

Liste des Figures

Figure 1 :Architecture de système d'aide à l'atterrissage aux instruments ILS	1
Figure 2:Réseau RCMS	4
Figure 3:Architecture du processus 2TUP.....	12
Figure 4:Diagramme de cas d'utilisation globale	17
Figure 5:Diagramme d'authentification	18
Figure 6:gestion des solutions des pannes techniques	19
Figure 7:Diagramme de Gestion des solutions des pannes techniques	20
Figure 8:Diagramme des tâches de la maintenance préventive	21
Figure 9:les notifications des tâches de la maintenance préventive	23
Figure 10:les notifications des tâches de la maintenance corrective	24
Figure 11:Diagramme de déploiement	25
Figure 12:Modèle MVC	27
Figure 13:Architecture trois-tiers de l'application	28
Figure 14:évidence l'architecture général de l'application et les dépendances entre les divers composants	29
Figure 15:Diagramme de classe en couche de l'application	30
Figure 16:Diagramme de classe de la couche présentation	31
Figure 17:Diagramme de classe de la couche logique applicative	32
Figure 18:Diagramme de classe de la couche modèle	33
Figure 19: Description matérielle de l'application	39
Figure 20:Interfaces de démarrage et d'authentification	40
Figure 21:Interfaces module administrateur	40
Figure 22:Interfaces d'ajout des pièces	41
Figure 23:Interfaces d'ajout des pannes	41
Figure 24:Interfaces d'indexer les solutions.....	42

Figure 25:Interface de consultation des pièces de rechange.....	43
Figure 26:Interfaces de consultation des pannes et leurs solutions	44



Liste des tableaux

Tableau 1: Relation entre UML et 2TUP	Erreur ! Signet non défini.	3
Tableau 2: Liste des cas d'utilisation	Erreur ! Signet non défini.	16
Tableau 3: Patron textuel du cas d'utilisation Indexer des solutions des pannes	Erreur ! Signet non défini.	19
Tableau 4 :Patron textuel de Consultation des notifications des tâches de la maintenance préventive	Erreur ! Signet non défini.	
Tableau 5:environnement materiels	Erreur ! Signet non défini.	33
Tableau 6 : Android Studio VS Eclipse.....		35



Liste des Acronymes

UML: Unified Modeling Language

MVC: Model-View-Controller

JSON: Java Script Object Notation

DAO: Data Access Object

GMAO: Gestion de Maintenance Assistée par Ordinateur

AITA : Aéroport International Tabarka Ain Drahem

DOM : Document Object Model

OACA : Office de l'Aviation Civile et des Aéroports

OACI : Organisation de l'Aviation Civile Internationale

ILS: Instrument Landing System

LLZ: Localizer

GP: Glide Path

VOR: VHF Omni Range

DME: Distance Measuring Equipment

RCMS: Remote Control Monitoring System

SDK : Software development kit

IDE : integrated development environment

ADT : Android Development Tools

Sommaire

INTRODUCTION GÉNÉRALE.....	1
CHAPITRE 1 : CADRE GÉNÉRAL.....	3
Introduction :	3
I PRÉSENTATION DE L'ENTREPRISE D'ACCUEIL.....	3
I.1 Présentation de l'Office de l'aviation civile et des aéroports (OACA).....	3
I.2 Etude de l'existant et Problématique.....	4
I.2.1 Unité de télécommande RCMS (Remote Control Monitoring System):.....	4
I.2.2 Absence d'un GMAO dans le service (Gestion de maintenance assistée par ordinateur).....	5
I.3 Solution proposée.....	5
I.4 Travail demandé.....	6
Conclusion :	6
CHAPITRE2 : ÉTUDE PRÉLIMINAIRE.....	7
INTRODUCTION :	7
I CONTEXTE DE PROJET:	7
I.1 Objectif (Aspect de la solution adoptée).....	7
I.2 Solution de mobilité.....	8
I.3 Choix de l'environnement de développement :	8
II Présentation de l'application :	9
III Spécification des besoins :	10
III.1 Besoins fonctionnels.....	10
III.2 Besoins non fonctionnels.....	10
IV processus de développement et méthodologie de conception:	11
IV.1 Le processus de développement unifié 2TUP :	11
IV.2 La méthodologie de conception UML :	13
CONCLUSION.....	14
CHAPITRE 3 : EUDE FONCTIONNELLE.....	15
INTRODUCTION :	15
I CAPTURE DES BESOINS FONCTIONNELS :	15
I.1 IDENTIFICATION DES CAS D'UTILISATION :	15
I.1.1 Présentation des cas d'utilisation :	17
I.1.2 DESCRIPTION DES CAS D'UTILISATION :	18
I.1.3 Le cas d'authentification	18
I.2 Le module d'administration :	18
I.2.1 Gestions des pannes techniques	18
I.2.3 Diagramme de séquence des solutions des pannes techniques	20
I.3 Module de la maintenance préventive :	21
I.3.1 Affinement du cas d'utilisation « consulter les notifications des tâches de la maintenance préventive » :	21
I.3.2 Diagramme de séquence des notifications des tâches de la maintenance préventive :	23
I.4. Diagramme de séquence de consultation des notifications des tâches de la maintenance corrective :	24
CONCLUSION.....	24
CHAPITRE 4 : ETUDE TECHNIQUE	25
INTRODUCTION	25
I Spécification d'architecture :	25
II ARCHITECTURE DE L'APPLICATION	26
II.1 Présentation du modèle MVC :	26
II.2. Présentation de l'architecture trois tiers de l'application :	27
III Diagramme de composant général :	29
CONCLUSION.....	29

CHAPITRE 5 : LA CONCEPTION	30
INTRODUCTION	30
I Les diagrammes de classe:.....	30
I.1 Le diagramme de classe en couche :	30
I.2 Le diagramme de classe de la couche présentation :	31
I.3 Le diagramme de classe de la couche logique applicative :	32
I.4 Le diagramme de classe de la couche modèle.....	33
CONCLUSION :	33
CHAPITRE 6: IMPLÉMENTATION	34
INTRODUCTION :	34
I. ASPECT TECHNIQUE	34
I.1. Choix de la plateforme.....	34
I.1.1. La plate-forme Androïde.....	34
II. ENVIRONNEMENT DU TRAVAIL :	35
II.1 Environnement matériel :	35
II.2 Environnement logiciel :	36
II.2.1 Outils d'infographie.....	36
II.2.2 Outils de conception	36
I.2.3 Outils de développement.....	36
I.2.4 Configuration pré-développement :	38
II Description matérielle :	39
II.1 Implémentation.....	40
I.1.2 Interfaces du module administrateur	40
I.1.3 Interfaces du module maintenance corrective :	43
CONCLUSION :	44
CONCLUSION GÉNÉRALE	45
BIBLIOGRAPHIE	46
WEBOGRAPHIE.....	47

Introduction Générale

La navigation aérienne est l'ensemble des techniques permettant à un pilote d'aéronef de maîtriser ses déplacements, elle lui permet de suivre une trajectoire appelée route aérienne.

L'ILS (Instrument Landing System) est un système d'aide à l'atterrissage aux instruments par visibilité réduite qui délivre au pilote des informations :

- Continues de position par rapport à l'axe de piste par un système appelé **Localizer**
- Continues de position par rapport à un plan oblique de descente aboutissant à la piste par un système appelé **Glide Path**
- discontinues de distance par rapport au seuil de piste par des **Markers**
- continues de distance par rapport au seuil de piste si l'ILS est couplé avec un système de mesure de distance **DME**

Ces équipements jouent un rôle indispensable dans le guidage de l'avion en route ainsi qu'en phase d'atterrissage d'où l'importance de leurs maintenance et dans ce cadre se situe notre projet à travers lequel nous avons essayé de proposer toute une stratégie de travail assurant la supervision à distance par une technologie mobile le déroulement de la maintenance des équipements des aides à la navigation aérienne.

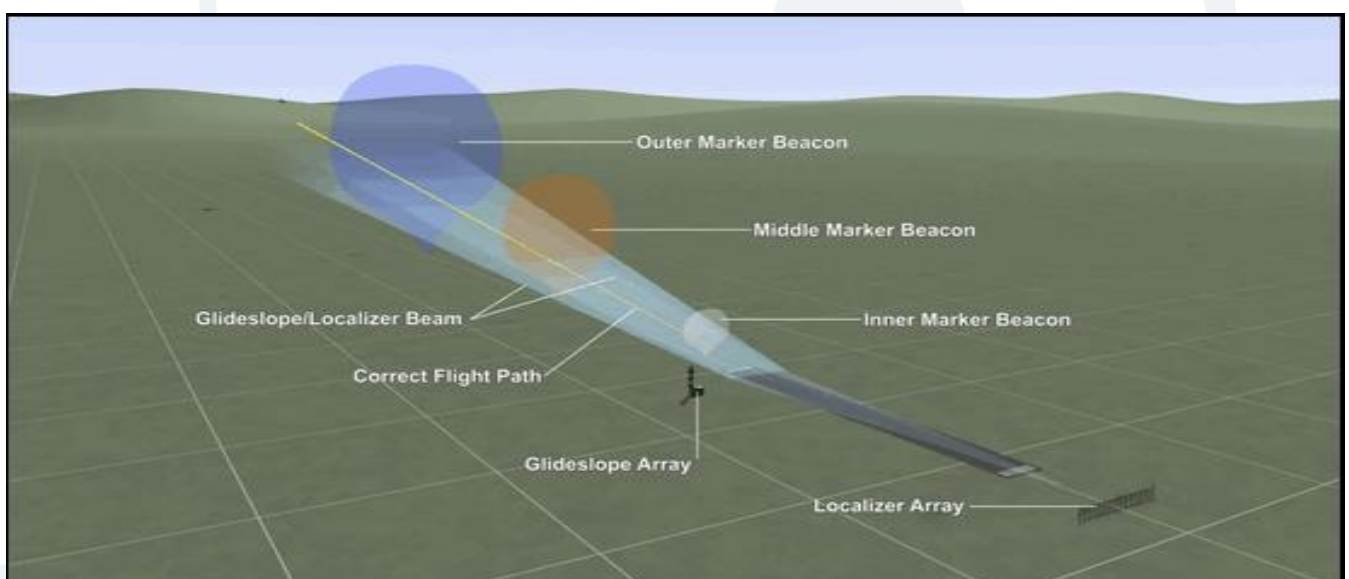


Figure 1 : Architecture de système d'aide à l'atterrissage aux instruments ILS

La situation actuelle au sein du notre service est dotée de quelques lacunes concernant la supervision et la gestion selon les normes OACI de point de vue efficacité de l'intervention en cas de panne technique pour la maintenance corrective ainsi que l'organisation des tâches de maintenance préventive d'où notre solution proposée ou nous allons remédier à tous ces inconvénients et donc augmenter la performance de la maintenance au sein du service technique de l'aéroport à fin d'assurer la sûreté de la navigation aérienne et la sécurité de l'espace aérien tunisien.

Depuis quelques années, les équipements mobiles sont de plus en plus complexes et robustes. Les appareils mobiles deviennent une combinaison de la technologie informatique et de la technologie du téléphone mobile. Dans l'avenir, les mobiles peuvent remplacer généralement le rôle de l'ordinateur portable. C'est pour quoi, les systèmes d'exploitation pour les mobiles (SE) de plus en plus deviennent importants, particulièrement pour les fournisseurs, les développeurs et les éditeurs d'application pour les mobiles.

Le présent rapport est organisé comme suit :

- **La première partie** a été consacrée à une présentation générale mettant l'accent sur la présentation de l'OACA qui représente l'organisme d'accueil de ce projet avant de passer à la déclaration de la problématique, la solution proposée pour y résoudre ainsi que le travail demandé pour assurer la réalisation de ce projet.
- **La deuxième partie** a été consacrée à l'étude préliminaire afin de mettre en œuvre les spécifications du besoin du système.
- **La troisième partie** a été réservée à l'étude fonctionnelle de projet
- **La quatrième partie** a été réservée à l'étude technique de projet
- **La cinquième partie** a été consacrée à la conception qui permet de mettre en œuvre les principales fonctionnalités du système.
- **Dans le dernier chapitre** nous présentons l'environnement logiciel et matériel et le passage vers le développement de l'application

Enfin ce rapport est clôturé par une conclusion générale donnant une synthèse à notre projet et exposant les perspectives qui visent l'amélioration de ce présent travail.

Chapitre 1 : Cadre général

Introduction :

Afin de mettre le travail dans son contexte général, nous présenterons dans ce chapitre le cadre de ce projet tout en présentant l'organisme d'accueil, les objectifs visés tout en déclarant la problématique ainsi que la solution proposée.

I Présentation de l'entreprise d'accueil

I.1 Présentation de l'Office de l'aviation civile et des aéroports (OACA):



L'Office de l'Aviation Civile et des Aéroports (OACA) , est un établissement public à caractère industriel et commercial sous tutelle du ministère du Transport, chargé de la gestion, de l'aménagement, de l'exploitation, de l'entretien et du développement des aéroports internationaux, ainsi que du contrôle régional et local de la navigation aérienne en Tunisie et des services de la navigabilité.

En outre, L'Office de l'Aviation Civile & des Aéroports (OACA) intervient dans :

- L'exploitation, l'aménagement et le développement des aéroports ainsi que l'accomplissement de toutes les opérations et services nécessaires aux voyageurs, au public, aux aéronefs, au fret et au courrier aérien dans les aéroports.
- Le contrôle régional et local de la navigation aérienne et la participation à l'exécution des plans de recherches et de sauvegarde.
- Les opérations de délivrance et de renouvellement des titres du personnel civil naviguant et des documents d'aéronefs.
- L'application des exigences réglementaires dans le domaine de la navigabilité des aéronefs.

L'Aéroport International Tabarka Ain Drahem (AITA) est l'un des aéroports qui sont sous la tutelle de l'OACA contenant un service technique de maintenance des aides à la navigation aérienne au sein duquel nous avons effectué notre stage de projet de fin d'études.

I.2 Etude de l'existant et Problématique :

Cette partie va être consacrée à dégager les lacunes de la situation actuelle au sein du service de la maintenance des équipements de radionavigation.

I.2.1 Unité de télécommande RCMS (Remote Control Monitoring System):

- L'unité de télécommande est utilisée dans la tour ou dans la salle de contrôle technique.
- Elle comporte des indicateurs d'état de fonctionnement ainsi que des alertes détaillés.
- Elle peut commander la mise en service ou hors service de l'équipement.

Cette unité utilise un système de moniteurs de télésurveillance dans lesquelles est installé le programme RMM (Remote Monitoring Monitor) qui nous donne les interfaces de fonctionnement des équipements de navigation.

A l'aéroport Tabarka, toute une installation assure l'acheminement des données des équipements des aides à la navigation aérienne de la piste vers la salle technique de supervision via fibre optique comme il est décrit dans la figure suivante:

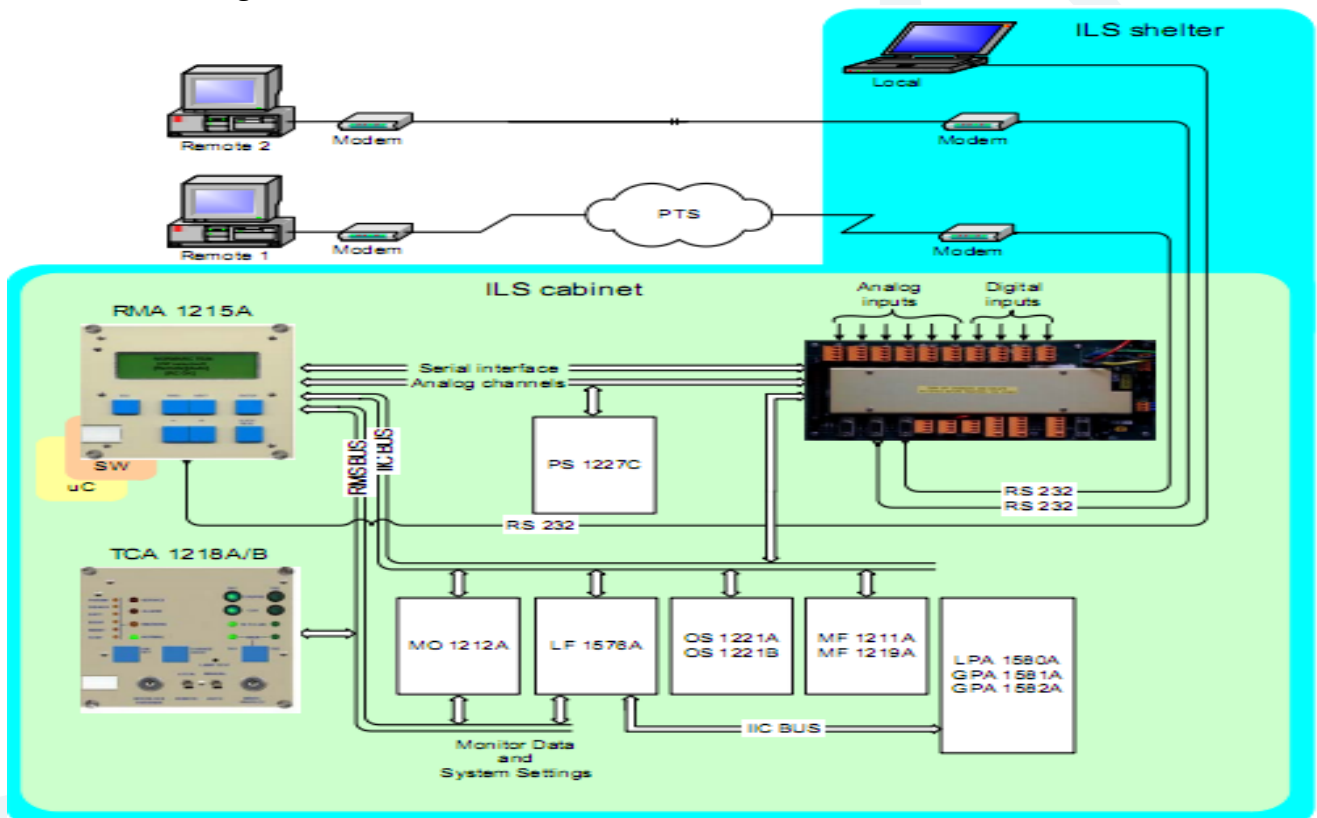


Figure 2:Réseau RCMS

Avantages:

- Données sur l'état des équipements en temps réel.
- Transmission des données rapide et avec redondance.

Inconvénients :

- Supervision limitée au niveau de la salle technique → Problème de mobilité.
- Grande distance entre le local de la supervision (tour de contrôle) et le lieu d'intervention (piste)
- Pas d'accès au système lorsqu'on a des problèmes au niveau de support de transmission (câblages détériorés) ou défaillances de certains équipements (modems, pc)
- Diagnostic et résolution lente des problèmes.

I.2.2 Absence d'un GMAO dans le service (Gestion de maintenance assistée par ordinateur) :

La gestion de maintenance assistée par ordinateur (souvent abrégée en GMAO) est une méthode de gestion assistée d'un logiciel destiné aux services de maintenance d'une entreprise afin de l'aider dans ses activités.

L'absence de ce système dans notre service nous a donné l'idée de créer cette application qui va nous aider à réaliser des tâches similaires à celles d'un GMAO.

I.3 Solution proposée :

Dans ce cadre, nous avons mis en œuvre ce projet pour répondre à des besoins dans le service de maintenance des équipements des aides à la navigation aérienne tout en résolvant les lacunes existantes. Le projet s'agit donc de concevoir et développer une application mobile sur une plateforme Android qui répond aux besoins de l'entreprise.

I.4 Travail demandé :

Nous sommes invitées à développer :

Une application mobile de Supervision et exploitation à distance des équipements aériens via le modèle client-serveur et à l'aide d'un smart phone comme protocole de communication

Ainsi, les fonctions et les bénéfices attendus les plus courantes de cette application sont :

- Gestion des équipements : inventaire des équipements
- Gestion de la maintenance : procédure de maintenance préventive et corrective
- Gestion des stocks : articles de rechange, listes de cartes installées
- Amélioration de la fiabilité et de la disponibilité des équipements
- Amélioration de la gestion des stocks (meilleur contrôle des sorties, aide aux inventaires)
- Amélioration de la planification des interventions
- Diagnostic et résolution rapide des problèmes.

Conclusion :

Ce chapitre a donné l'occasion de présenter l'organisme d'accueil, le cadre du stage ainsi que les objectifs visés ce que va nous permettre d'entamer le chapitre suivant ou nous allons étudier l'état de l'art du ce projet.

Chapitre2 : Étude préliminaire

Introduction :

L'étude préliminaire est le premier maillon dans la chaîne de l'implémentation du développement. Elle consiste en fait à faire dégager les divers besoins fonctionnels et non fonctionnels en considérant le système comme une boîte noire. Au cours de ce chapitre, nous essayerons de valoriser les entités externes interagissant directement avec le système ainsi que les différentes interactions.

I contexte de projet:

Depuis quelques années, les équipements mobiles sont de plus en plus complexes et robustes. Les appareils mobiles deviennent une combinaison de la technologie informatique et de la technologie du téléphone mobile. Dans l'avenir, les mobiles peuvent remplacer généralement le rôle de l'ordinateur portable. C'est pour quoi, les systèmes d'exploitation pour les mobiles (SE) deviennent de plus en plus importants, particulièrement pour les fournisseurs, les développeurs et les éditeurs d'applications. Grâce à la capacité mobile et le développement des équipements mobiles, on peut déployer les services pour les sociétés basé sur la technologie Smartphone. Actuellement, il existe plusieurs SE comme Windows Mobile, Palm OS, BlackBerry qui sont les systèmes propriétaires. De plus, il y a les plateformes libres ou code source ouvert comme Moblin.org, Ubuntu MID Edition, Androïde, etc. Chaque type de SE est souvent approprié à quelques modèles concrets des mobiles. Cependant, chaque type de SE a des avantages et des limitations. Il est donc très difficile de choisir la plateforme répondant bien à l'objectif d'une société ou à une personne.

I.1 Objectif (Aspect de la solution adoptée)

Après la phase d'étude de la situation existante du domaine de l'aviation Tunisien et précisément au sein de l'AITA, nous proposons une solution de supervision de maintenance des aides à la navigation aérienne. Notre solution consiste à implémenter toute une stratégie de travail par le biais d'une application mobile sur la plateforme **Androïde** pour l'amélioration de la performance de la maintenance des aides à la navigation tout en assurant la suivi des tâches de maintenance préventive (selon un plan annuel respectant les normes de l'OACI), la gestion du stock des pièces de rechanges des équipements, ainsi que la gestion des solutions adéquates des pannes techniques.....

1.2 Solution de mobilité

Depuis l'apparition de la notion de mobilité on assiste une redéfinition de l'entreprise tel que la disparition des frontières entre vie professionnelle et vie privée, entre domicile et lieu de travail. En fait, Une application mobile, est un programme téléchargeable conçu pour fonctionner sur un système d'exploitation mobile. Les services de mobilités permettent :

- D'accélérer la prise de l'information
- L'accès à l'application en temps réel, où qu'ils se trouvent

1.3 Choix de l'environnement de développement :

Androïde est un système d'exploitation ouvert (Open Source) pour tablettes tactiles, terminaux mobiles et TV connectées. Il a été conçu en 2007, par la société Androïde, une start-up rachetée par Google. Ses principaux concurrents sont Apple avec iPhone OS, RIM avec BlackBerry OS, Samsung avec Bada ou encore Microsoft et son Windows Phone os (ex Windows Mobile).

Il est doté de plusieurs avantages tels que la grande mémoire, la gestion de processus, le modèle de sécurité, le soutien de bibliothèque partagé, etc.

Dans cette partie, nous allons comparer les deux systèmes d'exploitation mobiles Androïde et IOS.

IPhone IOS VS Androïde

IPhone IOS

L'environnement de développement pour l'iPhone est Xcode. iPhone est basé sur le système MAC OS il n'est donc pas compatible avec Windows. Le développement d'applications pour iPhone OS se fait avec Objective C, une extension du langage C, orienté objet et réflexif.

Androïde

Le développement d'applications pour Androïde se fait en Java. Java est un puissant langage orienté objet, utilisé très largement dans le monde du développement. Le développement d'applications Androïde est possible aussi bien sur Eclipse, que sur Apache ou JDK. C'est pourquoi le développement d'applications Androïde est possible aussi bien sur Windows que sur Linux ou Mac.

Après la comparaison, nous avons choisi Androïde comme support d'application.

II Présentation de l'application :

C'est notre solution proposée pour assurer la coopération entre la supervision et la gestion tout en implémentant une application collaboratrice résolvant les contraintes de la mobilité de la maintenance, le temps consommé lors de l'intervention technique, la gestion des ressources, la suivi de toute l'opération de l'intervention et le choix adéquat des solutions des pannes, une solution qui respecte les normes OACI et met en évidence l'amélioration de l'efficacité de l'intervention technique et donc l'amélioration de la sécurité de la navigation aérienne tunisienne. Pour garantir l'efficacité de l'application en terme de la navigation aérienne à l'échelle national, nous avons conçu cette application pour gérer la maintenance des aides à la navigation aérienne au sein du l'AITA pour être au futur un système généralisable à tous les aéroports tunisiens.

Cette application comporte trois modules :

1/Module administrateur :

C'est un espace géré par l'administrateur de l'application et son rôle consiste à préparer tout un environnement d'intervention technique efficace en assurant l'extraction, l'ajout, la vision et la suppression des pièces de rechange et des solutions des pannes des différents équipements de la navigation aérienne, de notifier les différentes tâches de la maintenance préventive organisées selon un plan annuel et conforme aux normes OACI ainsi que de gérer la localisation du déplacement des personnels techniques et des voitures de services entre les divers équipement éparpillés de part et d'autre de la piste de l'aéroport à fin d'assurer une supervision complète de l'opération de l'intervention.

2/Module maintenance préventive :

C'est un espace géré par le superviseur grâce au droit lui est réservé, en fait le superviseur consulte les notifications périodiques illustrant les divers tâches de la maintenance préventive organisées selon un plan annuel détaillé et conforme à la norme de l'OACI. Cet espace fournit au superviseur aussi une formation continue tout en consultant les différents détails et caractéristiques techniques des équipements pour que sa supervision soit efficace.

3/Module maintenance corrective :

C'est l'espace le plus sensible dans le cycle de la maintenance puisqu'il permet aux techniciens qualifiés d'intervenir lors d'une panne d'une manière efficace et performante puisqu'il les permet de se bénéficier des solutions de la plupart des pannes techniques qui peuvent parvenir, déjà établit dans la

base de données à des statistiques faites mettant en évidence des dizaines d'années d'expérience des experts dans ce domaine.

III Spécification des besoins :

L'expression des besoins comme son nom l'indique a pour objectif d'assurer une compréhension des besoins et des exigences du client. Plus précisément, la spécification des besoins permet de :

- Recenser les besoins fonctionnels (de point de vue utilisateur).
- Appréhender les besoins non fonctionnels (technique) et livrer une liste des exigences.

III.1 Besoins fonctionnels :

Au niveau serveur :

- Le serveur reçoit une requête de client qui a mentionné une panne bien déterminée. En se basant sur cette dernière, le serveur consulte la base de données et renvoi la réponse correspondante ;
- Le serveur assure l'envoi d'une notification permanente et périodique des divers tâches de la maintenance préventive convenable aux normes OACI et organisée selon un plan clair ;
- Le serveur assure la localisation du déplacement des personnels et des voitures de service lors d'une intervention bien déterminée entre les différents équipements de la navigation aérienne distribués de part et d'autre de la piste de l'aéroport.

Au niveau client :

- Permet aux utilisateurs d'accéder aux informations et de bénéficier d'une idée globale sur tous les équipements de la navigation aérienne.
- Envoie une requête au serveur contenant les critères de recherche pour recevoir les pièces de rechange et les solutions des pannes adéquates.

————→ En recevant la réponse, le client va explorer les données reçus pour assurer son intervention et il peut à tout moment relancer une autre recherche.

III.2 Besoins non fonctionnels :

Besoins Matériels :

Afin de tester l'application, on a besoin du matériel suivant :

- Téléphone portable : Pour tester l'application il nous faut un téléphone portable supportant la technologie Androïde.
- Serveur connecté à internet d'une façon permanente pour la réception des requêtes envoyées par l'utilisateur.

Besoins logiciels :

- Un langage de programmation d'une application mobile.
- Une application serveur qui tourne d'une façon permanente pour recevoir et traiter les requêtes envoyées par l'utilisateur ou l'administrateur

Contraintes ergonomiques :

- Il faut que l'interface de l'application soit facile à manipuler.
- Il est nécessaire de faciliter l'utilisation des différentes fonctions fournies par l'interface tout en essayant de rendre l'application facile à utiliser pour gagner le maximum du temps
- Il est nécessaire que l'interface soit lisible. La lisibilité de l'application peut être atteinte par la bonne répartition des différents composants de l'interface tels que le menu, les boutons...
- Il faut utiliser un bon style et une bonne mise en forme pour que l'apparition soit claire et compréhensible.

Cohérence de données :

Le système est conçu pour être généralisé qui englobe tous les aéroports tunisiens, une contrainte de cohérence des données et nécessaire au niveau de la base de données pour que la fusion des différents systèmes sera possible.

IV processus de développement et méthodologie de conception:**IV.1 Le processus de développement unifié 2TUP :**

Nous avons choisi pour que notre projet soit bien organisé 2TUP comme étant un processus de développement logiciel qui implémente le processus unifié doté des plusieurs caractéristiques en effet il est itératif, piloté par les risques, orienté composant, orienté utilisateur et incrémental.

Définition :

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le

système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Avantage :

Pour résumer, il permet à la fois de capitaliser la connaissance métier sur la branche gauche et de réutiliser un savoir-faire technique sur la branche droite. En d'autres termes, un meilleur contrôle sur les capacités d'évolution et de correction de tels système.

Architecture:

- une branche technique
- une branche fonctionnelle
- une phase de réalisation

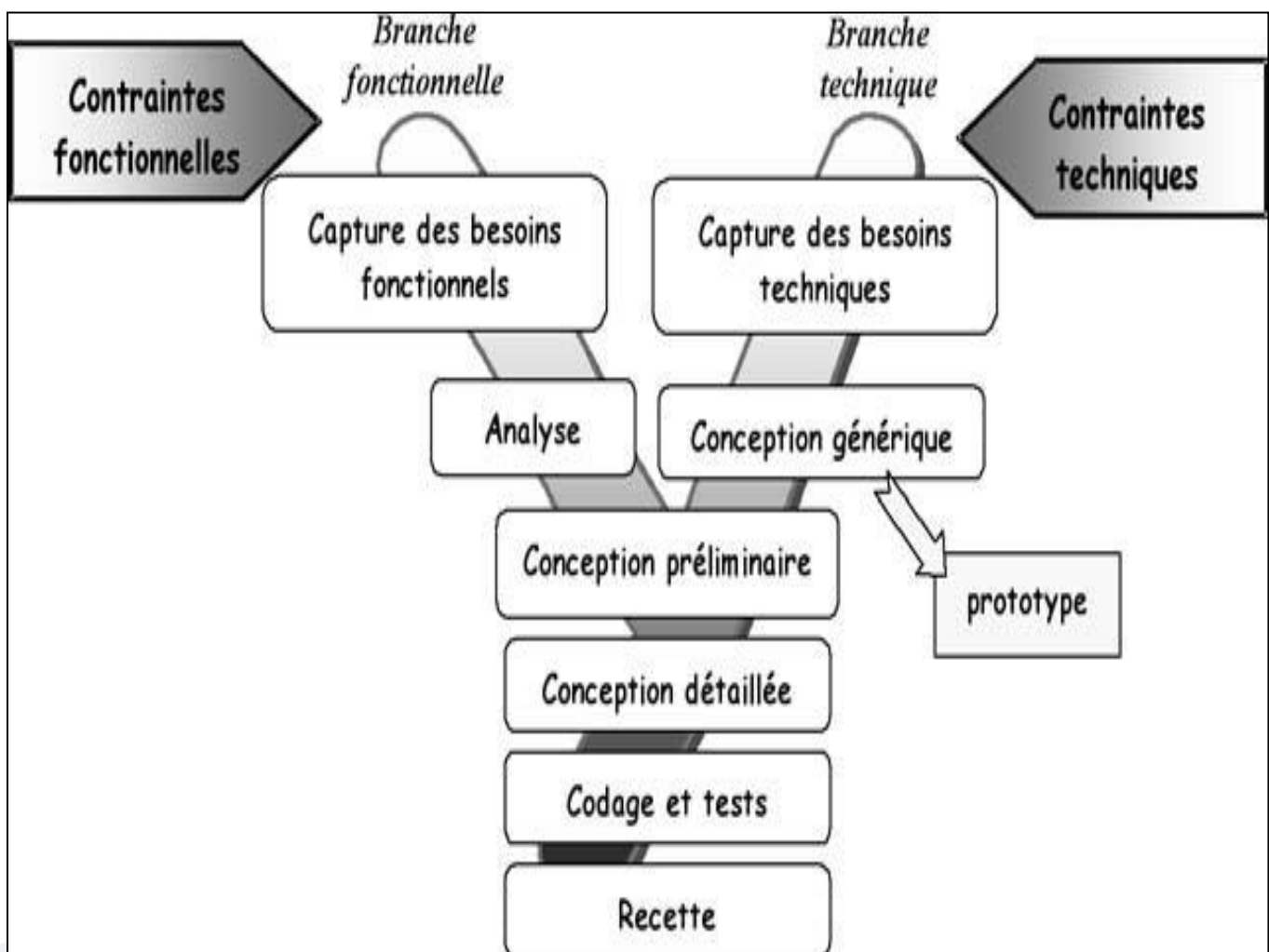


Figure 3:Architecture du processus 2TUP

La branche fonctionnelle :

La branche fonctionnelle capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.

La branche technique :

La branche technique capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système le sont indépendamment des fonctions à réaliser.

La phase de réalisation :

La phase de réalisation consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

IV.2 La méthodologie de conception UML :

- Notation graphique de modélisation à objet
- Permet de visualiser, spécifier, construire et construire les différentes parties d'un système logiciel
- Langage graphique basé sur des diagrammes

UML est capable de :

- Représenter des systèmes entiers (pas uniquement logiciels) par des concepts objets ;
- Pouvoir modéliser des systèmes à différents niveaux de granularité (pour permettre d'appréhender Des systèmes complexes

Il existe une correspondance entre les différents diagrammes d'UML et les étapes de 2TUP.

Tableau 1:Relation entre 2TUP et UML :

2TUP	UML
Capture des besoins fonctionnels	Diagramme des cas d'utilisation Diagrammes de séquence Diagrammes de collaboration
Analyse	Diagramme de classes Diagrammes d'états transition
Capture des besoins techniques	Diagramme des cas d'utilisation
Conception générique	Diagramme de déploiement
Conception préliminaire	Diagramme de composants Diagramme de déploiement
Conception détaillée	Diagramme de classes Diagramme de séquence Diagramme de collaboration Diagramme d'états Diagramme d'activités Diagramme de composants

Conclusion :

Au niveau de ce deuxième chapitre, nous avons essayé de préparer tous les facteurs pour entamer la partie conception tout en élaborant un cahier de charge bien détaillé et dégagant les divers besoins fonctionnels et non fonctionnels. Alors, le chapitre suivant va être consacré pour le traitement de l'architecture du système à réaliser ainsi que la conception détaillée à l'aide des différents diagrammes de conception UML.

Chapitre 3 : Eude fonctionnelle

Introduction :

La dissociation de la branche fonctionnelle de la branche technique est la principale caractéristique de la méthodologie 2TUP.

Ce chapitre sera consacré pour l'étude fonctionnelle ou on va entamer la phase d'analyse à travers les diagrammes de cas d'utilisation, les diagrammes de séquence système.

I Capture des besoins fonctionnels :

La capture des besoins fonctionnels va nous offrir une description claire des différentes façons qu'auront les acteurs d'utiliser le futur système.

I.1 Identification des cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases).

Ce tableau représente les différents cas d'utilisation du système, ainsi que les acteurs et les messages émis et reçus.

Tableau 2 : Liste des cas d'utilisation

Cas d'utilisation	Acteur principal	Message(s) émis reçus par les acteurs
Changer les paramètres d'accès.	Administrateur	Emet : authentification, demande de changement des paramètres d'accès. Reçoit : succès / échec d'authentification, enregistrement de modification
Gérer les pièces de rechange.	Administrateur	Emet : authentification, ajout, modification, suppression Reçoit : succès / échec
Gérer les solutions des pannes techniques.	Administrateur	Emet : authentification, ajout, modification, suppression Reçoit : succès / échec
Indexer les solutions des pannes.	Administrateur	Emet : authentification, demande d'indexation des solutions des pannes. Reçoit : succès / échec
Gérer la localisation et le suivi du déplacement des intervenants dans la piste.	Administrateur	Emet : authentification, demande de gestion de la localisation. Reçoit : succès / échec
Notifier les tâches de la maintenance préventive.	Administrateur	Emet : authentification, demande de notification des tâches. Reçoit : succès / échec
Consulter les informations sur les équipements.	Technicien superviseur	Emet : authentification, demande d'affichage de la liste des informations. Reçoit : Liste des informations.
Consulter les notifications des tâches de la maintenance préventive.	Technicien superviseur	Emet : authentification, demande d'affichage de la liste des notifications. Reçoit : Liste des notifications.
Consulter les pièces de rechange.	Technicien intervenant	Emet : authentification, demande d'affichage de la liste des pièces de rechange. Reçoit : Liste des pièces.
Consulter les solutions des pannes.	Technicien intervenant	Emet : authentification, demande d'affichage de la liste des solutions des pannes technique. Reçoit : Liste des solutions.

I.1.1 Présentation des cas d'utilisation :

Ce diagramme représente les différents cas d'utilisation du système :

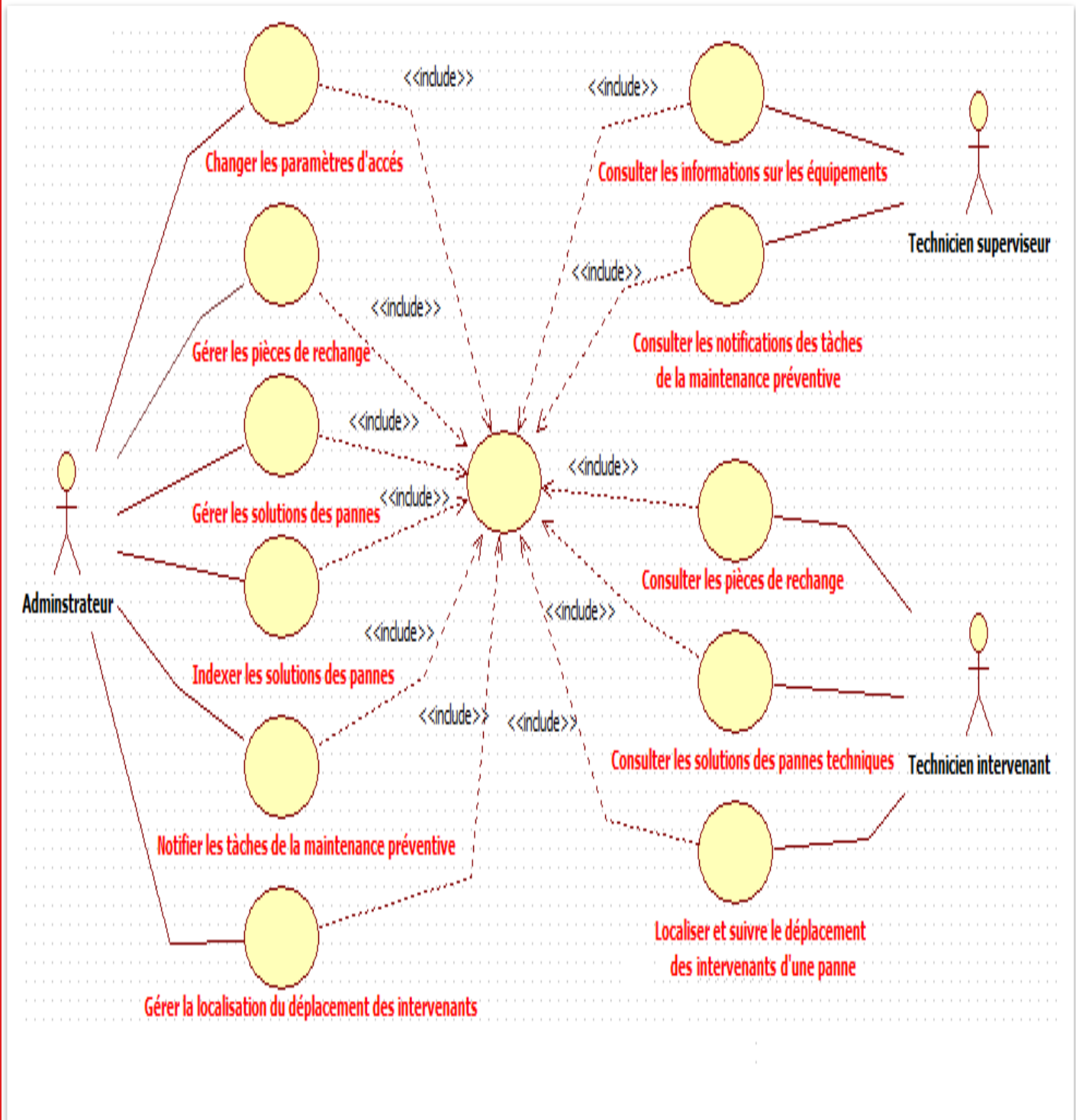


Figure 4:Diagramme de cas d'utilisation globale



I.1.2 Description des cas d'utilisation :

Pour assurer une communication claire et facile avec les utilisateurs, une documentation des cas d'utilisation par le biais du patron textuel enrichi par des diagrammes de séquence me paraît très efficace.

I.1.3 Le cas d'authentification :

Pour sécuriser l'application, il faut contrôler l'accès au système d'où la nécessité d'authentification de tous ses acteurs. La figure ci-dessous illustre l'authentification des différents acteurs :

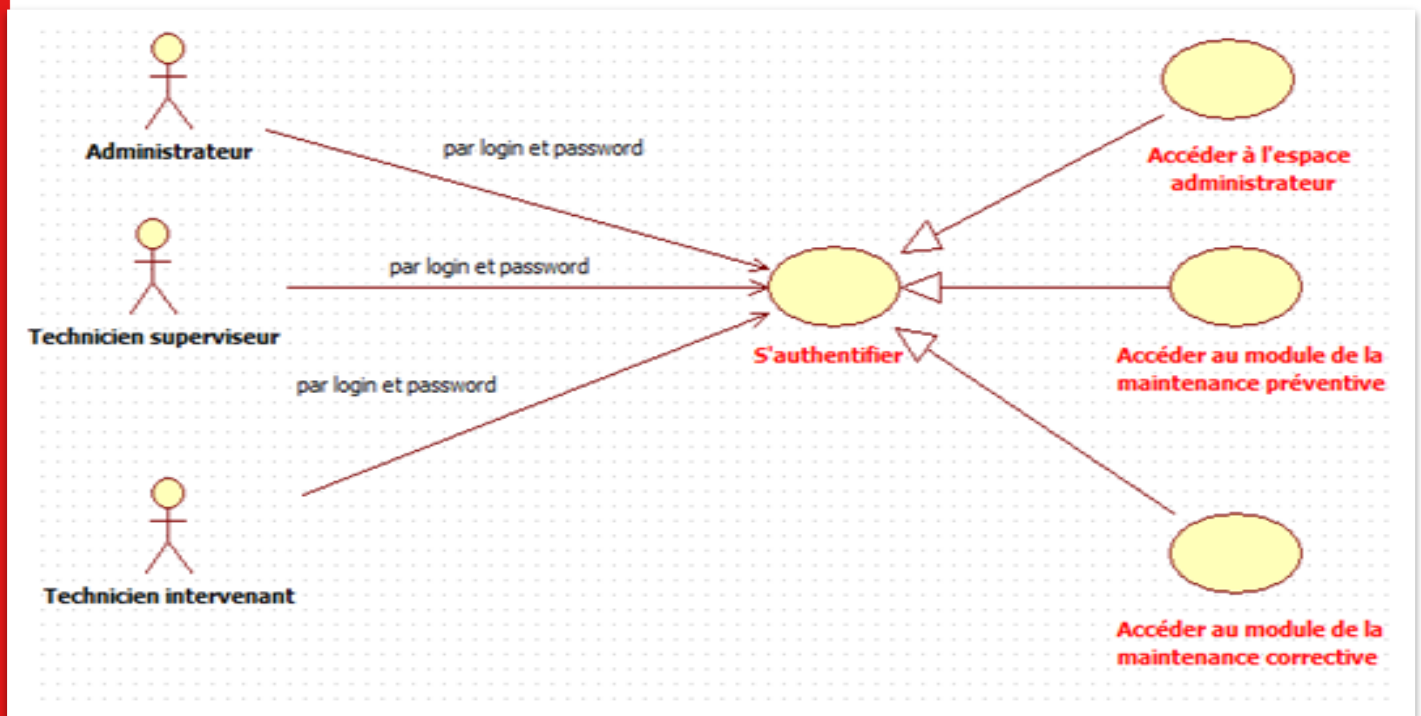


Figure 5:Diagramme d'authentification

I.2 Le module d'administration :

Pour le module d'administration, nous avons présenté le cas de gestion des solutions des pannes techniques.

I.2.1 Gestions des pannes techniques :

La figure ci-dessous décrit la procédure de gestion des solutions des pannes techniques ou l'administrateur une fois authentifié va ajouter une solution tout en attribuant une panne touchant un module dans un équipement à chaque solution non indexée. En cas de besoins, il peut mettre à jour ou supprimer cette solution.

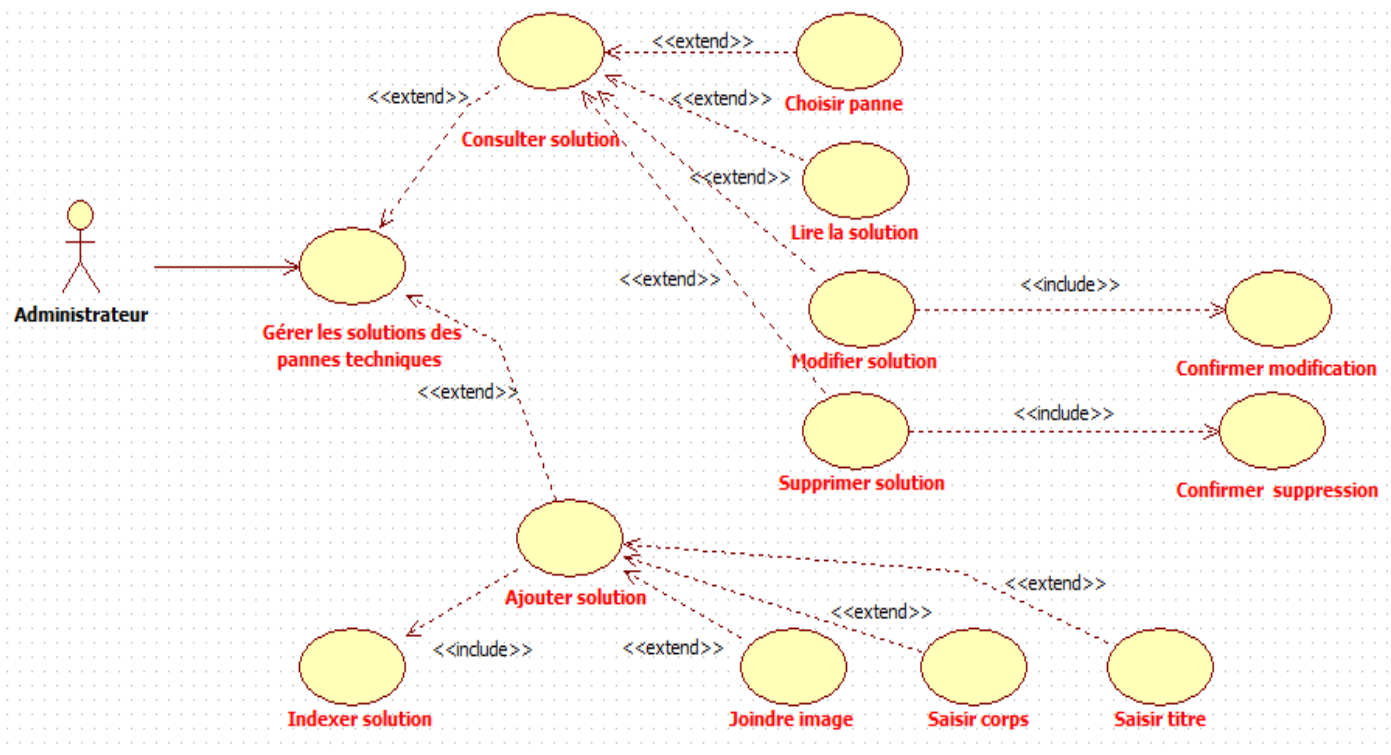


Figure 6:gestion des solutions des pannes techniques

Tableau 3 : Patron textuel du cas d'utilisation Indexer des solutions des pannes

Sommaire d'identification
Titre : Gérer les solutions des pannes techniques.
Résumé : Gestion des solutions des pannes, ajout, modification, suppression.
Acteurs : Administrateur.
Description des enchaînements
Pré-conditions : ➤ L'administrateur est authentifié. ➤ Il existe au moins une solution de panne pour l'ajouter. ➤ Il existe au moins une panne. ➤ Il existe un besoin pour la modification.
Scénario nominal :
Ce cas d'utilisation commence lorsque l'administrateur demande au système la main à la gestion.

Enchaînement alternatif :

1. Ajouter une solution tout en assurant l'indexation avec l'équipement, le module et la panne correspondants.

[Exception 1 : éléments non trouvés.]

2. Modifier la solution, en cas de besoin l'administrateur accède à la solution et effectue les modifications.

3. Supprimer la solution, il la sélectionne et la supprime.

Post-conditions :

➤ Solution est ajoutée ou mis à jour.

I.2.3 Diagramme de séquence des solutions des pannes techniques :

On peut représenter le diagramme de séquence du scénario nominal de cas d'utilisation gérer les solutions des pannes techniques comme suit :

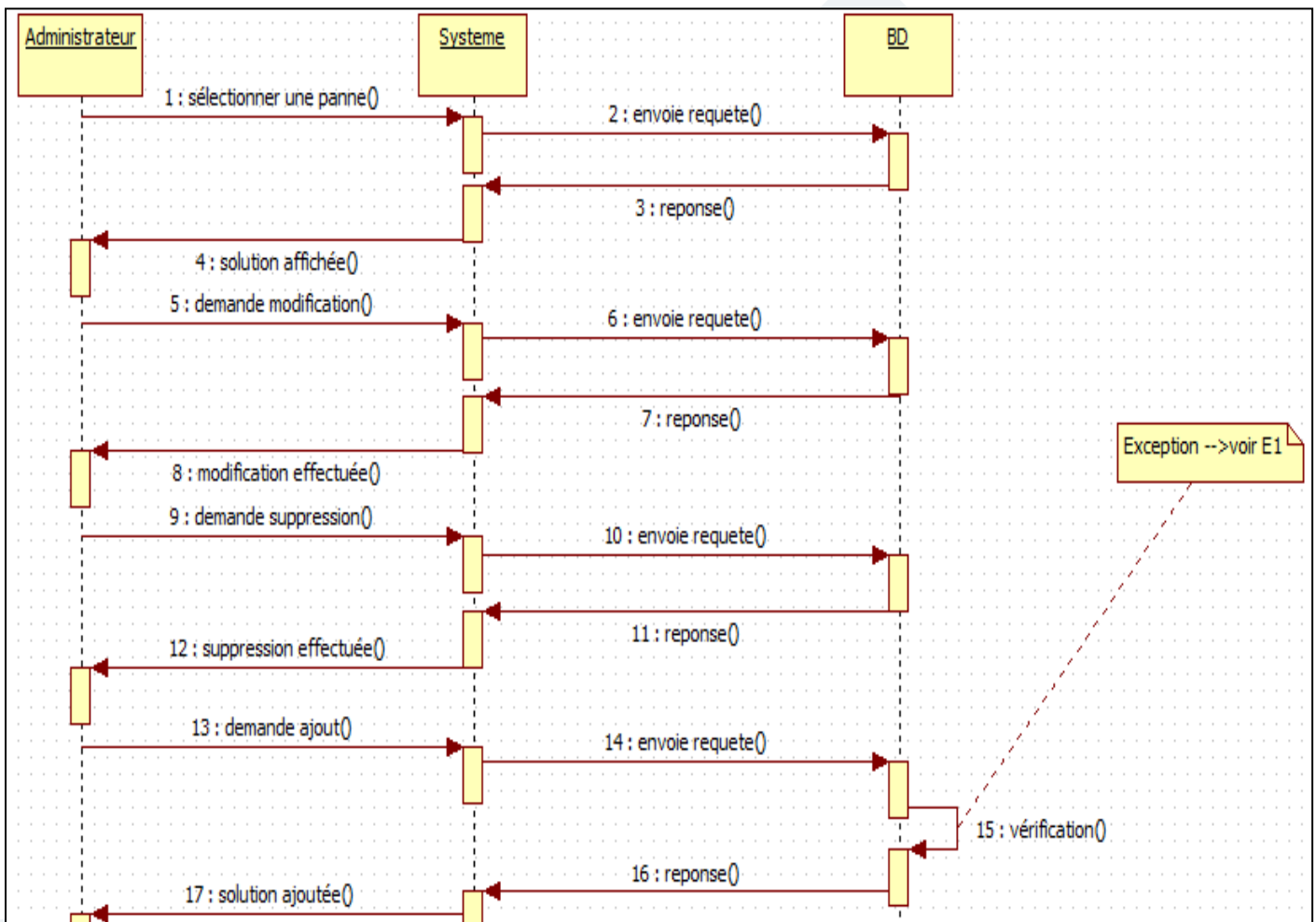


Figure 7:Diagramme de Gestion des solutions des pannes techniques

E1_: Lors de l'ajout d'une solution, le système vérifie si elle est déjà enregistrée dans la base de données, si elle existe l'exception est déclenchée.

I.3 Module de la maintenance préventive :

Ce module est géré par le technicien superviseur, à travers lequel il peut consulter les informations sur les équipements des aides à la navigation aérienne ainsi que les notifications des tâches de la maintenance préventive selon un plan annuel conforme aux normes OACI.

I.3.1 Affinement du cas d'utilisation « consulter les notifications des tâches de la maintenance préventive » :

Avant de commencer la consultation le technicien superviseur doit s'authentifier pour garantir la sécurité d'accès aux données sur les équipements de la navigation aérienne de l'AITA. Après l'authentification.

Le technicien peut lire les diverses tâches de la maintenance préventive organisé selon un plan annuel bien déterminée à une heure bien précise. Une fois informé, le technicien confirme sa lecture.

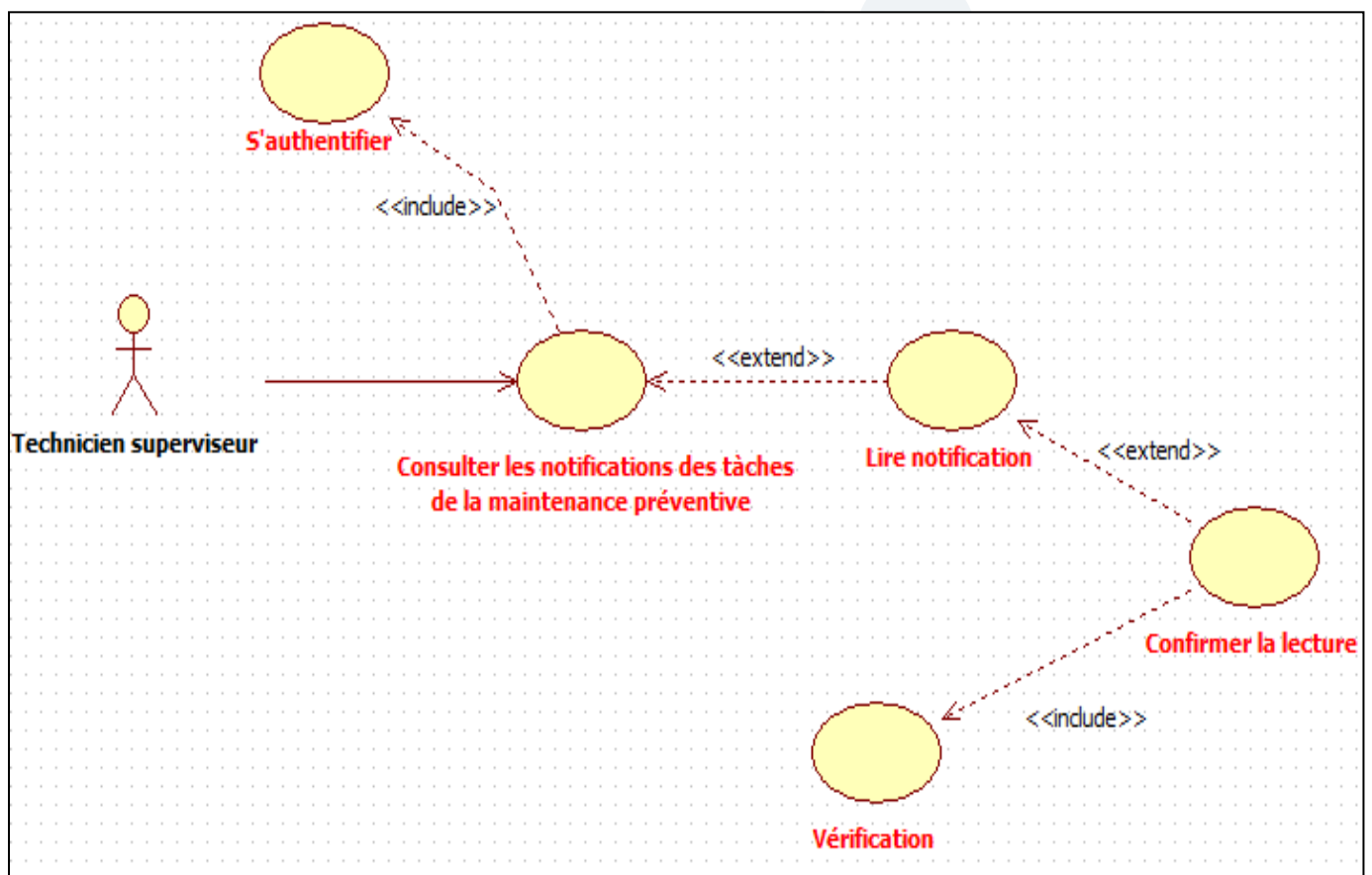


Figure 8:Diagramme des tâches de la maintenance préventive

Tableau 4 : Patron textuel de Consultation des notifications des tâches de la maintenance préventive

Sommaire d'identification
Titre : Consulter les notifications des tâches de la maintenance préventive.
Résumé : Le technicien peut lire la notification, être informé de la tâche à effectuer puis il confirme sa lecture.
Acteurs : Technicien superviseur.
Description des enchaînements
Pré-conditions : ➤ Le technicien superviseur est authentifié.
Scénario nominal :
Ce cas d'utilisation se déclenche par le système périodiquement et à une heure bien précise.
Enchaînement alternatif : 1. A l'heure désignée, la notification se déclenche et la tâche à faire sera affiché pour le technicien. [Exception 2 : notification non existante.] 2. Le technicien lit la tâche de la maintenance préventive à faire. 3. Confirmer la lecture. Post-conditions : ➤ Notification consultée.

I.3.2 Diagramme de séquence des notifications des tâches de la maintenance préventive :

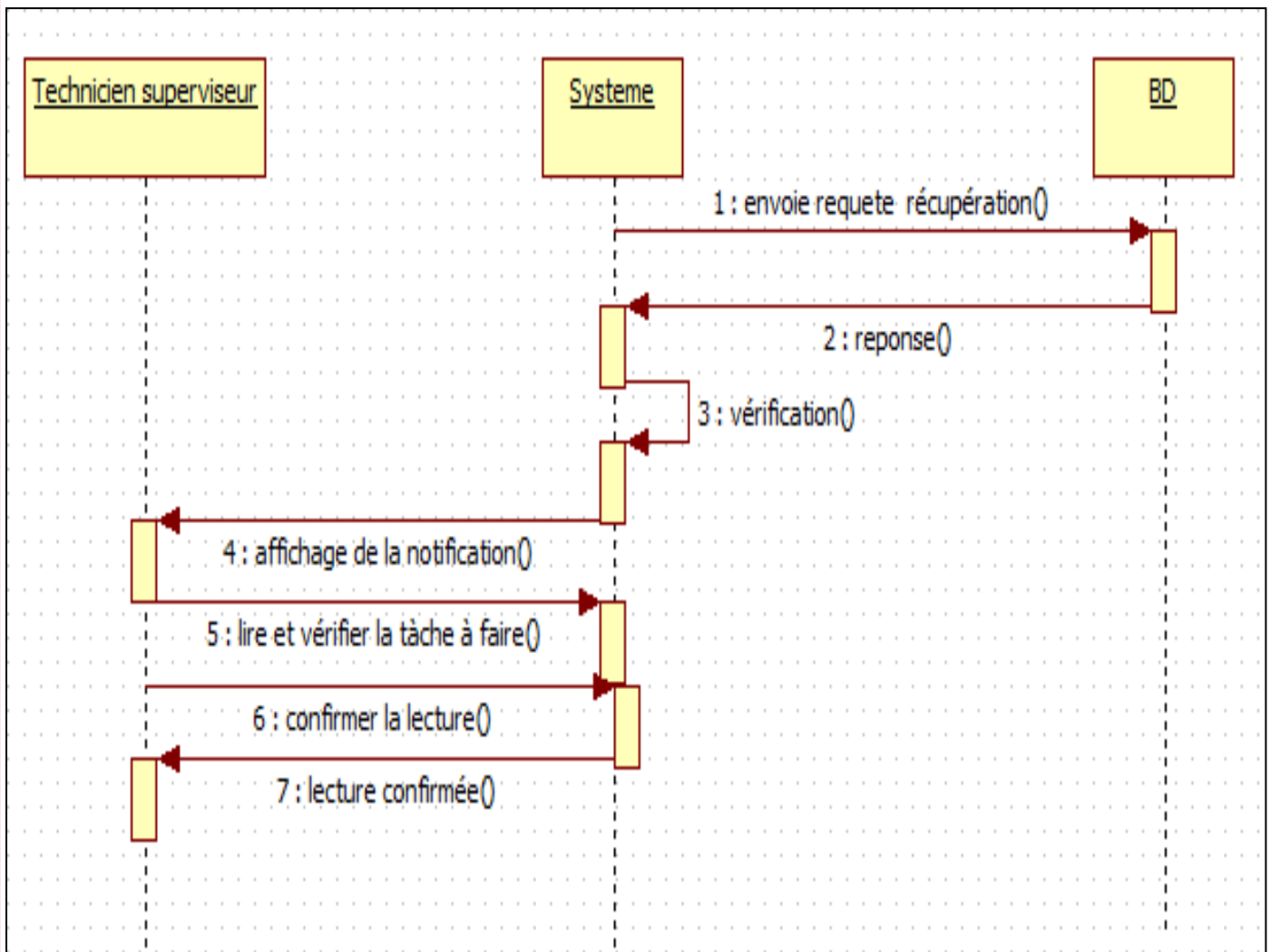


Figure 9:les notifications des tâches de la maintenance préventive

I.4. Diagramme de séquence de consultation des notifications des tâches de la maintenance corrective :

Nous pouvons représenter le scénario nominal déjà cité de ce cas d'utilisation par le diagramme de séquence suivant :

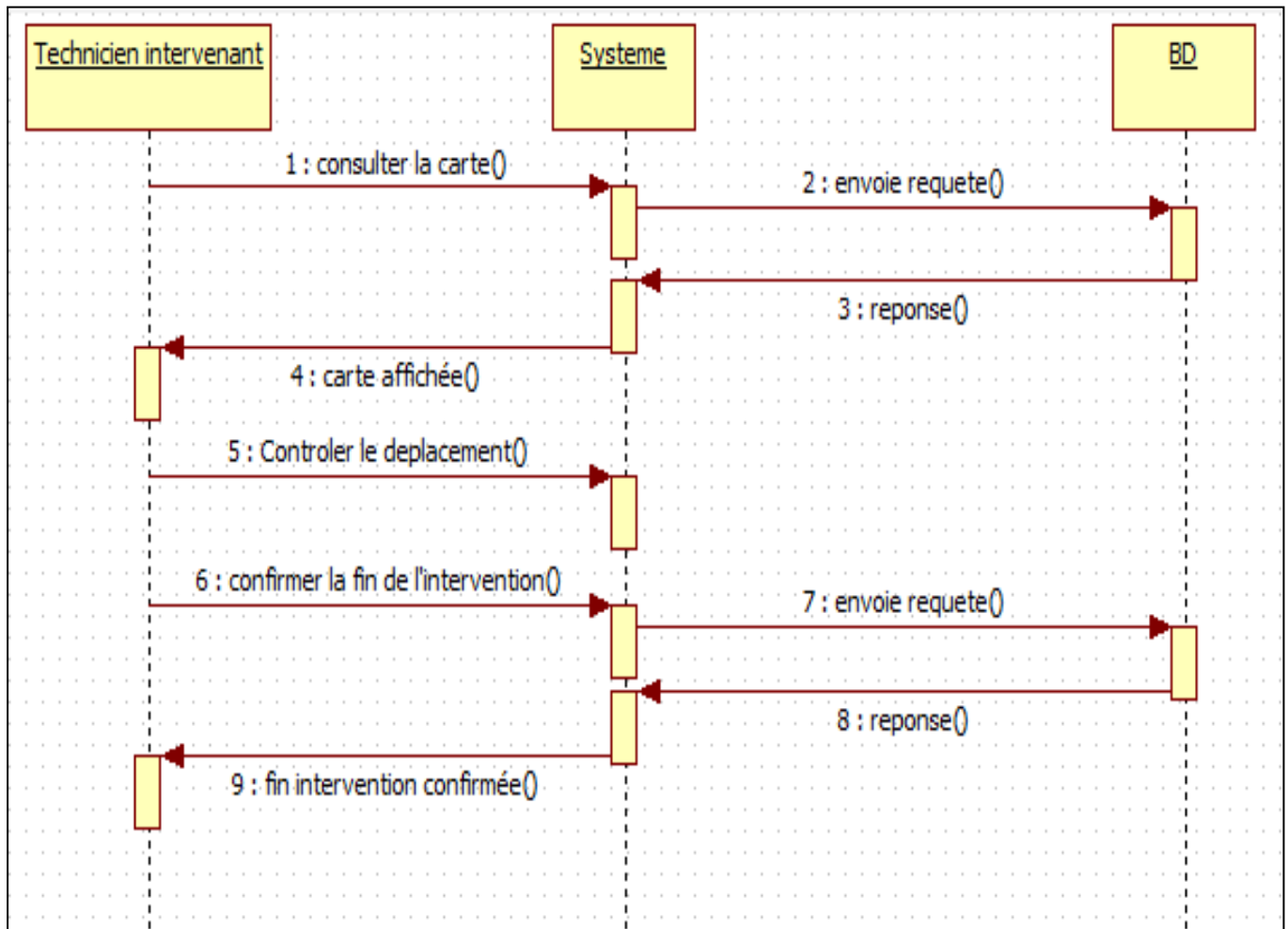


Figure 10:les notifications des tâches de la maintenance corrective

Conclusion

Après avoir décrit la branche fonctionnelle, le prochain chapitre est consacré pour la deuxième branche qui est la branche technique.

Chapitre 4 : Etude technique

Introduction

La capture des besoins techniques met en évidence le recensement des outils, des matériels et des technologies à utiliser ainsi que les contraintes (temps de réponse maximal, contraintes d'intégration avec l'existant) et tout cela va aboutir à une première conception de l'architecture technique. Sur le plan technique aussi, la conception générique est basée sur le découpage en composants nécessaires à la construction de l'architecture technique. Cette étape permet de minimiser l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles.

I Spécification d'architecture :

Le pattern architectural de notre application est composé comme suit :

- L'espace Administrateur.
- L'espace Maintenance préventive.
- L'espace Maintenance corrective.
- Le serveur Web.
- Le serveur Base de données.

La figure suivante présente le diagramme de déploiement du système qui sert à représenter les éléments matériels et la manière dont les composants du système sont répartis sur ces derniers et interagissent entre eux.

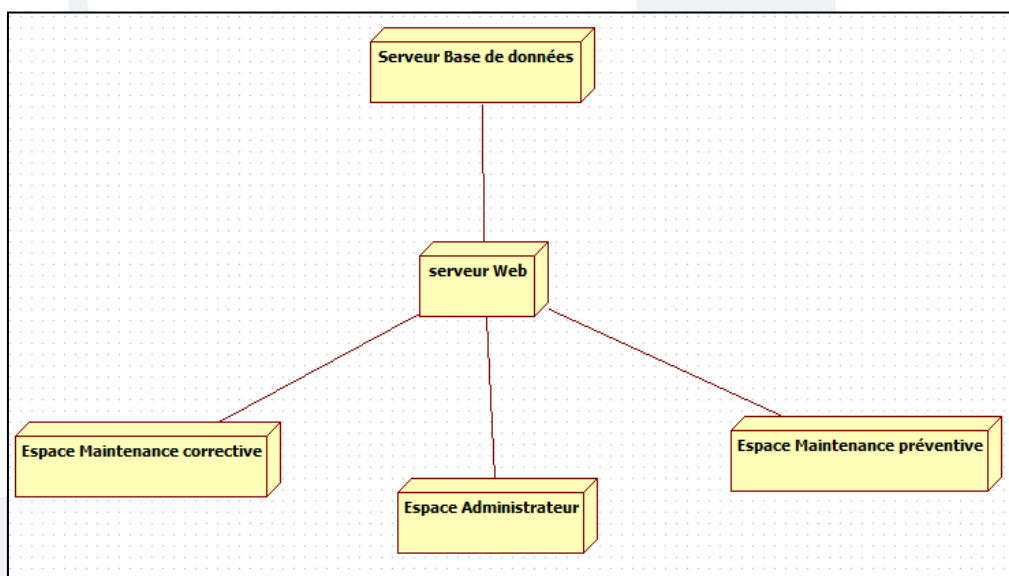


Figure 11:Diagramme de déploiement

II architecture de l'application

II.1 Présentation du modèle MVC :

Le modèle **MVC** est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective ce qui nous permet de bien gérer chacune d'eux indépendamment des autres.

Ce paradigme regroupe les fonctions nécessaires en trois catégories :

Les modèles : Représentent les données de l'application (les bases de données en fait partie) et définissent la logique de manipulation de ces données. C'est dans cette partie que s'effectuent les traitements, nous ne nous occupons absolument pas de la mise en forme, mais bien des données seules. Dans une application respectant les règles du modèle MVC, les données les plus importantes seront encapsulées dans les objets modèles.

Les vues : représentent la vue de l'application, l'interface avec laquelle l'utilisateur communique. Ils ne s'occupent pas de la gestion ou du stockage des données puisqu'aucun traitement ne doit être effectué dans cette partie. Mais, en contrepartie, ils affichent les résultats provenant des objets "modèles" et s'assurent que ces données sont correctement affichées.

Le contrôleur : gère l'interaction avec l'utilisateur. Ainsi c'est dans cette partie que se réalise l'interaction entre la vue et le modèle. En effet, les objets contrôleur reçoivent les requêtes utilisateur puis détermine quelles parties des objets vues et modèles sont requises. Ils constituent donc l'intermédiaire entre les deux autres types d'objets.

La figure ci-dessous présente l'architecture MVC ainsi que l'interaction entre ses composants

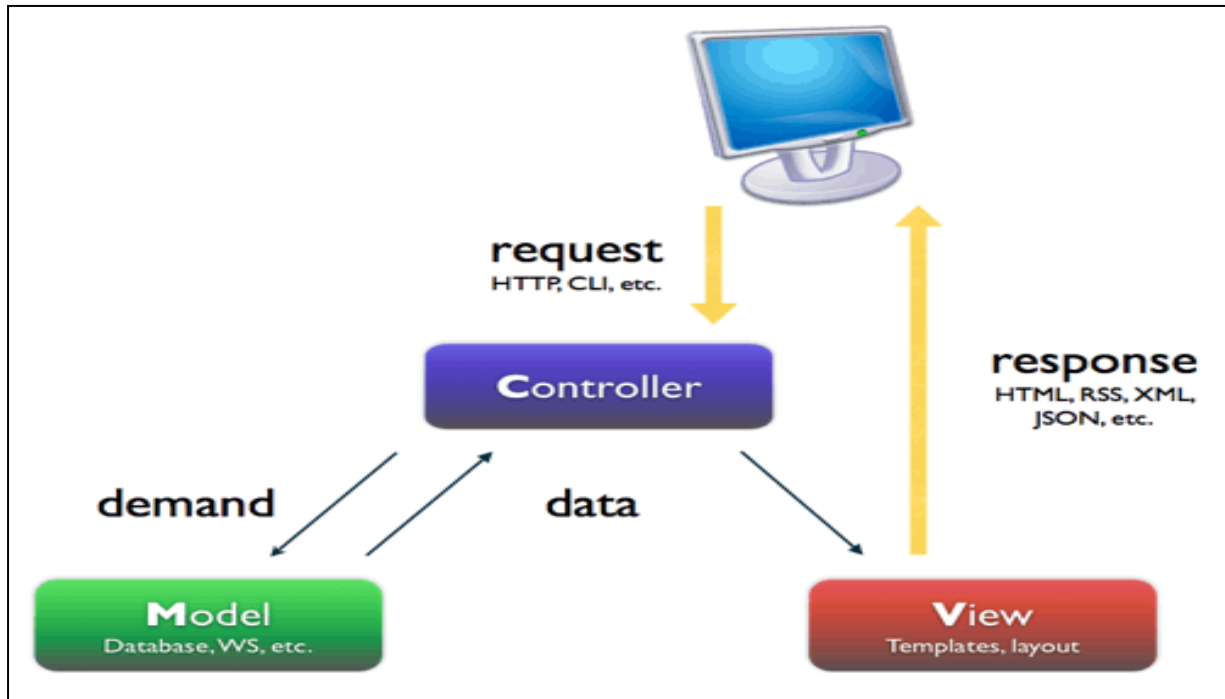


Figure 12:Modèle MVC

II.2. Présentation de l'architecture trois tiers de l'application :

Nous avons choisi l'architecture 3-tiers pour établir ce travail qui vise à assurer une séparation de trois couches logicielles au sein de l'application.

Description :

L'architecture 3-tiers est un modèle logique d'architecture applicative basée sur la séparation de trois couches :

- La présentation des données: correspondant à l'affichage, le dialogue avec l'utilisateur.
- Le traitement des données: correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- L'accès aux données: correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

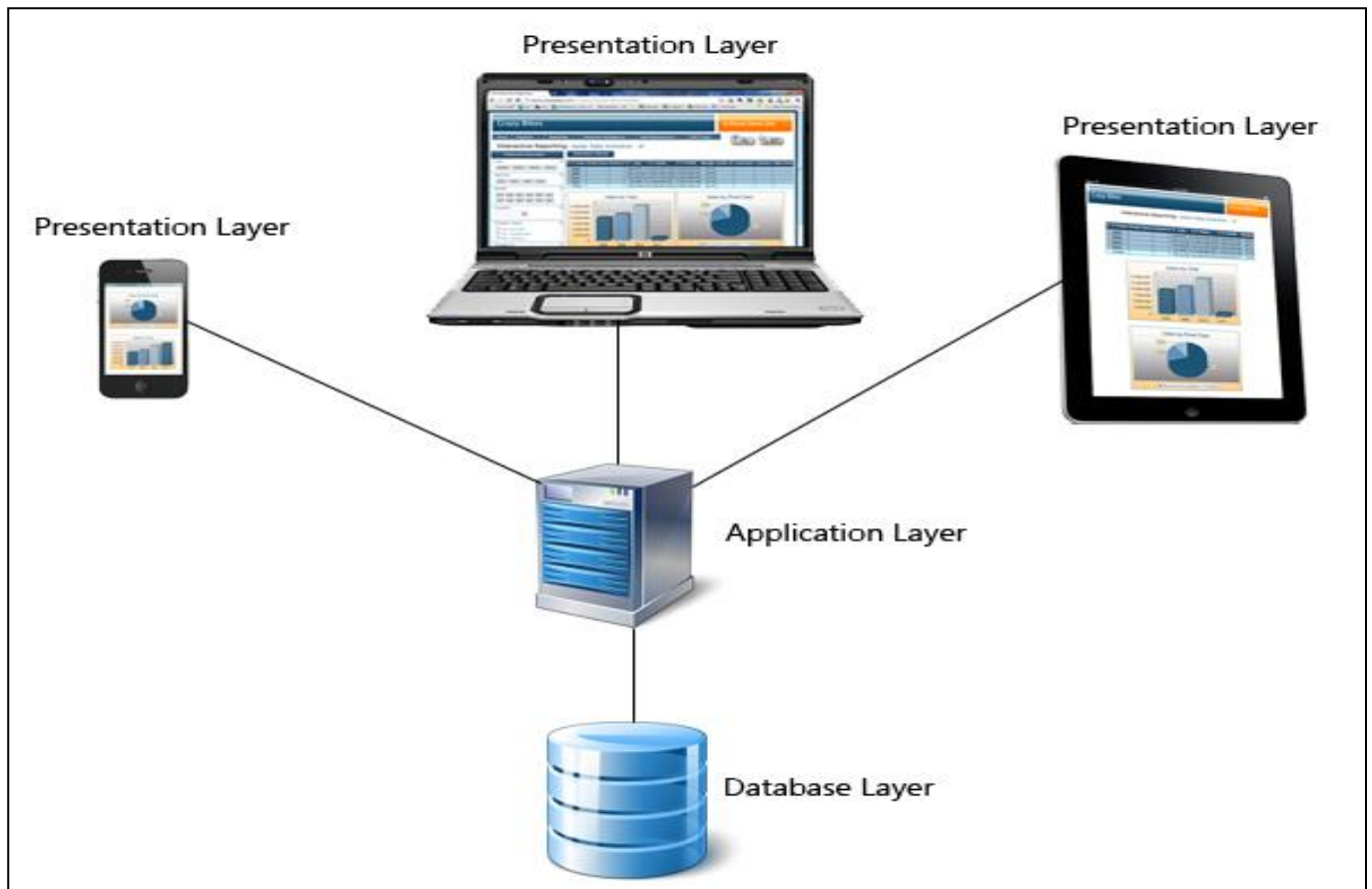


Figure 13:Architecture trois-tiers de l'application

Critères de choix :

L'architecture trois tiers est dotée de plusieurs avantages vis-à-vis de celle deux tiers tels que :

- Prise en compte de l'hétérogénéité des plateformes (serveurs, clients, langages, etc.).
- Amélioration de la sécurité des données, en supprimant le lien entre le client et les données.
- Meilleure répartition de la charge entre différents serveurs d'application.

III Diagramme de composant général :

Nous présentons maintenant le diagramme de composant qui vise à mettre en évidence l'architecture général de l'application et les dépendances entre les divers composants.

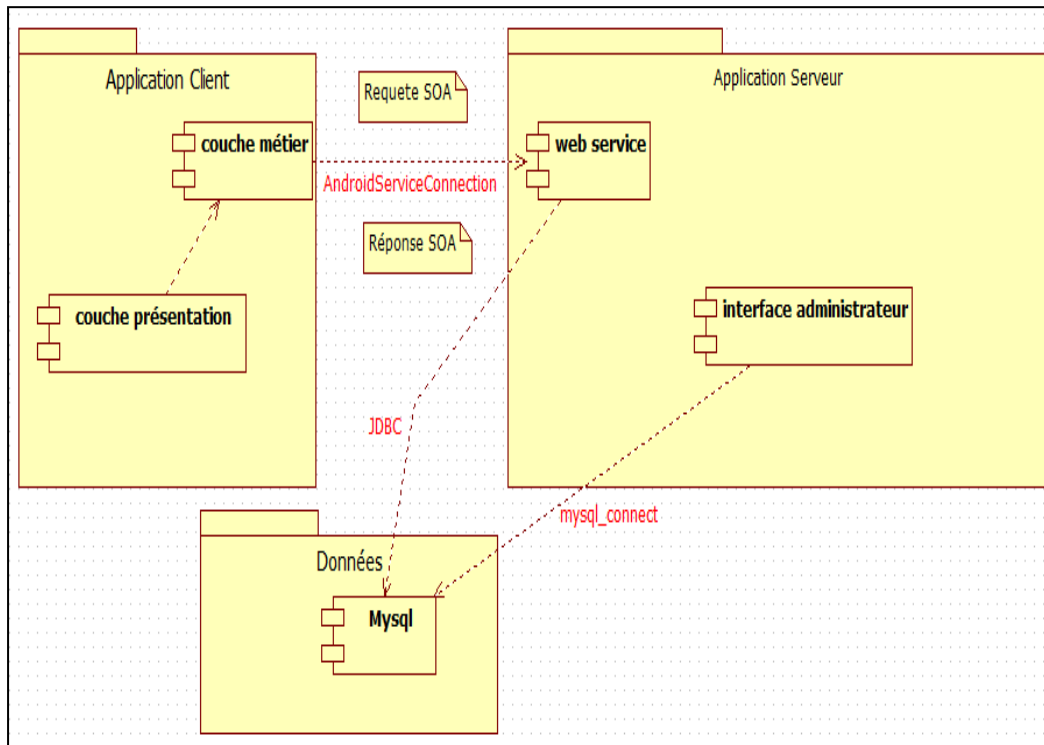


Figure 14:évidence l'architecture général de l'application et les dépendances entre les divers composants

Conclusion

À la fin de ce chapitre, on a terminé le processus en Y qui est composé d'une étude fonctionnelle et une étude technique.

Toutes les informations acquises après ces deux études vont être exploitées pour les prochains chapitres.

Chapitre 5 : la conception

Introduction

La conception est la première phase après les études fonctionnelles et techniques.

Dans le cadre de ce chapitre nous nous intéressons à la conception des cas d'utilisation que nous venons d'analyser dans les études précédents.

I Les diagrammes de classe:

Le diagramme de classes met en œuvre un ensemble de classes dotées des attributs et des opérations dont chacune représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques.

I.1 Le diagramme de classe en couche :

La figure suivante représente les différentes classes existantes dans chacune de trois couches de l'application :

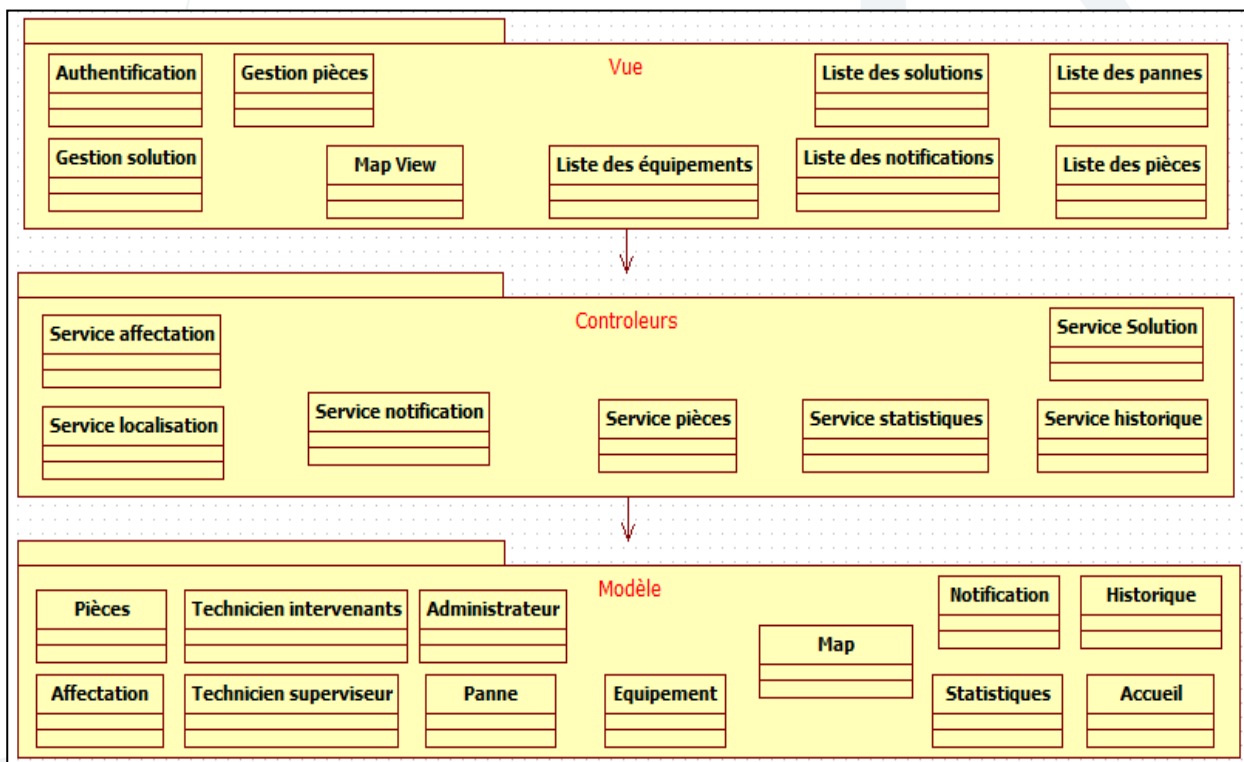


Figure 15:Diagramme de classe en couche de l'application

I.2 Le diagramme de classe de la couche présentation :

Nous trouvons ci-joint le diagramme de la classe de la couche présentation qui met l'accent sur l'interface d'affichage et d'interaction du client avec les données.

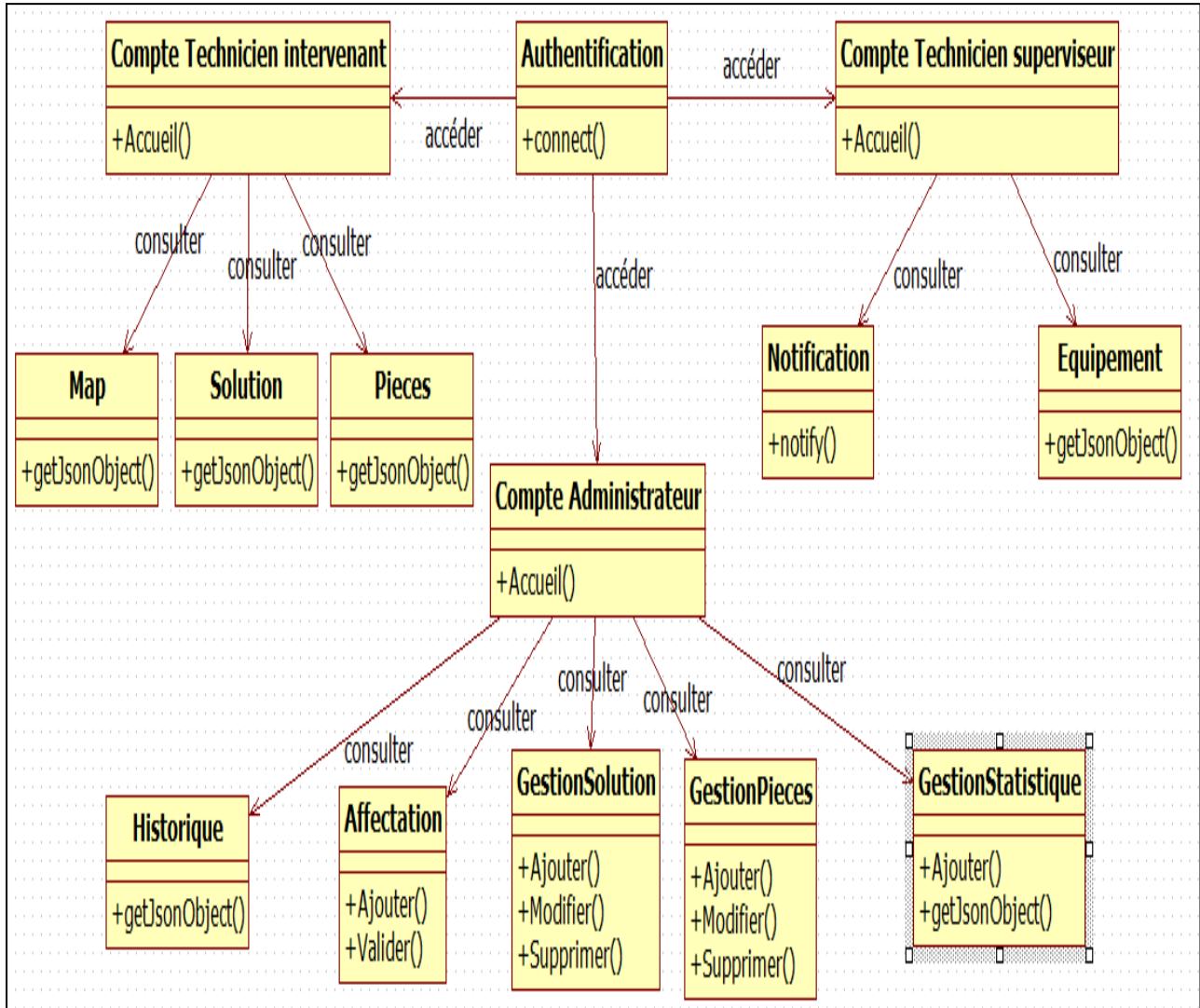


Figure 16:Diagramme de classe de la couche présentation

I.3 Le diagramme de classe de la couche logique applicative :

La figure suivante représente le diagramme de la couche logique applicative qui met l'accent sur les contrôleurs assurant la coopération entre les vues et le modèle.

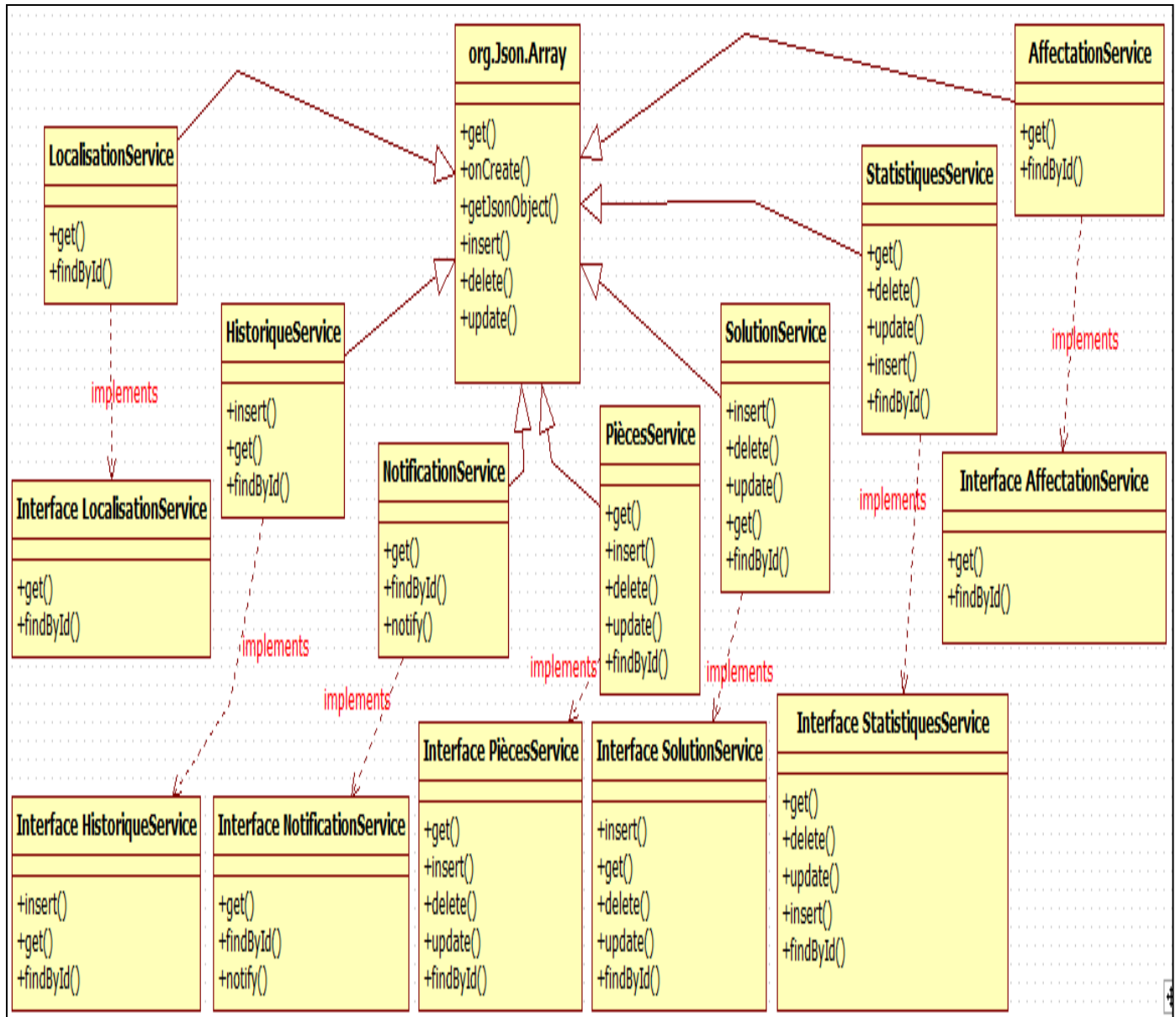


Figure 17:Diagramme de classe de la couche logique applicative

I.4 Le diagramme de classe de la couche modèle

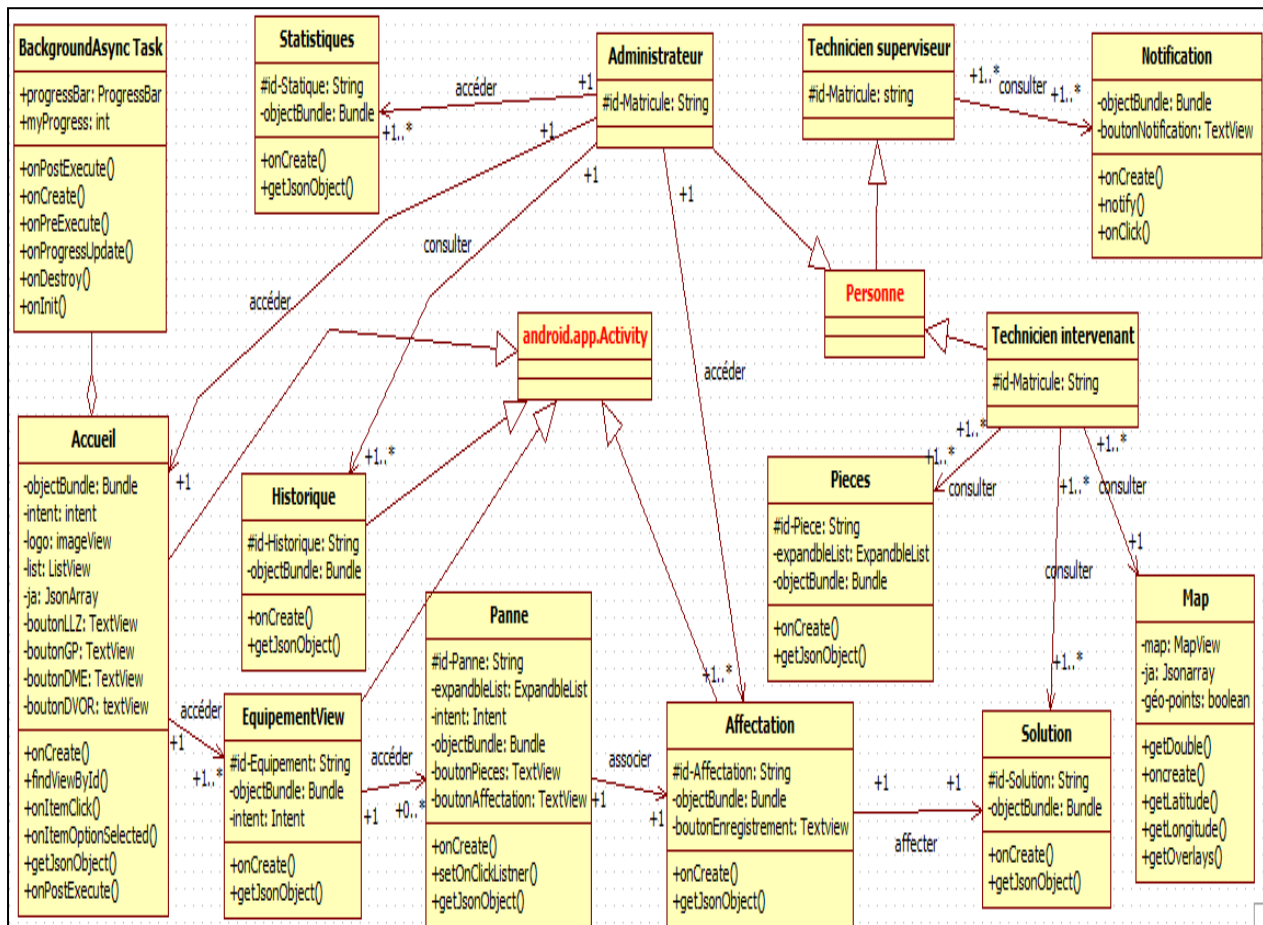


Figure 18:Diagramme de classe de la couche modèle

Conclusion :

Nous avons détaillé dans ce chapitre la conception de notre application tout en décrivant l'architecture de notre projet et détaillant les divers diagrammes de classe ce qui va nous permettre d'entamer la phase de la réalisation du projet.

Chapitre 6: Implémentation

Introduction :

Après la conception, nous allons entamer dans ce chapitre la dernière phase dans le cycle de vie d'un projet, la phase de développement. Ce chapitre sera consacré en premier lieu à la présentation de l'environnement matériel et logiciel avant de décrire le principe de fonctionnement via des captures d'écran.

I. Aspect technique

I.1. Choix de la plateforme



I.1.1. La plate-forme Androïde

Pour choisir la plateforme de développement adéquate, nous recourons à une étude faite en 2010 sur les plateformes mobiles.

Cette étude prouve que le choix des développeurs tourne vers l'iPhone (73%), suivi de l'IPad (35%), d'Androïde (28%) et de RIM (17%). De plus, l'étude montre que l'IPad et Androïde possèdent les meilleures évolutions avec respectivement 35 et 18 points en plus par rapport à 2009. Nous choisissons alors la plateforme Androïde du fait qu'elle est une plateforme puissante, moderne, sûre et ouverte. D'autre part, il est basé sur le noyau Linux, ce qui offre plus d'avantages tels que la grande mémoire, la gestion de processus, le modèle de sécurité, le soutien des bibliothèques partagées, etc. Androïde intègre aussi un émulateur gratuit, qui reproduit à l'identique le comportement du téléphone, à l'inverse de celui de l'iPhone. Elle est un système d'exploitation open source pour les Smartphones, développé par Google. Son système d'exploitation (OS) est basé sur le libre Linux kernel ce qui offre aux développeurs la possibilité de modifier et de personnaliser le système d'exploitation pour chaque téléphone. Par conséquent, différents téléphones basés sur Androïde peuvent avoir différentes interfaces utilisateur graphiques GUI, même si elles utilisent le même OS. Un appareil Androïde va offrir un niveau élevé de personnalisation et un monde des Smartphones incroyablement flexible avec une certaine souplesse du coût.

II. Environnement du travail :

II.1 Environnement matériel :

Le matériel utilisé pour la réalisation de l'application consiste en :

Tableau 4: Environnement matériels

- Un poste pour le développement supportant les caractéristiques suivantes :

Fabricant	Lenovo H530
RAM	4 GO
Processeur	Intel(R) Core (TM) i3.
OS	Windows 7 professionnel
Stockage	320 GO

- Un Smartphone avec les propriétés suivantes :

Fabricant	Evertex evermiracle nano
Mémoire interne	1 Go RAM
Processeur	Quad Core 1.3 Ghz
OS	Android 4.4 KitKat
Stockage	8Gb

- Un serveur

Fabricant	HP pro 3100
Mémoire interne	4TO RAM
Processeur	Quad Core 1.3 Ghz
OS	Windows server8
Stockage	32Gb

II.2 Environnement logiciel :

Pour réaliser ce concept, nous avons eu recours à quelques outils de conception, d'infographie et de programmation :

II.2.1 Outils d'infographie :

➤ Draw 9 patch :

Un graphique 9-patch est une image bitmap extensible, qui Android redimensionne automatiquement pour s'adapter au contenu de la vue dans laquelle elle est placée comme arrière-plan. Un exemple d'utilisation d'un 9-patch est les milieux utilisés par les boutons standards Android - boutons doivent s'étirer pour accueillir des chaînes de longueurs différentes. Un désignable 9-patch est une image PNG standard qui comprend une frontière supplémentaire de 1 pixel de large. Il doit être sauvegardé avec l'extension 0.9. Png.

II.2.2 Outils de conception :

➤ Star UML :

Nous choisissons UML puisqu'il est un support de communication performant. Il offre la possibilité d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation, aussi c'est un langage visuel, sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions. UML permet non seulement de représenter et de manipuler les concepts objet, mais aussi il permet de concevoir une solution objet de manière itérative, grâce aux diagrammes, qui supportent l'abstraction.

I.2.3 Outils de développement :

➤ Android Studio :

- Eclipse » était l'outil de développement le plus utilisé pour développer des applications Android jusqu'à mai 2013.
- Cet outil était interfacé avec le plug-in « ADT » (Android Development Tool) pour permettre un tel développement.
- Ce plug-in était disponible aussi pour les outils de développement « Netbeans » et « IntelliJ ».
- Google a annoncé, le 16 mai 2013, lors de la conférence « Google I/O », un nouvel outil de développement pour des applications Android, « Android Studio ».

- Android Studio est basé sur la version open-source (Community Edition) de « IntelliJIDEA » développée par « JetBrains ».
- La première version stable (1.0) de cet outil était disponible en décembre 2014.

Tableau 5 :Android Studio versus Eclipse

	ADT(éclipse)	Androïde studio
Facilité d'installation	moyen	simple
Langue	Nombreuses	Anglais
Performance	Peut être lourd	rapide
Éditeur d'interface graphique	oui	oui

➤ **XAMPP :**



SQL c'est bien mais:

- Peu de gens connaissent le SQL
- Toute insertion doit être faite manuellement
- Nécessité d'accéder au serveur SQL à distance

Solution possible: une interface web permettant d'interagir avec la base de données

XAMPP est une des solutions les plus utilisées

XAMPP est un ensemble de logiciels permettant de facilement créer une interface web interagissant avec une base de données SQL

X pour cross-plateforme (LAMPP pour Linux, WAMPP pour Windows,...)

A pour Apache

M pour MySQL

P pour PHP

P pour Perl

Le package open source XAMPP a été mis en place pour être incroyablement facile à installer et à utiliser.

➤ **Androïde SDK :**

C'est un kit de développement gratuit qui intègre une grande variété d'outils destinés à développer des applications, à dessiner des interfaces utilisateurs et à rechercher les bugs. Ce SDK propose aussi un plug-in pour l'IDE Java Eclipse qui aidera à générer les ressources XML (comme les animations ou les menus).

➤ **AVD 2.3.3 (Android Virtual Devices) :**

C'est un émulateur de tests qui offre pour chaque produit un espace de stockage propre, qui permet de réaliser des tests sur les fonctionnalités de l'application.

Langages :

➤ **Androïde :**



Androïde est un système d'exploitation open source utilisant le noyau Linux, pour Smartphone, tactiles, PDA et terminaux mobiles conçu par Androïde.

API :

➤ **JSON :**

JSON (JavaScript Object Notation) est un format de données léger, facile à lire et à écrire et compatible avec pas mal de langages de développement. Sa structure est composée d'objets et de tableaux. Sa flexibilité fait de JSON le parfait candidat pour retourner des données.

➤ **JFree Chart :**

JFreeChart est une API Java permettant de créer des graphiques et des diagrammes de très bonne qualité. Cette API est open source et sous licence LGPL. En revanche, la documentation est payante.

I.2.4 Configuration pré-développement :

Pour assurer le développement, une préparation de l'environnement de travail est indispensable. Le développement mobile de ce projet est sur la plateforme Eclipse donc pour la préparer il faut assurer ce qui suit :

- Installation Android studio .
- Installation SDK Android.
- Intégration SDK Android pour pouvoir créer des projets Android.
- Installation du serveur XAMPP.

II Description matérielle :

Nous allons mettre en évidence dans cette partie la description matérielle du système tout en traitant son architecture et analysant ses tâches assurées.

Nous avons traité dans ce projet une architecture client-serveur où l'appareil Android est le client et le serveur est le couple PHP/MySQL. En effet, pour réaliser la connexion à distance et à partir de mon appareil Android à ma base de données MySQL, nous avons eu recours à des PHP scripts et au protocole http du système Android pour récupérer des données depuis le serveur. Lorsque l'application Android va s'exécuter, il se connectera au script PHP qui va à son tour récupérer les données depuis la base de données MySQL. Ensuite les données seront encodées au format JSON et envoyées au système Android. Enfin, notre application Android va obtenir ces données codées il les analysera et les affichera sur l'appareil.



Figure 19: Description matérielle de l'application

- Pour le code PHP, il sera très simple à exploiter, en effet, il suffit de se connecter à la base de données, exécuter une requête SQL selon le contexte et en fonction des données des valeurs POST/GET et enfin l'envoi va être en format JSON
- Au niveau notre application Android on va assister à l'utilisation du protocole HttpPost pour obtenir les données avant de les convertir en chaîne et enfin analyser ces données JSON et les utiliser selon mon contexte.

II.1 Implémentation

Présentant alors les diverses interfaces de chaque module ou chaque utilisateur de ces derniers doit s'authentifier pour qu'il puisse accéder à l'application.

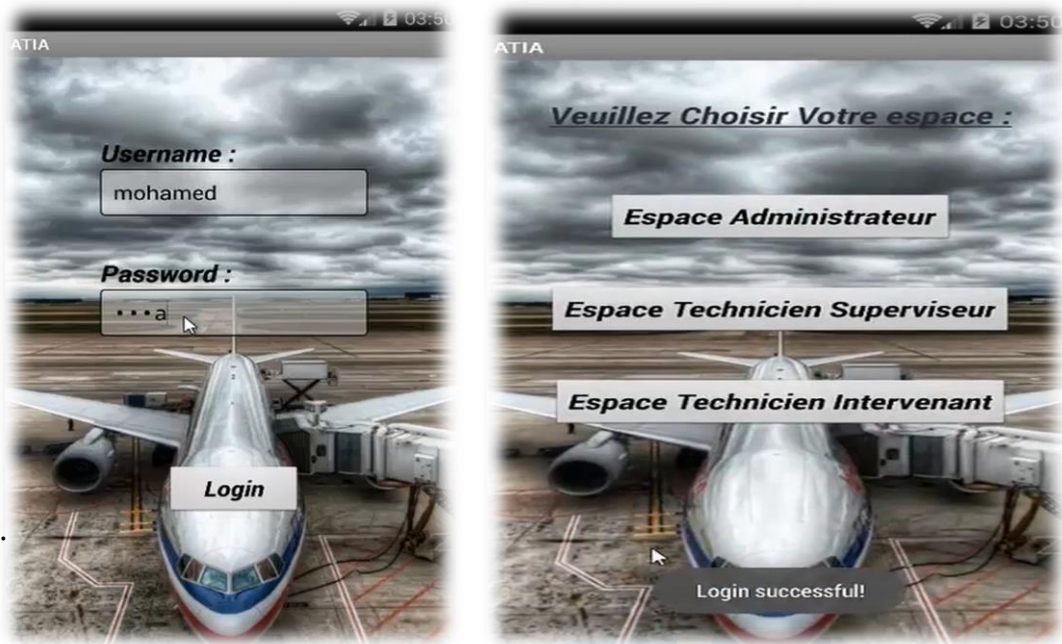


Figure 20:Interfaces de démarrage et d'authentification

I.1.2 Interfaces du module administrateur :

L'administrateur pilote est en fait tout le mécanisme de la maintenance. En effet, il contrôle le stock des pièces de rechange tout en ajoutant, supprimant ou mettant à jour les listes des cartes et modules de rechange sauvegardées dans la base.



Figure 21:Interfaces module administrateur

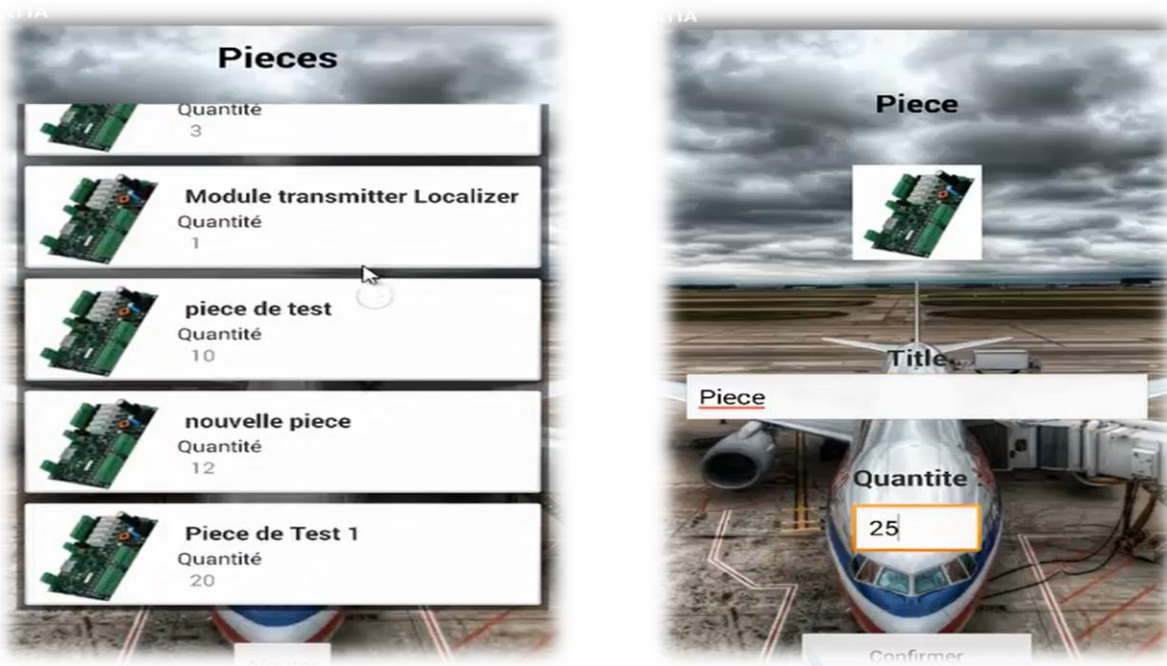


Figure 22: Interfaces d'ajout des pièces

De plus, l'administrateur organise l'opération de l'intervention de point de vue gestion des pannes par l'ajout de toute panne détectée ou des scénarios des pannes qui peuvent être réalisée et ça ce qui montre la figure ci-joint :

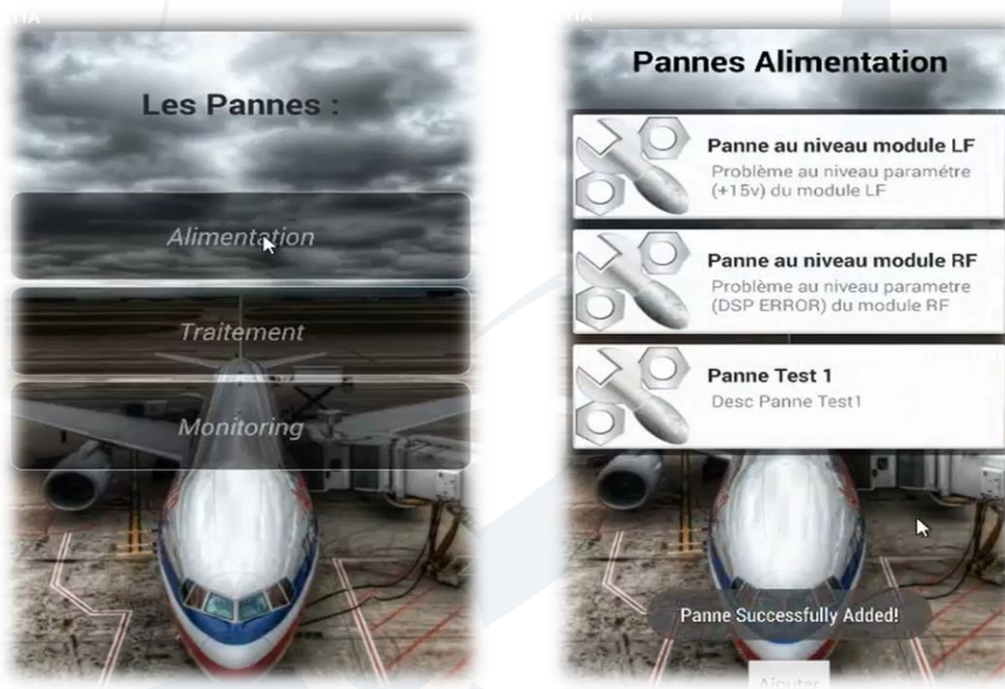


Figure 23: Interfaces d'ajout des pannes

Pour minimiser le temps de réponse afin d'augmenter l'efficacité de l'intervention, une indexation entre ces pannes et leurs solutions adéquates est mis en évidence, en effet, l'administrateur assure l'ajout, la suppression et la mise à jour des solutions des pannes convenables et qui répond aux besoins des techniciens lors de l'intervention. Les figures suivantes mis en œuvre la gestion des solutions des pannes.



Figure 24: Interfaces d'indexer les solutions

I.1.3 Interfaces du module maintenance corrective :

Le technicien intervenant s'occupe de la maintenance corrective. En effet, suite à une panne et afin de garantir l'efficacité de l'intervention, notre technicien doit avoir une idée claire sur le stock de rechange et cette application lui permet de consulter la liste des pièces et des cartes de rechange bien classées sous la catégorie des pièces voulue. La figure suivante montre l'interface de consultation des pièces de rechange :



Figure 25:Interface de consultation des pièces de rechange

Une fois qu'il a eu une idée claire sur l'état du stock, l'intervenant peut consulter la solution adéquate indexée à une panne sélectionnée et touchant une catégorie bien déterminée dans un temps limité d'où l'augmentation de l'efficacité de l'intervention. La figure suivante montre les interfaces de consultation des pannes ainsi que leurs solutions convenables :

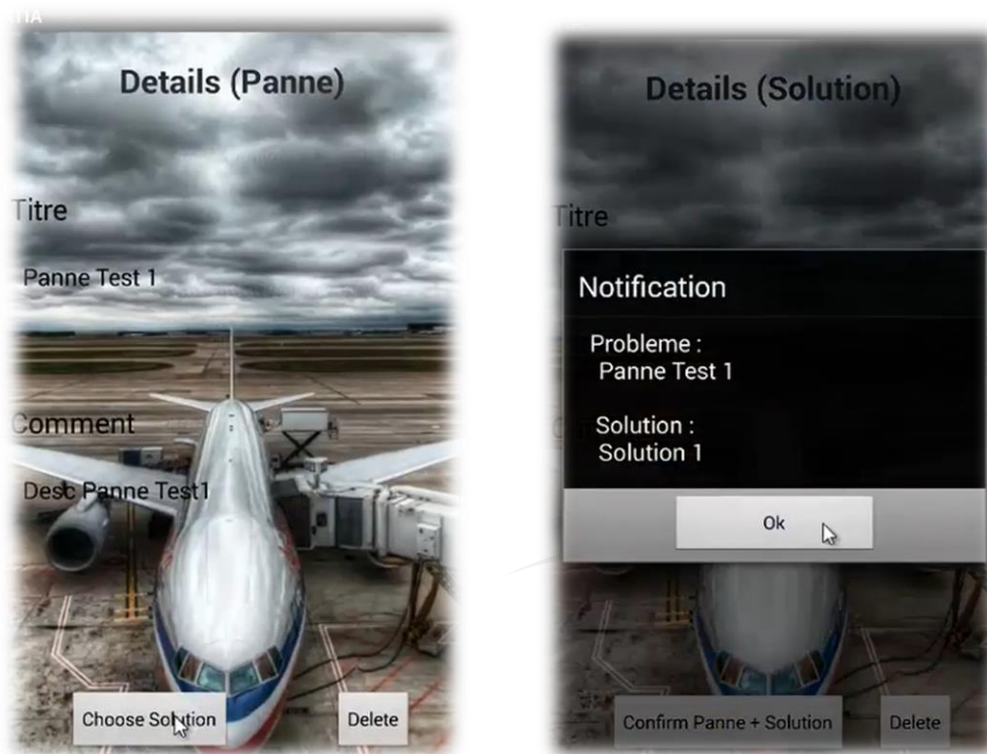


Figure 26: Interfaces de consultation des pannes et leurs solutions

Conclusion :

Nous avons mis en œuvre dans ce chapitre l'environnement du travail de notre projet : Côté matériel ainsi que logiciel et nous avons explicité son principe de fonctionnement par L'intermédiaire des captures d'écran.

Conclusion générale

Tout au long de ce projet, nous avons mis l'accent sur notre solution proposée pour améliorer les performances de la maintenance au sein du service technique dans l'aéroport Tabarka Ain Drahem et qui consiste à réaliser une application assurant la supervision de la maintenance des équipements des aides à la navigation aérienne à distance par la plateforme Androïde.

Alors pour ce faire, nous avons eu recours en premier lieu à une étude de l'existant pour dégager les insuffisances et les lacunes dans le service de la maintenance avant de passer à une étude préliminaire illustrant les divers besoins fonctionnels et non fonctionnels en s'appuyant sur le processus de développement unifié 2TUP ce qui nous a ramené à établir une conception détaillée et affinée basée sur UML comme un support de modélisation pour garantir une compréhension facile du système d'où l'implémentation et la réalisation adéquate du projet et donc la création d'une application répondant aux besoins fixés d'avance.

Ce projet est conçue pour être un système centralisé c'est-à-dire un système qui englobe tous les aéroports en Tunisie en organisant la maintenance des équipements des aides à la navigation aérienne à l'échelle national.

Bibliographie

- [B1] « Document 8071 Manuel sur la vérification des aides de radionavigation » Edition novembre 2010
- [B2] « Annexe 10/ Volume 5 télécommunication aéronautique deuxième édition juillet 2001» de l'OACI
- [B3] « Manuel Technique des aides à la navigation aérienne Normarc 7011B-12B »
- [B4] « Direction Générale de l'aviation civile Revue du service technique de la navigation aérienne N°59 Décembre 2000»

Webographie

- ✓ <http://www.enac.fr>
- ✓ <http://www.oaca.nat.tn>
- ✓ <http://www.atc-network.com>
- ✓ <http://stackoverflow.com/questions/tagged/android>
- ✓ <http://developer.android.com>
- ✓ <http://fr.wikipedia.org>
- ✓ <https://www.youtube.com/watch?v=kB7ioLMadBE>
- ✓ <https://www.youtube.com/watch?v=LfM1sNHfob8>
- ✓ https://www.youtube.com/watch?v=aFc56_1WQy4

Rapport-gratuit.com 
LE NUMERO 1 MONDIAL DU MÉMOIRES