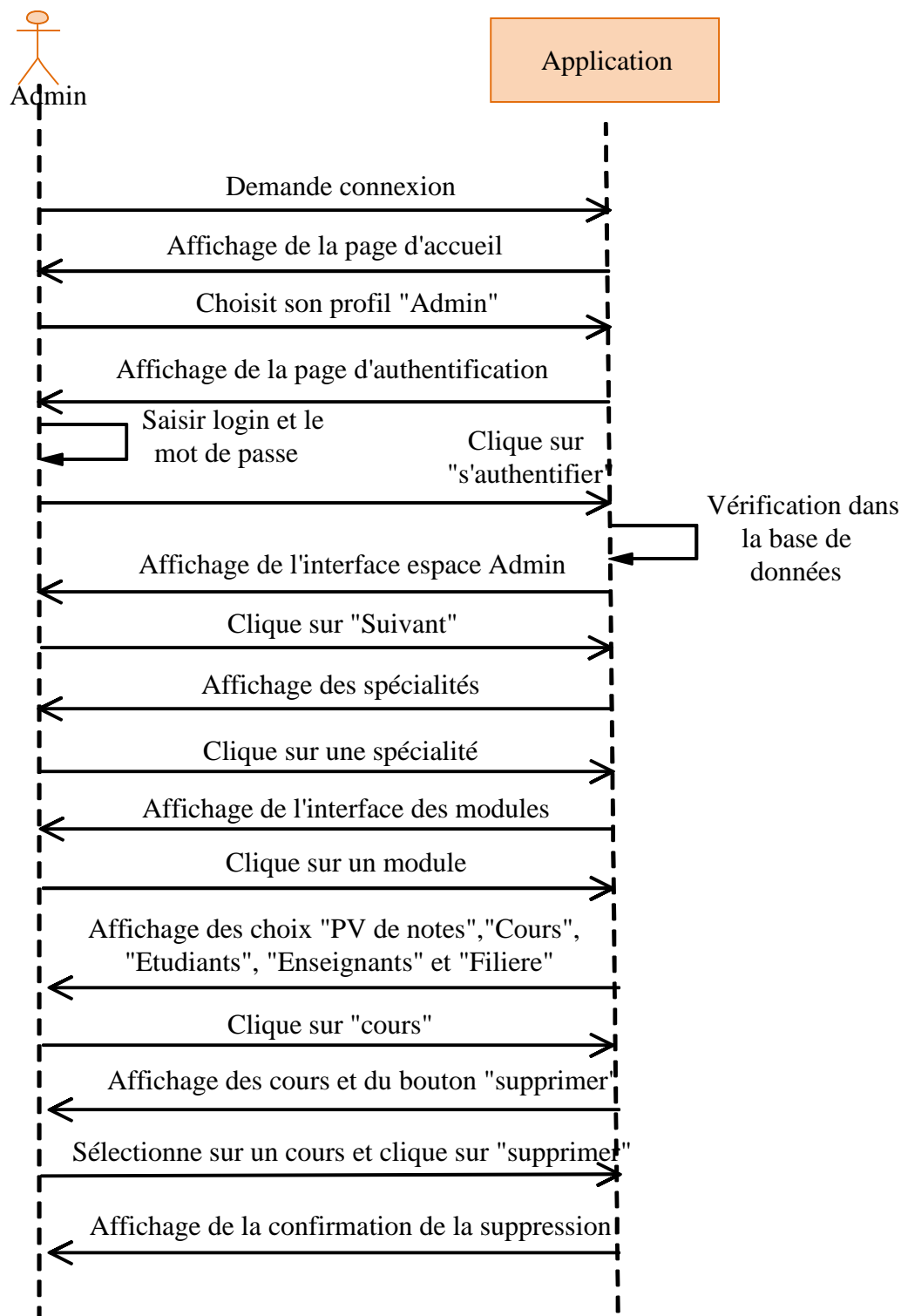


FIGURE 29 : Diagramme de séquence « Publier PV de notes »

**FIGURE 30 : Diagramme de séquence « Supprimer cours »**

VIII. La conception objet :

La conception est souvent considérée, à tort, comme un simple enrichissement des résultats obtenus dans l'analyse. Elle se base sur le savoir-faire qui peut être interne au projet ou acquis de l'extérieur sous forme d'outils, de composants réutilisables ou plus largement de cadres de développement. Elle vise principalement à préciser le modèle d'analyse de telle sorte qu'il puisse être implémenté avec les composants de l'architecture, c'est la phase la plus complexe du projet.

Cette partie de ce chapitre est consacrée à décrire la manière de concevoir la structure de la base de données.

VIII.1. Les bases de données

Une base de données est une collection de données sur un domaine d'application particulier, où les propriétés des données ainsi que les relations sémantiques entre ces données sont spécifiées en utilisant les concepts proposés par le modèle de données.

Le langage standard de requête structurée des bases de données relationnelles est le SQL.

SQL signifie Structured Query Language c'est-à-dire langage d'interrogation structuré. Il permet d'interroger une base de données, d'en modifier des informations. C'est un langage universel d'interrogation des bases de données, qui permet aussi à différents systèmes d'échanger des données entre eux.

A. Le niveau conceptuel de la base de données :

Après avoir élaborés les diagrammes de séquence des différents cas d'utilisations déjà décrits, nous allons élaborer le diagramme de classes général qui représente une vue conceptuel de la base de données.

❖ *Le diagramme de classe :*

Le diagramme de classe a toujours été le diagramme le plus important dans les méthodes orientées objets. Il représente la structure statique d'un système. Il contient principalement des classes, ainsi que leurs associations, mais on peut aussi y trouver des objets.

L'intérêt majeur du diagramme de classes est de modéliser les entités du système.

❖ Le schéma conceptuel :

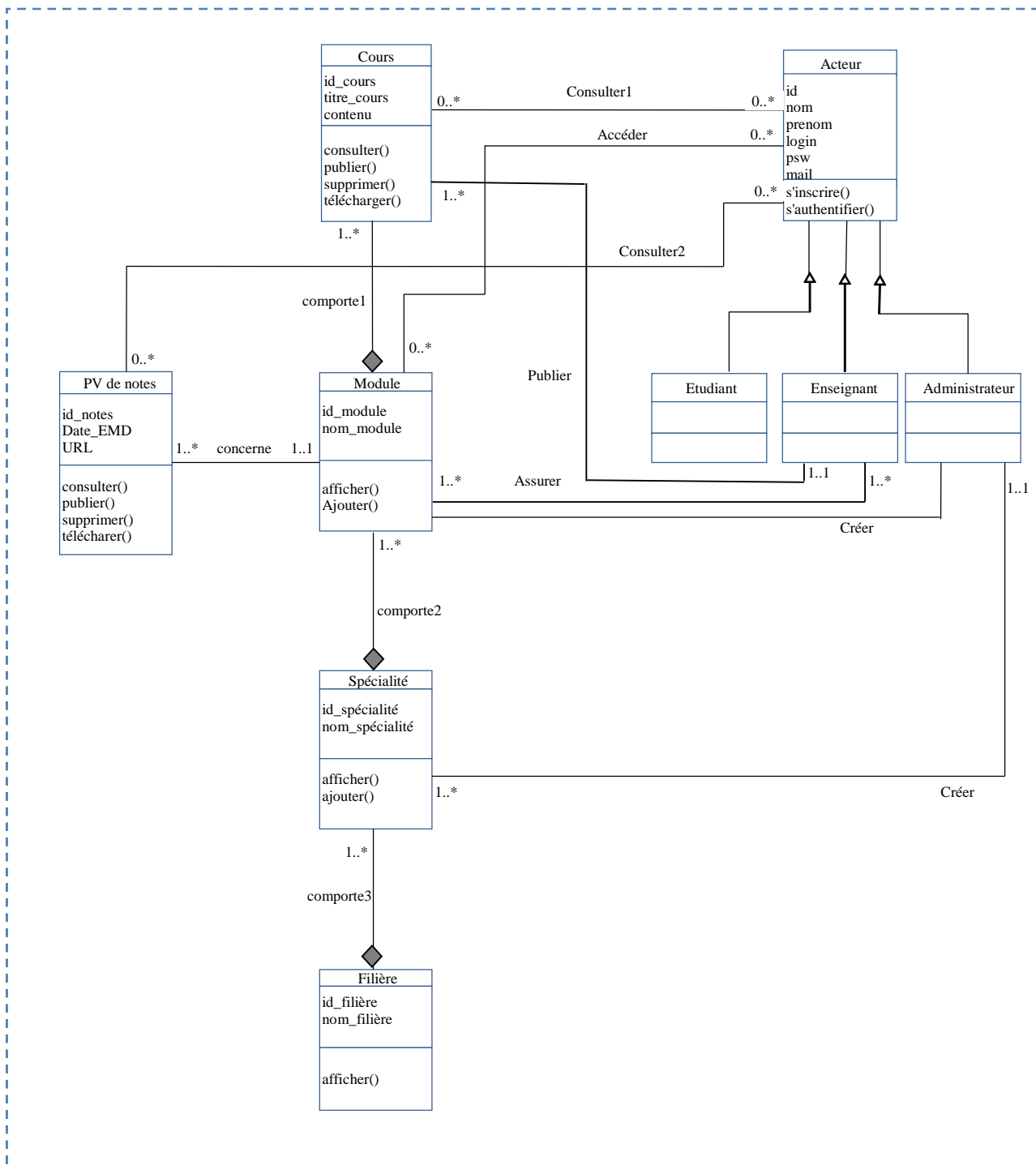


FIGURE 31 : Diagramme de classe général

B. Le niveau logique de la base de données :

Nous allons dans cette partie élaborer le schéma relationnel de la base de données dont le schéma conceptuel est déjà réalisé tout en respectant les règles de passage de niveau conceptuel au niveau logique.

○ Classe avec attributs

Chaque classe entité devient une relation. Les attributs de la classe deviennent des attributs de la relation. Si la classe possède un identifiant, il devient la clé primaire de la relation, sinon, il faut ajouter une clé primaire arbitraire.

○ Association 1 vers 1

Pour représenter une association 1 vers 1 entre deux relations, la clé primaire de l'une des relations doit figurer comme clé étrangère dans l'autre relation.

○ Association 1 vers plusieurs

Pour représenter une association 1 vers plusieurs, on procède comme pour une association 1 vers 1, excepté que c'est forcément la relation du côté plusieurs qui reçoit comme clé étrangère la clé primaire de la relation du côté 1.

○ Association plusieurs vers plusieurs

Pour représenter une association du type plusieurs vers plusieurs, il faut introduire une nouvelle relation dont les attributs sont les clés primaires des relations en association et dont la clé primaire est la concaténation de ces deux attributs.

❖ Le schéma relationnel :

Admin (id_admin, nom_admin, prenom_admin, login_admin, psw_admin).

Enseignant (id_ens, nom_ens, prenom_ens, login_ens, psw_ens).

Etudiant (id_etud, nom_etud, prenom_etud, login_etud, psw_etud).

Filiere (id_fil, nom_fil).

Specialite (id_spec, nom_spec, id_fil).

Module (id_mod, nom_mod).

PV_note (id_note, date_EMD, url_note).

Cours (id_cours, titre, url_cours, id_mod)

Cours_Etudiant (id_cours, id_etud).

Note_Etudiant (id_note, id_etud).

Module_Enseignant (id_mod, id_ens).

Module_Specialite (id_mod, id_spec).

Remarque :

- Les attributs soulignés sont des clés primaires.
- Les attributs doublement soulignés sont des clés étrangères.

C. Structure physique des tables de la base de données

La base de données que nous allons à présent construire contiendra les différentes tables qui comportent toutes les données nécessaires au bon fonctionnement de notre application.

Table 1 : « Etudiant »

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|---------------------------|-----------------|----------|
| id_etud | identifiant de l'étudiant | Int (5) | Primaire |
| nom_etud | nom de l'étudiant | Varchar (20) | |
| prenom_etud | prenom de l'étudiant | Varchar (20) | |
| login_etud | Nom utilisateur étudiant | Varchar (25) | |
| psw_etud | Mot de passe étudiant | Varchar (25) | |

Table 2 : «Enseignant»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|-----------------------------|-----------------|----------|
| id_ens | identifiant de l'enseignant | Int (3) | Primaire |
| nom_ens | nom de l'enseignant | Varchar (20) | |
| prenom_ens | prenom de l'enseignant | Varchar (20) | |
| login_ens | Nom utilisateur enseignant | Varchar (25) | |
| psw_ens | Mot de passe enseignant | Varchar (25) | |

Table 3 : «Admin»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|-------------------------|-----------------|----------|
| id_admin | identifiant de l' admin | Int (1) | Primaire |
| nom_admin | nom de l' admin | Varchar (20) | |
| prenom_admin | prenom de l' admin | Varchar (20) | |
| login_admin | Nom utilisateur admin | Varchar (25) | |
| psw_admin | Mot de passe admin | Varchar (25) | |

Table 4 : «Filière»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|---------------------------|-----------------|----------|
| id_fil | identifiant de la filière | Int (3) | Primaire |
| nom_fil | nom de la filière | Varchar (20) | |

Table 5: «Spécialité»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|------------------------------|-----------------|-----------|
| id_spec | identifiant de la spécialité | Int (3) | Primaire |
| id_fil | identifiant de la filière | Int (3) | Etrangère |
| nom_spec | nom de la spécialité | Varchar (20) | |

Table 6 : «Module»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|-----------------------|-----------------|----------|
| id_mod | identifiant du module | Int (3) | Primaire |
| nom_mod | nom de du module | Varchar (20) | |

Table 7 : «PV de notes»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|----------------------------|-----------------|----------|
| id_note | identifiant du PV de notes | Int (3) | Primaire |
| Date_EMD | Date de l'EMD | DATE | |
| url_note | url du PV de notes | Varchar (50) | |

Table 8 : «Cours»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|-----------------------|-----------------|-----------|
| id_ cours | identifiant du cours | Int (3) | Primaire |
| Id_mod | identifiant du module | Int (3) | Etrangère |
| titre | titre du cours | Varchar (50) | |
| url_ cours | url du cours | Varchar (50) | |

Table 9 : «Cours_Etudiant»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|---------------------------|-----------------|---------------|
| id_ cours | identifiant du cours | Int (3) | Etrangère (1) |
| id_ etud | identifiant de l'étudiant | Int (3) | Etrangère (2) |

Table 10 : «Note_Etudiant »

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|----------------------------|-----------------|---------------|
| id_ note | identifiant du PV de notes | Int (3) | Etrangère (1) |
| id_ etud | identifiant de l'étudiant | Int (3) | Etrangère (2) |

Table 11 : «Module_ Enseignant»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|-----------------------------|-----------------|---------------|
| id_ mod | identifiant du module | Int (3) | Etrangère (1) |
| id_ ens | identifiant de l'enseignant | Int (3) | Etrangère (2) |

Table 12 : «Module_ Spécialité»

| Nom du champ | Description | Type de données | Clé (s) |
|--------------|------------------------------|-----------------|---------------|
| id_ mod | identifiant du module | Int (3) | Etrangère (1) |
| id_ spec | identifiant de la spécialité | Int (3) | Etrangère (2) |

Remarque : (1)+ (2) : Clé primaire.

IX. Le diagramme de déploiement :

Un diagramme de déploiement permet de donner la structure d'une plate-forme technique, mais aussi de spécifier la localisation des nœuds constitués par des unités distribuées, de préciser où se trouvent les processus et de montrer comment les objets se créent et se déplacent dans une architecture distribuée.

Notre diagramme de déploiement permet de représenter l'environnement de développement de notre application dont l'implémentation se base sur une architecture client/serveur trois tiers :

- ✓ Le premier niveau de cette architecture qui est le niveau présentation est constitué d'un navigateur Web tel que Mozilla Firefox.
- ✓ Le deuxième niveau est le niveau applicatif (logique applicative) qui est pris en charge par le serveur Apache Tomcat.
- ✓ Le troisième niveau qui fournit au niveau intermédiaire les données dont il a besoin, est pris en charge dans notre cas par le SGBD MySQL.

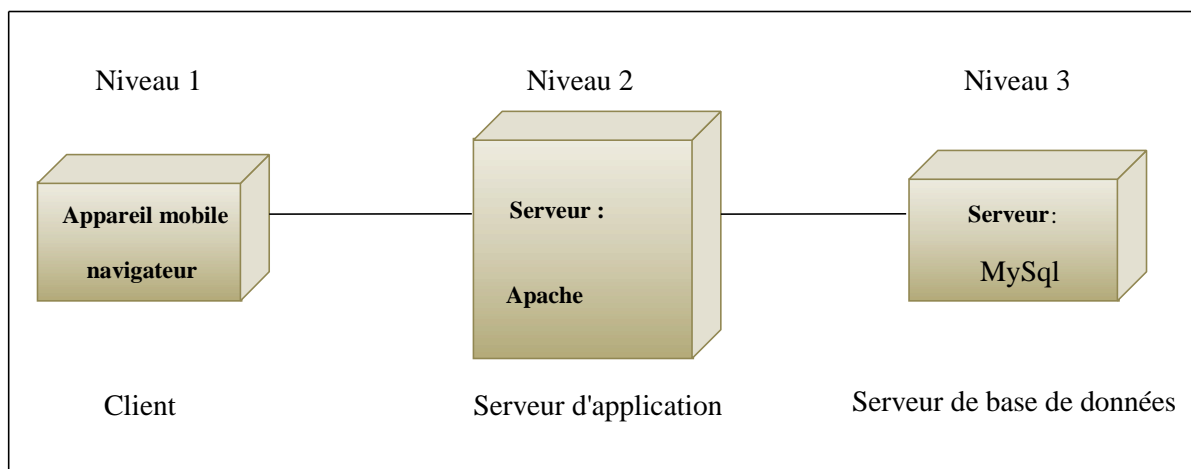


FIGURE 32: Diagramme de déploiement de l'application

X. Conclusion :

Ce chapitre est consacré à l'analyse et à la conception de l'application avec le langage UML pour le web.

Nous avons donc commencé par la définition des besoins qui ont été traduits par la suite en formalismes graphiques à l'aide des diagrammes offerts par ce langage.

La partie conception a mis en évidence les différents composants de l'application et cela dans le but de construire notre base de données.

Donc à ce stade nous sommes assez armé pour mettre sur pied notre application, ce qui va être l'objet du chapitre suivant tout en exposant l'environnement de développement.








I. Introduction :

Après avoir fait une étude détaillée des différents modules constituant notre application, nous allons nous pencher sur la mise en œuvre de notre système ; en choisissant les différents outils de développement correspondants à des critères bien définis pour atteindre des objectifs bien précis.

Dans le chapitre précédent, nous avons détaillé l'étape de conception de notre système. Les résultats de cette étape seront employés dans la partie réalisation que nous détaillerons dans ce présent chapitre.

Nous commençons par la présentation des différents services web implémentés au cours de la réalisation, de l'environnement de développement puis nous présentons quelques interfaces de notre application e_learning déployée sur mobile.

Comme nous l'avons signalé d'emblé, notre solution se base sur les services web (nous avons présenté ce concept en détails dans le chapitre III). Dans le souci de répondre aux différents besoins fonctionnels de notre système nous avons opté pour le découpage suivant :

-  Le service web **EtudiantFacadeREST**
-  Le service web **EnseignantFacadeREST**
-  Le service web **AdminFacadeREST**
-  Le service web **CoursFacadeREST**
-  Le service web **NoteFacadeREST**
-  Le service web **ModuleFacadeREST**
-  Le service web **SpecialiteFacadeREST**

Dans ce qui suit nous présentons une description détaillée des différents services web implémentés.

II. Description des différents services web :

❖ Le service web : **EtudiantFacadeREST**

Ce service web implémente cinq web méthodes, ses fonctions principales sont :

Inscrire les étudiants, vérifier les informations introduites par un étudiant dans le but de s'authentifier, supprimer un étudiant à travers son id, fournir des informations sur un étudiant donné d'après son id et permettre de récupérer les informations personnelles de l'ensemble des étudiants inscrits.

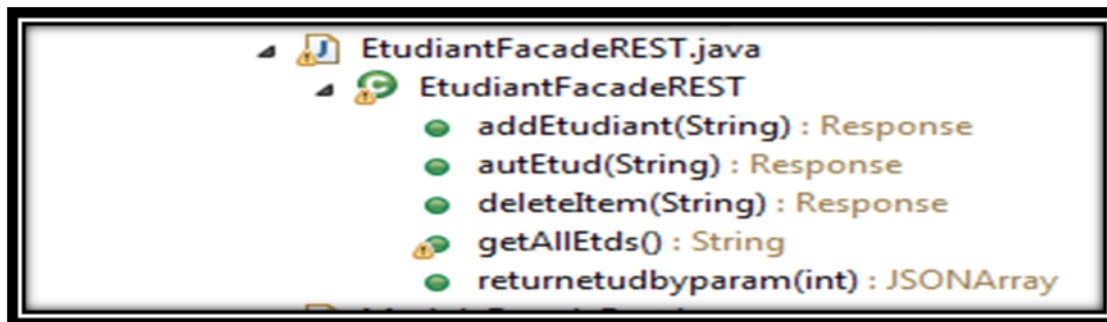


FIGURE 33: Fonctionnalités du service web **EtudiantFacadeREST**

Exemple :

Test de la web méthode getAllEtds() qui retourne les informations personnelles concernant tout les étudiants au format Json.

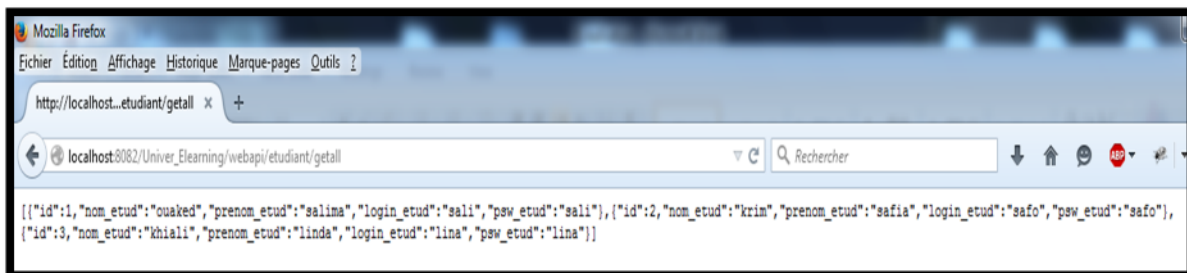


FIGURE 34: Test de la web méthode getAllEtds()

❖ Le service web : **EnseignantFacadeREST**

Ce service web implémente cinq web méthodes. Principalement il gère l'inscription, l'authentification et la suppression des enseignants, ainsi que la récupération des informations personnelles de tout les enseignants déjà inscrits ou bien celle d'un enseignant spécifique d'après son identifiant. ;

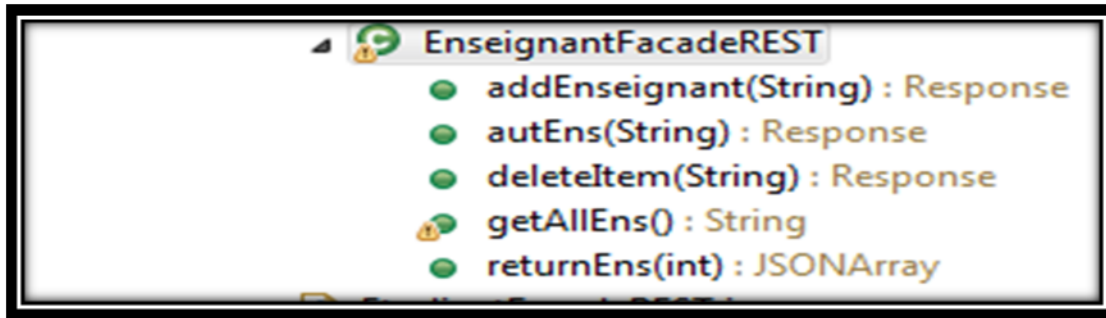


FIGURE 35: Fonctionnalités du service web **EnseignantFacadeREST**

Exemple :

Test de la web méthode returnEns() qui prend en entrée l'id d'un enseignant et retourne ses informations personnelles au format Json.

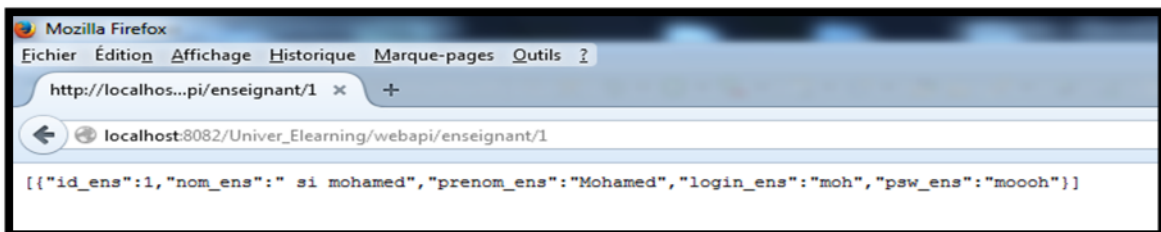


FIGURE 36: Test de la web méthode returnEns()

❖ Le service web : **AdminFacadeREST**

Le service web AdminFacadeREST implémente une seule fonctionnalité, le principal rôle de ce service est de permettre à l'administrateur de s'authentifier.

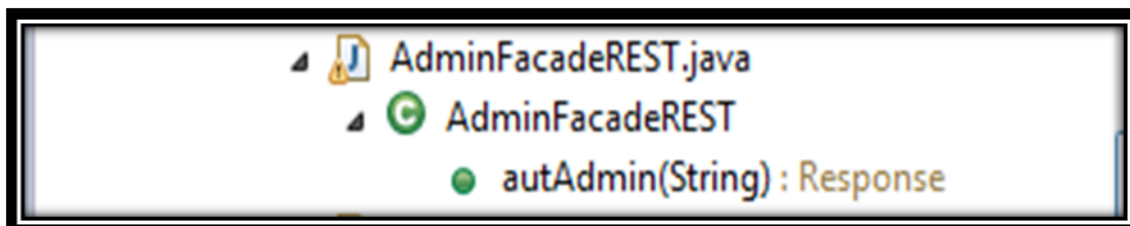


FIGURE 37: Fonctionnalités du service web **AdminFacadeREST**

❖ Le service web : **CoursFacadeREST**

Ce service web implémente quatre web méthodes, ses fonctions principales sont :

Créer un nouveau cours, supprimer un cours à travers son identifiant, récupérer un cours ou encore retourner l'ensemble de tout les cours du campus numérique publiés par les enseignants.



FIGURE 38: Fonctionnalités du service web **CoursFacadeREST**

❖ Le service web : **NoteFacadeREST**

Ce service web implémente quatre web méthodes, ces dernières permettent de créer un nouveau PV de note, supprimer un PV de note à travers son identifiant, récupérer un PV de note ou encore retourner l'ensemble de tout les PV de note publiés par les enseignants.

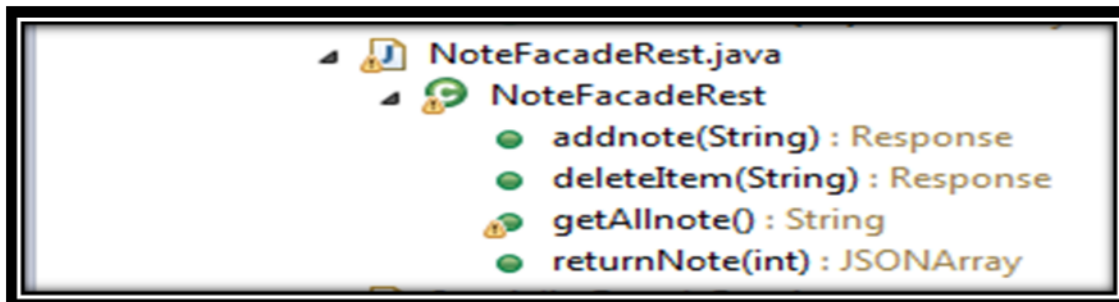


FIGURE 39: Fonctionnalités du service web **NoteFacadeREST**

❖ Le service web : **ModuleFacadeREST**

Ce service web implémente trois fonctionnalités, sa fonction principale est de créer un nouveau module pour une spécialité donné, ainsi que de retourner l'ensemble des modules existants ou bien seulement un module d'après son identifiant.

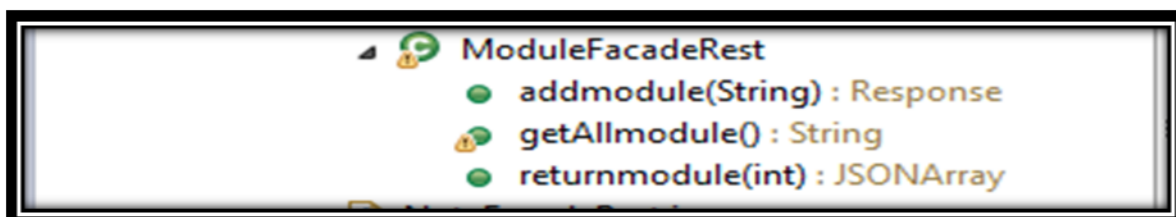


FIGURE 40: Fonctionnalités du service web **ModuleFacadeREST**

❖ Le service web : **SpecialitéFacadeREST**

Ce service web implémente trois fonctionnalités. Son principal rôle est de créer une nouvelle spécialité pour une filière donné, ainsi que de retourner l'ensemble des spécialités existantes ou bien seulement une d'après son identifiant.

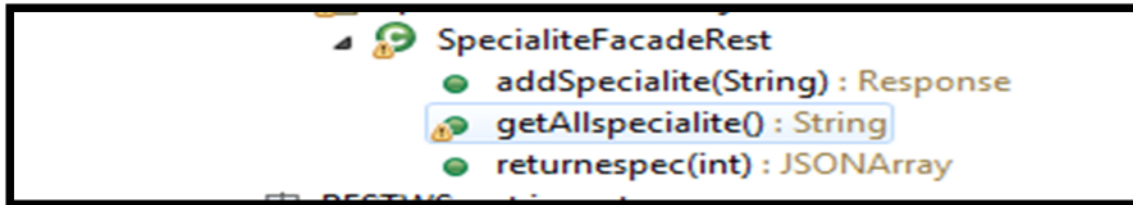


FIGURE 41: Fonctionnalités du service web **SpecialitéFacadeREST**

III. L'environnement technique de développement :

III.1. Outils de développement

Notre environnement de développement est constitué d'un ensemble d'outils à savoir :

Le serveur web apache : Tout développement de site Web requiert un serveur Web qui s'occupe du traitement des requêtes des clients et l'exécution des programmes sur la machine serveur.

Le serveur Web Apache est développé par un groupe de personne autonome. L'objectif était de développer un serveur http (Web) puissant et utilisable gratuitement.

Le serveur web apache est très bien placé parmi d'autres serveurs, puisqu'il :

- représente plus de deux tiers des serveurs actuellement installés.
- Il présente un niveau de performances élevé pour des exigences matérielles modestes
- Il est gratuit et robuste.

Le module Tomcat : Le conteneur de servlets choisi est le moteur Tomcat 8.0 développé par la fondation Apache. Le dialogue entre le moteur de servlet et le serveur Web s'effectue à l'aide d'un module logiciel appelé connecteur. Tomcat peut fonctionner sur d'autre serveur Web mais seul le couple Tomcat/Apache a été testé voir (*figure 8- Apache/Tomcat*). Le module Tomcat du serveur Apache a été développé à partir des sources de Sun Microsystems. Il représente une implémentation de référence pour les servlets. Tomcat peut fonctionner seul, mais cela n'est pas une solution efficace. En exploitation il est préférable d'associer Tomcat avec un serveur http plus puissant, qui se chargera du contenu statique.

Le serveur de bases de données :

- Le serveur MySQL version.
- Une interface graphique PhpMyAdmin.

➤ Le serveur MySQL :

MySQL est un véritable serveur de base de données SQL Multiutilisateurs et multitraitements. Cela permet d'établir des connexions rapides et d'utiliser la même mémoire cache pour plusieurs requêtes.

MySQL est basé sur une bibliothèque de gestion de données éprouvée depuis de nombreuses années et faisant appel à des index d'arbres binaires. Grâce à cela, le coeur du système peut afficher une performance remarquable, tout particulièrement dans les accès indexes.

Les principaux objectifs de MySQL sont la rapidité, la robustesse et la facilité d'utilisation.

fonctionnalité de MySQL :

Multitraitement : MySQL est multitraitement en utilisant les threads du noyau. Il peut utiliser plusieurs CPU.

Langues : Le serveur peut fournir au client les messages d'erreurs en plusieurs langues.

APIs : Les applications de base de données MySQL peuvent être écrites en C, C++, Eiffel, java, perl, PHP, python et TCL.

Multi plates-formes : Prise en charge de plus de 20 plates formes de système d'exploitation (Win95/98 et NT, UNIX, LINUX).

Tables : MySQL stocke chaque table sous forme de fichier distinct dans le répertoire de la base de données. La taille maximale d'une table est comprise entre 4 GO et la taille maximale du fichier accepté par le système d'exploitation.

➤ L'Interface graphique PhpMyAdmin :

C'est une interface conviviale gratuite réalisée en langage PHP, permettant d'administrer des bases de données MySQL via un navigateur web. PhpMyAdmin permet de :

- Créer /supprimer de nouvelles bases.
- Créer /modifier/supprimer /copier des tables.
- Afficher /ajouter /modifier /supprimer des tuples dans les tables.
- Effectuer des sauvegardes de la structure et /ou des données.
- Effectuer n'importe quelle requête.
- Gérer les privilèges des utilisateurs.

L'IDE Eclipse : Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plateforme modulaire pour permettre de réaliser des développements informatiques.

I.B.M. est à l'origine du développement d'Eclipse qui est d'ailleurs toujours le coeur de son outil Websphere Studio Workbench (WSW), lui même à la base de la famille des derniers outils de développement en Java d'I.B.M. Tout le code d'Eclipse a été donné à la communauté par I.B.M afin de poursuivre son développement.

Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse.

Dreamweaver CS6 : C'est une plateforme de développement pour la création de sites et d'applications Web. Le logiciel propose d'utiliser des outils d'inspection CSS pour la conception et de développer et améliorer des systèmes de gestion de. Adobe Dreamweaver dispose également d'un module de tests de compatibilité avec les navigateurs grâce à l'intégration avec Adobe BrowserLab. Au chapitre des nouveautés, on retrouve un outil de mise en forme à "grille fluide" réactive pour CSS3, de meilleures performances pour les transferts FTP, ainsi qu'une prise en charge de PhoneGap pour le développement de site optimisés Android et iOS. Adobe Dreamweaver dispose de fonctionnalités d'affichage en direct et d'aperçu sur écrans multiples et intègre de nouvelles transitions CSS3.

Phonegap : C'est un outil permettant de créer des applications mobiles multiplateformes en utilisant les standards du web. PhoneGap est développé par Adobe, qui a racheté la société Nitobi à l'origine du développement de PhoneGap.

Les applications réalisées avec PhoneGap sont des applications web utilisant les technologies HTML 5, CSS 3 et JavaScript. Elles sont packagées par l'outil de façon être exécutées par le moteur de rendu du navigateur natif du smartphone ou de la tablette cible. PhoneGap propose également une API JavaScript qui permet l'appel, de manière unifiée parmi les plate-formes cibles, aux fonctionnalités avancées des smartphones (GPS, accéléromètre, caméra...).

PhoneGap supporte de nombreuses plates-formes dont iOS, Android, BlackBerry 10 (et 6), Windows Phone 7 et 8, Windows 8, Tizen, WebOS et Symbian.

C'est donc une solution complète et techniquement pertinente pour le développement d'application mobiles multiplateformes.

Adobe propose par ailleurs un service PhoneGap Build permettant la génération des applications en ligne, s'affranchissant de l'installation des différents SDK.

III.2. Les langages utilisés

J2EE : Le langage de programmation utilisé est le J2EE de SUN, un langage orienté objet qui combine toutes les possibilités de manipulation des données proposées par SQL, et ce grâce à l'API JDBC. Il offre de nombreux avantages :

- C'est un langage de programmation évolué qui facilite tout les accès à la base grâce à la puissance de SQL.
- Performant, facile à la programmation et permet la portabilité.
- Supporte le modèle Client-Serveur.
- Supporte la programmation Orienté Objet.
- Permet la modularité et l'ouverture sur d'autres langages L3G.

SQL : C'est un langage de manipulation de base de données mis au point dans les années 70 par IBM. Il permet notamment :

- La manipulation des tables (création, suppression, modification de la structure des tables).
- La manipulation des bases de données (sélection, modification et suppression d'enregistrements).
- La gestion des droits d'accès aux modifications.

HTML : L'*HyperText Markup Language*, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du *Standard Generalized Markup Language* (SGML).

HTML5 (*HyperText Markup Language 5*) est la dernière révision majeure d'HTML. Cette version a été finalisée le 28 octobre 2014. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM : HTML5 et XHTML5. Le langage comprend également une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme. Le travail a été repris par le W3C en mars 2007 après avoir été lancé par le WHATWG. Les deux organisations travaillent en

parallèle sur le même document afin de maintenir une version unique de la technologie. Le W3C vise la clôture des ajouts de fonctionnalités le 22 mai 2011 et une finalisation de la spécification en 2014¹, et encourage les développeurs Web à utiliser HTML 5 dès maintenant.

Les feuilles de style (CSS) : Elles ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent. Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier !

Elles sont appelées « *feuilles de style en cascade* » car il est possible d'en définir plusieurs et que les styles peuvent être hérités en cascade.

JavaScript : C'est un langage de scripts qui est incorporé aux balises Html, permet d'améliorer la présentation et l'interactivité des pages web. JavaScript est donc une extension du code Html des pages web. Les scripts qui s'ajoutent aux balises Html, peuvent en quelques sorte être comparés aux macros d'un traitement de texte. Ces scripts vont être gérés et exécutés par le navigateur lui-même sans devoir faire appel aux ressources du serveur.

Son objectif principal est d'introduire de l'interactivité avec les pages Html et effectuer des traitements simples sur le poste de travail de l'utilisateur tel que les contrôles de saisis pour valider les champs d'un formulaire, ouvrir ou fermer des nouvelles fenêtre ou encore gérer des éléments graphiques.

Les balises annonçant un code JavaScript sont les suivantes :

```
<SCRIPT langage= "JavaScript">
```

Mettez ici votre code

```
</SCRIPT>.
```

jQuery mobile : C'est un Framework d'interface optimisé pour les appareils mobiles tactiles. Son objectif est de permettre de rapidement développer des applications mobiles ou des applications web multi-plateformes (cross-devices), en réponse à la grande diversité des

Smartphones et tablettes sur le marché. Il est développé par l'équipe du projet JQuery. Comme son nom l'indique, il est basé sur jQuery, ainsi que sur HTML5 et CSS3.

Le Framework jQuery Mobile est compatible avec d'autres Frameworks mobiles tels que PhoneGap.

Ses atouts sont très nombreux, en voici les principaux :

- Une documentation complète et bien mise à jour est disponible : [jQuery Mobile Doc](#).
- Une très grande facilité d'utilisation et d'apprentissage : il n'y a pas besoin d'énormément de connaissances ni d'efforts pour en tirer la substance.
- Il est testé et a un rendu optimal sur de nombreuses plateformes mobiles, comme iOS, Android, etc.
- Il permet de créer des sites adaptés aux tablettes et aux smartphones (en responsive).
- Même si les styles sont facilement bidouillables, on peut (presque) totalement s'en passer, la base est très complète et bien pensée.
- Le look de jQuery Mobile peut facilement être modifié.

IV. Présentation des interfaces :

L'objectif de notre projet est de réaliser une application multiplateforme c'est-à-dire la développer une seule fois et pouvoir la faire tourner sous diverses plateformes suivant le slogan « Write Once, Run Any Where » sans avoir à programmer autant de fois que de plateformes ciblées.

Pour concrétiser notre objectif, nous avons opté pour l'approche hybride en se servant du service phonegap build, qui nous a compilé notre application et généré des packages relatifs à chacun des systèmes d'exploitations mobiles supportés.

Voici l'illustration d'accès au service phonegap build depuis deux types de smartphone :

- ✓ Sony Xperia (Android 4.4.4).
- ✓ Nokia Lumia 520 (Windows phone 8.0).

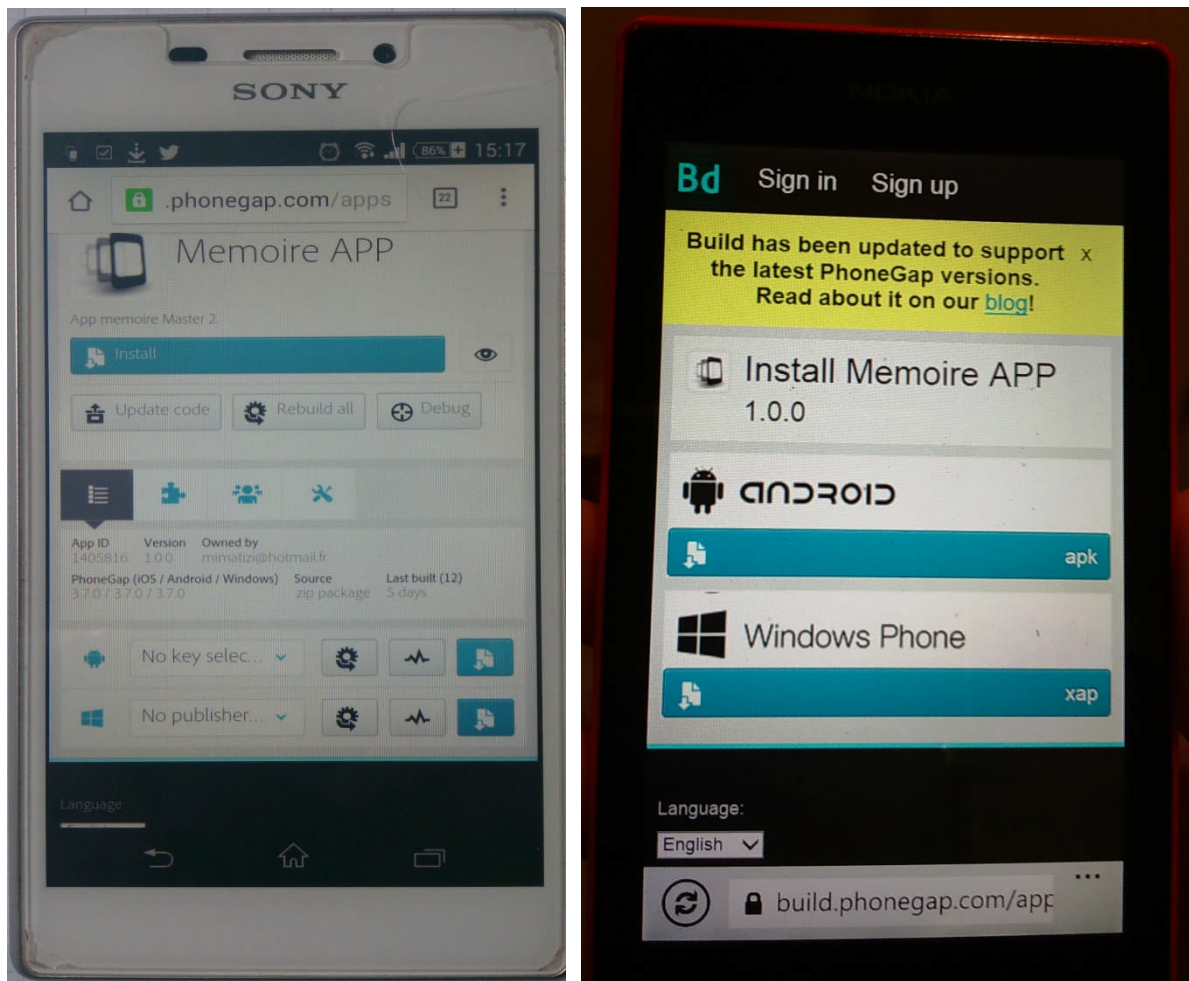


FIGURE 42: Capture des Smartphones Sony(Android) et Nokia (Windows phone).

Dans notre cas, après compilation de notre application, voici les fichiers **.apk** et **.xap** générés respectivement pour Android et Windows phone qui servent à installer des applications sur les terminaux appropriés.

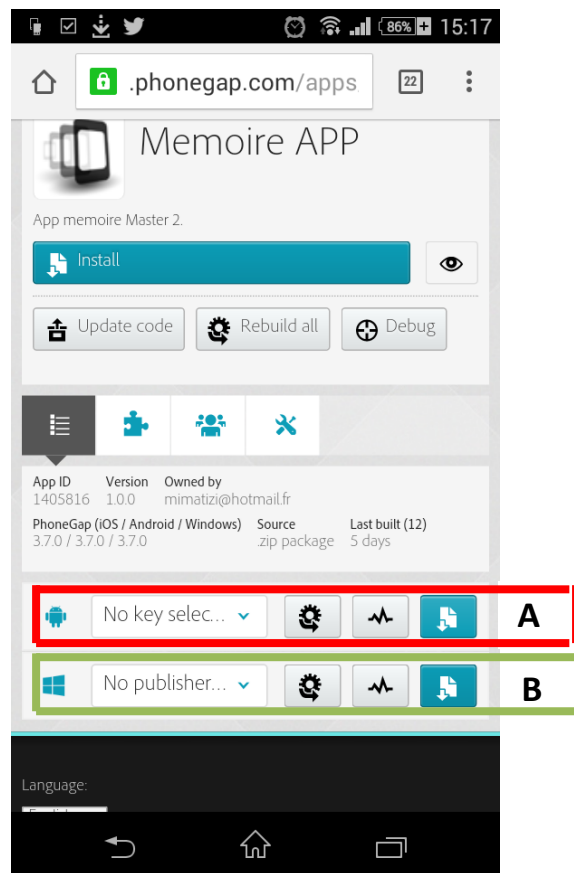


FIGURE 43: capture des packages générés après compilation de l'application

Ensuite, la procédure d'installation est la suivante :

1. Télécharger :

A. l'.APK généré pour Android.

B. XAP généré pour Windows phone.

2. Lancer l'explorateur de l'appareil pour le repérer ou il se trouve.

3. Enfin, l'exécuter, et confirmer la procédure d'installation.

4. Après l'installation de l'application, supprimer le fichier apk/xap afin de faire de la place sur l'appareil.

Nous vous présentons, dans ce qui suit, quelques interfaces de notre application :

La page d'accueil :

C'est le point d'entrée à la plateforme. Elle permet aux différents utilisateurs d'accéder à leur profil.

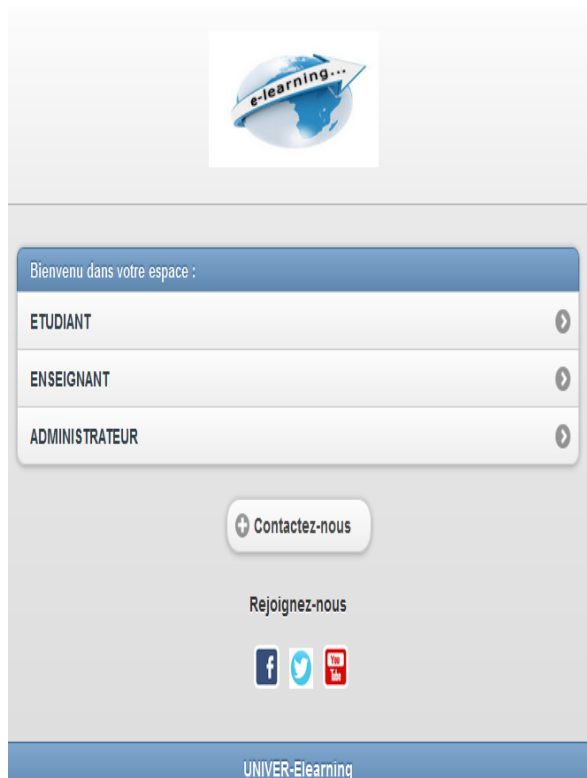
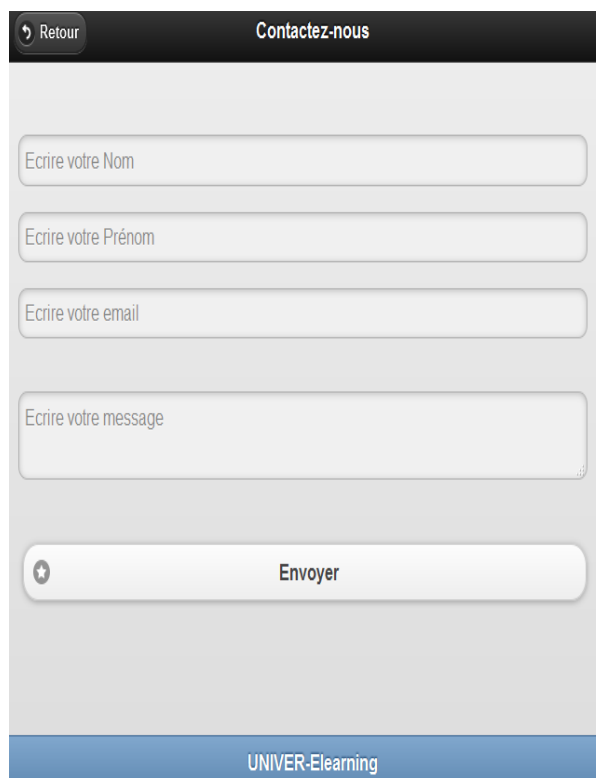


FIGURE 44: la page d'accueil

Elle contient aussi :

-Des liens vers les réseaux sociaux qui représentent un outil d'interactivité entre les différents acteurs de la plateforme.

-Un bouton « contactez-nous » qui permet à l'utilisateur de contacter l'administrateur de la plateforme.

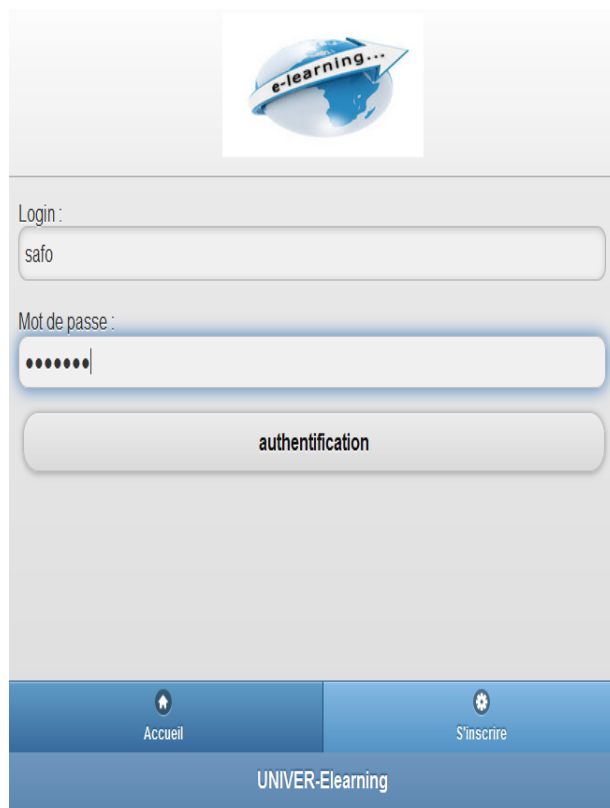


The image shows a mobile application interface for a contact form. At the top, there is a dark header bar with a back arrow and the text 'Retour' on the left, and 'Contactez-nous' in the center. Below the header, there are four text input fields stacked vertically, each with a light gray border and rounded corners. The labels for these fields are 'Ecrire votre Nom', 'Ecrire votre Prénom', 'Ecrire votre email', and 'Ecrire votre message'. Below the message field is a large, light gray button with rounded corners, containing a plus sign icon on the left and the text 'Envoyer' in the center. At the bottom of the screen, there is a blue footer bar with the text 'UNIVER-Elearning' in white.

FIGURE 45: Contact de l'administrateur

Le profil Etudiant :

Lorsqu'on clique sur le bouton « Etudiant » de la page d'accueil, on accède directement à la page d'authentification permettant à un étudiant déjà inscrit de se connecter en renseignant son login et mot de passe.



The screenshot shows the authentication page of the UNIVER-Elearning system. At the top center is a logo featuring a globe with the text "e-learning...". Below the logo, there are two input fields: "Login :" with the text "safo" and "Mot de passe :" with a masked password of seven dots. A large button labeled "authentification" is positioned below the password field. At the bottom of the page, there is a navigation bar with two buttons: "Accueil" (Home) and "S'inscrire" (Sign up), and the text "UNIVER-Elearning" centered below them.

FIGURE 46: la page d'authentification

Si les informations introduites sont erronées, un message d'erreur lui est affiché.

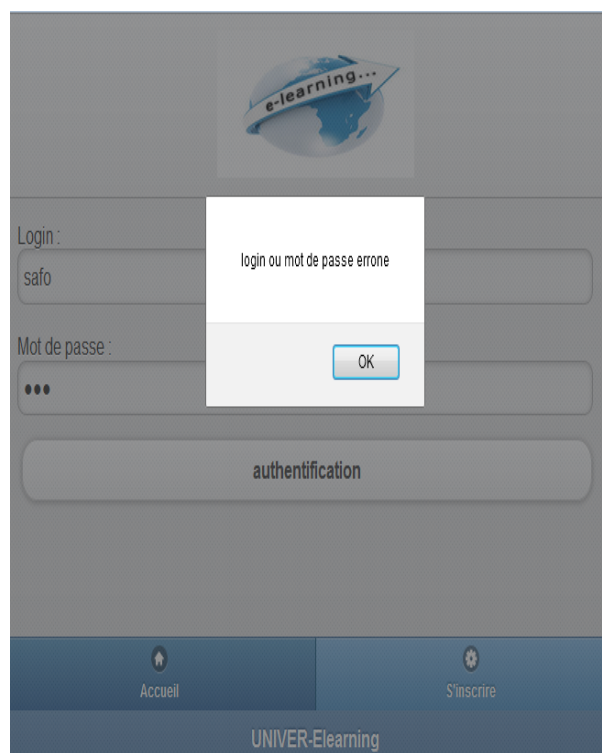


FIGURE 47: message d'erreur dans l'authentification

S'il s'agit d'une première visite, il choisit de s'inscrire en cliquant sur le bouton « s'inscrire », ensuite il doit remplir le formulaire de la page suivant et soumettre.

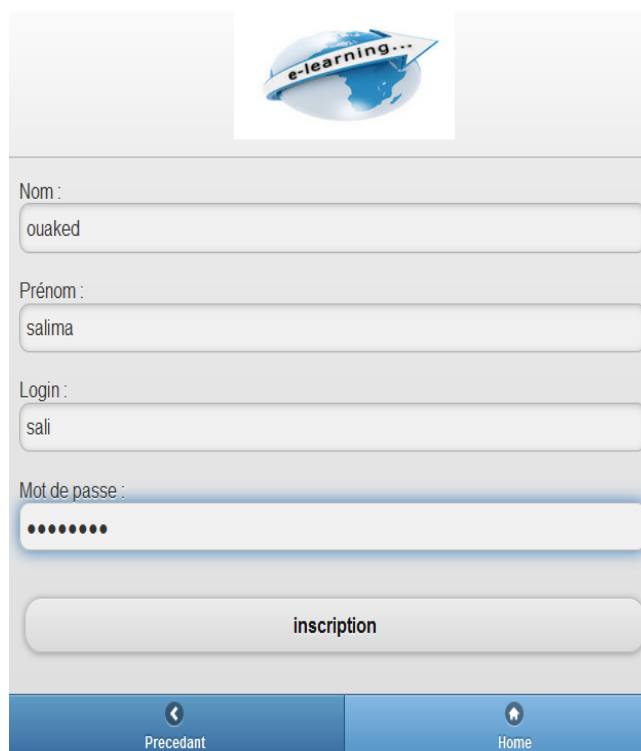


FIGURE 48: page d'inscription

Après authentification, la première page de son espace personnel apparaît.



FIGURE 49: page espace étudiant

En cliquant sur suivant, la page de toutes les filières apparaît.

Ensuite l'étudiant clique sur sa filière pour accéder à sa spécialité.

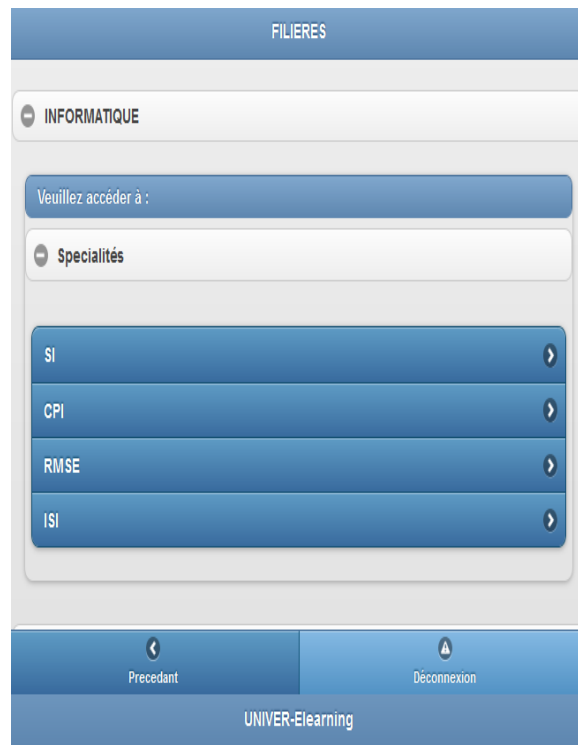


FIGURE 50: page des spécialités

En cliquant sur l'une des spécialités, l'étudiant pourra accéder à l'ensemble des modules.

Pour un module donné, il a le choix de consulter les cours ou bien les PV de notes.

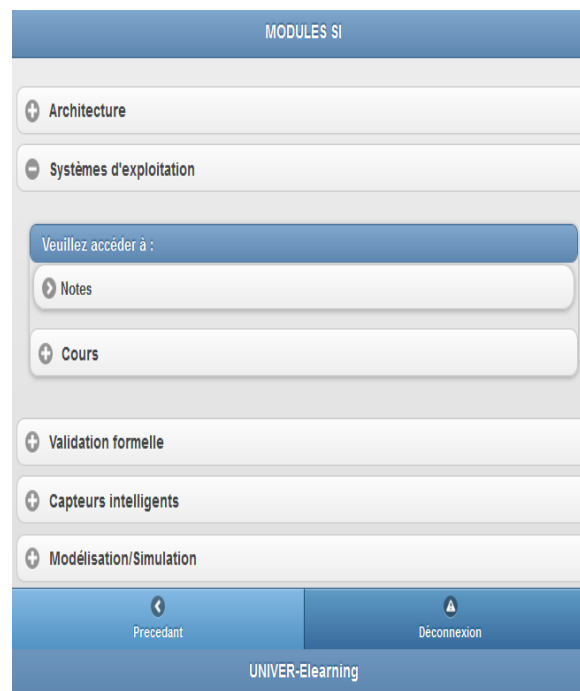
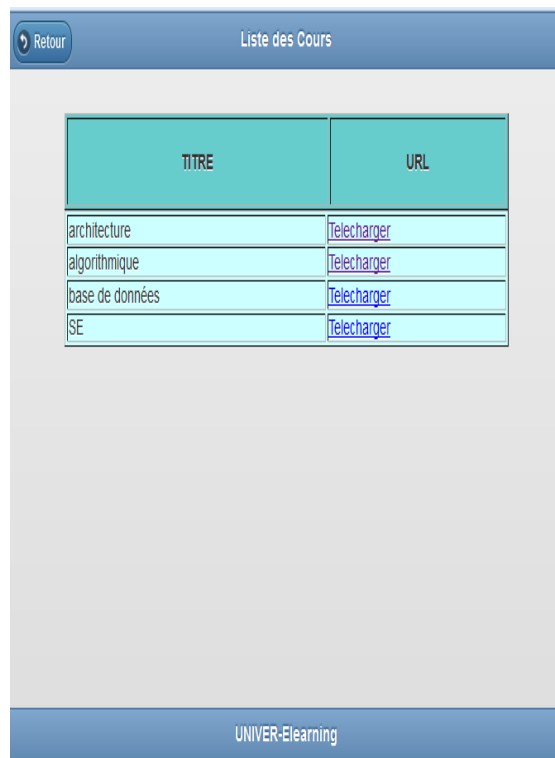


FIGURE 51: page des modules

En choisissant de consulter les cours publiés par ses enseignants, il aura la page suivante.



| TITRE | URL |
|-----------------|-----------------------------|
| architecture | Telecharger |
| algorithmique | Telecharger |
| base de données | Telecharger |
| SE | Telecharger |

FIGURE 52: liste des cours

Profil Enseignant :

Après authentification, l'enseignant atteint son espace personnel qui lui permettra d'accéder à la page de son module, pour publier des cours, des PV de notes ou bien de consulter les cours et PV de notes déjà publiés.

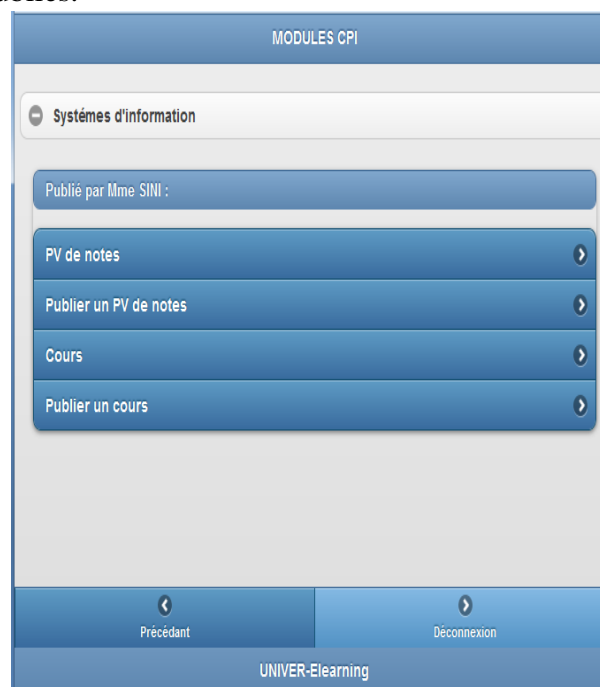


FIGURE 53: page du module de l'enseignant

L'enseignant décide de publier un cours en cliquant sur « publier un cours »

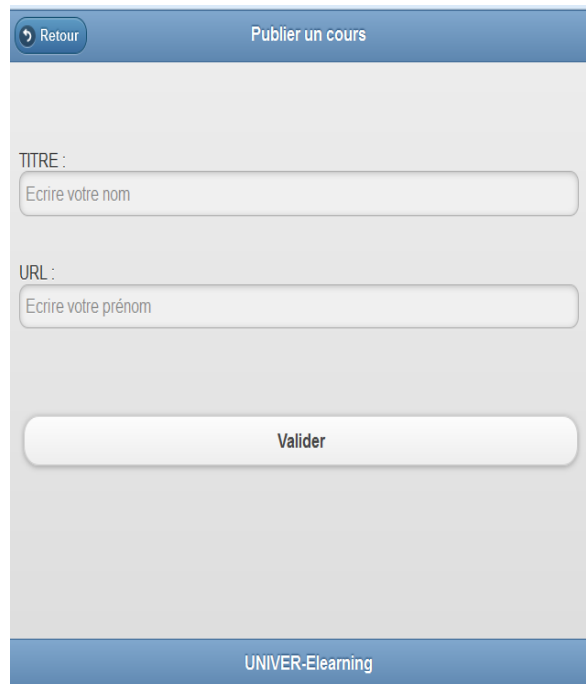


FIGURE 54 : page de création d'un cours

Profil administrateur

L'administrateur dans son espace réservé, possède une multitude de choix de traitements, entre la gestion des étudiants, des enseignants, des cours, des PV de notes ainsi que des filières.

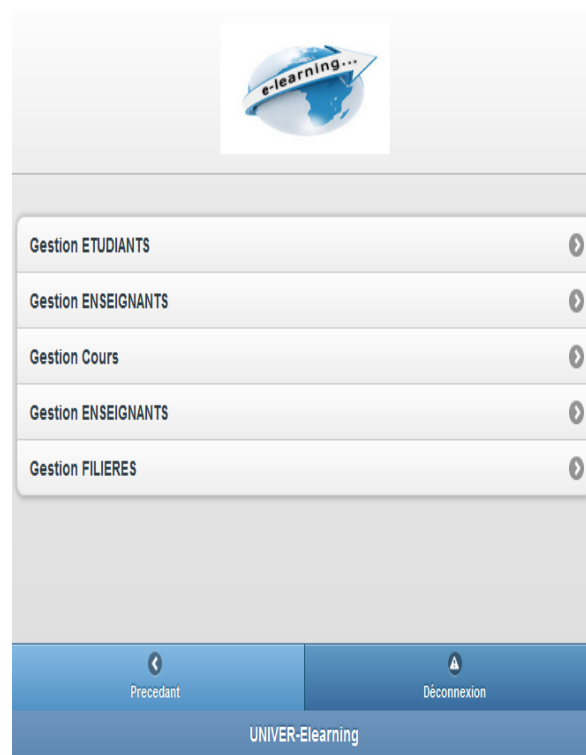


FIGURE 55: page espace admin

Dans la gestion des étudiants, l'administrateur peut en supprimer un.

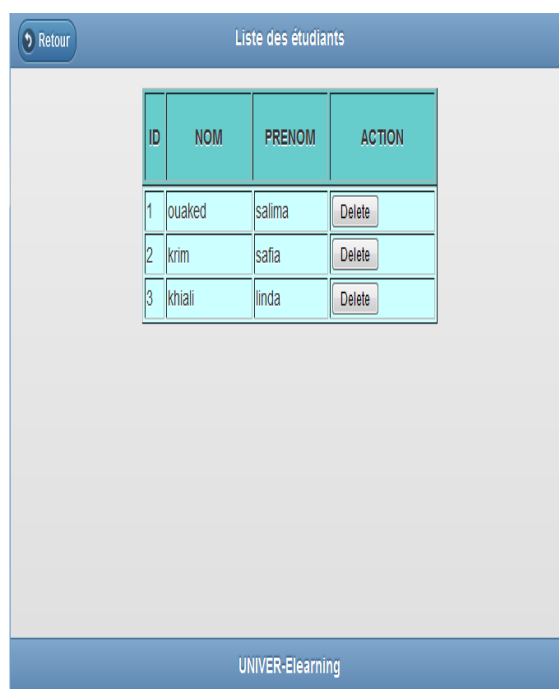


FIGURE 56: page de suppression d'un étudiant

V. Conclusion :

Dans ce chapitre nous avons présenté l'environnement technique dans lequel nous avons développé notre application ainsi que les outils que nous avons utilisés. Nous avons aussi décrit quelques interfaces de notre application.

Conclusion générale :

Nous avons présenté à travers ce travail une étude relative à la conception et la réalisation d'une application d'enseignement à distance accessible depuis différents supports mobiles tout en essayant de répondre à la question suivante : quel est l'intérêt du développement multiplateforme mobile ?

L'intégration de la technologie mobile dans le domaine de l'enseignement à distance engendre d'importants avantages tels que : la possibilité de travailler dans des points géographiquement dispersés, ce qui permet à l'utilisateur d'accéder à l'information où il en a besoin et de gagner du temps sans se soucier du type du terminal d'accès. Ceci permet au domaine de l'e-learning de devenir de plus en plus large qu'auparavant.

Ce travail nous a permis :

- ✓ De nous initier à l'utilisation du support mobile.
- ✓ De nous familiariser avec les technologies et approches de développement multiplateforme.
- ✓ D'approfondir nos connaissances théoriques et pratiques en rapport avec les services web, les réseaux et les bases de données ...
- ✓ D'acquérir de nouvelles connaissances sur les langages HTML, JavaScript, J2EE, JQueryMobile, le langage de modélisation UML, et aussi d'utiliser plusieurs logiciels tels que, Dreamweaver CS6 et le système de gestion de bases de données MySQL.
- ✓ D'avoir les éléments de base d'une nouvelle discipline qui ne cesse de se développer aussi bien dans le secteur public que dans le secteur privé : L'e-learning ou enseignement à distance.

L'application que nous avons réalisée correspond aux objectifs que nous nous étions fixés au départ. Ce projet étant notre première expérience, nous a permis de mettre en pratique les connaissances que nous avons acquies tout au long de notre cursus universitaire.

Nous espérons que notre travail pourra servir de référence aux prochaines promotions, ce projet reste ouvert à toute amélioration et extension notamment celle concernant :

- Enrichissement de la plateforme pédagogique avec les outils de collaboration (forum, chat ..).
- Adoption d'une autre approche de développement multiplateforme.
- Choix du type des services web à utiliser.

Webographie

<http://miratech.fr/fragmentation-technologique-recette-test-utilisateur-mobile>

www.netapsys.fr

<http://revue-eti.net/document.php?id=354>

<http://blogs.msdn.com/b/emargraff/archive/2014/09/11/3-approches-pour-cr-233-er-son-application-mobile-sur-toutes-les-plateformes.aspx>

<http://www.jmdoudoux.fr/java/dej/chap-service-web.htm>

<https://openclassrooms.com/courses/les-services-web>

<http://www.jmdoudoux.fr/java/dej/chap-service-web.htm>

<https://software.intel.com/fr-fr/articles/construire-des-applications-multiplateformes-avec-html5-0>

<http://blog.cedriccornelis.fr/blog/2013/02/25/phonegap-jquery-mobile-et-dreamweaver-un-trio-gagnant/>

<http://blog.dihaw.com/grails/tuto-4-web-service-rest-sous-grails/>

<http://www.croes.org/gerald/blog/qu-est-ce-que-rest/447/>

<http://coenraets.org/blog/2011/10/sample-application-with-jquery-mobile-and-phonegap/>

<https://netbeans.org/kb/docs/websvc/rest.html>

<http://mbaron.developpez.com/tutoriels/soa/developpement-services-web-rest-jaxrs-netbeans/>

<http://www.developpez.net/forums/d582694/general-developpement/alm/architecture/rest/implementer-service-web-restful-java/>

https://apexapps.oracle.com/pls/apex/f?p=44785:24:1748912573189001::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:6084,2

<https://abderrahmanhamila.wordpress.com/2013/08/18/developpement-mobile-natif-ou-multiplateforme/>

I) Langage de modélisation UML

1. Introduction

La démarche adoptée nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

Pour bien représenter l'outil qu'on va concevoir, un recours à un langage bien structuré et complet s'avère indispensable. UML (Unified Modeling language), défini par l'OMG, est à présent un standard de modélisation.

2. Définition

UML (en anglais *Unified Modeling Language* ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toute sorte de système ne se limitant pas au domaine informatique.

La modélisation UML utilise des diagrammes qui permettent de préciser d'une manière claire la structure et le comportement d'un système.

UML comporte treize diagrammes, diagramme de classes, diagramme de composants, diagramme de structurer composite, diagramme de déploiement, diagramme d'objet, diagramme des paquetages, diagramme d'activité, diagramme d'état, diagramme de cas d'utilisation, diagramme de séquences, diagramme global d'interaction et diagramme de temps.

Nous n'allons pas serte étudier tous les diagrammes existant, mais nous allons juste spécifier les diagrammes qui vont nous permettre de voir comment interagit l'application que nous allons développer dans son contexte de fonctionnement. Pour se faire la connaissance de l'entreprise (ou bien d'un service de l'entreprise) pour laquelle l'application est prévue est nécessaire.

Les éléments de modélisation

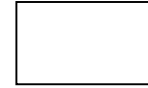
- Les objets :

Une entité d'un monde réel ou virtuel



- Les classes :

La description d'un ensemble d'objets



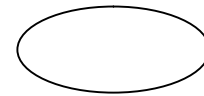
- Les états :

Une étape de la vie d'un objet



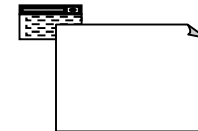
- Les cas d'utilisation :

Une manière dont un acteur utilise le système



- Les notes

Un commentaire, une explication ou une annotation.



Les relations

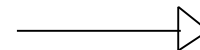
- L'association :

Une connexion sémantique entre instances.



- La généralisation :

Une relation de classification.



- La dépendance :

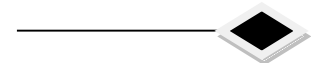
L'utilisation d'un élément par un autre.



- Composition :

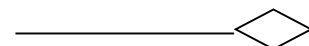
C'est une relation qui signifie que

les données de la classe cible (composante) sont des données de la classe source (composite).



- Agrégation :

Est une composition faible.



- Etat initial

L'état initial marque le point d'entrée de la première activité.

Il est représenté par un cercle plein. Il ne peut y avoir qu'un seul état initial sur un diagramme.



- Etat final :

L'état final marque la fin du déroulement de l'opération finale multiple sur un diagramme. Ils sont représentés par un cercle plein entouré d'un autre cercle.



- Barre de Synchronisation :

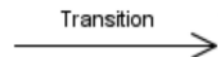
Souvent, certaines activités peuvent être faites en parallèle. Pour dédoubler le traitement "Fork", ou le reprendre quand des activités multiples ont été accomplies ("join"), des barres de synchronisation sont utilisées. Celles-ci sont modélisées par des rectangles pleins, avec des transitions multiples entrantes ou sortantes



- La transition :

Quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions sont utilisées pour marquer ce passage.

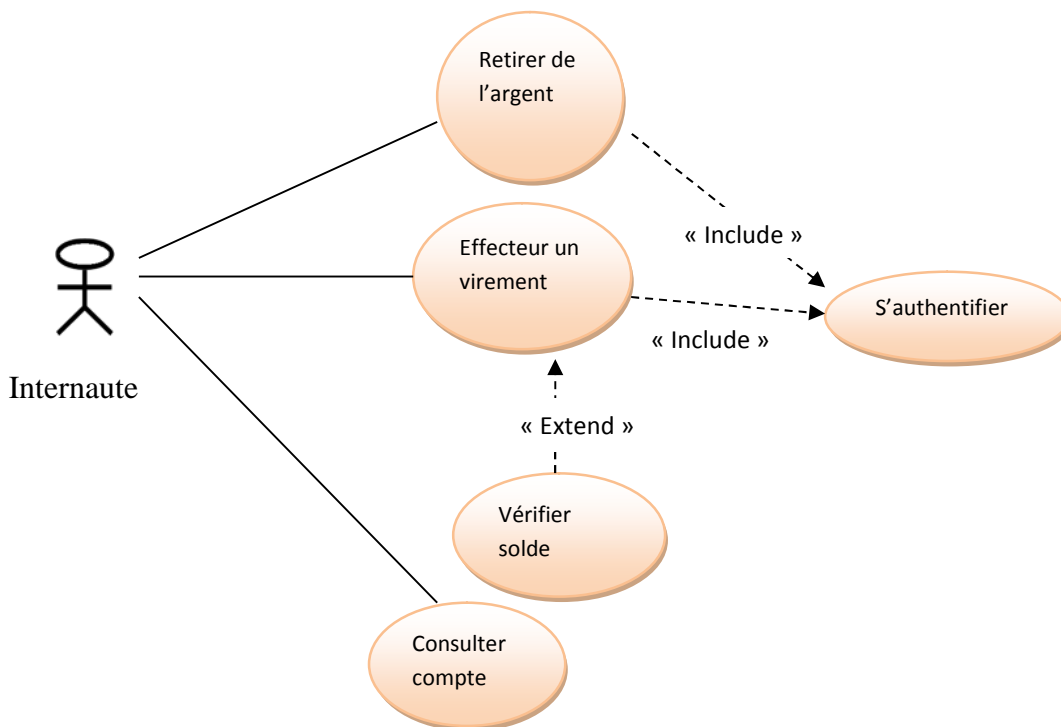
Les transitions sont modélisées par des flèches.



3. Les diagrammes utilisés

- **Diagramme de cas d'utilisation (use case) :** il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

Exemple :



Commentaires :

▪ Acteur: entité externe qui agit sur le système (opérateur, autre système...). Un acteur peut consulter ou modifier l'état du système. En réponse à l'action de cet acteur, le système fournit un service qui correspond à son besoin.

Exemple: Internaute est un Acteur

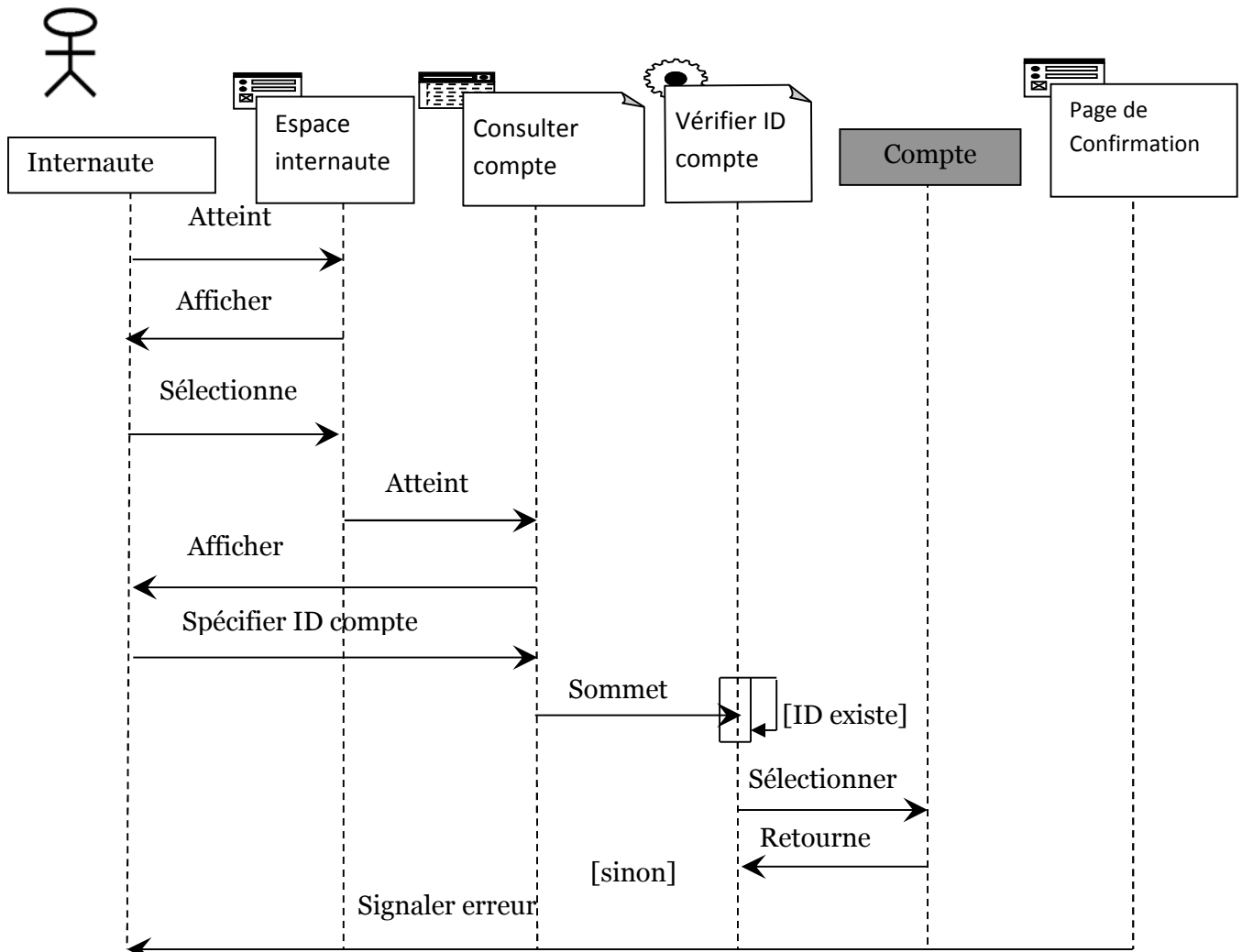
▪ La relation extend : est une relation entre 2 instances de cas d'utilisation telle que A extend B signifie que le comportement de B peut être complété par le comportement de A. La relation extend indique une possibilité, un complément possible.

▪ La relation Include : est une relation entre 2 instances de cas d'utilisation telle que la réalisation de l'un nécessite la réalisation de l'autre.

- **Diagramme de séquence** : Il représente la succession chronologique des opérations réalisées par un acteur : saisir une donnée, consulter une donnée et lancer un traitement. Le diagramme de séquence indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. Dans un diagramme de séquence, les objets sont associés à une ligne de vie. La dimension verticale de celle-ci représente l'écoulement du temps (du haut vers

le bas). Notons que la disposition des objets sur l'axe horizontal n'est pas importante dans ce type de diagrammes. Les messages sont représentés par des flèches et leur ordre est donné par leurs positions sur la ligne de vie. Le concept de message unifie toutes les formes de communication entre objets.

Exemple : consulter une compte.



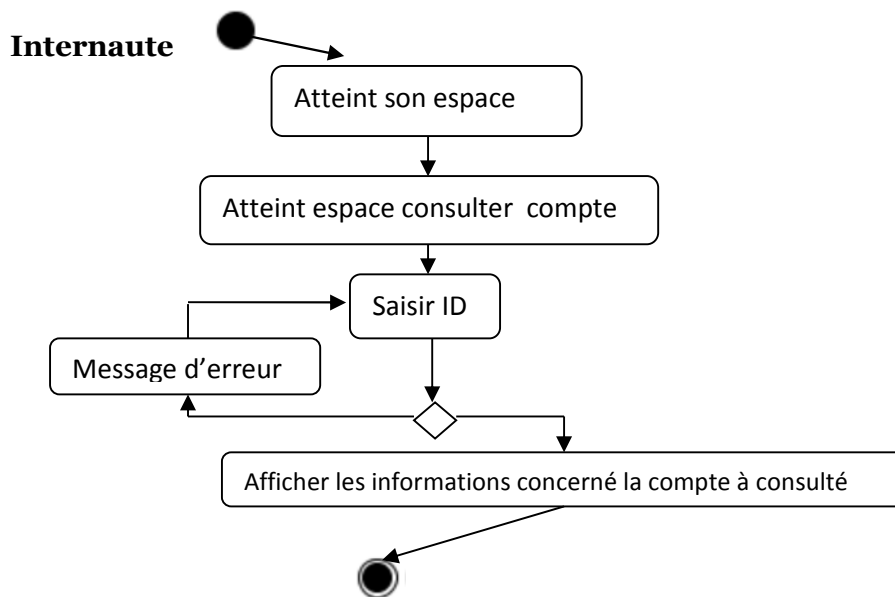
1. L'internaute atteint son espace.
2. Le système affiche son contenu.
3. L'internaute atteint le formulaire consulter compte.
4. Le système affiche son contenu.
5. Le manager sélectionne consulter compte.
6. Le système affiche son contenu.

7. Le manager spécifie la compte à consulter (en donnant son identifiant).

8. Le système spécifie si l'identifiant existe dans le cas positif le système sélectionne la compte dans la base de données et la retourne au internaute sinon le système renvoie un message d'erreur.

- **Diagramme d'activité:** permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

Exemple :



- **Diagramme de classe:** Le diagramme de classe représente l'architecture conceptuelle du système. Il décrit les classes que le système utilise ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation).

Exemple :

