

UNIVERSITE SIDI MOHAMMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES



PROJET DE FIN D'ETUDES
Licence Sciences et Techniques Génie Informatique

Réalisation d'une application de gestion de parc informatique



Lieu de stage : Inventis

Réalisé par :
Larfaoui Ghizlane

Encadré par :
Pr. Lamrini Loubna

Soutenu le 15/06/2015 devant le jury composé de :

Pr Loubna Lamrini
Pr Azzedine Zahi
Pr Rachid Benabbou

[Année Universitaire 2014-2015](#)

Dédicace

A mes chers parents

Pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon instruction et mon bien-être.

A travers vos encouragements et vos critiques que je me suis réalisé. Que dieu tout puissant vous garde et vous procure santé, bonheur et longue vie.

A mes frères

A toute la famille

A tous mes amis

Remerciements

A terme de ce projet, je tiens à exprimer ma profonde gratitude et mon immense respect à tout nos professeurs pour leur générosité en matière de formation, et nos encadrants Mr Zahi et Mme Lamrini, pour leur disponibilité, leurs avis éclairés et leurs judicieux conseils.

Je tiens à remercier Mr le directeur d'Inventis ainsi toute l'équipe pour leur accueil chaleureux, leurs conseils et leur aide.

Enfin je témoigne ici à tous les membres du jury, toute ma reconnaissance et le respect que j'ai pour eux d'avoir accepté d'évaluer mon travail.

Sommaire

Liste des figures	6
Abréviations	7
Introduction générale.....	8
Chapitre1 : Contexte du projet	9
1. Présentation de l'organisme d'accueil.....	10
1.1. Présentation d'Inventis.....	10
2. Présentation du projet.....	10
2.1. Cahier de charge	11
2.2. Problématique.....	11
2.3. Etude de l'existant.....	11
2.3.1. Clarilog.....	11
2.3.2. H-Inventory.....	12
2.4. Critique de l'existant.....	12
2.5. Solution proposée.....	12
3. Conclusion	12
Chapitre 2 : Modélisation et conception de la solution proposée	13
1. Introduction.....	14
2. Méthodologie d'analyse.....	14
2.1. le langage UML.....	14
2.2. le modèle incrémental et itératif.....	14
2.2.1. les avantages du modèle incrémental et itératif.....	15
2.2.2. les inconvénients modèle incrémental et itératif.....	15
2.2.3. les incréments principaux du projet.....	16
2.3. Le modèle MVC.....	16
2.3.1. Définition	16
2.3.1.1. Le modèle.....	16
2.3.1.2. La contrôleur.....	17
2.3.1.3. La vue	17
2.3.2. Avantages du MVC.....	17



3.	Modélisation du contexte.....	17
3.1.	Description des acteurs.....	17
3.2.	Branche fonctionnelle.....	19
3.2.1.	Besoins fonctionnels.....	19
3.2.2.	Besoins non fonctionnels.....	20
4.	Analyse et conception.....	21
4.1.	Diagramme de package.....	21
4.2.	Diagramme de cas d'utilisation.....	21
4.3.	Diagramme de séquence.....	25
4.4.	Diagramme de classe.....	26
5.	Conclusion.....	27
Chapitre 3 : Réalisation		28
1.	Introduction.....	29
2.	Outils et technologies de développement.....	29
3.	Présentation de l'application.....	31
3.1.	Schéma principal de l'application.....	31
3.2.	Authentification.....	32
3.3.	Gestion de tickets.....	34
3.4.	Fonctionnalités générales.....	36
4.	Conclusion.....	37
Conclusion et perspectives.....		38
Bibliographe et Webographie.....		39

Liste des figures

Figure 1: Organigramme Inventis.....	10
Figure 2: L'application Clarilog	11
Figure 3 :L'application H-Inventory.....	12
Figure 4 : Cycle de vie.....	14
Figure 5: Les incréments du Modèle Incrémental et Itératif	15
Figure 6: Architecture du modèle MVC	16
Figure 7: Acteurs de l'application	18
Figure 8:Interaction des acteurs avec le système.....	18
Figure 9: Diagramme de package	21
Figure 10: Diagramme des cas d'utilisation Administrateur.....	22
Figure 11: Diagramme des cas d'utilisation Utilisateur.....	22
Figure 12: Diagramme des cas d'utilisation Technicien.....	23
Figure 13: Le cas d'utilisation »gestion des tickets » pour l'utilisateur.....	23
Figure 14: Le cas d'utilisation »gestion des tickets » pour l'administrateur.....	24
Figure 15: Le cas d'utilisation »gestion des tickets » pour le technicien.....	24
Figure 16: Diagramme de séquences authentification	25
Figure 17: Diagramme de séquences ajouter poste	26
Figure 18: Modèle Conceptuel de Donnée.....	26
Figure 19: Schéma général de l'application.....	32
Figure 20: Fenêtre d'Authentification	33
Figure 21: Fenêtre d'authentification lors d'une connexion erronée	33
Figure 22: Fenêtre d'ajout d'un ticket	34
Figure 23: Alertes	34
Figure 24: Table des tickets chez l'administrateur	35
Figure 25: Fenetre affectation du technicien.....	35
Figure 26: Table des tickets chez le technicien.....	35
Figure 27: Fenetre traitement du ticket	36
Figure 28: Agenda	36
Figure 29: E-mail	37



Abréviations

GPinfo	: Gestion de Parc Informatique
UML	: Langage Unifié pour la Modélisation objet.
PHP	: Hypertext Preprocessor.
MVC	: Modèle Vue Contrôleur.
JS	: JavaScript.
HTML	: Hypertext Markup Language
CMS	: Content Management System
E-Commerce	: Electronic Commerce
GNU	: GNU's Not UNIX
GPL	: General Public License



Introduction générale

Ce document a pour objectif de présenter le travail réalisé lors de mon stage qui s'inscrit dans le cadre du projet de fin d'étude effectué au sein de l'agence web Inventis, en vue d'obtention du diplôme Licence de la Faculté des sciences et techniques Fès.

Ce projet a pour but de réaliser une application web pour la gestion du parc informatique avec une interface simple à manipuler.

En effet, la période du stage est une étape très importante dans le processus de formation, qui enrichit les connaissances et surtout qui aide à découvrir de plus près la vie professionnelle.

Durant mon projet, on aurait pour mission dans un premier temps de cerner le sujet. Après une analyse approfondie de la problématique, on a élaboré les différents diagrammes. Ensuite, on a abordé la phase de la mise en œuvre et de l'implémentation de la solution. La dernière étape a fait l'objet du déploiement des tests et de validation. Pour bien mener mon projet, on a choisi de suivre un cycle de développement Modèle Incrémental et Itératif. C'est une démarche qui a fait ses preuves dans le domaine des projets informatiques de grande taille.

Durant une période de stage allant du 15 Avril au 15 Juin 2015, on a élaboré 3 grandes parties. La première partie définit le contexte général du projet en présentant l'organigramme d'accueil, en définissant la problématique du projet, l'étude et critique de l'existant ainsi la solution proposée. Dans la deuxième partie, on présente la modélisation et conception de la solution proposée en décrivant les fonctionnalités du système ainsi les différents diagrammes d'UML^[1]. La troisième partie sera consacrée aux outils et langages de développement utilisés, à la réalisation du projet et la présentation de l'application. Enfin une conclusion et des perspectives du travail seront citées.



Chapitre 1 : Contexte du projet

1. Présentation de l'organisme d'accueil

1.1. Présentation d'Inventis

Inventis [1] est l'une des agences web leaders au Maroc dans la ville de Fès, créée depuis 7 ans.

C'est une agence web au Maroc qui crée des sites internet, le référencement web, l'e-mailing, l'hébergement, le mobile et le multimédia

Parmi les outils qu'emploient l'équipe de cette agence :

- Design et conception graphique (Photoshop, Illustrator, Indesign),
- CMS (Joomla, WordPress, Drupal)
- E-Commerce: (Prestashop, Magento)
- Frameworks (Zend, Symfony)
- Mobile : PhoneGap, iOS, Android.

1.2. Organigramme d'Inventis

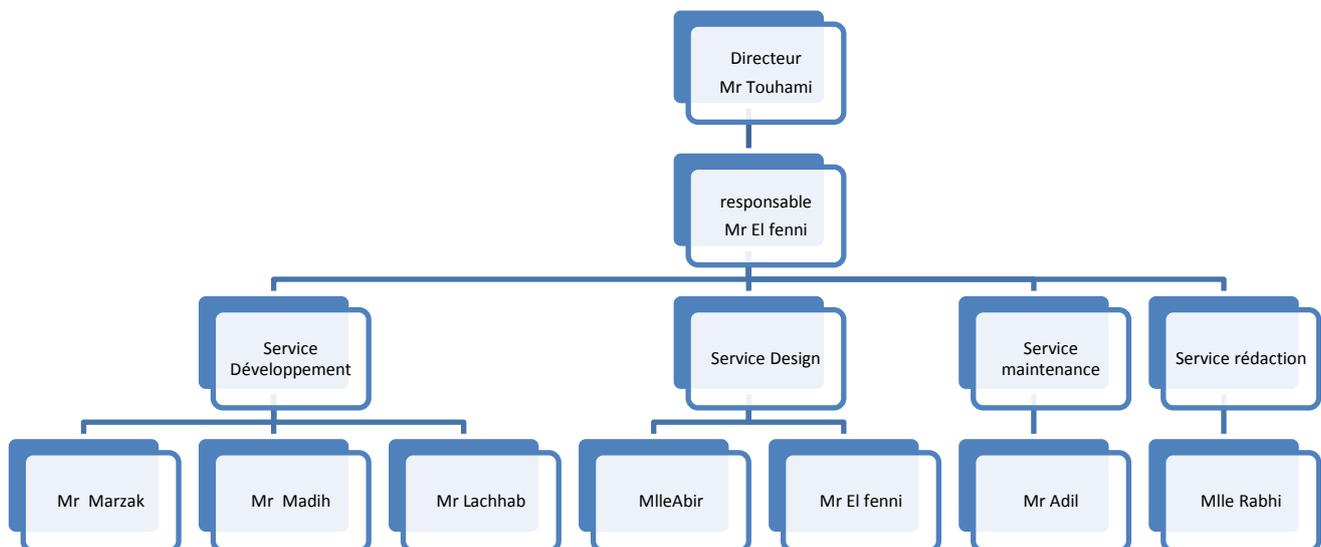


Figure 1: Organigramme d'Inventis

2. Présentation du projet

2.1. Cahier de charge



Un poste est caractérisé par son numéro d'inventaire, son adresse réseau (adresse IP), la date de son acquisition. Sur chaque poste est installé un ensemble de logiciels et matériels. Chaque poste peut être utilisé par plusieurs utilisateurs, Un utilisateur peut occuper plus qu'un poste.

Lors d'une panne, l'utilisateur peut créer un ticket défini par un sujet : matériel , logiciel ou réseau ,une description et un niveau d'urgence . Les tickets sont gérés par l'administrateur qui affecte un technicien selon son spécialité pour le traitement du ticket.

A la création d'un ticket, il prend le statut Nouveau. Lorsqu'un technicien est chargé de le traiter, il passe alors à l'état En attente (attribué).. Quand une solution est apportée au ticket, alors il prend le statut En attente (planifié). Enfin, quand le technicien valide la solution proposée, alors il change l'état en résolu. Le technicien peut changer le statut à tout moment, notamment pour mettre le ticket En attente.

2.2. Problématique

Avant de plonger dans la solution, il est indispensable de prendre du recul et de faire un résumé de quelques problèmes concrets que rencontrent au jour le jour nos différents acteurs.

En tête de liste, le manque de traçabilité et de suivi est le problème le plus récurrent chez la plupart des administrateurs .en effet, le nombre croissant d'équipements et l'hétérogénéité du parc ne permettent pas à ceux-ci de maîtriser tous les systèmes, logiciels et matériaux installés.

Ainsi que le contrôle et le suivi des opérations techniques dans les entreprises se font mais pas de manière automatisée et sécurisée.

L'application proposée devra donc être à mesure d'apporter une solution concrète à la prise en charge des différents problèmes ci-dessus.

2.3. Etude de l'existant

Parmi les produits existants sur marché, il y'a :

2.3.1. Clarilog

Cette application a été créée par l'entreprise **Clarilog** France et permet entre autre l'audit du parc informatique en utilisant le module clarilog Fast Inventory permet de récolter les données sans déploiement d'agent, permet aussi une cartographie complète des équipements du parc.



Figure 2: Application Clarilog

qui



2.3.2. H-inventory

Sous licence GNU GPL, H-inventory propose
Les fonctionnalités suivantes :

- inventories les machines d'un parc informatique :
- gérer les incidents (HelpDesk)
- Faire un audit réseau (scan map)
- déployer automatiquement des applications Windows et linux
- effectuer du monitoring sur les services (alertes, mail...)

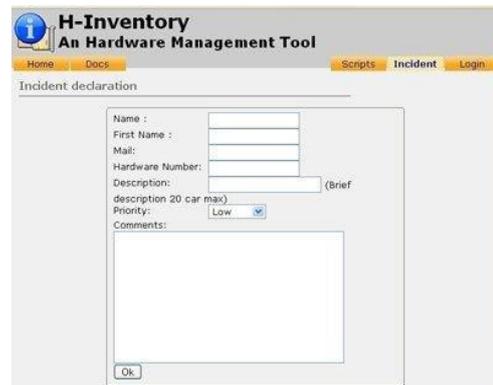


Figure 3: Application H-inventory

2.4. Critique de l'existant

Une analyse des solutions existantes montre que la plupart de ces applications offrent des fonctionnalités de base de gestion d'un parc informatique à savoir l'inventaire, l'accès au Helpdesk et le scan du réseau. Au regard de ces informations, nous pouvons relever qu'elles répondent au besoin principal des utilisateurs, Néanmoins, nous pouvons aussi noter les inconvénients suivants :

-**Clarilog** n'est pas une application open source

-**H-inventory** n'offre pas une gestion administrative et ne permet pas une traçabilité et un suivi des tâches administratives effectuées.

2.5. Solution proposée

Après une étude comparative sur les différentes solutions existantes, il est donc primordial au regard des inconvénients recensés de proposer une application qui pourra répondre à nos besoins. L'idée générale du projet consiste à concevoir un outil applicatif qui pourra de façon concrète à un utilisateur de circonscrire une demande de service ou une demande d'intervention nommée ticket. L'administrateur utilisera cet outil pour gérer ces demandes d'une part et d'autre part effectuer la supervision en se basant sur l'inventaire du parc. Le technicien sera concerné une fois choisi pour traiter un ticket. On a choisi de travailler avec **GPinfo**. Cet outil est capable de fournir une liste de ressources via un inventaire et permet ainsi à l'utilisateur de maîtriser les équipements de son parc ainsi permet une gestion administrative, une traçabilité et un suivi de tâches administratives effectuées.

3. Conclusion

Dans ce chapitre, on a présenté l'organisme d'accueil et le projet à réaliser. Ainsi on a présenté une idée sur les applications existantes et la solution retenue. Une étude plus approfondie sera présentée dans les prochains chapitres. L'analyse des besoins et la spécification feront l'objet du chapitre suivant.



Chapitre2 :

Modélisation et conception de la solution proposée

1. Introduction

La phase modélisation et conception présente une étape primordiale dans le cycle de développement d'un projet. On présente d'abord la méthodologie d'analyse puis il sera question de présenter les principaux acteurs et leurs rôles. Ensuite dans la branche fonctionnelle, on va définir les besoins fonctionnels et non fonctionnels, puis présenter les différents diagrammes d'UML.

2. Méthodologie d'analyse

2.1. Le langage UML

UML[2] ou Langage de Modélisation Unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML est couramment utilisé dans les projets logiciels. Les différents éléments sont :

- Activité d'un objet/logiciel.
- Acteurs.
- Processus.
- Schéma.
- Composants logiciels.
- Réutilisation de composants.

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie code, par exemple en langage Java, à partir des divers documents réalisés.

2.2. Le Modèle Incrémental et Itératif

La phase d'étude est la partie la plus importante pour tout projet réussi. On s'est basé durant la réalisation de notre application à des normes universelles durant la conception, en particulier le respect des principes du Modèle Incrémental.

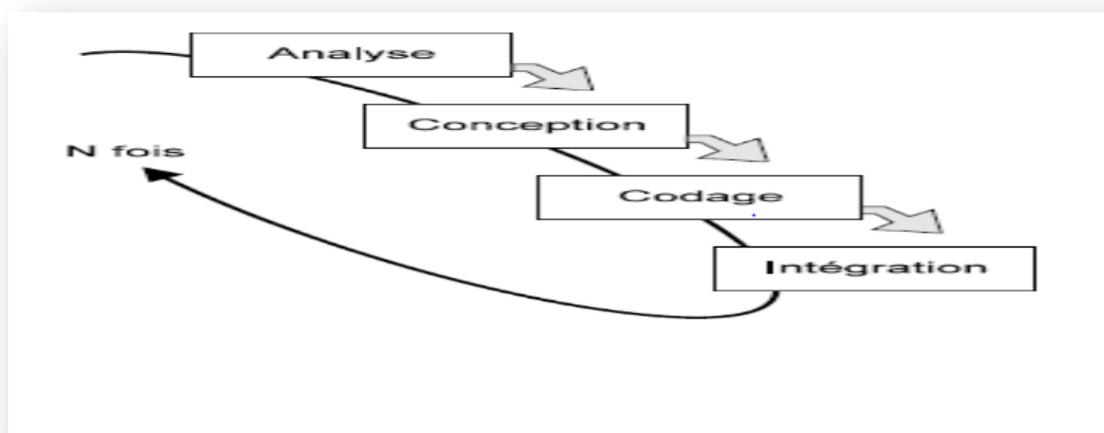


Figure 4: Cycle de vie Modèle Incrémental et Itératif



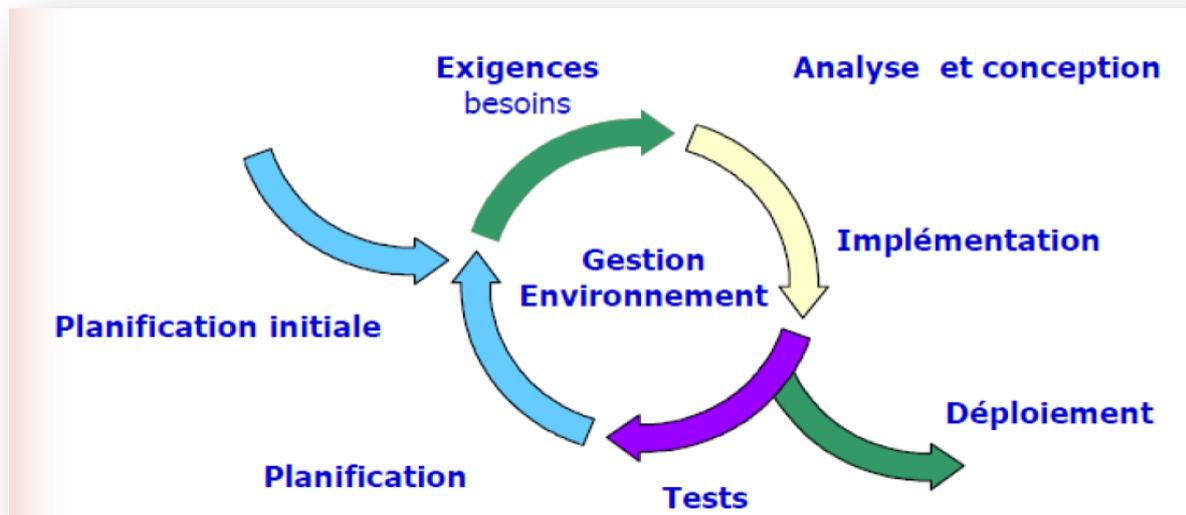


Figure 5: Les incréments du Modèle Incrémental et Itératif

- Le projet de développement est découpé en plusieurs petits projets.
- Chaque projet représente une itération qui:
 - Donne lieu à un incrément (version du produit).
 - Prend en charge une partie des besoins.
 - Répond à un ensemble de risques.
- Le développement se déroule en plusieurs itérations.
- Le projet est décomposé en un noyau et plusieurs incréments.
- Chaque incrément est développé séparément ou en parallèle

2.2.1. Les avantages du Modèle Incrémental et Itératif

- Flexibilité (agilité) vis à vis de nouveaux besoins ou des changements.
- Pas de blocage en cas de spécifications incomplètes.
- Meilleure testabilité.
- Découverte de malentendu assez tôt pour les corriger.
- Répartition de l'effort dans le temps.
- Objectifs réduits et clairs.
- Utilisation de l'approche «diviser pour régner».
- Le client rentre en relation avec le produit très tôt.

2.2.2 Les inconvénients du Modèle Incrémental et Itératif

- Difficultés de gestion du projet.

- Difficultés de contrôle qualité.
- Exigence d'une bonne planification et d'une bonne conception.
- Exigence d'une vision sur le produit fini pour bien diviser en incréments.

2.2.3. Les incréments principaux du projet

Notre projet est constitué des incréments principaux suivants:

- Incrément gestion d'inventaire
 - Gestion des matériels
 - Gestion des logiciels
 - Gestion des périphériques
 - Gestion de réseau
- Incrément gestion d'utilisateurs.
- Incrément gestion de techniciens.
- Incrément gestion de postes.
- Incrément gestion de tickets.

2.3. Le Modèle MVC (Modèle-Vue-Contrôleur)

L'architecture MVC (modèle, vue et contrôleur) [3] est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur), ce qui assure la clarté de l'architecture et simplifie la tâche du développeur responsable de la maintenance et de l'amélioration du projet. Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure suivante.

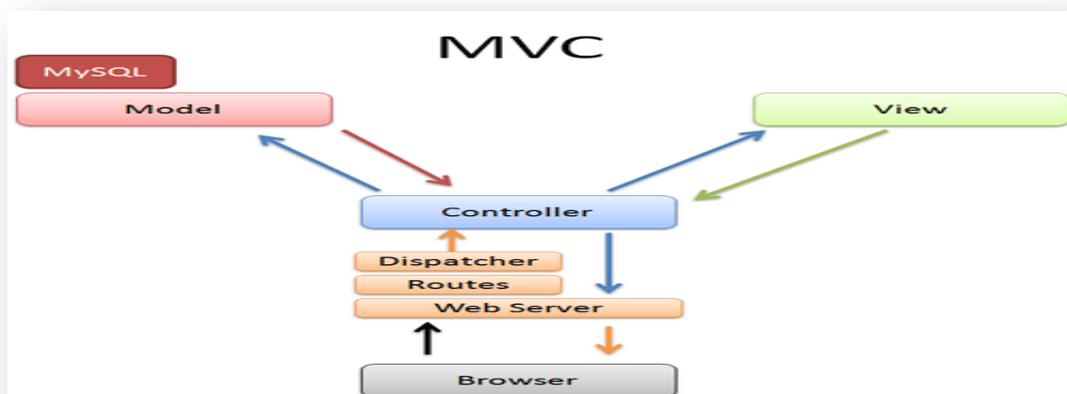


Figure 6: Architecture du modèle MVC

2.3.1. Définition MVC (Modèle-Vue-Contrôleur)

2.3.1.1. Le modèle :



Le modèle représente le cœur de l'application : traitements des données, interactions avec la base de données. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation, Le modèle ne contient aucun lien direct vers la vue.

2.3.1.2. Le contrôleur :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et déclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ce dernier notifie la vue que les données ont changée pour qu'elle se mette à jour. D'après le patron de conception observateur/observable, la vue est un « observateur » du modèle qui est « observable ». Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier. Le contrôleur n'effectue aucun traitement, ne modifie aucune donnée, il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

2.3.1.3. La Vue :

C'est avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle, sa seconde tâche est de recevoir toute action de l'utilisateur (clic de souris, sélection d'un bouton radio, coche d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur.

La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

2.3.2 Avantages du MVC

- Une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur.
- Un gain de temps de maintenance et d'évolution du site.
- Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l'affichage et des actions)

3. Modélisation du contexte

3.1. Description des acteurs

C'est l'étape initiale qui consiste à faire un recensement sur les différents acteurs qui vont interagir de près ou de loin avec le système. Mais avant d'aller plus loin, il est important de définir certains termes importants pour la suite.



- **acteurs** : constitue une entité physique capable d'interagir avec le système .Notons juste que cette entité peut être humaine ou matérielle
- **système** : c'est l'entité à concevoir, dans notre cas il s'agit de l'application en elle-même.

Cette section a pour objet de présenter les acteurs et leurs fonctionnalités à lesquelles doit répondre notre application. Nous commençons notre analyse par identifier les acteurs qui agissent sur ce système à savoir :

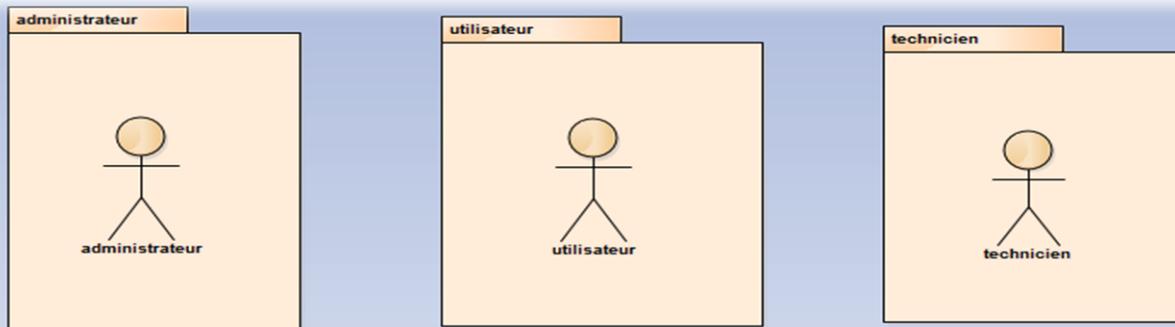


Figure 7: Les acteurs de l'application

- ✚ **L'administrateur** : Joue un rôle primordial dans l'assurance du bon fonctionnement du système. C'est la personne qui prend en charge la gestion de tous les incréments de l'application.
- ✚ **L'utilisateur** : A le droit de visualiser les composants de son poste au niveau du matériel , logiciel et réseau ainsi de créer des tickets et visualiser leurs suivis .
- ✚ **Le technicien** : Prends en charge le traitement des tickets.

La figure ci-dessous illustre parfaitement ces différents acteurs ainsi leurs interactions avec le système qui se représente sous forme de trait reliant les acteurs concernés au système.

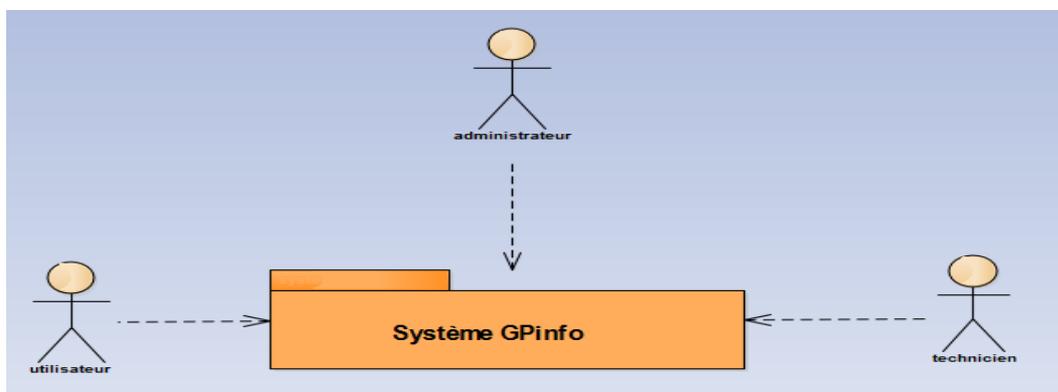


Figure 8: Interaction des acteurs avec le système

Cette représentation est incomplète car le système ainsi représenté constitue une sorte de « boîte



noire » dont on ne connaît en rien le fonctionnement. Il est donc nécessaire de passer par une représentation plus détaillée, c'est ce qui fait l'objet de la partie suivante.

3.2. La branche fonctionnelle

3.2.1. Besoins fonctionnels

Cette étape décrira ce que nous attendons de notre application, puis tout ceci sera modélisé sous forme de diagramme à l'aide du langage de modélisation UML.

ACTEURS	BESOINS FONCTIONNELS
Administrateur	<ul style="list-style-type: none">• Gestion d'inventaire<ul style="list-style-type: none">▪ Gestion de matériels▪ Gestion de logiciels▪ Gestion de réseau• Gestion des utilisateurs• Gestion des tickets<ul style="list-style-type: none">▪ Faire le suivi des tickets▪ Affecter technicien au ticket• Gestion de techniciens• Envoyer un email• Créer un événement dans l'agenda
Utilisateur	<ul style="list-style-type: none">• Gestion des tickets<ul style="list-style-type: none">▪ Créer, modifier un ticket▪ Imprimer, exporter la liste des tickets▪ Visualiser le suivi de ses tickets• Visualiser les composants matériels et logiciels de son poste.• Créer un événement dans l'agenda• Envoyer un email
Technicien	<ul style="list-style-type: none">• Traitement des tickets<ul style="list-style-type: none">▪ Associer des solutions aux tickets▪ Ajouter un suivi aux tickets

- Mentionner la date de traitement du ticket sur machine dans l'agenda
- Envoyer un E-mail

3.2.2. Besoins non fonctionnels

Après avoir déterminé les besoins fonctionnels, nous présentons ci-dessous l'ensemble des contraintes à respecter pour garantir le bon fonctionnement du système. Il est à noter cependant qu'ils peuvent être exprimés en matière de performance, de type de matériel ou le type de conception.

Dans le cadre de notre travail, on peut citer par exemple :

Besoins non fonctionnels	
Performance	Afin d'être acceptée par les utilisateurs l'application doit respecter ce critère tout en assurant un temps de réponse minimum et des fonctionnalités répondant aux besoins de tout utilisateur.
La sécurité	l'application doit assurer la sécurité des utilisateurs. D'où la nécessité de procéder à l'authentification des utilisateurs en assurant la confidentialité de leurs données.
L'extensibilité	la possibilité d'ajouter de nouvelles fonctionnalités ou de modifier celles existantes.
L'ergonomie de l'interface	Les interfaces doivent être simples et conviviales, pour que l'utilisateur puisse l'exploiter sans se référer à des connaissances particulières, Ainsi l'application doit être lisible et facile à manipuler par n'importe quel utilisateur.



4. Analyse et Conception

Nous parvenons à une étape clé du processus. C'est elle qui grâce à l'étude réalisée dans la partie précédente mettra en valeur le rôle de chaque acteur du système ainsi que les fonctionnalités présentées en haut.

4.1. Diagramme de package

C'est un moyen pour regrouper les différents éléments de la modélisation. Il permet de représenter les relations entre les différents profils de l'application. Il rassemble les cas d'utilisations propre à chaque acteur de façon cohérente.

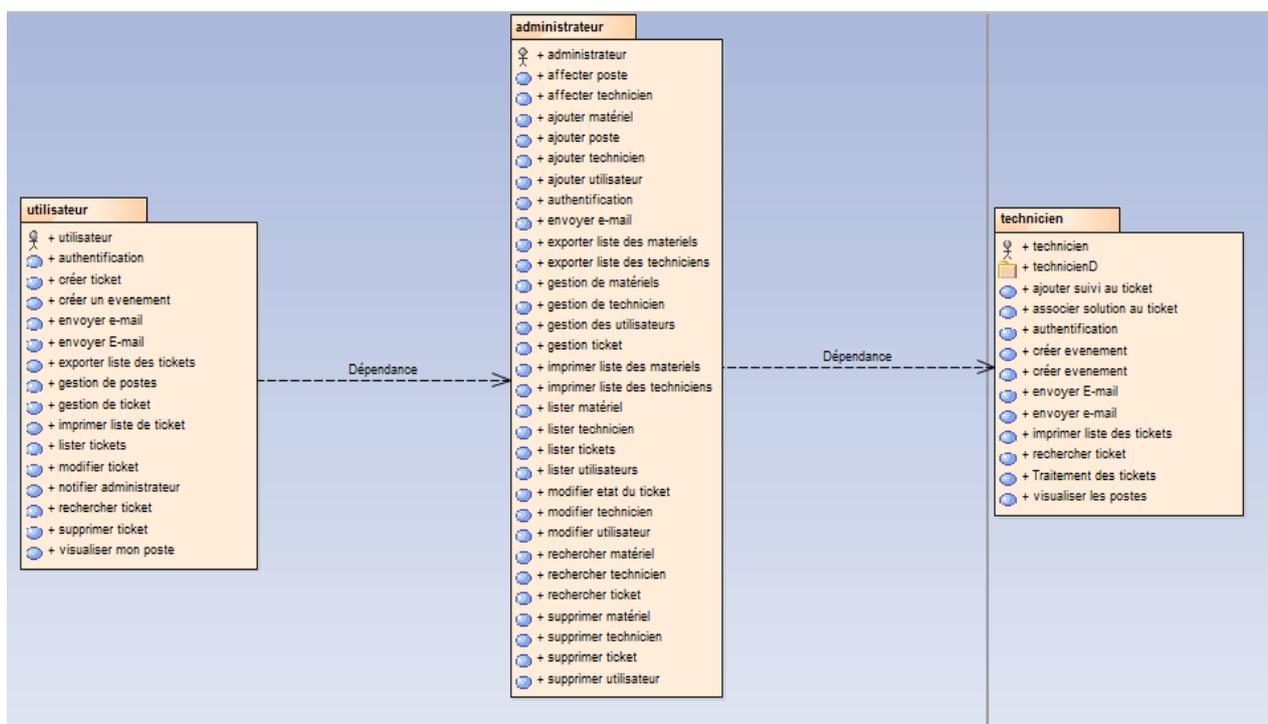


Figure 9: Diagramme de package

4.2. Diagramme de cas d'utilisation

Chaque usage que les acteurs font du système est représenté par un cas d'utilisation. Chaque cas d'utilisation représente une fonctionnalité qui leur est offerte afin de produire le résultat attendu. Ainsi, « le diagramme de cas d'utilisation décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur ».

Administrateur



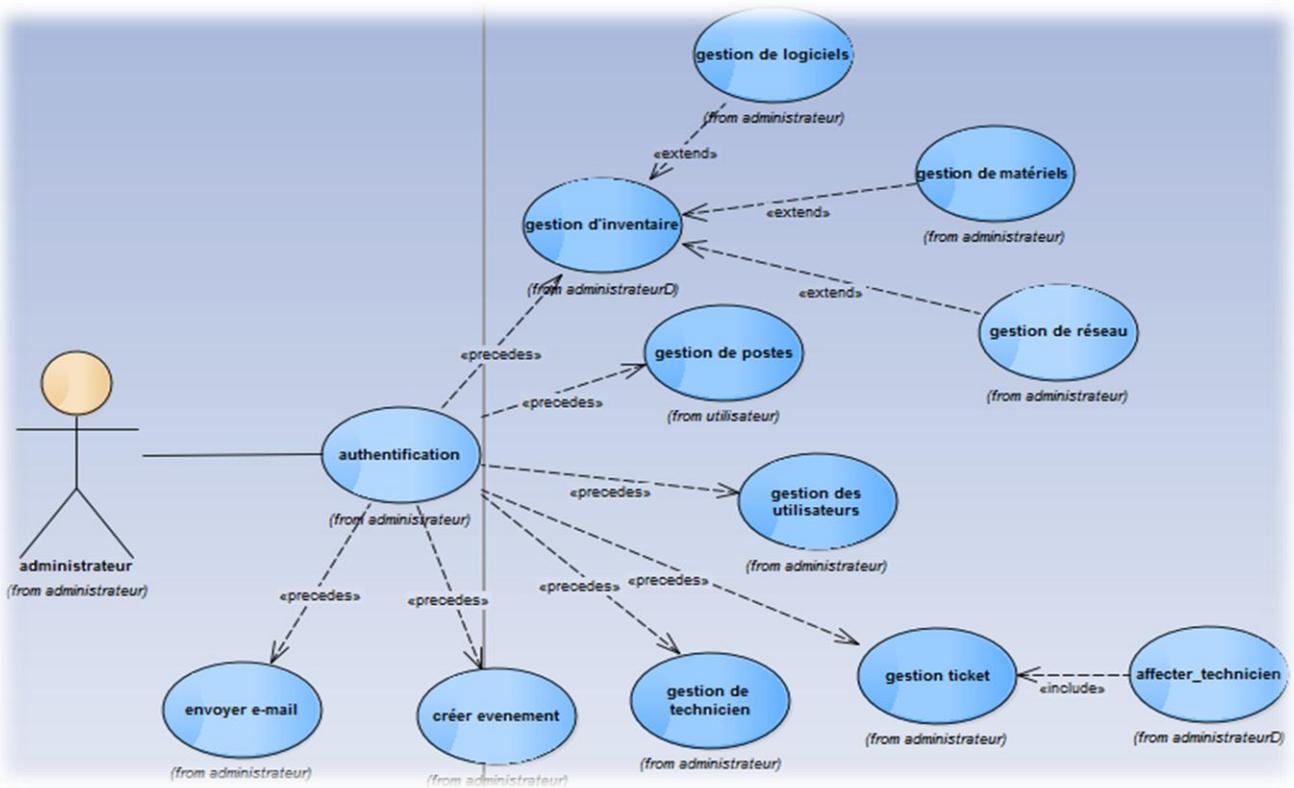


Figure 10: Diagramme de cas d'utilisation pour l'administrateur

. Utilisateur

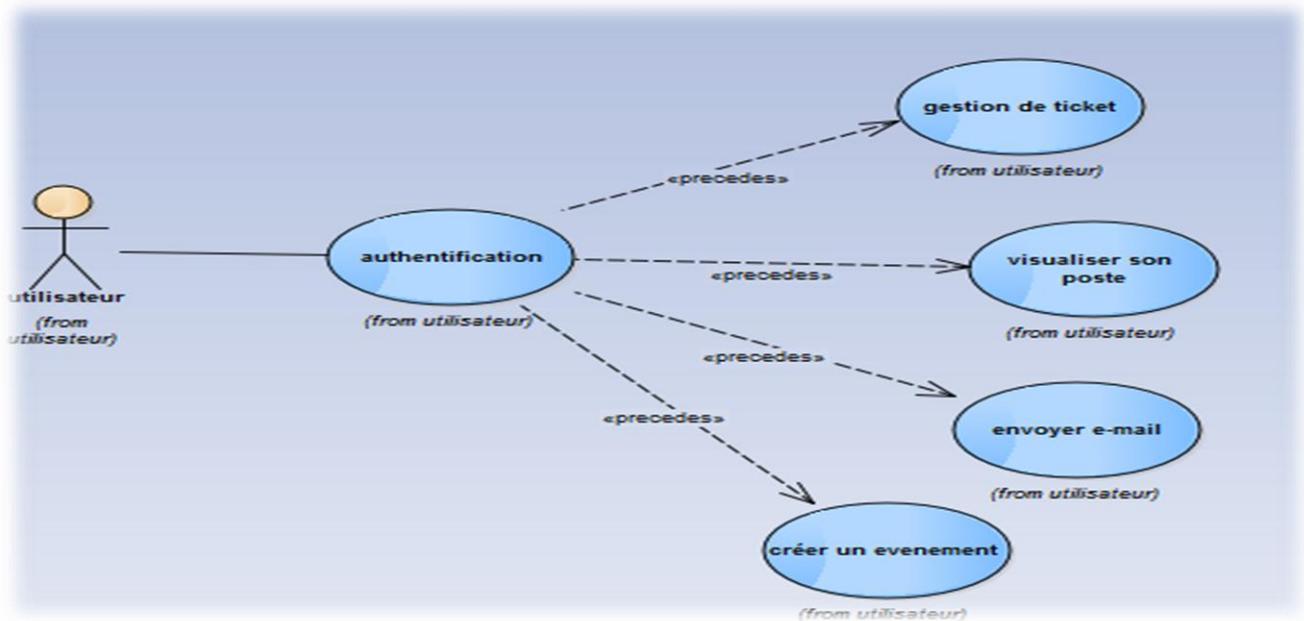


Figure 11: Diagramme de cas d'utilisation pour l'utilisateur



Technicien

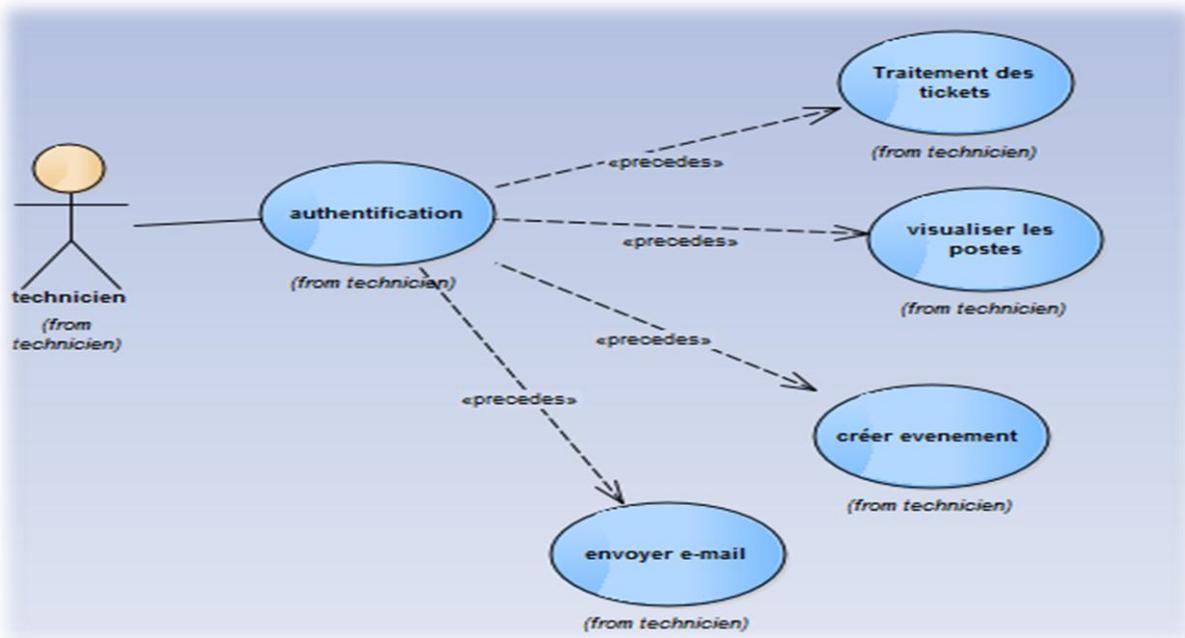


Figure 12: Diagramme de cas d'utilisation pour le technicien

. Cas d'utilisation « Gestion de tickets »

Les figures suivantes nous présentent de façon plus détaillée le cas d'utilisation « gestion de tickets » selon les trois acteurs

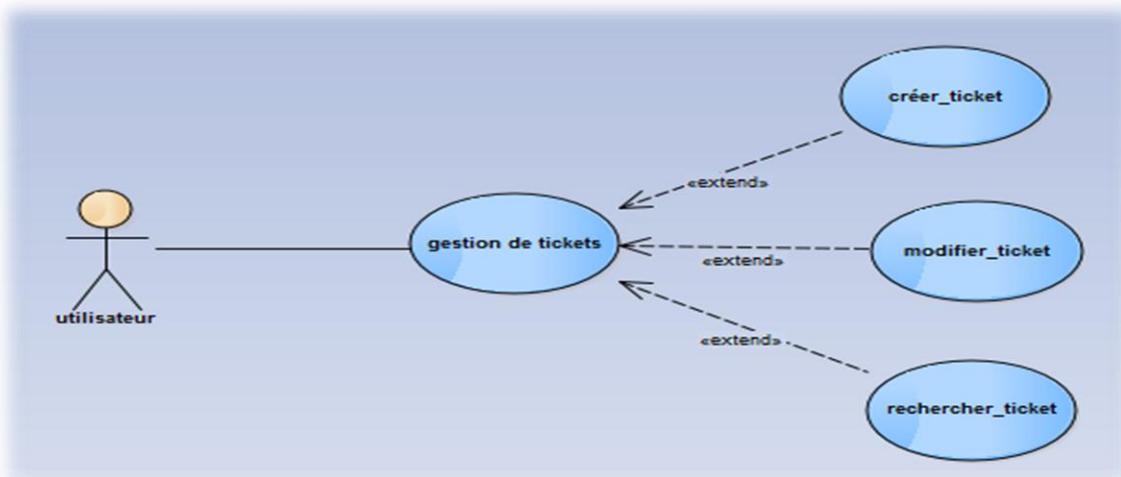


Figure 13: Cas d'utilisation « gestion de tickets » pour l'utilisateur

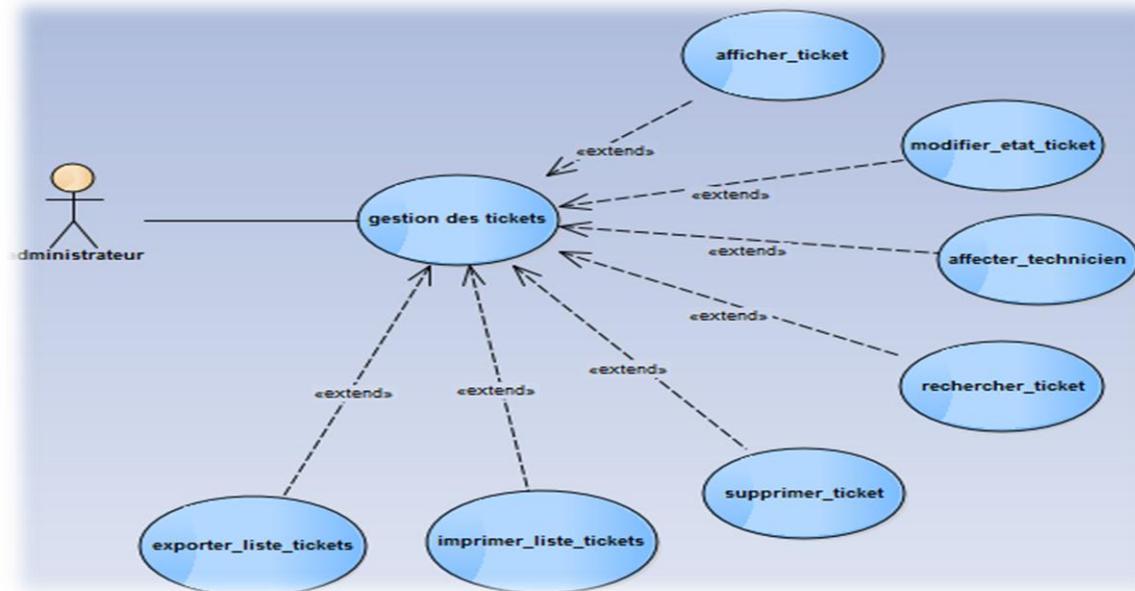


Figure 14 : Cas d'utilisation « gestion de tickets » pour l'administrateur

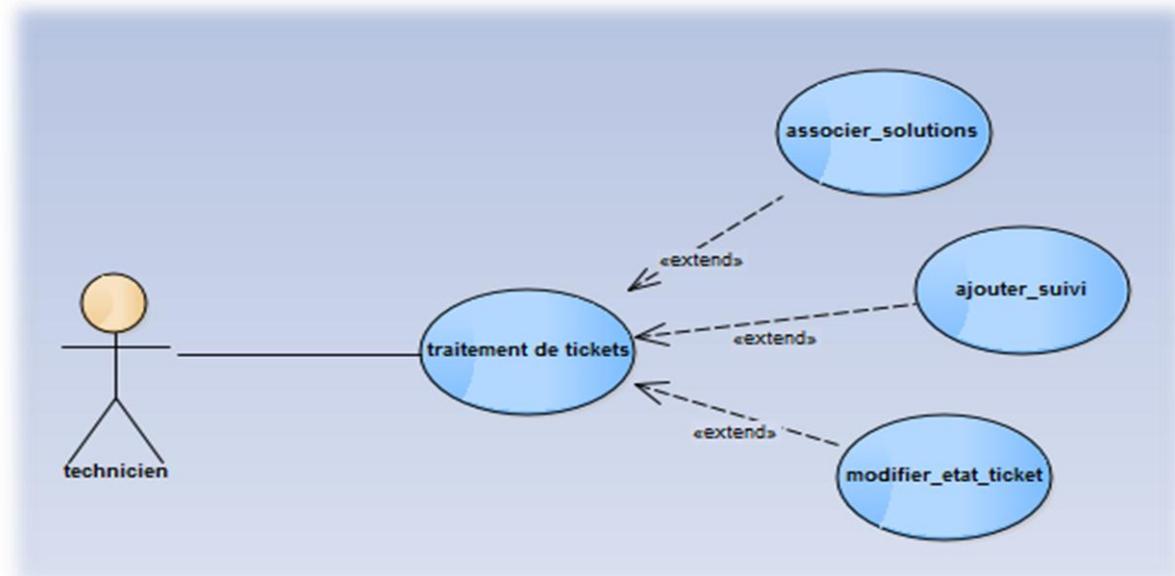


Figure 15 : Cas d'utilisation « gestion de tickets » pour le technicien

4.3. Diagramme de séquence

Le diagramme de séquence permet d'illustrer les cas d'utilisation et de représenter les interactions dans le temps entre les objets du système.

Authentification



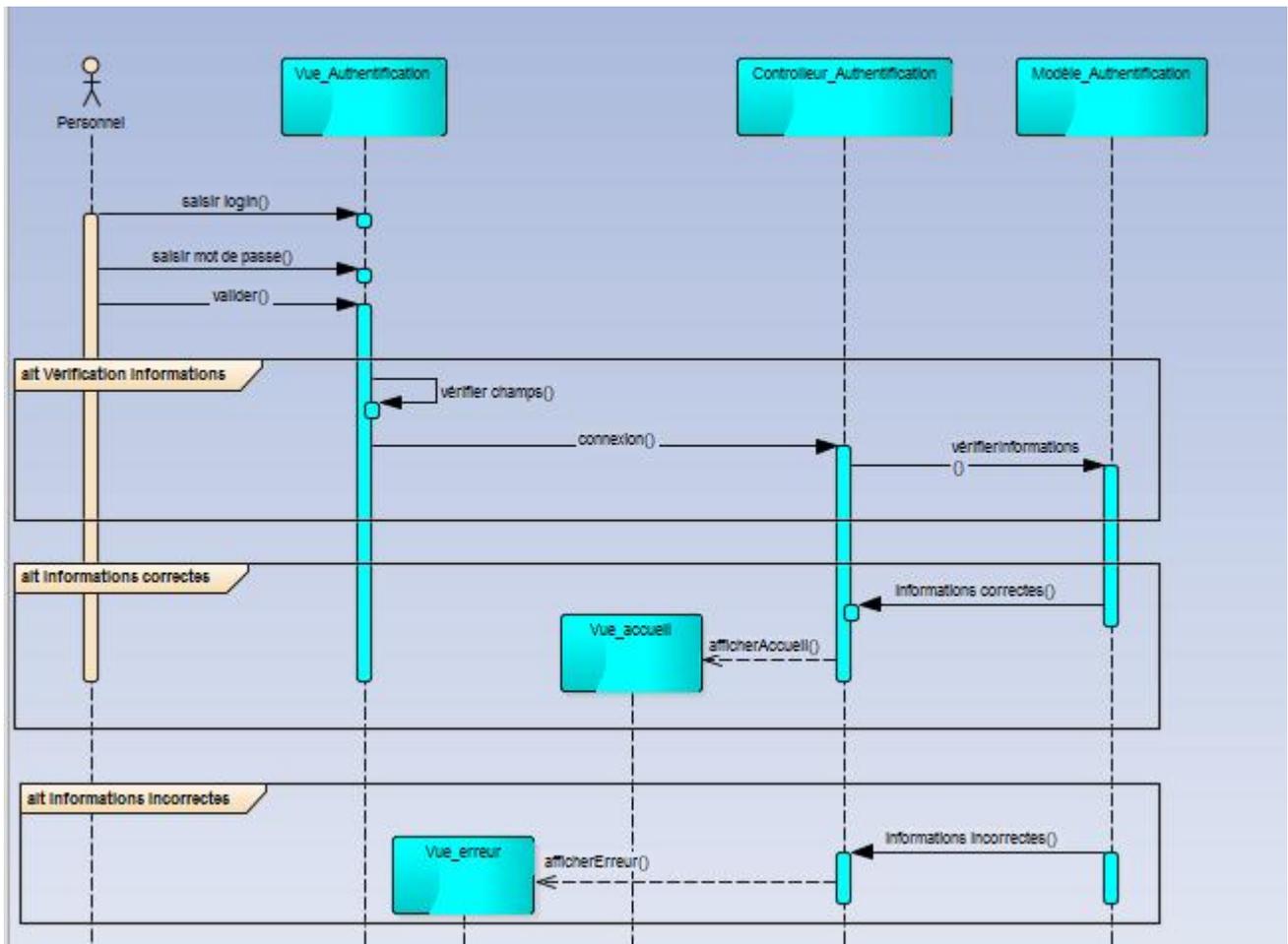


Figure 16 : Diagramme de séquence authentification

Ajout d'un poste

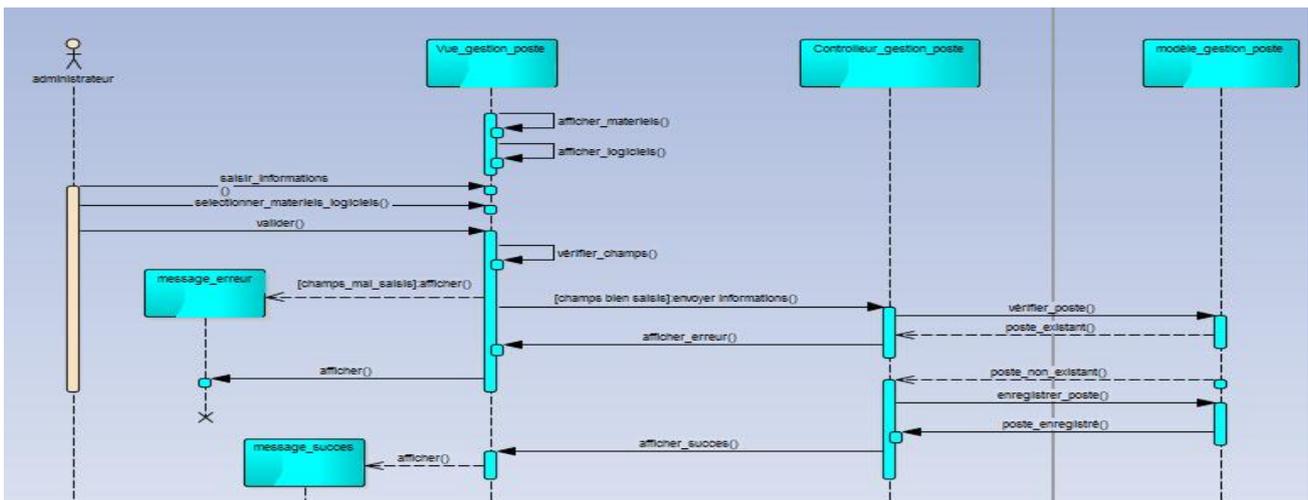


Figure 17 : Diagramme de séquence ajouter poste

4.4. Diagramme de classe

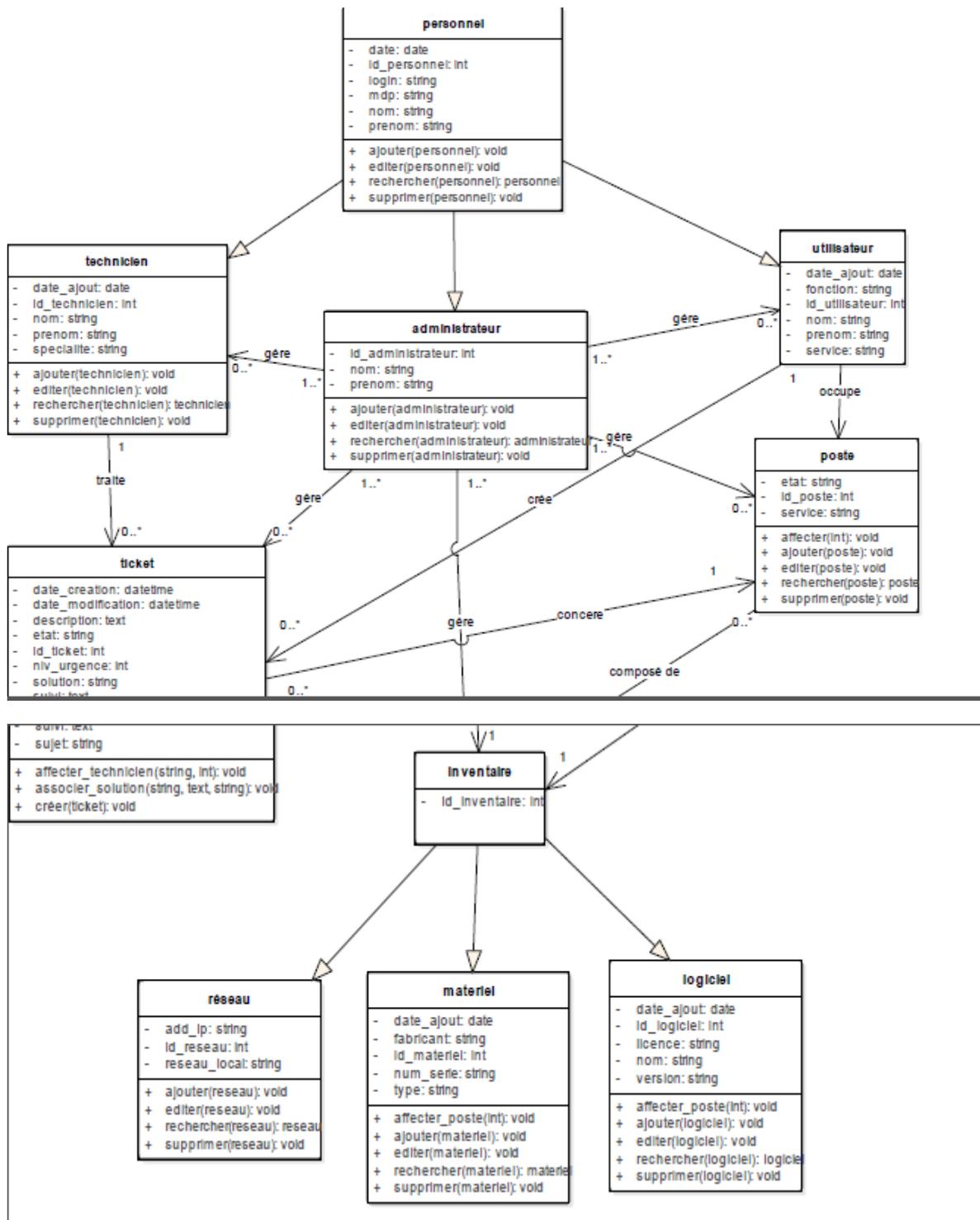


Figure 18 : Diagramme de classe



Description du diagramme de classe :

La classe personnel généralise les trois classes (administrateur, utilisateur et technicien),

L'administrateur gère les classes utilisateur, technicien , poste ,inventaire et ticket

Un utilisateur ou plusieurs utilisateurs peuvent occuper un ou plusieurs postes.

Chaque poste contient au moins un matériel et un logiciel et une seule adresse réseau.

Chaque matériel peut appartenir à plus qu'un poste, ainsi pour un logiciel.

L'inventaire contient les matériels , logiciels et les adresses réseau.

Un utilisateur peut créer un ou plusieurs tickets.

Chaque ticket peut être géré par un ou plusieurs administrateurs et traité par un seul technicien.

Un ticket concerne un seul poste.

5. Conclusion :

Dans ce chapitre, nous avons énuméré les différents besoins fonctionnels et non fonctionnels de notre application. Ensuite, on a fait une étude des différents cas d'utilisation de ma solution. Ce chapitre a été d'une importance cruciale surtout pour la compréhension des besoins et attentes du client.

Chapitre3 :

Réalisation



1. Introduction

Nous avons tout au long des chapitres précédents introduits le projet, énuméré les étapes nécessaires à sa mise en œuvre, analysé l'ensemble des besoins à satisfaire et conçu l'architecture du système. A présent nous entamerons la phase de réalisation qui est l'étape où nous traduisons la conception et les règles par un langage de programmation afin d'aboutir à une automatisation des besoins tels qu'ils ont été définis dans la spécification. Ainsi donc, ce chapitre sera divisé en quatre parties majeures. Premièrement, on va commencer par une description de l'environnement de travail à savoir l'environnement logiciel.

2. Outils et technologies de développement

	<p>EasyPHP[7] fut le premier package WAMP à voir le jour (1999). Il s'agit d'une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL phpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs. Il permet donc d'installer en une seule fois tout le nécessaire au développement local du PHP.</p>
	<p>Apache est un serveur http créé et maintenu au sein de la fondation Apache. C'est le serveur http populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.</p>
	<p>MySQL Workbench[4] est un logiciel de gestion et d'administration de bases de données MySQL créé en 2004. Via une interface graphique intuitive, il permet entre autres de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL.</p>
	<p>Notepad++[6] est un programme spécialement conçu pour l'édition de code source. Il est compatible avec plusieurs langages de programmation</p>
	<p>Enterprise Architect[5] est un outil d'analyse de création UML, couvrant le développement des logiciels de rassemblement d'exigences, en passant par</p>

	<p>les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien.</p> <p>Cet outil permet de bien schématiser notre application, pour passer de la conception vers la réalisation. Il facilite la représentation des diagrammes UML tels que le diagramme des cas d'utilisation, des séquences et des classes.</p>
	<p>L'Hypertext Markup Language[8], généralement abrégé HTML, est le format de données conçu pour représenter les pages. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages</p> <p>Les feuilles de style en cascade[9], généralement appelées CSS, forment un langage informatique qui décrit la présentation des documents HTML et XML.</p>
	<p>Hypertext Preprocessor[10], plus connu sous son sigle PHP est un langage de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté-objet.</p>
	<p>Ajax [11] (Asynchronous JavaScript and XML) est une architecture informatique basée essentiellement sur le JavaScript, permet d'accéder de manière asynchrone avec les actions de l'utilisateur à la base de données et ce en utilisant en plus du JavaScript, une classe XML http Request, qui comporte des méthodes permettant de communiquer avec le serveur, offrant ainsi à l'utilisateur une réponse rapide et instantanée</p>
	<p>JS est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype.</p>
	<p>jQuery[12] est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.</p>
	<p>Bootstrap[13] est une collection d'outils utile à la création de sites web et applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub (GitHub est un service web d'hébergement et de gestion de développement de logiciels).</p>





CodeIgniter [14] est un Framework libre écrit en PHP. IL suit le motif de conception MVC. La documentation de CodeIgniter est complète. La communauté du Framework est très active ce qui permet de trouver de l'aide très rapidement. De plus, les membres de la communauté de CodeIgniter ont développé de nombreuses bibliothèques réutilisables.

Justification du choix de CodeIgniter :

Le Framework CodeIgniter a été choisi parmi plusieurs Framework .Ce choix est justifié par le fait qu'il est plus adapté à mon application et qu'il permet de développer des projets plus rapides, il possède une documentation importante, permet d'organiser le travail.

3. Présentation de l'application

3.1. Schéma général de l'application

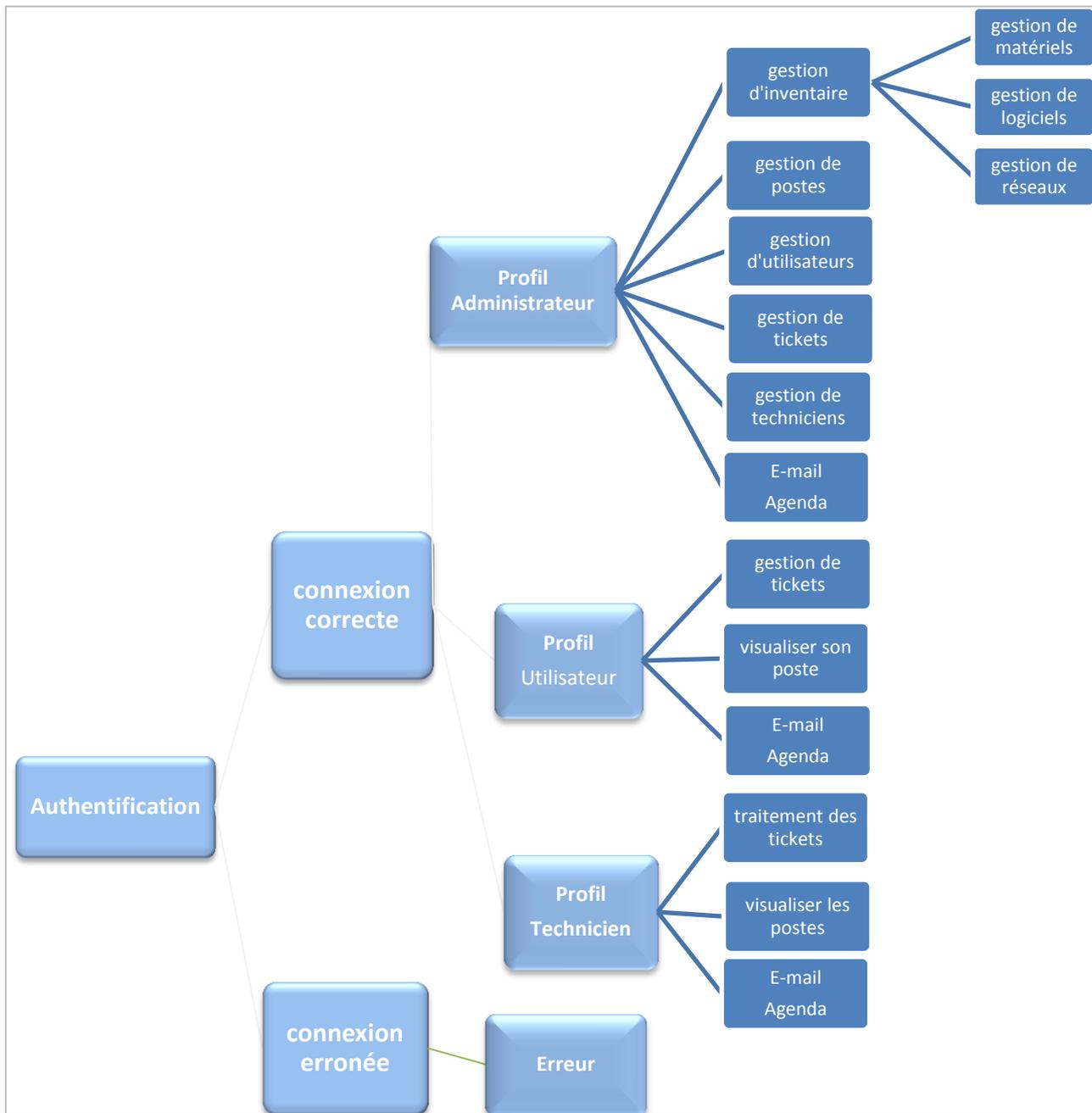
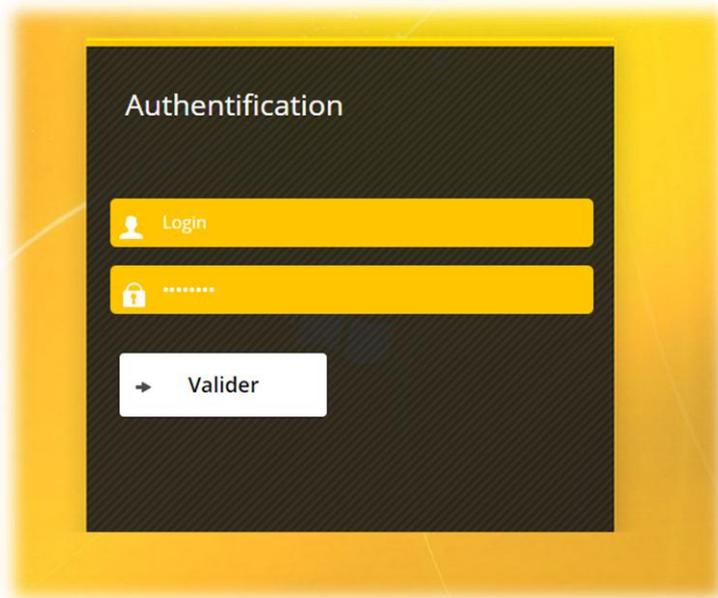


Figure 19 : Schéma général de l'application

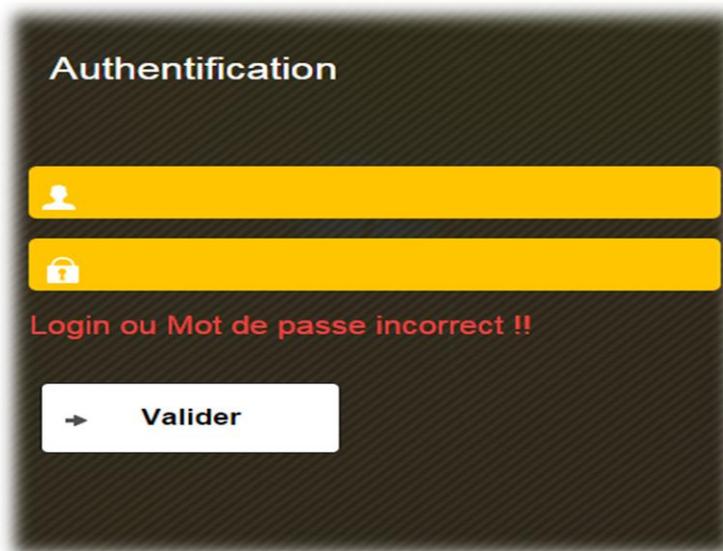
3.2. Authentification





[Figure 20 : Fenêtre d'authentification](#)

Cette page permet de s'authentifier et de faire une redirection vers l'accueil associé à chacun des acteurs. Si le login ou mot de passe incorrecte , la page affichera le message d'erreur comme le suivant :



[Figure 21 : Fausse authentification](#)

3.3. Gestion de tickets

Partie utilisateur

L'utilisateur peut créer un ticket en spécifiant principalement le sujet du ticket matériel ,logiciel ou réseau et une description , une fois validé , le ticket prend l'état nouveau et s'envoie à l'administrateur.

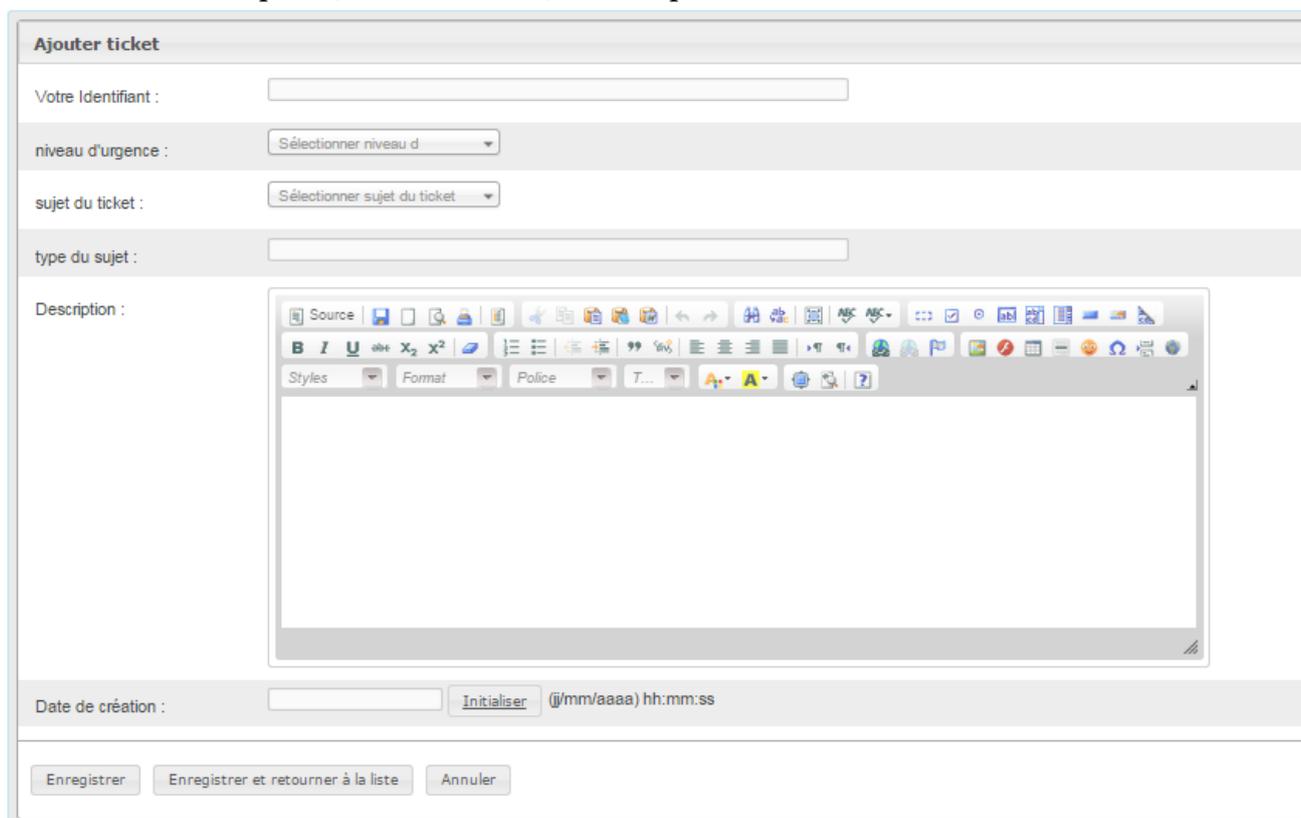


Figure 22 : Fenêtre d'ajout d'un ticket

Partie administrateur

L'administrateur reçoit **3 alertes** sur son accueil, la première lors de création d'un ticket urgent, la deuxième lors de l'ajout d'un nouveau ticket et la troisième lors de résolution d'un ticket.

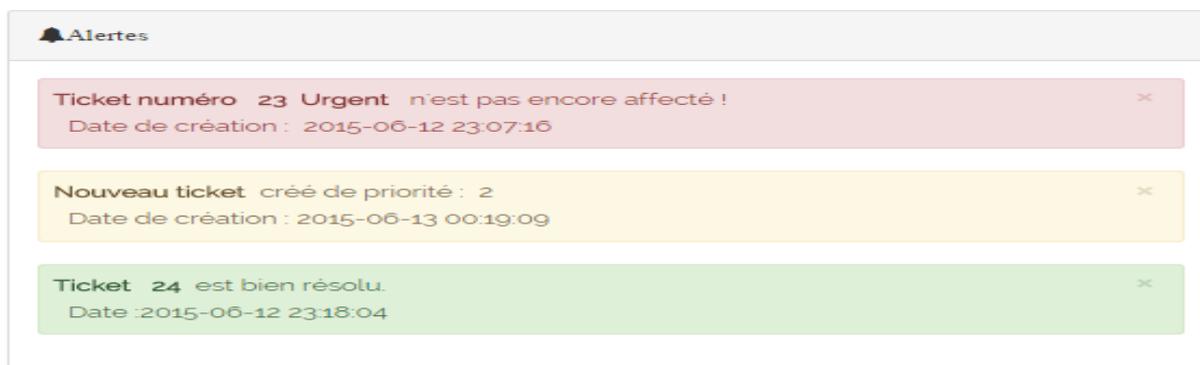


Figure 23 : Alertes

La figure suivante montre la table des tickets chez l'administrateur :



Date création	Sujet	Priorité	état	Technicien	Derniere modification	Actions
12/06/2015 - 23:07	materiel	1	nouveau	id : - 0		Affecter technicien Voir Éditer Supprimer
13/06/2015 - 00:19	logiciel	2	nouveau	id : - 0		Affecter technicien Voir Éditer Supprimer
20/05/2015 - 13:15	logiciel	1	En attente (affecté)	id : 2 - alaoui (installation)	20/05/2015 - 13:31	Affecter technicien Voir Éditer Supprimer

Affichage de 1 à 3 de 3 enregistrements

Figure 24 : Table des tickets pour l'administrateur

Pour affecter un technicien, la page suivante qui s'ouvre lui permet de sélectionner un technicien et changer l'état du ticket.

Sujet : logiciel

Détails : **Bonjour,**
J'ai besoin du logiciel MATLAB dans mon poste.

Priorité : 1

Technicien : id : 2 - alaoui (installation)

état : Sélectionner état

Derniere modification : 20/05/2015 13:31:44 Initialiser (jj/mm/aaaa) hh:mm:ss

Sauvegarder les modifications Mettre à jour et retourner à la liste Annuler

Figure 25 : Fenêtre d'affectation de technicien

Partie technicien

Le technicien choisi, reçoit le ticket dans sa table comme le montre la figure suivante :

Dernier ticket	Niveau d'urgence	Sujet	Etat	Solution proposée	Suivi	Derniere modification	Actions
20/05/2015 - 13:15	1	logiciel	en attente(affecté)				Associer solution Voir Éditer

Affichage de 1 à 1 de 1 enregistrements

Figure 26 : table des tickets chez le technicien

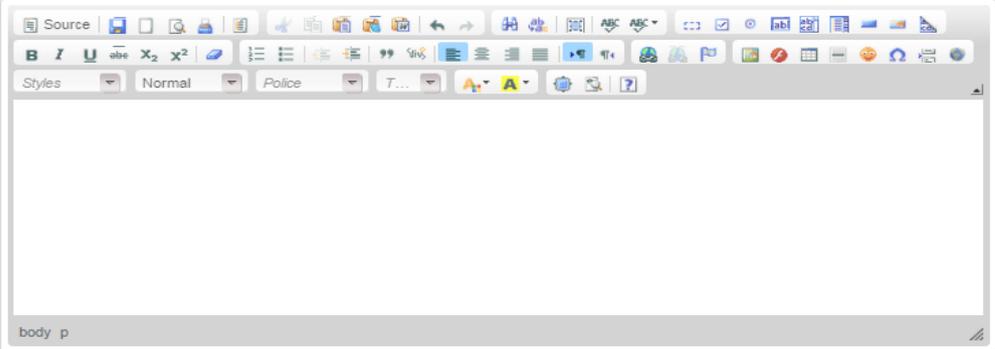


Sujet : logiciel

Description : **Bonjour,**
J'ai besoin du logiciel MATLAB dans mon poste.

Niveau d'urgence : 1

Solution proposée :

Suivi : 

Etat : Sélectionner Etat

Derniere modification : (jj/mm/aaaa) hh:mm:ss

Figure 27 : Fenêtre traitement ticket

4.1. Fonctionnalités générales

Agenda

Tous les acteurs ont le droit de créer des événements dans un agenda partagé .Pour le technicien, il peut spécifier l'heure de résolution d'un ticket sur machine.

Evenements

- ticket urgent
- ticket résolu
- nouveau suivi
- nouveau ticket

Supprimer après selection

Créer evenement

< >
Aujourd'hui
June 2015
Mois Semaine Jour

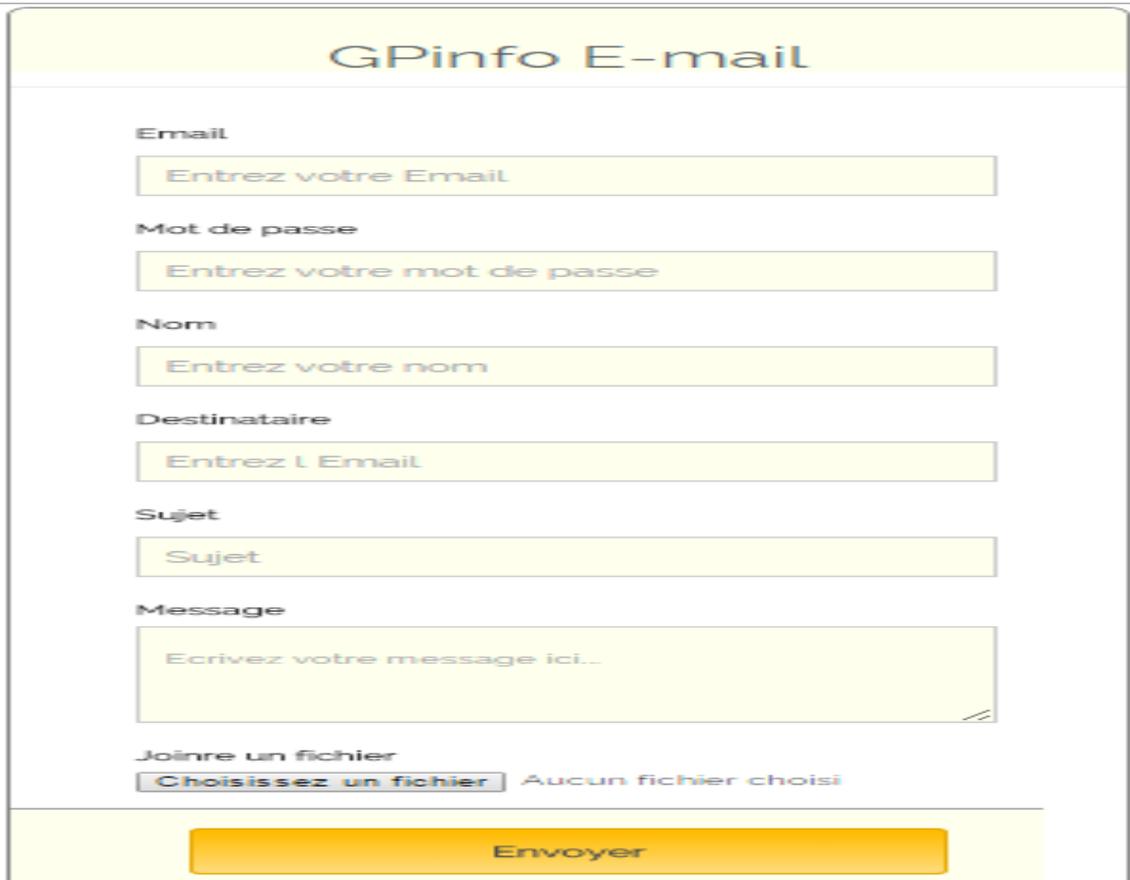
Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8 ticket urgent	9	10 nouveau ticket	11	12	13
14	15	16	17 ticket résolu	18	19	20
21	22	23 nouveau suivi	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Figure 28 : Agenda



E-mail

Tous les acteurs peuvent envoyer un E-mail .



The screenshot shows a web form titled "GPinfo E-mail". The form contains several input fields and a submit button:

- Email:** A text input field with the placeholder text "Entrez votre Email".
- Mot de passe:** A text input field with the placeholder text "Entrez votre mot de passe".
- Nom:** A text input field with the placeholder text "Entrez votre nom".
- Destinataire:** A text input field with the placeholder text "Entrez l Email".
- Sujet:** A text input field with the placeholder text "Sujet".
- Message:** A large text area with the placeholder text "Ecrivez votre message ici...".
- Joindre un fichier:** A button labeled "Choisissez un fichier" followed by the text "Aucun fichier choisi".
- Envoyer:** A large orange button at the bottom of the form.

Figure 29 : E-mail

4. Conclusion

L'objectif principal de ce chapitre était de présenter quelques parties de l'application obtenue. A cet effet on a tour à tour présenté les outils et technologies de développement, puis on a décrit la partie gestion ticket de la vue des trois acteurs enfin on a présenté les fonctionnalités générales utilisables par les différents acteurs.



Conclusion et perspectives

Au cours de la période du stage de fin d'études, on a eu l'opportunité de mettre en œuvre différentes connaissances acquises durant nos études à la faculté des sciences et techniques de Fès et acquérir de nouveaux outils de développement tels que l'architecture MVC, Ajax, PHP (orienté objet) et jQuery.

Notre travail s'est fixé comme objectifs de satisfaire le maximum des besoins du cahier de charge et faciliter les tâches à l'administrateur et aux utilisateurs. Le projet se présente en trois parties. La première partie s'est intéressée au lieu de stage et à la problématique, la deuxième partie à la méthodologie de l'analyse et à la conception UML, la troisième partie aux technologies et outils utilisés et à la présentation de l'application.

Les difficultés majeures qu'on a rencontrées résident essentiellement dans la nouveauté des technologies avec lesquelles on a travaillé et la contrainte du temps pour les maîtriser.

Comme perspectives on peut envisager par la suite la création d'un système de communication interne, , mettre en œuvre une cartographie des réseaux



Bibliographie et Webographie

Webographie

- [1] [Inventis.ma](http://inventis.ma)
- [2] <http://fr.wikipedia.org/wiki/UML>
- [3] http://fr.wikipedia.org/wiki/Modèle_vue_contrôleur
- [4] [http://fr.wikipedia.org/wiki/MySQL Workbench](http://fr.wikipedia.org/wiki/MySQL_Workbench)
- [5] [http://fr.wikipedia.org/wiki/Entreprise Architect](http://fr.wikipedia.org/wiki/Entreprise_Architect)
- [6] <http://fr.wikipedia.org/wiki/Notepad++>
- [7] <http://fr.wikipedia.org/wiki/EasyPHP>
- [8] <http://fr.wikipedia.org/wiki/HTML>
- [9] <http://fr.wikipedia.org/wiki/CSS>
- [10] <http://fr.wikipedia.org/wiki/PHP>
- [11] <http://fr.wikipedia.org/wiki/Ajax>
- [12] jquery.com
- [13] getbootstrap.com
- [14] Ellislab.com/CodeIgniter

Bibliographie

- Cours Pr Ilham Chaker Génie logiciel chapitre2 le processus logiciel page 78
- Cours Techniques web du Pr. Ouazarf (2014/2015)
- Cours UML de Mr. BENNABOU (2014-2015)

