

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Faculté Des Sciences Exactes
Département d'informatique

MÉMOIRE DE FIN DE CYCLE

En vue d'obtention de diplôme de Master professionnel en
informatique

Spécialité : ADMINISTRATION ET SÉCURITÉ DES
RÉSEAUX

Thème

*Conception et Réalisation d'une Application
Android pour la Gestion des Mutuelles
Cas : MTEE (Mutuelle des Travailleurs de
l'Environnements Economiques)*

Présenté par :

M'SILI Souhila & AREZKI Mounira

Devant le jury :

Président	Mr MIR Foudil
Examinateur	Mr ELSAKAAN Nadim
Promoteur	Mr OUZEGGANE Redouane

Promotion : 2015-2016

Dédicaces

Ce modeste travail est dédié à nos très chers parents pour tout les sacrifices, et les encouragement, conseils et leurs amour durant toute la période de nos études, nous profitons cette occasion pour leurs dire que Dieu les protègent pour nous et leur donnent tout le bonheur du monde qu'il méritent.

À nos frères et sœurs en leur souhaitant un avenir radieux et plein de réussite et toute nous fammille.

À tous nos amis qui nous avons partagés les meilleurs moments de notre vie.

À notre encadreur et à tous les enseignants et le personnel de l'université de Bejaia.

Remerciements

avant de commencer cette oeuvre, nous tenons à remercier le bon Dieu qui nous à donner la volonté et le courage pour réalisé ce modeste travail.

Un grand merci pour nos familles, surtout nos parents qui nous ont épaulés, soutenus et suivis tout au long de ce projet.

Nous tenons à exprimer notre vive gratitude à notre promoteur **M. OUZEGGANE Redouane** pour nous avoir honoré par son encadrement, pour sa disponibilité, ses orientations, ses précieux conseils et ses encouragements qui nous ont permis de mener à bien ce travail.

Nos remerciements vont également aux membres de jury **M. MIR Foudil** et **M. ELSAKAAN Nadim** qui nous ont fait l'honneur de juger notre travail.

Nos sincères remerciements s'adressent aussi au personnel du **MTEE** particulièrement **MME. AISSOUkarima** pour leur aide précieuse et conseils qu'ils nous ont données tout au long de notre travail.

Enfin, nous remercions toutes les personnes ayant contribué de près ou de loin à la réalisation de ce travail.

TABLE DES MATIÈRES

Table des matières	IV
Liste des figures	V
Liste des tableaux	VI
Listes des abréviations	VII
Introduction générale	1
1 Etat de l'Art sur les Système Android	2
1.1 Introduction	2
1.2 L'informatique mobile	2
1.2.1 présentation	2
1.2.2 Les Smartphones	3
1.3 Le Système d'exploitation Android	3
1.3.1 Présentation	3
1.4 Fonctionnalités d'Android	3
1.5 Architecture Android	4
1.6 Cycle du vie d'une application Android	4
1.7 Conclusion	5
2 Etude Preliminaire Et Specification Des Besions	6
2.1 Introduction	6
2.2 Définition de la MTEE	6

2.3	Les principes de base de la mutuelle	7
2.4	Etat de prestation de la mutuelle MTTE	7
2.4.1	Intitulé des Prestations	7
2.4.1.1	Prestations Individuelles	7
2.4.1.2	Prestations Facultatives	8
2.5	Organisation de la mutuelle MTEE	8
2.5.1	Les organes statutaires de la mutuelle	8
2.6	Situation Informatique de la MTEE	10
2.7	Présentation du Travail et problématique	11
2.8	Solution proposée	12
2.9	conduite de projet	13
2.9.1	Présentation du langage UML et processus UP	13
2.9.1.1	Définition d'UML	13
2.9.1.2	Diagramme de l'UML	13
2.9.1.3	Le processus unifié	14
2.10	Spécification des besoins	15
2.10.1	Besoins fonctionnels	15
2.10.2	Besoins non fonctionnels	15
2.11	Conclusion	16
3	Analyse Des Besoins et Conception	17
3.1	Introduction	17
3.2	Analyse Des Besoins	18
3.2.1	Identification des acteurs	18
3.2.1.1	Définition	18
3.2.1.2	Identifier les acteurs de notre système	18
3.2.2	Identification des messages	18
3.2.3	Diagramme de contexte	19
3.2.4	Identifications des cas d'utilisation	20
3.2.4.1	Définition	20
3.2.5	Diagramme de cas d'utilisation	20
3.2.5.1	«le diagramme de cas d'utilisation »	20
3.2.5.2	Description textuelle des cas d'utilisation :	21
3.2.6	Diagrammes de séquence	22

3.2.6.1	Diagramme de séquence «s’authentifier»	23
3.2.6.2	Diagramme de séquence «Afficher les statistiques de cotisations»	24
3.2.6.3	Diagramme de séquence «Afficher les statistiques de prestations»	24
3.2.6.4	Diagramme de séquence «Afficher les statistiques par adhérent»	25
3.3	Conception	26
3.3.1	Diagrammes de séquence détaillé	26
3.3.1.1	Diagrammes de séquence détaillé«s’authentifier»	26
3.3.1.2	Diagrammes de séquence détaillé«Afficher les statistiques de cotisations »	27
3.3.1.3	Diagrammes de séquence détaillé«Afficher les statistiques de prestations »	28
3.3.1.4	Diagrammes de séquence détaillé«Afficher les statistiques par adhérent »	29
3.3.2	Diagramme de classes	30
3.3.3	Les éléments de base d’un diagramme de classe	30
3.4	Conclusion	33
4	Réalisation	34
4.1	Introduction	34
4.2	Applications mobiles	34
4.2.1	Définition des applications mobiles	34
4.2.2	Types d’applications mobiles	35
4.3	Implémentation de l’application	35
4.3.1	Outils et langages de développement	35
4.3.1.1	Java	35
4.3.1.2	XML	36
4.3.1.3	IDE Eclipse	36
4.3.1.4	Le plugin ADT (Android Développement Tools)	37
4.3.1.5	SDK Android)	37
4.3.2	Librairies utilisées	37
4.3.2.1	JSON	37
4.3.2.2	JDBC	37

4.4	Architecteur de notre application	38
4.4.1	Diagramme déploiement	38
4.4.2	Architecture globale du Système	39
4.5	Conclusion	40
	Conclusion générale	41
	Bibliographie	42

TABLE DES FIGURES

2.1	Organigramme de la MTEE	10
2.2	Architecteur de communication entre l'application de gestion MTEE et le serveur d'application	11
3.1	Diagramme de contexte	19
3.2	Diagramme de cas d'utilisation globale	20
3.3	Diagramme de séquence de cas d'utilisation «S'authentifier»	23
3.4	Diagramme de séquence de cas d'utilisation «Afficher les statistiques cotisations »	24
3.5	Diagramme de séquence de cas d'utilisation «Afficher les statistiques prestations»	24
3.6	Diagramme de séquence de cas d'utilisation «Afficher les statistiques par adhérent»	25
3.7	Diagramme de séquence système détaillé «S'authentifier»	26
3.8	Diagramme de séquence système détaillé «Afficher les cotisations »	27
3.9	Diagramme de séquence système détaillé «Afficher les prestations »	28
3.10	Diagramme de séquence système détaillé «Afficher les statistiques par adhérent »	29
3.11	Diagrammes de classes	32
4.1	Diagramme Déploiement	38
4.2	Architecture globale du Système	39

LISTE DES TABLEAUX

3.1	représente les messages échangées entre acteur et système	19
3.2	Cas d'utilisation s'authentifier	21
3.3	Cas d'utilisation afficher statistiques cotisations	21
3.4	Cas d'utilisation afficher statistiques prestations	22
3.5	Cas d'utilisation afficher statistiques par adhérent	22

LISTES DES ABRÉVIATIONS

ADT : **A**ndroid **D**evelopment **T**ools.

AG : **A**ssemblée **G**énérale.

CA : **C**onseil d'**A**dministration.

CC : **C**onseil de **C**ontrolé .

CMA : **C**aisse **M**utualiste d' **A**lgérie.

DG : **D**irection **G**énérale.

IDE : **I**ntegrated **D**evelopment **E**nvironnement.

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

JSON : **J**ava **S**cript **O**bjct **N**otation.

JVM : **J**ava **V**irtuel **M**achine.

HTTP : **H**epert**T**ext **T**ransfer **P**rotocol.

MTEE : **M**utuelle des **T**ravailleurs de l' **E**nvironnement **E**conomique.

UP : **U**nified **P**rocess.

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

SDK : **S**oftwore **D**evelopment **K**it.

XML : **E**xtensible **M**arkup **L**anguage.

INTRODUCTION GÉNÉRALE

Les progrès technologiques récents ont permis l'apparition d'une grande variété de nouveaux moyens permettant à un utilisateur d'accéder et d'utiliser l'information qui l'intéresse en tout lieu couvert par le réseau et à tout moment. L'accès au contenu ne s'effectue plus exclusivement de la même façon ni par les mêmes appareils qu'il y a quelques années. Ces nouveaux appareils, fruits d'une véritable révolution technologique, ont pour noms : assistants personnels, téléphone cellulaires, smartphone, etc. Le nombre d'utilisateurs de ces nouveaux appareils continue sa croissance exponentielle. Les moyens d'accès au contenu ont également évolué, avec de nouveaux réseaux tels que les réseaux sans fils (Wifi, GPRS, etc.).

L'objectif de ce projet, est de concevoir et d'implémenter une application mobile, pour le système d'exploitation ANDROID, accessible à travers des smartphones ou des tablettes, qui permettent la gestion de la MTEE en offrant sous forme d'un tableau de bord toutes les informations nécessaires pour la direction de la mutuelle, ainsi assurer une rapidité d'accès aux informations et une mobilité totale qui assurera une meilleure gestion de la MTEE.

Le présent mémoire est structuré en quatre principaux chapitres. Après cette introduction, nous définissons, dans le premier chapitre, quelques généralités sur la gestion des mutuelles, nous avons ensuite consacré le deuxième chapitre à l'étude préliminaire et la spécification des besoins, le troisième chapitre présente l'analyse des besoins et la conception de l'application. Le quatrième chapitre est dédié à la phase de réalisation et implémentation de notre application, en décrivant les outils et langages utilisés, qui ont permis l'implémentation de notre application suivi d'une présentation de quelques interfaces développées.

Enfin, nous concluons ce travail avec une conclusion générale qui synthétise tout le travail qui a été fait ainsi que quelques perspectives.

CHAPITRE 1

ETAT DE L'ART SUR LES SYSTÈME ANDROID

1.1 Introduction

Nous allons présenté dans ce chapitre une définition globale de l'informatique mobile et du système d'exploitation Android, ainsi que ces fonctionnalités, son architecture et le cycle de vie d'une activité Android.

1.2 L'informatique mobile

1.2.1 présentation

l'informatique mobile sinscrit sous l'approche globale de l'informatique ubiquitaire. Historiquement, elle a été autour depuis 1992. Depuis, elle constitue un outil de communications très puissant pour les entreprises et les utilisations personnel[1]. Les supports qui lui ont été développés ont repris l'industrie sans fil. Il s'agissait des entités mobiles communicantes et parfois de très petites tailles permettant de se connecter aux différents types de réseaux tant qu'ils disposent des périphériques appropriés dont les Smartphones font partie[1].

1.2.2 Les Smartphones

Un Smartphone est un téléphone «intelligent», appelé aussi ordi-phone, c'est un appareil dédié aux communications mobiles, disposant d'un Système d'exploitation ouvert et adoptant des applications tierces développées par fabricant, par l'opérateur ou par un éditeur de logiciel.

1.3 Le Système d'exploitation Android

1.3.1 Présentation

L'histoire de la naissance d'Android a commencé en aout 2005, lorsque Google a acquis Android INC, une startup qui développent des applications pour téléphones mobiles. Depuis lors, Andy Rubin, étant un ancien d'Android INC, a entamé son travail sur un système d'exploitation basé sur un noyau linux dédié aux appareils mobiles. En 2007, le 5 novembre, l'Open Handset Alliance fut officiellement annoncée. c'est un consortium qui réunit un ensemble de sociétés, ayant pour but de développer des standards open sources pour terminaux mobiles. Le 15 novembre, le premier standard a été annoncé. Il s'agissait d'android, une plateforme pour appareils mobiles. En effet, Android représente un système d'exploitation open source dédié pour Smartphone, PDA et terminaux mobiles.

Il est basé essentiellement sur la simplicité d'utilisation et surtout sur une capacité de personnalisation importante présentant un argument commercial de poids. Pour promouvoir ce système d'exploitation open source, Google lui a conféré les alliés puissants réunis au sein de l'Open Handset Alliance tel que samsung, Motorola, Sony Ericsson et LG. les principaux concurrents d'Android sont apple avec iPhone OS qui équipe l'iPhone, Research IN Motion avec BlackBerry OS, Palm avec Nova ou web OS, Microsoft et son Windows Mobile, etc.[3].

1.4 Fonctionnalités d'Android

Android a été conçu pour mieux intégrer au mieux les applications existantes de Google comme le service de courrier Gmail, l'agenda Google calendrier ou encore la cartographie. Les fonctionnalités proposées par Android sont différentes d'une version à une autre, on peut citer celle de la version utilisée dans le cadre de notre travail (version 2.2)

- Augmentation de la performance et de la vitesse ;

- Fonctionnalité de Hot spot Wifi ;
- Partage du contact sur Bluetooth ;
- Mise à jour automatique des applications[2].

1.5 Architecture Android

Android bénéficie d'une architecture en couche complète faisant de lui une plateforme riche, dédiée aux appareils mobiles. Il est basé sur le noyau Linux profitant des services système de base tel que la sécurité, la gestion mémoire, gestion du processus, etc. Ici ce trouve un ensemble de bibliothèques écrites en C/C++ jouant le rôle d'un middleware (on en cite le système bibliothèque C, les médiathèques, le SGL, etc.). C'est sur cette couche que se greffe l'Android Runtime, comprenant la machine virtuelle de java et ses bibliothèques. Vient ensuite la plateforme logicielle, nommée aussi framework de développement, écrite en java et permettant de mutualiser les ressources entre les applications java. Elle offre aux développeurs la possibilité de produire des applications diverses et innovantes à travers à travers un ensemble d'API. Enfin, à un niveau plus supérieur se situe un ensemble d'applications sous forme de paquets APK (Android Package). Les applications fournies par Android sont telles qu'un navigateur web, un client-email, un calendrier, un gestionnaire de contacts, etc[2].

1.6 Cycle du vie d'une application Android

Les technologies mobiles prennent de plus en plus de place sur le marché. Les Smartphones sont considérés comme des petits ordinateurs dotés d'un système d'exploitation s'appuyant sur un noyau Linux. Cependant, ils diffèrent des ordinateurs classiques par le cycle de vie d'une application. Sous Android, une application est composée d'une ou plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateurs. Afin de faciliter la cinématique de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité [3].

1.7 Conclusion

Dans ce chapitre, nous avons fait une étude du système d'Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et son architecture, à savoir les principaux composants du système. Ceci nous a permis de comprendre et de les exploiter pour proposer une extension à l'application existante de la MTEE (Mutuelle des Travailleurs de l'Environnement économique).

CHAPITRE 2

ETUDE PRELIMINAIRE ET SPECIFICATION DES BESIONS

2.1 Introduction

Avant de concevoir un système informatique, il est nécessaire d'effectuer une étude préliminaire du système dans le but d'identifier les besoins et exigences liées aux futurs utilisateurs. Nous aurons à étudier un ensemble de documents qui décrivent le système d'une manière formelle et exhaustive par des textes et des diagrammes, mais avant cela une étude préalable décrivant l'existant sera réalisée.

2.2 Définition de la MTEE

La Mutuelle des travailleurs de l'environnement économique (MTEE) est une personne morale régie par les dispositions de la loi 15/02 du 04/01/2015 et les textes y afférents sous l'égide du Ministère du travail, de l'emploi et de la sécurité sociale. Elle est un organisme de solidarité et d'entraide à vocation nationale, a pour objet d'assurer à ses adhérents et à leurs ayants droit des prestations multiples et agir par conviction pour le bien de tous. Elle regroupe actuellement plus de trente(30) organismes différents secteurs d'activités étalés sur le territoire national (entreprise économique, administrations publiques et fonction libérale).

2.3 Les principes de base de la mutuelle

la mutuelle s'articule autour des principes suivants :

- Tous les membres ont les mêmes droits et obligations ;
- Tous doivent participer, directement ou indirectement, par l'intermédiaire de leurs représentants au fonctionnement de la mutuelle ;
- Elle permet d'adapter les services offerts aux besoins de ses membres ;
- Les considérations économiques et les principes de bonne gestion ne doivent pas être négligés car ils contribuent à la réalisation du bien-être collectif ;
- L'adhérent se doit d'observer une certaine loyauté envers la MTEE et envers les autres membres ;
- Les membres de l'organe dirigeant exercent leurs fonctions gratuitement.

2.4 Etat de prestation de la mutuelle MTTE

2.4.1 Intitulé des Prestations

2.4.1.1 Prestations Individuelles

- **Frais médicaux :**
 - Produit Pharmaceutique ;
 - Consultation Généraliste ;
 - Consultation Spécialiste ;
 - Analyse Médicale ;
 - Radio (Scanner, Mammographie) ;
 - Séjours Clinique ;
 - Intervention Chirurgicale ;
 - Congé de maladie ;
- **Indemnité :**
 - Lunetterie Monture ;
 - Lunetterie Verre ;
 - Prothèse Dentaire (1 Mâchoire) ;
 - Prothèse Dentaire (2 Mâchoires) ;
 - Bridge et soin dentaire ;

- **Prime :**
 - Prime de Naissance ;
 - Prime de Circoncision ;
 - Prime de Scolarité ;
 - Prime de Bac ;
 - Prime de mariage ;
- **Allocation :**
 - Allocation d'handicape ;

2.4.1.2 Prestations Facultatives

- **Décès :**
 - Capital décès ;
 - Frais Funéraires (Père, Mère, Conjoint et Enfant) ;
- **Retraite Complémentaire**
- **Prêt social :**
 - Prêt logement ;
 - Prêt Mariage ;
- **Aide Sociale :**
 - Pèlerinage (OMRA) ;
 - Assurance tous risque du nouveau véhicule utilitaire ;
- **Activités Culturelles et Sportives**
- **Formation Continue ou qualifiante**

2.5 Organisation de la mutuelle MTEE

2.5.1 Les organes statutaires de la mutuelle

La mutuelle des travailleurs de l'environnement économique est placée sous l'égide du ministère du travail de l'emploi et de la sécurité sociale, direction générale de l'emploi et de la sécurité social.Elle est administrée par une équipe homogène et dynamique, elle s'articule autour de quatre principaux organes légalement constitués conformément aux règles de loi en vigueurs qui sont :

- **Assemblée Générale (AG) :**

Elle est l'instance suprême de la MTEE et prend toute les décisions relatives à sa vie, elle est composée de l'ensemble des représentants élus des organismes adhérents à la mutuelle.

– **Conseil d'administration (CA) :**

La mutuelle est administrée par un conseil d'administration composé de six (06) membres élus par l'assemblée générale en son sein, a pour mission de superviser, au nom de l'Assemblée Générale la gestion de la Mutuelle.

Le conseil d'administration est constitué de Bureau du conseil d'administration.

– **Bureau du conseil d'administration :**

Le bureau du conseil d'administration est constitué de trois (03) membres élus par le conseil d'administration en son sein. Il est chargé d'assurer l'exécution des décisions du conseil d'administration.

– **Conseil de Contrôle (CC) :**

Il est composé des membres élus par l'AG et a pour mission de :

- Contrôler l'exactitude de la comptabilité et de régularité ;
- Contrôler l'exécution des décisions de l'AG ;
- Elaborer un rapport de contrôle directement à l'Assemblée Générale.

– **Structure de Gestion (Direction Générale) :**

La MTEE est dotée d'une structure de gestion administrative et comptable fonctionnant à plein temps organisée comme suit :

- Directeur général ;
- Assistant(e) du directeur général ;
- Service administration et finance (comptable) ;
- Service prestations ;
- Service étude et développement.

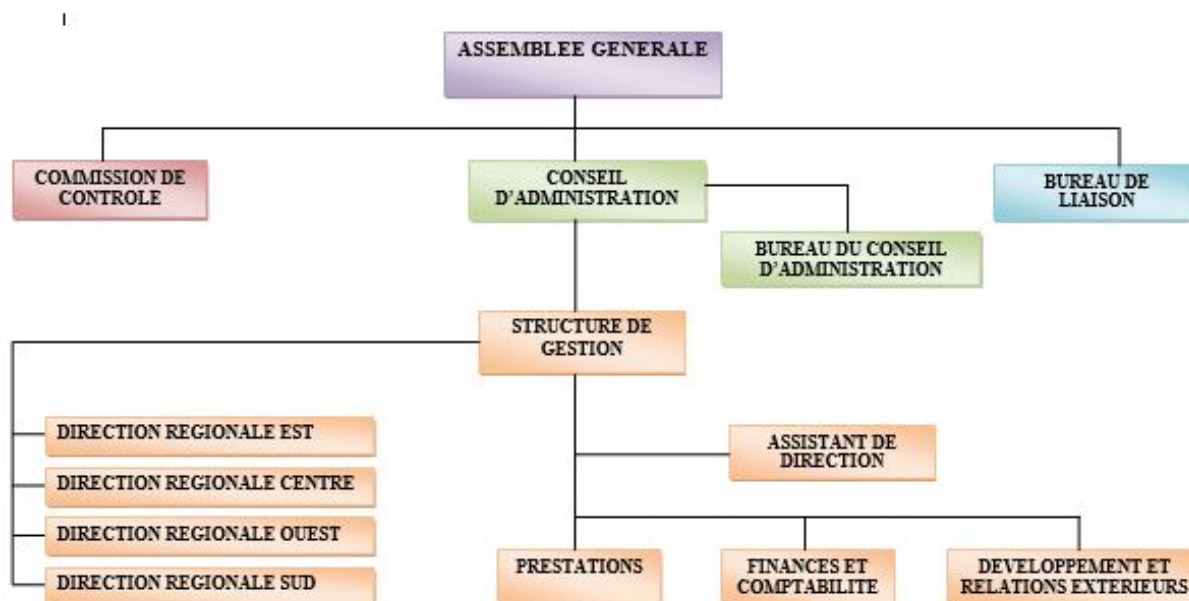


FIGURE 2.1 – Organigramme de la MTEE

2.6 Situation Informatique de la MTEE

La mutuelle MTEE dispose de quatre ordinateurs de travail qui sont connectés à une machine serveur via un réseau sans fil. cette dernière (la machine serveur), permet d'exécuter les différents serveur nécessaire pour l'application à savoir :

- Serveur de données MySQL ;
- Serveur d'application dédié, qui permet d'accéder à la base de données pour répondre au différentes requêtes applicatives. Les ordinateurs du travail permettent d'exécuter une application lourde (Java/Swing) qui permet la gestion de la mutuelle (gestion des ordinateurs, adhérents, cotisation, prestations les virement,...). Cette application communique avec le serveur pour obtenir les données nécessaires ou réaliser des traitements spécifiques.

La figure 2.2 illustre cette architecture de communication :

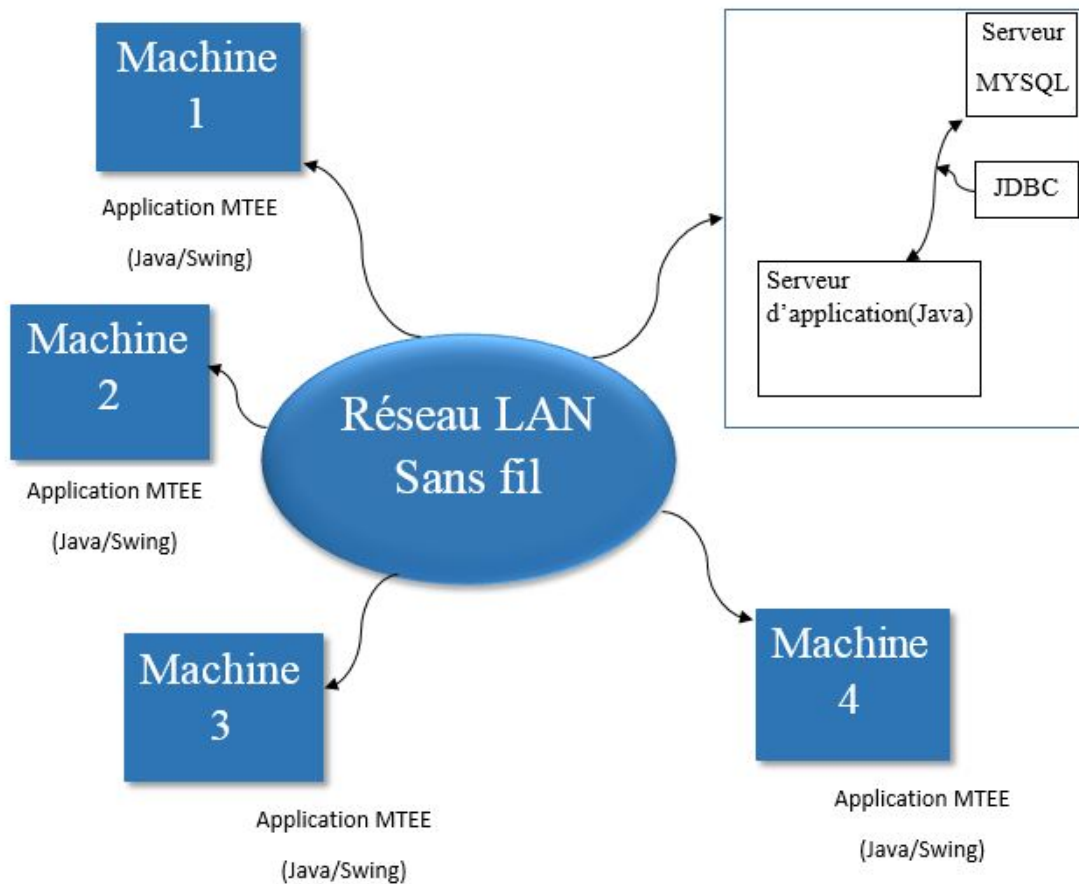


FIGURE 2.2 – Architecture de communication entre l'application de gestion MTEE et le serveur d'application

2.7 Présentation du Travail et problématique

Durant notre stage dans la MTEE nous avons constaté que l'application utilisée ne permet pas des statistiques globales et détaillées de la MTEE. Ces statistiques permettent de constituer un tableau de bord indispensable aux décideurs (directeur ou comptable). Pour remédier à ce problème notre travail consiste à concevoir et à développer une application mobile permettant d'afficher, sous forme d'un tableau de bord, toutes les informations nécessaires pour les décideurs (la direction).

La mobilité permet au responsable d'accéder à ce tableau de bord sans contrainte temporelle ou spatiale.

2.8 Solution proposée

Compte tenu des problèmes déduits de notre étude préalable et dans le souci de trouver des solutions appropriées, nous proposons de concevoir une application mobile permettant d'afficher, ce forme d'un tableau de bord, toutes les informations nécessaire pour les décideurs (la direction).

En effet, nous souhaitons simplifier la gestion de la MTEE, et pouvoir communiquer l'information de décision sous forme de graphe et de courbe.

Notre étude préalable consiste à étudier le fonctionnement du système existant pour établir un diagnostic précis de la situation, nous a conduits à proposer une solution qui vise à réaliser les objectifs suivants :

- **Tableau de bord** : qui donne une vision générale sur l'état du service ainsi que les différents statistiques sur les prestations, cotisations, virements (mensuelles, annuelles) ;
- **Rapidité d'accès à l'information** : la recherche d'une information devient plus facile et avec des critères désirés contrairement à la manière traditionnelle ;

2.9 conduite de projet

Tout au long de la réalisation de notre projet nous avons veillé à respecter le langage de modélisation UML et le processus unifié en suivant ces branches.

2.9.1 Présentation du langage UML et processus UP

2.9.1.1 Définition d'UML

UML est un ensemble de notations et de règles permettant de modéliser de manière claire et précise la structure ainsi que le comportement d'un système indépendamment de toute méthode ou de tout langage de programmation. Les créateurs de l'UML insistent tout particulièrement sur le fait qu'UML soit un langage de modélisation et non une méthode[4].

2.9.1.2 Diagramme de l'UML

UML2 repose sur une approche pédagogique de l'aspect normatif, cette dernière est une démarche d'élaboration des diagrammes couvrant l'analyse et la conception des systèmes d'information. UML2 propose sur treize diagrammes qui sont répartis en deux catégories, Chaque diagramme étant dédié à la présentation des concepts particuliers d'un système logiciel [5] :

– Diagramme structurel

- Diagramme de classes : identifie la structure des classes d'un système y compris les propriétés et les méthodes de chaque classe.
- Diagramme d'objets : montre les instances des éléments structurels et leurs liens à l'exécution.
- Diagramme de package : spécifie l'organisation logique du modèle et les relations entre packages.
- Diagramme de structure composite : désigne l'organisation interne d'un élément statique complexe.
- Diagramme de composants : indique des structures complexes avec leurs interfaces fournies et requises.
- Diagramme de déploiement : définit le déploiement physique des "artefacts" (objet) sur les ressources matérielles.

– Diagramme comportement

- Diagramme de cas d'utilisation (uses case diagram) : permet d'identifier les possibilités d'interactions entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
- Diagramme de vue d'ensemble des interactions : fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.
- Diagramme de séquence : montre la séquence verticale des messages passés entre objets au sein d'une interaction.
- Diagramme de communication : désigne la communication entre objets dans le plan au sein d'une interaction.
- Diagramme de temps : fusionne les diagrammes d'états et de séquences pour montrer l'évolution de l'état d'un objet au cours du temps.
- Diagramme d'activité : montre l'enchaînement des actions et décisions au sein d'une activité.
- Diagramme d'états-transition (state machine diagram) : permet de décrire sous forme de machines à l'état finis le comportement du système ou de ses composants.

2.9.1.3 Le processus unifié

Le processus unifié est un processus construit sur l'UML (unified modeling language), il représente un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. Ses activités de développements sont définies comme suit : [5]

- La capture des besoins ;
- L'analyse ;
- La conception ;
- l'implémentation ;
- test.

2.10 Spécification des besoins

Suivant le processus UP, la première étape de la spécification des besoins comporte les besoins fonctionnels et techniques.

2.10.1 Besoins fonctionnels

Les besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande, c'est à dire les sorties qui sont produit pour un ensemble de donnée d'entrées. Les services proposés par notre application se résument en deux actions majeures :

- **visualiser les statistiques** : L'application doit permettre d'offrir un tableau de bord qui permet de visualiser ou d'afficher les taux des statistiques suivantes :
 - afficher les statistiques des cotisations ;
 - afficher les statistiques des prestations ;
 - afficher les statistiques par adhérent ;
- Gérer simplement et efficacement les cotisations et les prestations ;

2.10.2 Besoins non fonctionnels

- **Authentification** :L'application doit permettre une authentification pour déterminer le rôle de l'utilisateur et la sécurité des données.
- **Fonctionnement** : L'application fonctionne sous réseau ;
- **Ergonomie** : cohérence de l'application et interfaces simples facile à manipuler par les utilisateurs ;
- **Sécurité et confidentialité** : l'accès aux informations est sécurisé par une authentification obligatoire.

2.11 Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil là où nous avons effectué notre stage et situation informatique de la MTEE, Nous sommes ensuite passés à la présentation du travail et la définition de la problématique, après nous avons proposée des solustion ainsi que la spécification des besoins du système. Ce recueil des besoins fonctionnels et non fonctionnel sera le point du départ pour le second chapitre qui fera l'objet d'analyse des besoins et conception.

CHAPITRE 3

ANAYSE DES BESOINS ET CONCEPTION

3.1 Introduction

Quel que soit le projet, l'analyse des besoins et la définition d'une politique de conception sont les étapes capitales qui précèdent la mise en place d'une application et d'une solution de gestion.

Dans cette partie nous allons analyser les informations recueillies lors de notre stage au sein de la MTEE (Mutuelle des travailleurs de l'environnement économiques) pour concevoir une application mobile pour le système Android, ou nous avons pu déterminer les caractéristique de la gestion du suivi des prestataires et cotisations, Pour atteindre les objectifs de notre projet il est essentiel que nous parvenions à une vue claire des différents besoins escomptés et de déterminer au moindre détail les fonctionnalités attendues et les différentes interactions entre les acteurs du système avant d'entamer la conception. Pour cela, nous nous sommes basés dans cette conception sur la méthode UP (processus unifié), cette dernière qui utilise UML comme langage de modélisation et qui offre une souplesse remarquable qui s'exprime par la possibilité d'obtenir des modèles de systèmes reflètent la réalité en utilisant des diagrammes.

3.2 Analyse Des Besoins

3.2.1 Identification des acteurs

3.2.1.1 Définition

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.[6]

3.2.1.2 Identifier les acteurs de notre système

- **l'utilisateur** : il peut seulement s'authentifier, représente toute personne qui n'est pas encore identifiée par le système, par conséquent il peut uniquement s'authentifier vis-à-vis du système.
- **Directeur** : c'est un utilisateur du système allant pour objectif de consulter les différentes statistiques.
- **Serveur MTEE** : une application serveur qui s'exécute sur machine serveur, et qui fournit des données à l'application mobile.

3.2.2 Identification des messages

Un message représente la spécification d'une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur. Un message est normalement associé à deux occurrences d'événements : un événement d'envoi et un événement de réception.[7]

D'après la phase de l'étude de l'existant nous avons présumé les messages échangés entre les acteurs et le système dans le tableau suivant :

Acteur	Msg envoyé acteur-Système	Msg envoyé Système-acteur
Utilisateur	M1 :demander d'accéder au système	formulaire d'authentification
Directeur	M1 M3 : Consulter	M2 M4 : Espace tableau de bord
Serveur MTEE	M6 :envoyer les données	M5 : envoyer la requete

TABLE 3.1 – représente les messages échangés entre acteur et système

3.2.3 Diagramme de contexte

Le diagramme de contexte a pour but de représenter les flux d'informations entre le système et les acteurs externes. Il permet d'isoler le système en le délimitant et ainsi définir les éléments externes avec lesquels il interagit[8]. C'est ainsi que lon modélise le contexte général du projet par le diagramme de contexte ci-dessous :

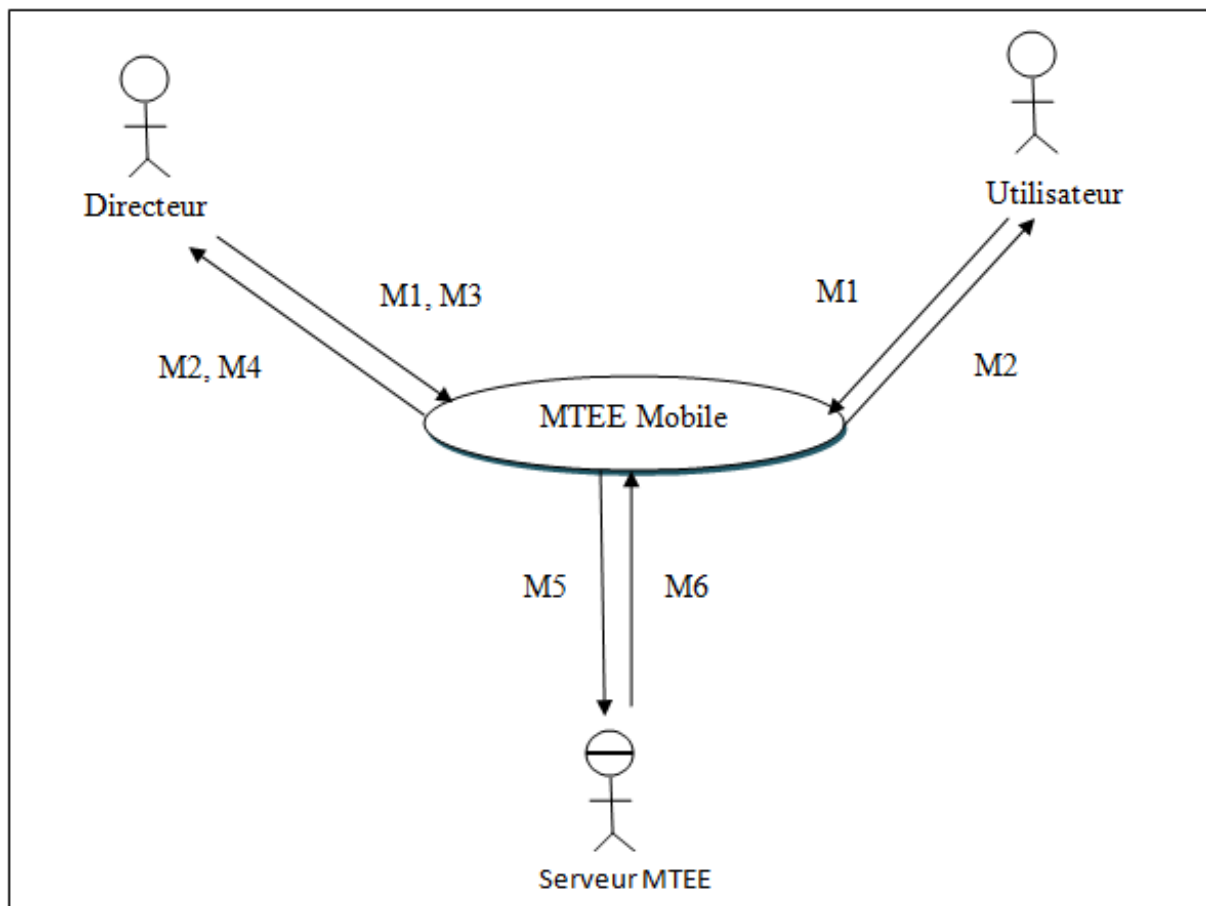


FIGURE 3.1 – Diagramme de contexte

3.2.4 Identifications des cas d'utilisation

3.2.4.1 Définition

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée " notable " à l'acteur concerné [6].

3.2.5 Diagramme de cas d'utilisation

3.2.5.1 «le diagramme de cas d'utilisation »

La figure 3.2 représente le diagramme de cas d'utilisation globale.

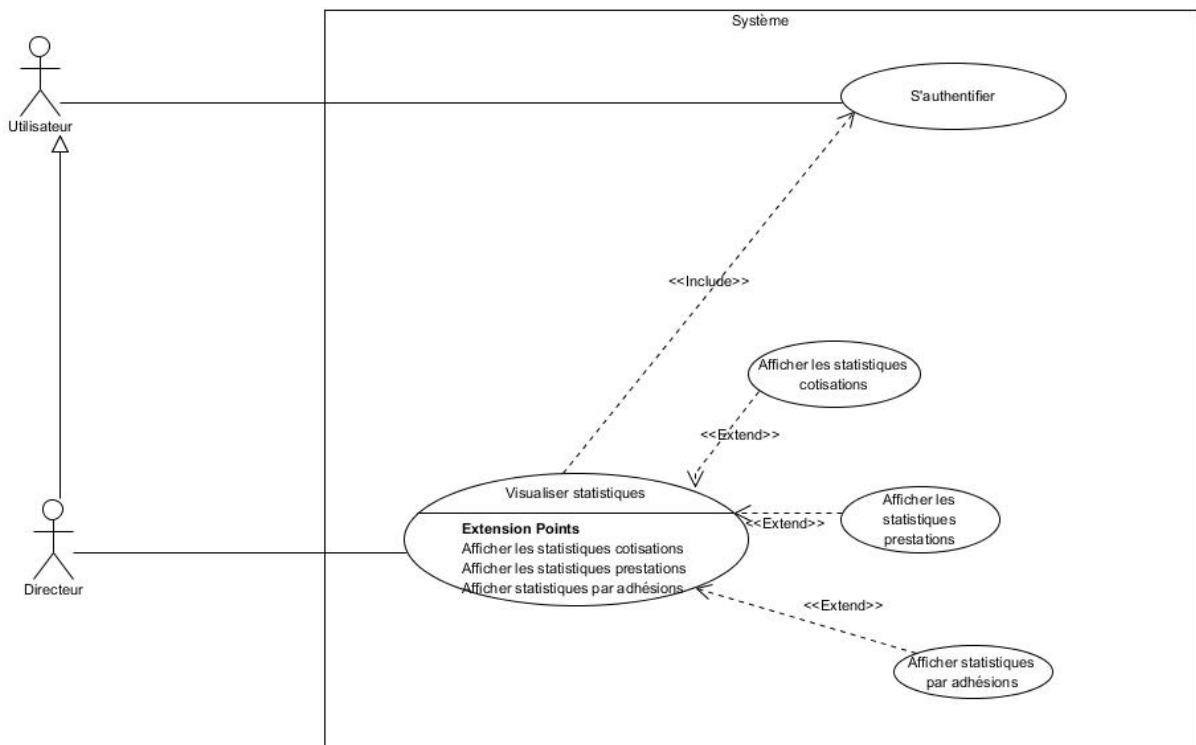


FIGURE 3.2 – Diagramme de cas d'utilisation globale

3.2.5.2 Description textuelle des cas d'utilisation :

– **Cas d'utilisation «S'authentifier» :**

Le tableau suivant représente la description textuelle du cas d'utilisation authentification.

Cas d'utilisation	S'authentifier.
Acteur	Utilisateur Directeur
Objectif	Accès de l'utilisateur aux fonctions qu'il lui sont réservés.
Scénario nominal	l'utilisateur accède à l'application. Le système affiche l'interface d'authentification. l'utilsateur saisit son identifiant et son mot de passe. Le système vérifie l'existence du compte. Le système donne l'accès à l'interface correspondante.
Alternative	Si les informations saisies sont fausses la systèma affiche un message d'erreur et re-affiche l'interface d'authentification.

TABLE 3.2 – Cas d'utilisation s'authentifier

– **Cas d'utilisation «Afficher les statistiques cotisations» :**

Le tableau suivant représente la description textuelle du cas d'utilisation afficher les statistiques cotisations.

Cas d'utilisation	Afficher les statistiques de cotisations.
Acteur	Directeur
Objectif	Accès de l'utilisateur aux fonctions qu'il lui sont réservés.
Scénario nominal	L'utilisateur accède à l'application. Le système affiche l'interface des statistiques cotisations. Le système donne l'accès à l'interface correspondante.
Alternative	/

TABLE 3.3 – Cas d'utilisation afficher statistiques cotisations

– **Cas d'utilisation «Afficher les statistiques prestations» :**

Le tableau suivant représente la description textuelle du cas d'utilisation afficher

les statistiques des prestations.

Cas d'utilisation	afficher les statistiques prestations.
Acteur	Directeur
Objectif	Accès de l'utilisateur aux fonctions qu'il lui sont réservés.
Scénario nominal	L'utilisateur accède à l'application. Le système affiche l'interface des statistiques prestations. Le système donne l'accès à l'interface correspondante.
Alternative	/

TABLE 3.4 – Cas d'utilisation afficher statistiques prestations

– **Cas d'utilisation «Afficher les statistiques par adhérent» :**

Le tableau suivant représente la description textuelle du cas d'utilisation afficher les statistiques par adhérent.

Cas d'utilisation	afficher les statistiques par adhérent.
Acteur	Directeur
Objectif	Accès de l'utilisateur aux fonctions qu'il lui sont réservés.
Scénario nominal	L'utilisateur accède à l'application. Le système affiche l'interface des statistiques par adhérent. Le système donne l'accès à l'interface correspondante.
Alternative	/

TABLE 3.5 – Cas d'utilisation afficher statistiques par adhérent

3.2.6 Diagrammes de séquence

Les cas d'utilisation décrivent les interactions des acteurs avec le système que nous voulons spécifier et concevoir. Lors de ces interactions, les acteurs produisent des messages qui affectent le système informatique et appellent généralement une réponse de celui-ci. Nous allons isoler ces messages et les représenter graphiquement sur des diagrammes de séquence UML.

Pour les messages propres à un cas d'utilisation, les diagrammes de séquence système montrent non seulement les acteurs externes qui interagissent directement avec le système, mais également ce système (en tant que boîte noire) et les événements système déclenchés par les acteurs. L'ordre chronologique se déroule vers le bas et l'ordre des messages doit suivre la séquence décrite dans le cas d'utilisation [6].

Nous allons représenter le diagramme de séquence d'un scénario représentatif de quelques cas d'utilisations décrits précédemment.

3.2.6.1 Diagramme de séquence «s'authentifier»

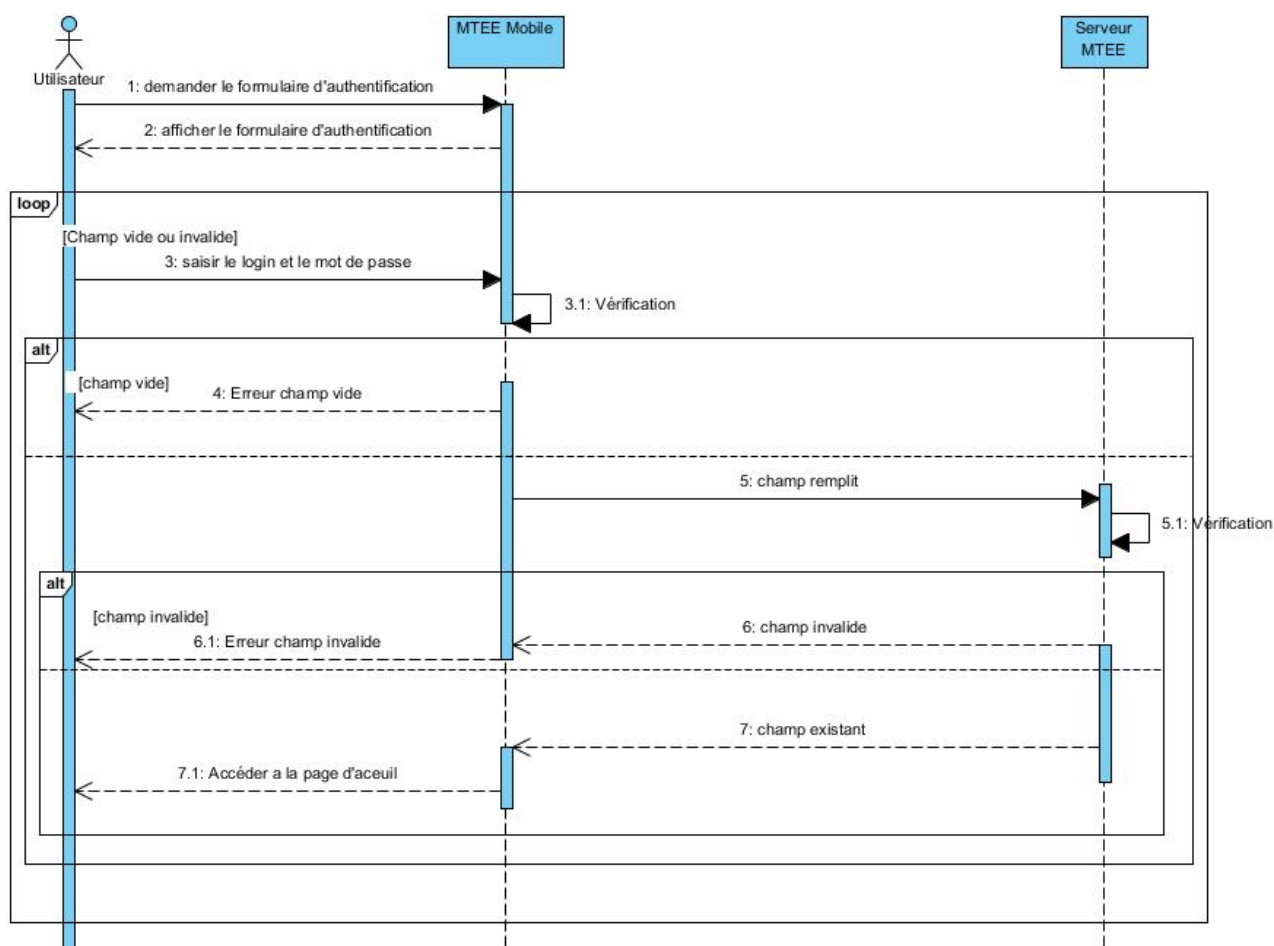


FIGURE 3.3 – Diagramme de séquence de cas d'utilisation «S'authentifier»

3.2.6.2 Diagramme de séquence «Afficher les statistiques de cotisations»

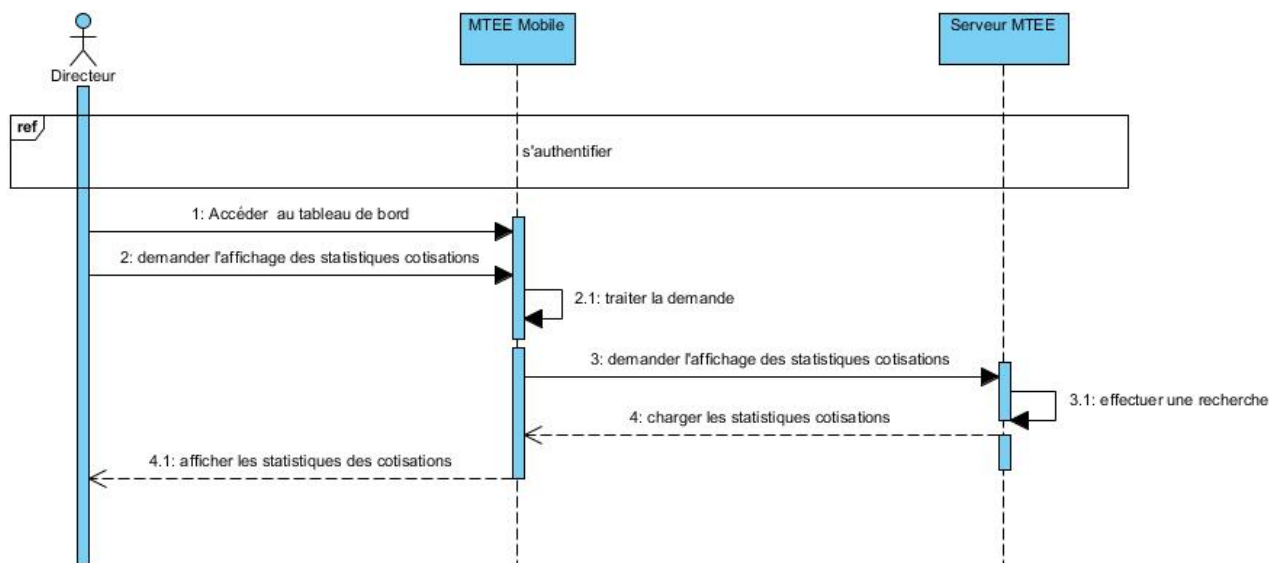


FIGURE 3.4 – Diagramme de séquence de cas d'utilisation «Afficher les statistiques cotisations »

3.2.6.3 Diagramme de séquence «Afficher les statistiques de prestations»

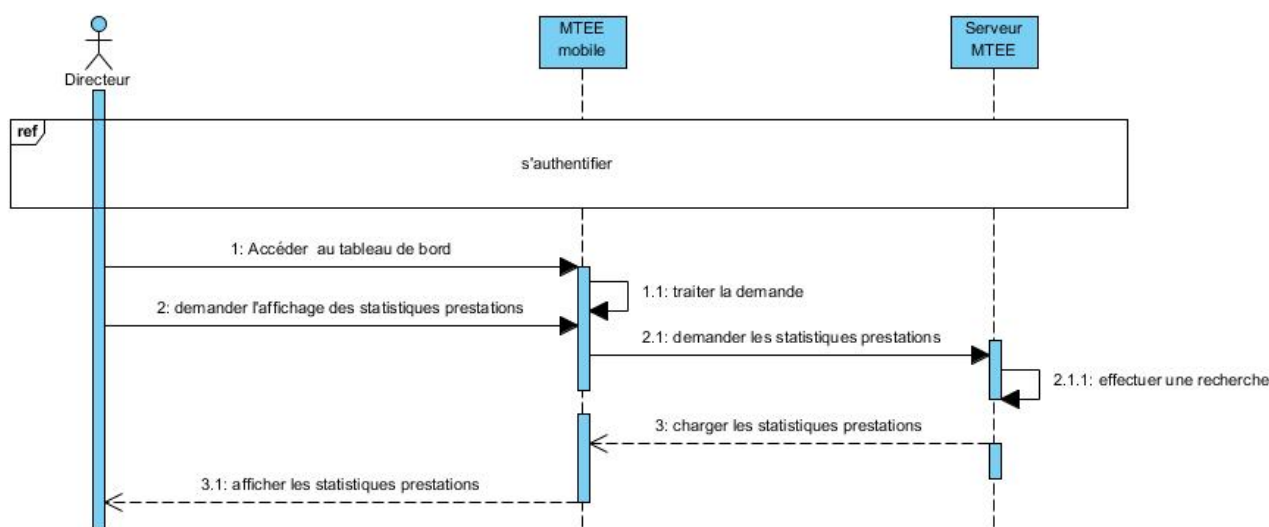


FIGURE 3.5 – Diagramme de séquence de cas d'utilisation «Afficher les statistiques prestations»

3.2.6.4 Diagramme de séquence «Afficher les statistiques par adhérent»

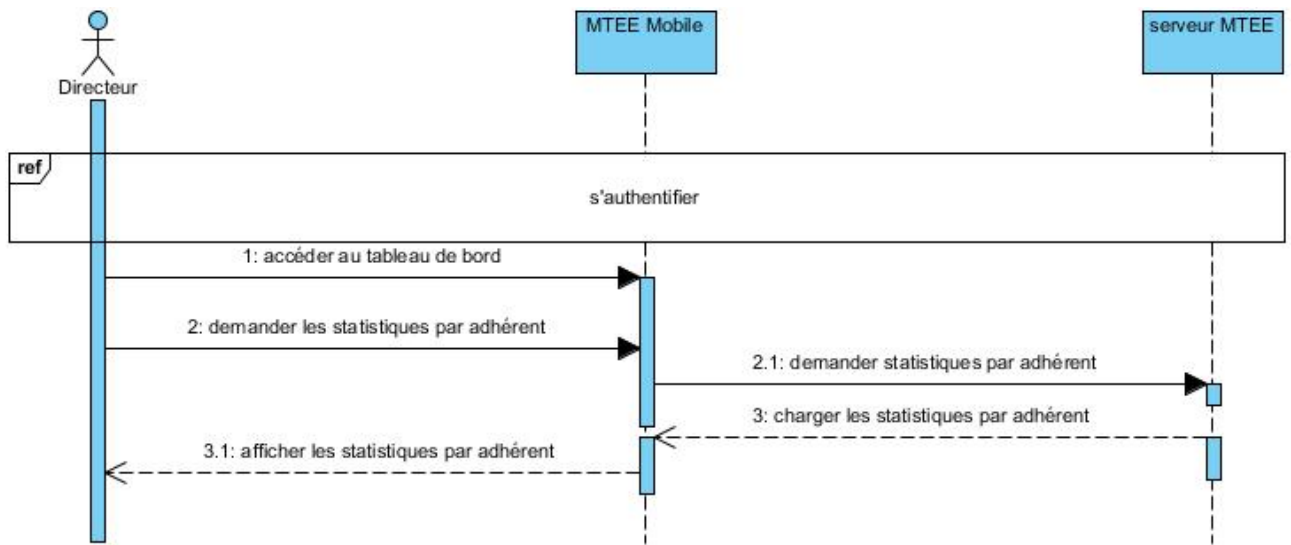


FIGURE 3.6 – Diagramme de séquence de cas d'utilisation «Afficher les statistiques par adhérent»

3.3 Conception

3.3.1 Diagrammes de séquence détaillé

pour produire un diagramme de séquence détaillé, il faut focaliser son attention sur un sous ensemble d'éléments du système et étudier leur façon d'interagir pour décrire un comportement particulier.

Dans cette partie, nous allons décrire les scénarios les plus important ainsi que leurs représentations par les diagrammes de séquence détaillés.

3.3.1.1 Diagrammes de séquence détaillé «s'authentifier»

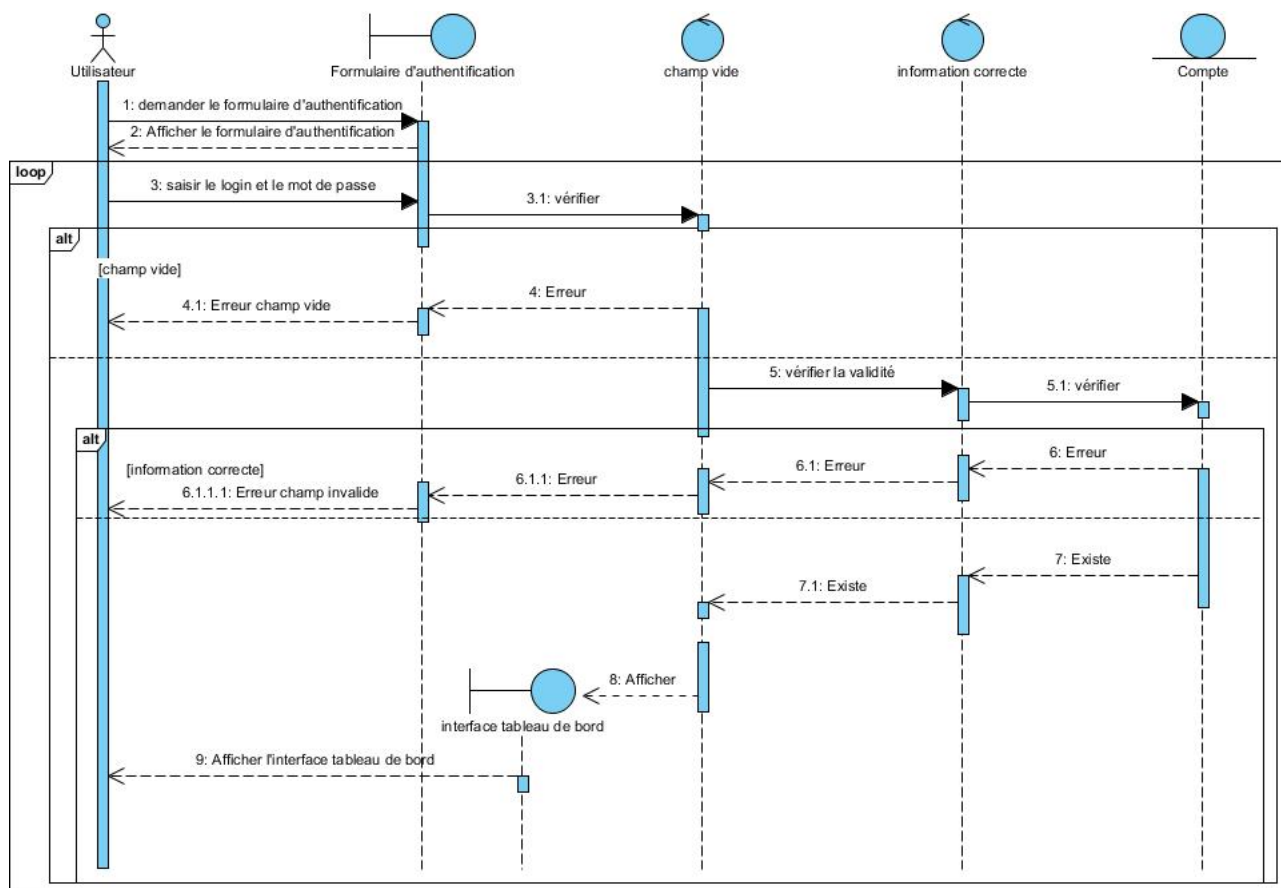


FIGURE 3.7 – Diagramme de séquence système détaillé «S'authentifier»

3.3.1.2 Diagrammes de séquence détaillé «Afficher les statistiques de cotisations »

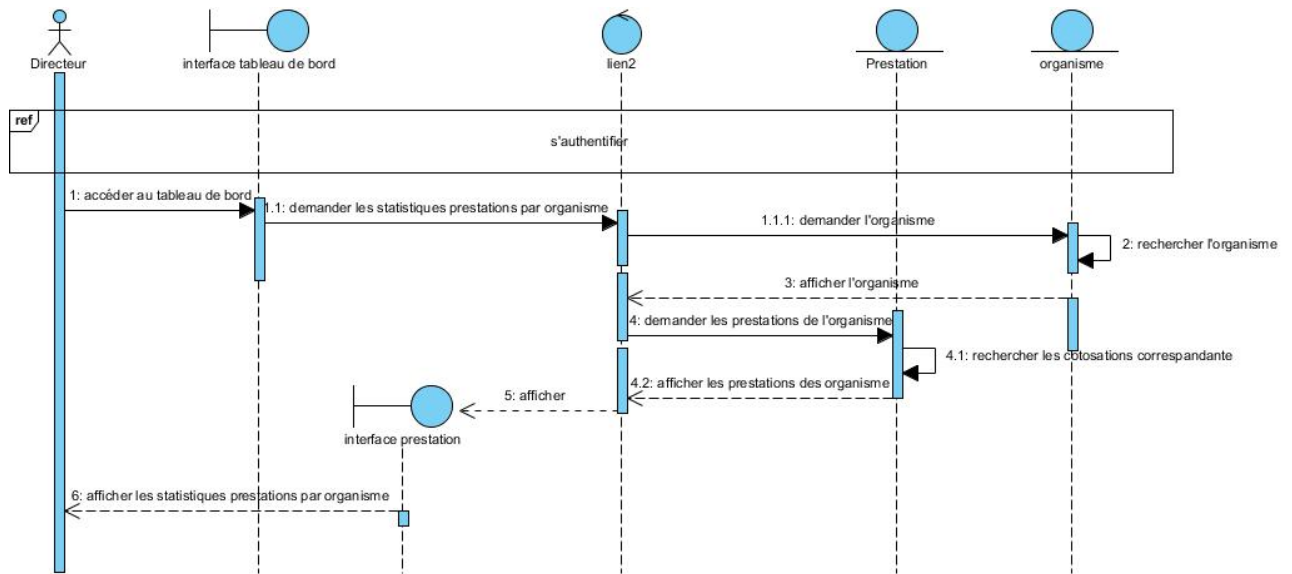


FIGURE 3.8 – Diagramme de séquence système détaillé «Afficher les cotisations »

3.3.1.3 Diagrammes de séquence détaillé «Afficher les statistiques de prestations »

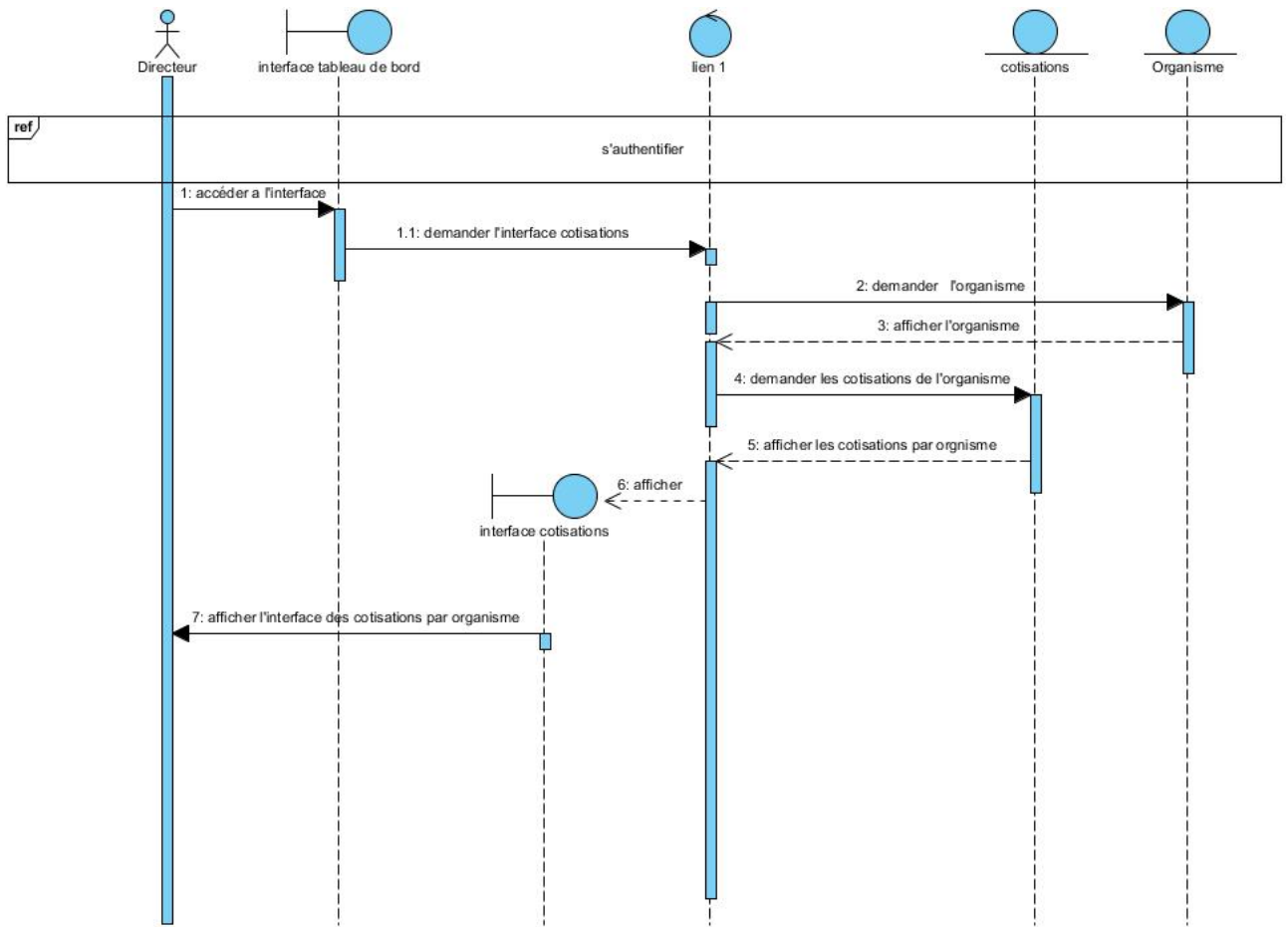


FIGURE 3.9 – Diagramme de séquence système détaillé «Afficher les prestations »

3.3.1.4 Diagrammes de séquence détaillé «Afficher les statistiques par adhérent »

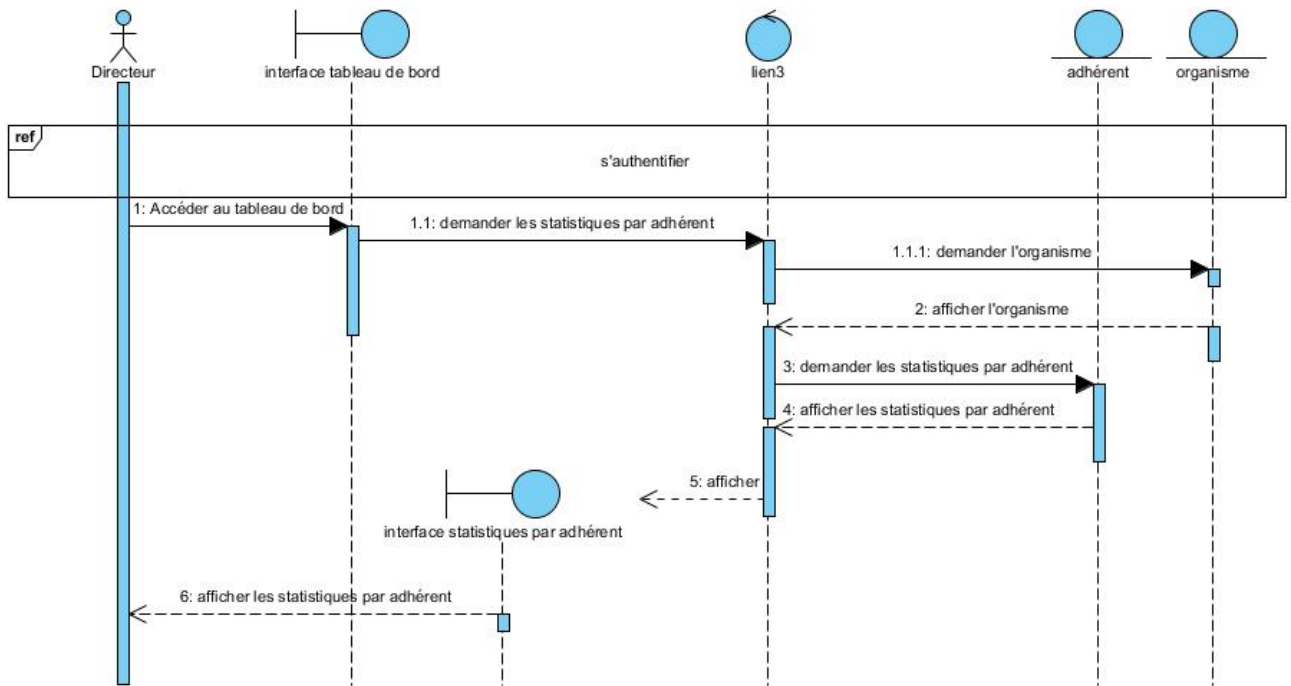


FIGURE 3.10 – Diagramme de séquence système détaillé «Afficher les statistiques par adhérent »

3.3.2 Diagramme de classes

Après l'étude détaillée des cas d'utilisations et des diagrammes de séquences et les diagrammes de séquence détaillés, nous avons déduit le diagramme de classe globale de notre système. Ce diagramme est considéré comme la phase finale de la conception théorique de ce dernier et sera pris comme la référence à partir de laquelle va se dérouler l'écriture du code source de notre application.

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateur.

Ces entités sont extraites, en phase d'analyse du domaine, en appliquant les règles de gestion ainsi que les diagrammes de séquence préalablement établis. Chaque objet de classe entité devient va apparaitre dans le diagramme de classes final [7].

3.3.3 Les éléments de base d'un diagramme de classe

Dans le langage UML, Un diagramme de classe est constitué des éléments suivants : [7]

- **Classe** : est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes.
- **Objet** : est une instance d'une classe. C'est une entité discrète dotée d'une identité, d'un état et d'un comportement que l'on peut invoquer. Les objets sont des éléments individuels d'un système en cours d'exécution.
- **Attribut** : Les attributs définissent des informations qu'une classe ou un objet doivent connaître. Ils représentent les données encapsulées dans les objets de cette classe. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisée. Le nom de l'attribut doit être unique dans la classe.
- **Operation** : Une opération est une fonction applicable aux objets d'une classe et permet de décrire le comportement de ces objets.
- **Classe d'association** : Permet de décrire soit des attributs, soit des opérations propres à l'association.

- **Relation** : Il existe plusieurs types de relation parmi elles :
- association : Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu'il peut y avoir des liens entre des instances des classes associées.
- La composition : Est une relation d'agrégation dans le quelle existe une contrainte de la durée de vie, entre la classe " composant " et la (ou les) classe(s).
- héritage : La généralisation décrit une relation entre une classe générale (classe de base ou classe parent) et une classe spécialisée (sous-classe). La classe spécialisée est intégralement cohérente avec la classe de base, mais comporte des informations supplémentaires (attributs, opérations, associations). Un objet de la classe spécialisée peut être utilisé partout où un objet de la classe de base est autorisé.
- **Multiplicité ou cardinalité** : La multiplicité associée à une terminaison d'association, d'agrégation ou de composition déclare le nombre d'objets susceptibles d'occuper la position définie par la terminaison d'association.

La figure 3.5 représente le diagramme de classe global de notre système.

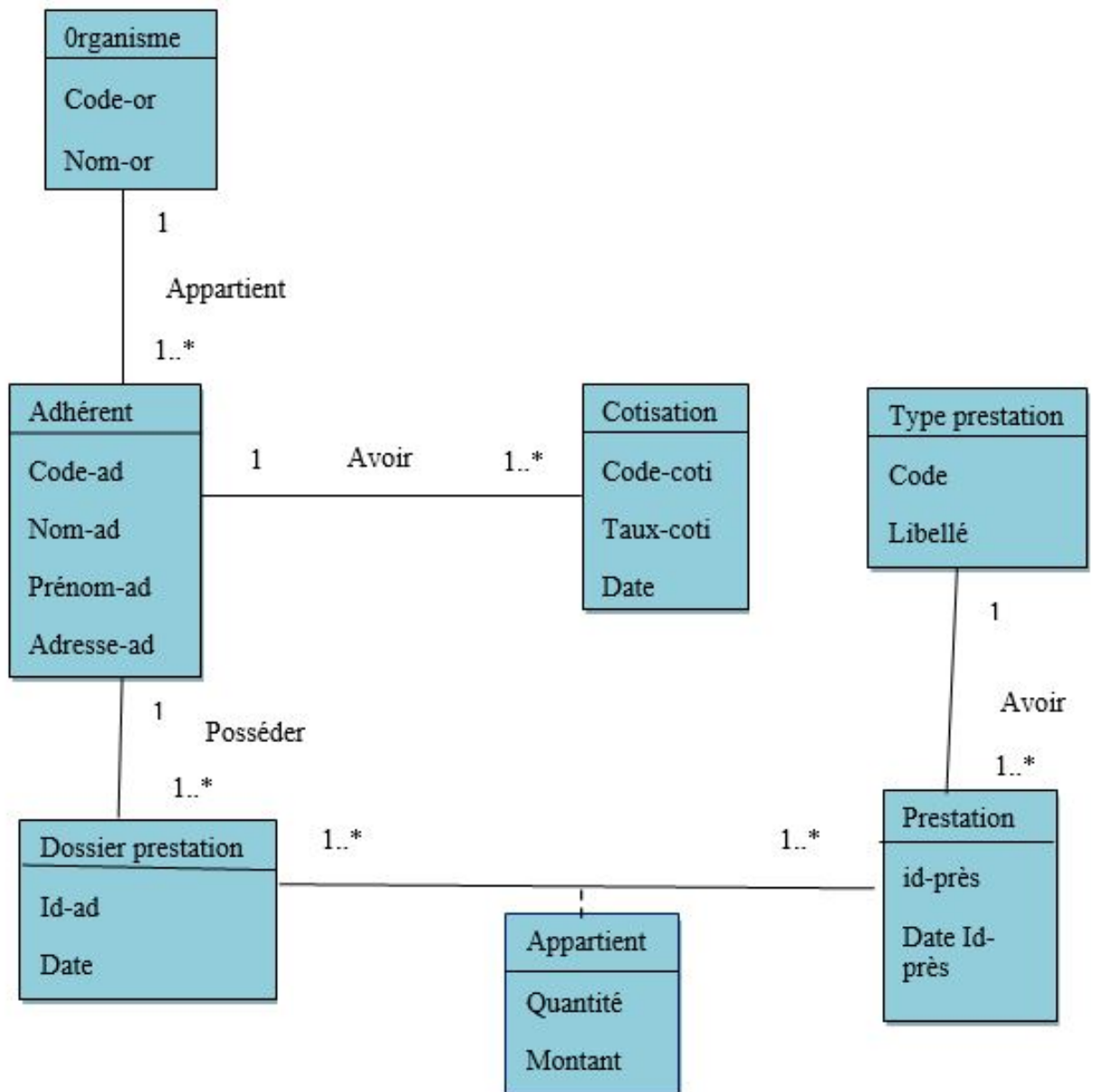


FIGURE 3.11 – Diagrammes de classes

3.4 Conclusion

Dans ce chapitre nous avons abordé en détail l'étape d'analyse et conception, tout en passant par UML et ses différents diagrammes : le diagramme de contexte, le diagramme de cas d'utilisation ainsi que les différents diagrammes de séquences. Enfin nous avons basculé vers les diagrammes de séquence détaillés et le diagramme de classe.

Le choix des outils et les langages de développements, des composants techniques, graphiques, et les interfaces des différentes fonctionnalités de l'application seront développés en détail dans le chapitre suivant.

CHAPITRE 4

RÉALISATION

4.1 Introduction

Suivant immédiatement l'étape de la conception, la réalisation représente la toute dernière étape de notre projet et la concrétisation du travail fait. Dans ce chapitre nous commençons par une partie consacrée aux applications mobiles puis nous présentons les outils, langages, et environnement de développement dont on a eu besoin pour la réalisation. Ensuite nous expliquerons le fonctionnement global de l'application suivi de l'exposition des captures d'écran des principales interfaces.

4.2 Applications mobiles

4.2.1 Définition des applications mobiles

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation d'un smartphone ou d'une tablette. Les applications mobiles sont adaptées aux différents environnements techniques des smartphones et à leurs contraintes et possibilités ergonomiques (écran tactile notamment). Elles permettent généralement un accès plus confortable et plus efficace à des sites ou services accessibles par ailleurs en versions mobile ou web. L'essentiel du temps passé sur les smartphones est

consacré à des applications mobiles [9].

4.2.2 Types d'applications mobiles

- **Une application Web** : est une application qui utilise les technologies Web (HTTP, HTML, XML, JSP, Web services, ..) et à laquelle on accède en utilisant un navigateur (Firefox, Google Chrome, ...), telle que les applications des réseaux sociaux [10].
- **Une application client/serveur (hybrides)** : est une application qui est installée sur un ou plusieurs appareils et qui se connecte à un(ou plusieurs) serveur(s). Elle repose à la fois sur :
 - Un serveur PHP alimenté par une base de données SQL ;
 - Un système de stockage interne de l'appareil mobile ;
 - L'application en elle-même [10].

4.3 Implémentation de l'application

4.3.1 Outils et langages de développement

4.3.1.1 Java

La plateforme Java EE s'appuie entièrement sur le langage Java. Java EE est donc une norme, qui permet à des développeurs, entreprises de développer leur propre application qui implémente en totalité ou partiellement de SUN. EN simplifiant, il est possible de représenter Java EE comme un ensemble de spécifications d'API, une architecture, une méthode de packaging et de déploiement d'application et la gestion d'applications déployées sur un serveur compatible Java.

- La machine virtuelle Java(JVM) : permet d'exécuter des applications Java. Elle constitue une passerelle et permet une portabilité entre les architectures (Windows, Linux, Mac...).
- La bibliothèque de classes Java : un ensemble de composants logiciels prêt à l'emploi.
- Les outils de développement : le compilateur javac, un interpréteur java nommé java, le generateur de documentation javadoc, la console de supervision Jcon-

sole...La plate forme Java EE est une extension de plateforme J2SE.Elle permet un développement d'application qui vont s'exécuter sur un serveur d'applications. Les seront utilisées par des clients léger (comme des navigateur Web) ou bien des applications lourdes (IHM). La dernière version stable de Java EE est la version Java EE 5.0 et fonctionne avec le JDK 5.0 et 6.0 [11].

4.3.1.2 XML

XML est l'acronyme de "Extensible Markup Language", expression signifiant "langage de balisage extensible". Il s'agit d'un langage de création de fichiers de données ayant la particularité de permettre la création et la définition de balises à la demande (d'où l'origine de l'expression "extensible"). XML a été conçu pour permettre les transferts d'informations d'une plateforme à une autre (deux applications non interfacées, par exemple). Il évite ainsi les fastidieux et coûteux développements d'interfaces [12].

En Android, grâce à ce langage, nous décrivons les interfaces dans un format spécial, et Android le converti automatiquement en objets Java qui seront par la suite disponibles comme tout autre objet dans notre code. Il offre ainsi plus de souplesse de développement, facilite les modifications du code et assure la séparation entre la présentation et le comportement des objets [12].

4.3.1.3 IDE Eclipse

L'éclipse IDE (Integrated Développement Environnement) est un environnement de développement libre permettant de créer des programmes dans de nombreux langages de programmation tels que Java, C++, PHP, etc.Il possède plusieurs caractéristiques dont :

- Les systèmes basés sur Eclipse sont enrichis grâce au nombre impressionnant de plugins qui existent ;
- Portabilité : Eclipse fournit un support à un grand nombre hétérogène d'environnements clients, du PC traditionnels à des supports plus légers tels que les tablettes, ou des plateformes mobiles comme les smartphones et les PDA. A partir du moment où une JVM peut être installée, Eclipse est utilisé [13].

4.3.1.4 Le plugin ADT (Android Développement Tools)

Pour développer une application Android, nous avons à installer le plugin Android qui rajoutera à Eclipse les fonctionnalités spécialisées dans le développement sous Android.

4.3.1.5 SDK Android)

Android permet aux développeurs de maximiser les compétences des appareils mobiles et d'accéder à des applications déjà existantes ou même d'en créer de plus riches. Android est basé sur le noyau Linux qui utilise une machine virtuelle personnalisée conçue pour optimiser la mémoire et les ressources matérielles d'un environnement mobile [14].

4.3.2 Bibliothèques utilisées

4.3.2.1 JSON

JSON est une bibliothèque Java qui peut être utilisée pour convertir des objets Java dans leur représentation JSON. Il peut également être utilisé pour convertir une chaîne JSON à un objet Java équivalent. JSON est un projet open-source hébergé .

4.3.2.2 JDBC

La technologie JDBC (Java DataBase Connectivity) est un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données (SGBD).

L'API JDBC permet de faciliter l'obtention de connexions JDBC vers des sources de données. L'API fournit les bibliothèques pour se connecter aux bases de données et pour la gestion des transactions [11].

4.4 Architecteur de notre application

4.4.1 Diagramme déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nud, le diagramme de déploiement précise comment les composants sont répartis sur les nuds et quelles sont les connexions entre les composants ou les nuds. Les diagrammes de déploiement existent sous deux formes : spécification et instance [15].

La figure suivante présente les différentes composantes de l'application ainsi que leurs configurations géographiques. Notre application est constituée de trois composantes, à savoir : la couche présentation qui est représentée par les pages mobile, la couche métiers qui jouent le rôle de contrôleurs et la couche données qui représenté par le serveur MTEE.

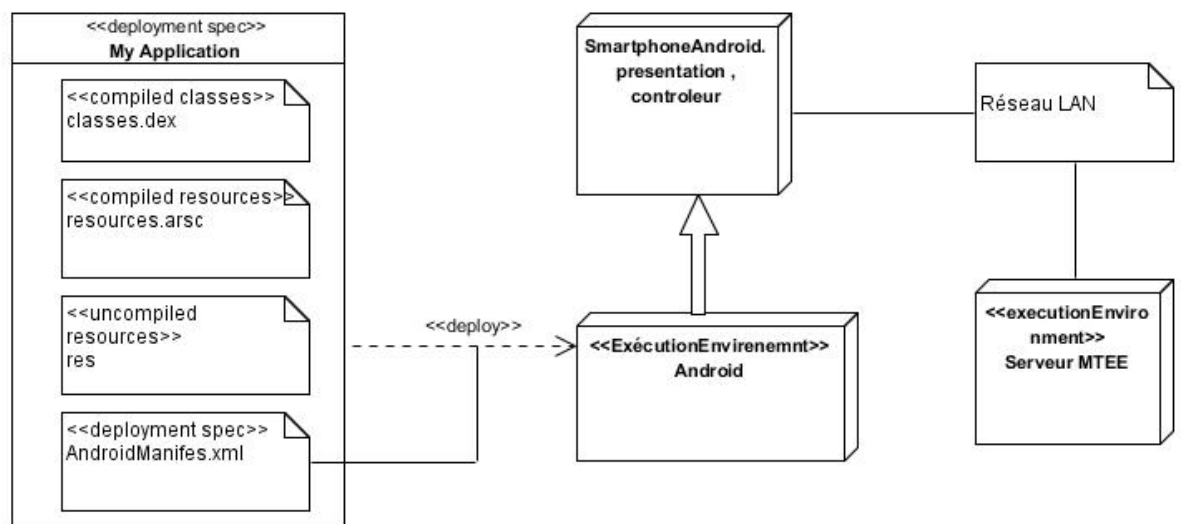


FIGURE 4.1 – Diagramme Déploiement

4.4.2 Architecture globale du Système

Après avoir réalisé l'extension de l'application de gestion de la mutuelle MTEE, nous aurons un système fonctionnant selon l'architecture globale suivante.

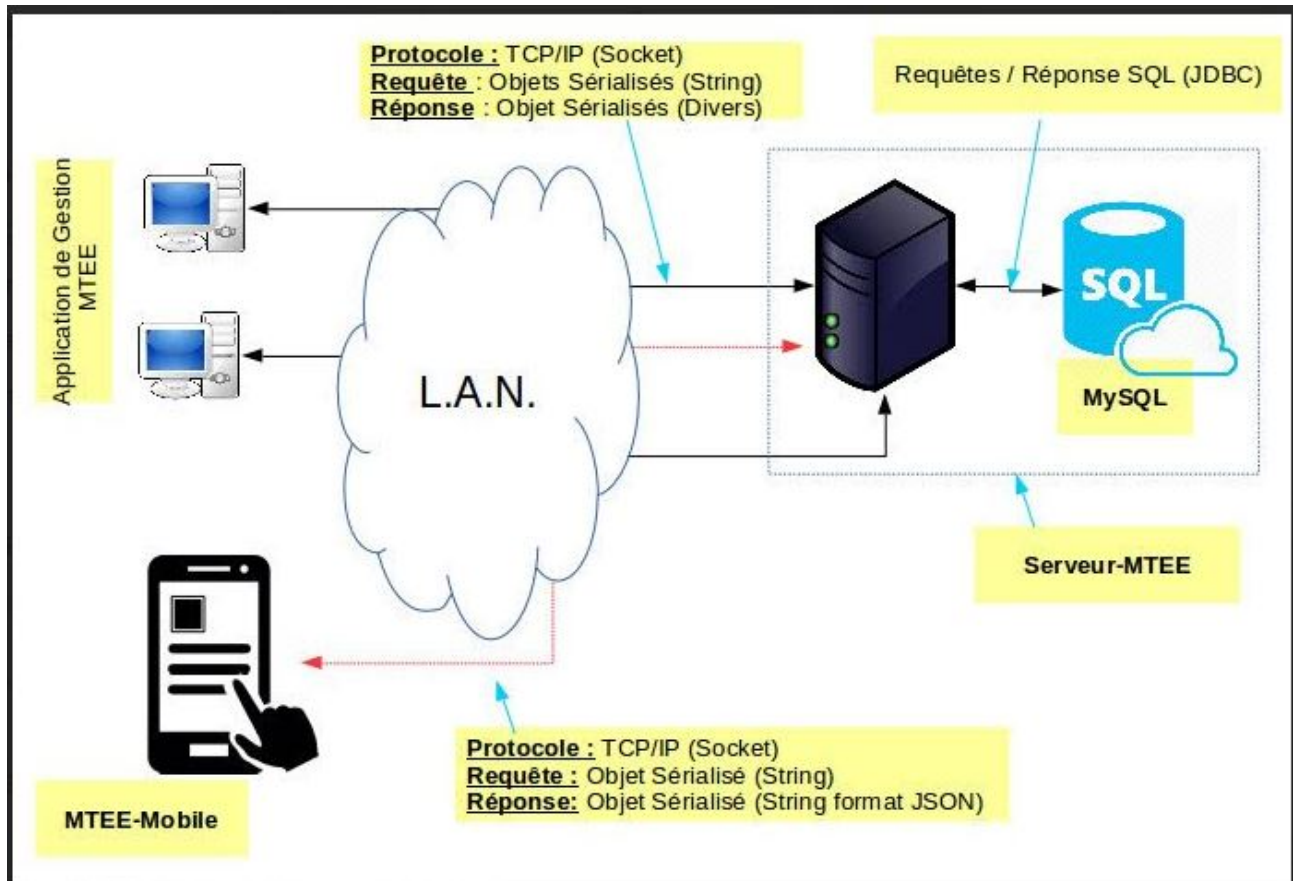


FIGURE 4.2 – Architecture globale du Système

D'après la figure ci-dessus, nous pouvons considérer l'application que nous avons réalisé comme une application basée sur une architecture client-serveur 2-tiers (Client est le smartphone et le Serveur est MTEE-Server). Cependant, si on décortique le serveur MTEE-Server, nous auront, d'une façon globale, une architecture 3-tiers, dans laquelle nous aurons deux serveurs, à savoir, le serveur d'application et le serveur de données MySQL.

4.5 Conclusion

Dans ce chapitre, nous avons présenté les outils et l'environnement sur lesquels nous nous sommes basés pour implémenter au mieux notre application. Nous avons par la suite, présenté les interfaces les plus significatives de notre application.

La mobilité est un élément très important dans les systèmes d'informations, elle permet de réaliser la saisie des données, des consultations et divers tache sans être obligé d'être physiquement à un endroit donné.

Le système Android représente un système d'exploitation mobile, libre et open source, permettent de construire des applications mobiles dans de multiples domaines. D'après les recommandations de la MTEE, qui consistent à favoriser les logiciels et solutions libres, nous avons opté pour les systèmes Android à fin de développer une application mobile qui permet de simplifier la gestion de la MTEE, ainsi d'améliorer et faciliter considérablement le travail pour les décideurs.

Pour réaliser cette application nous nous sommes basée sur le processus UP (Unified process) ainsi que sur le formalisme d'UML (Unified Modeling Language) dont nous avons utilisé quatre différents diagrammes : le diagramme de cas d'utilisation, le diagramme de séquence, diagramme de séquence détaillé et le diagramme de classe. Après l'analyse et la conception, nous avons utilisé pour le développement de l'application le langage Java, vu ses avantages dans la portabilité et la mise sous-réseau. Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'exploiter et améliorer nos compétences dans le domaine de la conception et la programmation.

Notre travail comme, comme tout travail humain, possède des insuffisances, pour cela nous avons dressé quelques perspectives :

- Enrichir l'application avec d'autres types des statistiques ;
- Ajouter d'autres acteurs comme (comptable, gestionnaire des prestations).

BIBLIOGRAPHIE

- [1] Sayed Y. Hachimi and Satya Komatineni,(Appress.Pro.Android), Juin.2009
- [2] L.Jordan. et P.Gryling, (Practical Android Projects), Books for professionals, Edition Appress 22 février 2011.
- [3] R. Meier, (Profesional Android Application Developement), Wiley Publishing 2009.
- [4] P.Muller.(éd.)(2000). Modélisation objet avec uml(2ème édition). Eyrolles.
- [5] D.Gabay.et J.Gubay.(éd.)(2008). UML2 Analyse et Conception.Paris : Dunod
- [6] Pascal Roques.(éd.)(2008).UML2 Modéliser une application web(4émé édition).Eyrolles.
- [7] F.Vallée.et P.Roques.(éd.)(2007).UML 2 en action de l’analyse des besoins à la conception(4ème édition).Eyrolles.
- [8] C.Belleil. Le langage UML 2.0 Diagramme de Contexte. Nante :Université de Nantes.
- [9] <http://www.definitions-marketing.com/definition/application-mobile/>, date de consultation 28/05/2016.
- [10] <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-mulx-glossaire-informatique-et-multimedia/528-definition.html>, date de consultation : 30/05/2016.
- [11] Jérôme LAFOSSE, Java EE Guide de développement d’application web en java, ENI Editions, février 2009.

- [12] <http://www.iphonedevoloppeur.fr/2011/04/fonctionnement-dune-application-iphone>, date de consultation 28/05/2016.
- [13] S.Darlot.(2014).pourquoi utiliser éclipse rcp pour développer une application.
- [14] http://www.openhandsetalliance.com/android_overview.html , date de consultation : 30/05/2016.
- [15] <http://laurent-audibert.developpez.com/Cours-UML/?page=diagrammes-composants-deploiement> , date de consultation : 30/05/2016.

Résumé

Aujourd'hui, l'informatique mobile a atteint une prodigieuse évolution technologique dans différents domaines . Cette évolution est nécessaire pour remédier aux problèmes rencontrés dans la vie courante. Dans ce contexte, notre travail consiste à concevoir et réaliser une application mobile pour le système Android permettant de présenter un tableau de bord pour la mutuelle de la MTEE. A vrai dire notre projet consiste à faire une extension à une application lourde existante dans l'établissement d'accueil.

Mots Clés

Application mobile, Android, UP, UML, XML, JSON, socket, Java.

Abstract

Today, mobile computing has reached a prodigious technological developments in different areas. This change is necessary to remedy the problems encountered in daily life. In this context, our work is to design and develop a mobile application for the Android system to provide a dashboard for the mutual MTEE . Actually our project consists of an extension to an application in the heavy existante Home Etablissement .

Keywords

Mobile application, Android, UP, UML, XML, JSON, socket, Java.