



PROJET DE FIN D'ÉTUDES

**MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX**

**CONCEPTION ET RÉALISATION D'UNE APPLICATION DE
GESTION DES CONTRATS DE MAINTENANCE ET D'ANALYSE
DE PERFORMANCE DES ÉQUIPES SAV**



LIEU DE STAGE : M2ASOFT

RÉALISÉ PAR : ZAKARIA HAOUAZIM

SOUTENU LE 15/06/2017

ENCADRÉ PAR :

PR. ABDERRAHIM BENABBOU

MR. ABDESLAM ESSAHRAOUI

DEVANT LE JURY COMPOSÉ DE :

PR. MOHAMED OUZARF PRÉSIDENT

PR. FATIHA MRABTI EXAMINATRICE

PR. AHLAME BEGDOURI EXAMINATRICE

PR. ABDERRAHIM BENABBOU ENCADRANT

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿ قالوا سبحانك لا علم لنا الا ما علمتنا

﴿ انك انت العليم الحكيم

صدق الله العظيم
الآيه (32) سورة البقره

Dédicace

A ALLAH, le tout puissant, le clément, le bienfaiteur miséricordieux, qui fait chaque jour des merveilles pour moi.

A mes très chers **parents**

Aucune dédicace ne saurait exprimer l'estime, le dévouement, le respect, l'affection et l'amour que je porte pour vous.

Personne au monde ne pourrait compenser les sacrifices que vous avez endurés tout le long de mes études. Veuillez trouver, dans ce travail, le fruit de votre Dévouement.

A mes **sœurs**, mes **frères**, et tous mes **amis**

Mes sincères remerciements pour le soutien, l'aide, les sacrifices et l'encouragement que vous m'apportez.

Que ce travail soit pour vous le témoignage de ma grande considération.

Pour tous les agréables instants qu'on a vécus, je vous dédie ce travail.

Remerciements

Tout d'abord, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué de près ou de loin à la réalisation de ce travail.

Je tiens à exprimer ma reconnaissance envers Monsieur le Professeur **Abderrahim Benabbou**, pour ses conseils, sa disponibilité et son aide constant durant la réalisation de ce travail sachant que sans lui ce mémoire n'aurait jamais vu le jour.

Je remercie également Monsieur **Abdeslam Essahraoui**, qui m'a consacré beaucoup de temps malgré ses obligations précieuses de recherche, pour ses fructueux conseils, ses commentaires et ses critiques constructives et très utiles pour structurer ce travail qui m'a mis dans le vif du sujet.

C'est un honneur et une grande fierté d'exprimer ma gratitude et ma reconnaissance à mes Professeurs, qui sont aujourd'hui des membres de ce jury les Professeurs **Mohamed Ouzarf**, **Fatiha Mrabti** et **Ahlame Begdouri**, à qui j'adresse mes vifs remerciements pour avoir accepté de siéger dans ce jury.

Résumé

Le projet s'inscrit dans le cadre d'un ensemble de projets de l'équipe de développement spécifique et destiné aux besoins de M2aSoft. Ce projet a une grande valeur pour M2aSoft dans la mesure où il permet de gérer le processus SAV (Service Après-Vente), en gérant 3 modules qui sont indispensables de ce service (gestion des contrats de maintenance, gestion des équipes SAV, gestion de processus SAV).

Ce stage consiste à participer dans la réalisation d'une application de gestion des contrats de maintenance et d'analyse de performance des équipes SAV. Afin d'évaluer la productivité des équipes (individus) de service SAV, la rentabilité d'un contrat et par conséquent on peut avoir un service SAV plus productive.

Le déroulement du travail réalisé est le suivant :

- Recenser les besoins fonctionnels et non fonctionnels du projet.
- L'étude technique et la conception détaillée de l'application.
- Réalisation de l'application.

Pour mener à bien le développement de ce projet, la méthodologie agile semble plus adéquate à notre contexte, avec le langage UML (Unified Modeling Language) comme langage de modélisation de ce système.

Relativement au volet technique, il y avait eu recours aux nouvelles technologies et aux bonnes pratiques de la programmation, depuis Maven qui nous définit la structure du projet, aux Framework et technologie : Spring MVC (Model View Controller), Spring Sécurité, Hibernate/JPA (Java Persistence API) et JAVA/JEE (Java Enterprise Edition)... etc.

Ma mission consiste à développer une partie de l'application pour la gestion des contrats de maintenance, la gestion des équipes SAV et la gestion de processus SAV.

Table des abréviations

Abréviation	Description
AGL	Atelier de Génie Logiciel
SAGE	Solution Adapté au Gestion de l'entreprise
SAV	Service Après-Vente
CRM	Customer Relationship Management
IT	Information Technolgy
SQL	Structured Query Language
UML	Unified Modeling Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
XML	Extensible Markup Language
API	Application Programming Interface
JPA	Java Persistance API
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
JS	JavaScript
AJAX	acronyme d'Asynchronous Javascript and Xml
DAO	Data Access Object
MVC	Model View Controller
ORM	Object Relational Mapping
JEE	Java Entreprise Edition
Ioc	Inversion Of Control

Liste des Figures

Figure 1: Partenaires M2aSoft.....	11
Figure 2: Référence M2aSoft.....	11
Figure 3: scénario d'intervenir une réclamation.....	14
Figure 4: Planning du projet.....	20
Figure 5: scénario de gestion des tickets et des interventions.....	23
Figure 6: diagramme d'activité de gestion des contrats.....	27
Figure 7: diagramme d'activité gestion des équipes SAV.....	28
Figure 8: diagramme d'activité gestion de processus SAV.....	29
Figure 9: digramme de packages générale.....	30
Figure 10: cas d'utilisation package de gestion des contrats de maintenance-Administrateur.....	32
Figure 11: package de gestion des contrats de maintenance-Intervenant.....	32
Figure 12: package de gestion des équipes SAV-Administrateur.....	33
Figure 13: package de gestion des équipes SAV-Chef d'équipe.....	34
Figure 14: package gestion de processus SAV-Tiers.....	34
Figure 15: cas d'utilisation package de gestion de processus SAV-Administrateur.....	35
Figure 16: package de gestion de processus SAV-Chef d'équipe, Intervenant.....	36
Figure 17: packages du diagramme de classes.....	37
Figure 18: diagramme de classes globale de l'application.....	38
Figure 19: Diagramme de séquence créer contrat.....	41
Figure 20: diagramme de séquence créer ticket.....	42
Figure 21: diagramme de composants.....	43
Figure 22: diagramme de déploiement.....	44
Figure 23: architecture générale de l'application.....	46
Figure 24: schéma Spring MVC.....	50

Table des matières

Dédicace	2
Remerciements	3
Résumé	4
Table des abréviations	5
Liste des Figures.....	6
Introduction Générale.....	9
Chapitre 1 : Contexte général du projet	10
1. Introduction.....	10
2. Organisme d'accueil : M2aSoft.....	10
2.1 Présentation	10
2.2 Partenaires M2aSoft.....	11
2.3 Références M2aSoft :	11
2.4 Equipe de développement spécifique.....	12
3. Cahier de charges	12
3.1 Présentation Général du Projet.....	12
3.2 Spécification des besoins.....	15
3.3 Solution attendues	18
3.4 Objectif	18
3.5 Livrables.....	19
4. Démarche de projet	19
4.1 Gestion de projet Agile.....	19
4.2 Agilité pratique	20
4.3 Planning du projet	20
4.4 Equipe de réalisation de projet	20
5. Conclusion	21
Chapitre 2 : Analyse et Conception.....	22
1. Introduction.....	22
2. Analyse de projet.....	22
2.1 Etude d'existant.....	22
2.2 Critique de l'existant	23
2.3 Solution : M2aSAV	24
3. UML	25
3.1 Choix d'UML	25
3.2 Les besoins.....	25
3.3 La maîtrise d'œuvre.....	25
3.4 Visual Pradigm	26

3.5	Identifications des acteurs	26
3.6	Diagramme d'activité	27
3.7	Diagramme de packages	29
3.8	Diagramme de cas d'utilisation	30
3.9	Diagramme de classe.....	36
	37
3.10	Diagramme de séquence.....	40
3.11	Diagramme de composants.....	43
3.12	Diagramme de déploiement.....	44
4.	Conclusion	44
Chapitre 3 : Études technique et Réalisation.....		45
1.	Introduction.....	45
2.	L'architecture générale de l'application.....	45
2.1	Back-end	46
2.2	Front-end.....	47
2.3	Fonctionnement	47
3.	Technologies et Framework de développement.....	48
3.1	Java	48
3.2	La technologie JEE	49
3.3	XML.....	49
3.4	Spring.....	49
3.5	JPA/Hibernate.....	51
3.6	Jackson.....	51
3.7	Bootstrap 3	51
3.8	jQuery / Ajax	52
3.9	Environnement de développement	52
4.	Réalisation	54
4.1	Fruit du travail	54
4.2	Test	66
5.	Conclusion	66
Conclusion Générale		67
Webliographie		68

Introduction Générale

Dans un monde actif et continuellement évolutif, la motivation d'avoir des moyens performants et efficaces de communication et d'échange d'information devient une notion fondamentale. Cette motivation à pousser M2aSoft qui est une société spécialisée en domaine d'ingénierie des systèmes et réseaux à développer une application de **gestion des contrats de maintenance et d'analyse des équipes SAV**.

Du un ensemble assez grand de tiers abonnés à M2aSoft, le service SAV a de plus en plus de réclamations pour la maintenance des services que couvre cette entreprise. D'autre part pour répondre à ces réclamations et fournir des meilleurs services, l'organisation et la gestion de ce service est une étape primordiale pour mener à bien la rentabilité et la productivité de service SAV.

C'est dans cet objectif que j'ai effectué mon stage PFE à M2aSoft.

La finalité de ce projet consiste à développer des modules qui permet de :

- Gérer les contrats de maintenance.
- Gérer les équipes SAV.
- Gérer le processus SAV.

Le présent rapport décline les différentes phases que j'ai suivies pour la réalisation du projet ainsi que les moyens auxquels j'ai eu recours pour atteindre les objectifs fixés.

La présentation de ce rapport est divisée en 3 chapitres :

- ❖ **Contexte générale du projet** : dans la première partie, on présente l'organisme d'accueil M2aSoft. Ensuite la présentation de l'étude fonctionnelle du projet sera l'objet de la deuxième partie et finalement la troisième partie concernera la démarche de projet.
- ❖ **Analyse et Conception** : Avant d'aborder la conception dans ce chapitre, on présente dans un premier temps une analyse de projet et solution M2aSAV.
- ❖ **Etude technique et Réalisation** : dans ce chapitre, on présente l'architecture technique du projet et les outils et Framework de développement et finalement on présente le fruit de travail en des captures d'écran ainsi que les types de tests effectués.

Chapitre 1 : Contexte général du projet

1. Introduction

Dans ce chapitre nous présentons dans la première partie l'organisme d'accueil ou j'ai effectué mon stage de projet de fin d'études, en passant par l'équipe de développement spécifique à laquelle j'étais intégré. Le cahier de charge est décrit dans la deuxième partie qui comporte une description sur le projet et une présentation du besoin, solution attendus et son objectif. Ce chapitre inclut également la planification employée dans le but de se mettre dans le contexte du projet pour le bon déroulement de sa réalisation.

2. Organisme d'accueil : M2aSoft

2.1 Présentation

Implantée à Casablanca depuis 2009, M2asoft société spécialisée en ingénierie des systèmes et réseaux et revendeur SAGE (solution adapté à la gestion de l'entreprise), propose une gamme complète de solutions et de services pour l'entreprise en intervenant à tous les stades des projets, depuis l'étude préalable jusqu'à la réalisation complète.

Pour répondre et interagir aux besoins des clients M2aSoft fournit les services suivants :

- Conseil
- Formation
- Développement spécifique
- Infrastructure et infogérance
- Virtualisation et Cloud
- Assistance

M2aSoft est également revendeur des solutions SAGE couvrant l'ensemble des besoins des entreprises en Gestion Comptable et Financière, Gestion Commerciale, Paie et CRM (customer relationship management) et e-business dont elle apporte toute son expérience et son expertise des solutions de gestion et de finance.

2.2 Partenaires M2aSoft

Comme toute entreprise leader de domaine d'IT (Information Technology) et de Consulting M2aSoft a des partenaires internationaux afin de fournir les meilleurs services aux clients, voici une figure illustrant quelques partenaires :



Figure 1: Partenaires M2aSoft

2.3 Références M2aSoft :

Parmi les clients qui sont satisfaits et font confiance aux services et produits fournis il y en a qui sont devenus des références pour M2aSoft, voici des exemples dans la figure suivante :



Figure 2: Référence M2aSoft

2.4 Equipe de développement spécifique

L'équipe de développement spécifique fait partie de l'ensemble des équipes techniques qui font enrichir la qualité des produits et des services fournis par M2aSoft.

L'équipe est constituée d'un ensemble d'individus ayant l'expertise nécessaire qui permet d'améliorer le développement spécifique autour des systèmes d'informations.

Cette équipe a pour objectif d'adapter des solutions sur-mesure tout en développant des fonctions spécifiques jusqu'à des solutions d'intégration complètes selon les besoins des clients.

Enfin Les outils de développement adoptés s'appuient sur le concept d'**Atelier de Génie Logiciel (AGL)** normalisé sur l'environnement Windows et Linux.

3. Cahier de charges

3.1 Présentation Général du Projet

3.1.1 Contexte du projet

Le projet de gestion des contrats de maintenance et analyse de performance des équipes SAV s'inscrit dans le cadre de l'ensemble des projets portés par l'équipe de développement spécifique et destiné aux besoins de M2aSoft.

Ce projet concerne dans un premier temps le développement d'une application web pour la gestion des contrats de maintenance des tiers abonnés tout en suivant les opérations et les traces des individus des équipes SAV, de la réclamation de la problématique par un tiers jusqu'à la résolution par les individus des équipes SAV. A partir de ces interventions l'analyse des équipes SAV durant la maintenance de ces réclamations liés à un contrat sera effectuée.

Les motivations principales sont de pérenniser l'information SAV en terme d'accès et d'analyse ainsi que d'organiser et informatiser la relation clientèle et l'analyse de déroulement des opérations de service SAV pour faciliter les prises de décisions.

En résumé le projet définit 3 volets majeurs dans son fonctionnement métier :

- **Volet 1** : Gestion des contrats de maintenance.

- **Volet 2** : Gestion des équipes SAV
- **Volet 3** : Gestion de processus SAV.

3.1.1.1 Les contrats de maintenance

Pour bénéficier des services SAV pour une durée bien déterminé le tiers doit disposer d'un contrat de maintenance ou un contrat d'assistance SAGE.

Un contrat de maintenance peut être un contrat de Matériel ou de logiciel.

Pour chaque contrat de maintenance M2aSoft doit avoir un inventaire des produits qui font l'objet d'une réclamation de maintenance que ce soit pour un contrat matériel ou logiciel.

Un contrat d'assistance SAGE est un contrat qui permet aux tiers d'avoir le suivi et l'accompagne aux différentes modules et solutions SAGE dans le cadre de partenariat SAGE Partner.

Un contrat d'assistance SAGE peut être lié un ensemble de modules ou progiciels.

3.1.1.2 Les équipes SAV

Les équipes SAV sont chargées de répondre aux réclamations des tiers qui ont des contrats de maintenance avec M2aSoft.

Chaque équipe a sa spécialité et constituée d'un ensemble d'individus, parmi eux un chef d'équipe et un ensemble d'intervenant.

3.1.1.3 Le processus SAV

Après l'enregistrement des tiers et l'affectation des contrats (nouvelle contrat ou renouvellement) et les services sous inscrits, les équipes SAV sont chargées d'intervenir et de répondre aux réclamations des tiers.

Les acteurs concernés sont :

- ❖ **Tiers** : c'est le client qui possède un contrat avec M2aSoft
- ❖ **Service d'administration** : le service chargé d'interagir avec les tiers et les autres services d'entreprise.

❖ **Equipes SAV** : sont les équipes chargées de services après ventes.

❖ **Intervenant** : celui qui fait intervenir le problème de tiers.

Pour une réclamation le processus s'effectue de la manière suivante :

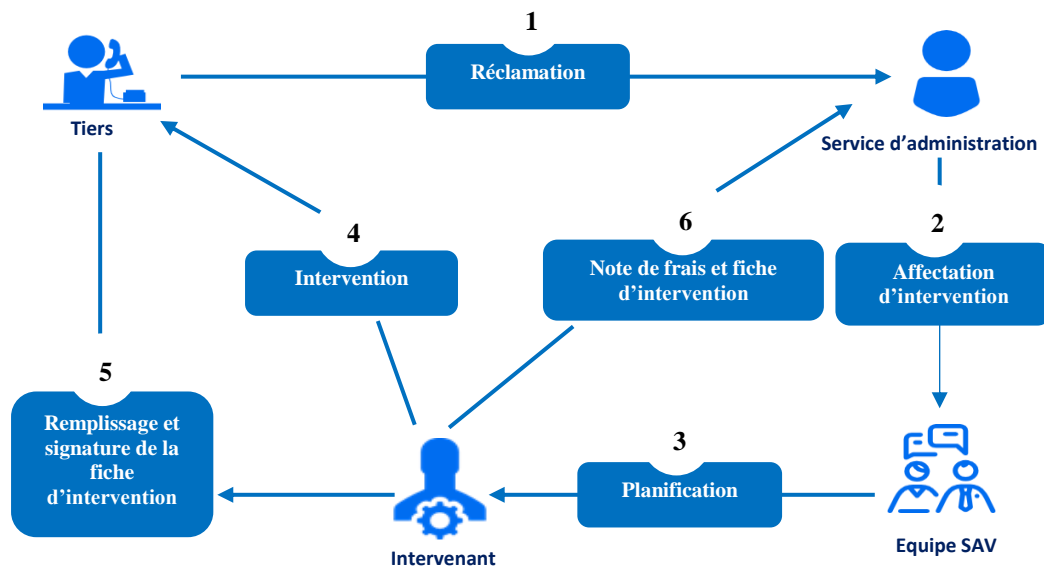


Figure 3: scénario d'intervenir une réclamation

1. Le tiers valide une réclamation auprès de service d'administration.
2. Le service de d'administration affecte l'intervention à une équipe SAV suivant les spécialités.
3. L'équipe SAV concerné planifie l'intervention pour l'intervenant qui est disponible.
4. l'intervenant fait intervenir la problématique de tiers.
5. A la fin d'intervention l'intervenant et le tiers signe une fiche d'intervention dont on a un rapport écrit par l'intervenant.
6. En fin la fiche d'intervention est déposée auprès de service d'administration et la déclaration de note de frais par l'intervenant.

3.2 Spécification des besoins

Dans cette partie nous allons spécifier les besoins fonctionnels et non fonctionnel du projet.

3.2.1 Périmètre des besoins

Le périmètre des besoins est délimité principalement par les activités des services suivants :

❖ **Service administratif :**

- Gestion des contrats de maintenance.
- Gestion des équipes SAV.
- Gestion de processus SAV.

❖ **Les équipes SAV :**

- **Chef d'équipe :** Gestion de son équipe et de processus SAV
- **Les intervenants :** Gestion de processus SAV.

❖ **Tiers :** Réclamation de problème, suivie des interventions.

3.2.2 Les besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système.

L'application doit permettre à l'utilisateur les fonctionnalités suivantes :

3.2.2.1 Gestion des contrats de maintenance

Ce module doit fournir au service administratif les fonctionnalités suivantes :

- ❖ La gestion des tiers y compris toutes les opérations classiques.
- ❖ La gestion de tous les types des contrats (maintenance, d'assistance SAGE) :
 - Enregistrement des contrats en spécifiant le type de contrat et les informations nécessaires.
 - Gestion de la validité des contrats en gros et par contrat.
 - Gestion de La résiliation des contrats.

- Gestion de fiche d'inventaire pour les contrats de maintenance.
- Gestion des progiciels ou modules liés aux contrats SAGE.

3.2.2.2 Gestion des équipes SAV

Il s'agit de la gestion des individus et l'affectation de ces derniers aux équipes SAV.

❖ Gestion des individus :

- Enregistrement des individus en spécifiant les coordonnées nécessaires.
- Enregistrement des compétences des individus.
- Suivre des équipes par individu.

❖ Gestion des équipes :

- Gestion des affectations (affecter, désaffecter) des individus aux équipes en spécifiant leurs rôles.
- Suivre des individus par équipe

3.2.2.3 Gestion de processus SAV.

Dans ce module l'application doit assurer la gestion de processus SAV de la réclamation de maintenance par les tiers en créant un ticket à la résolution de la problématique et la fermeture de ticket.

Une réclamation est un ticket crée par le tiers dont il exprime son besoin de maintenance.

Les fonctionnalités principales sont :

❖ Gestion des tickets de maintenance :

- Création des tickets.
- Annulation des tickets.
- Affectation des tickets aux équipes.
- Affectation des tickets aux intervenants.
- La clôture des tickets

❖ Gestion des interventions :

- Ouverture d'intervention.
- Gestion des planifications pour une intervention.

- Déclaration et validation des notes de frais.
- Fermeture d'intervention.
- Impression de fiche d'intervention.

Ces besoins fonctionnels de cette partie ont pour objectif d'avoir une vision générale sur la productivité d'un individu par équipes en particulier dans une période déterminée entre deux dates.

❖ **Individu :**

- Avoir le taux de réussite (le pourcentage des tickets résolus).
- La moyenne de temps pour ouvrir une intervention.
- La moyenne de temps de résolutions des tickets.

❖ **Chef d'équipe :** La moyenne de temps pour affecter un ticket.

3.2.3 Les besoins non fonctionnels

Les besoins non fonctionnels décrivent les propriétés que le système doit avoir.

Voici la description des besoins non fonctionnels que doit satisfaire ce système :

Exigence	Critère	Description
Exploitabilité	Efficacité	Les impacts en termes de performance doivent être pris en compte lors des développements. La consommation des ressources (CPU, mémoire, etc.) doit être minimisée.
	Lisibilité	Le code et la documentation doivent être compréhensibles par simple lecture, en respectant les normes de développement et les règles de gestion documentaire.
Maintenabilité	Modularité	L'application doit être découpée fonctionnellement
	Traçabilité	Toute modification apportée au code ou à la documentation doit être tracée et expliquée.
	Adaptabilité	

		L'ajout, modification, correction ou suppression de fonctionnalités doivent être facilement réalisables.
Capacité fonctionnelle et conviviale	Aptitude	Les composants développés doivent respecter les spécifications fournies par M2aSoft.
	Complétude	Les composants développés doivent couvrir tout le périmètre des spécifications sans omission.
	Ergonomie	L'application doit être facilement utilisable et disposer d'interfaces conviviales.
Sécurité	Intégrité et cohérence des données	Les données ne doivent subir aucune altération ou perte lors de leur manipulation.
	Disponibilité	L'application doit être disponible en tout moment pour les utilisateurs.
	Confidentialité et gestion des rôles	L'application doit garantir la confidentialité des données des utilisateurs et des rôles.

3.3 Solution attendues

D'après la présentation de projet et la spécification des besoins, notre solution consiste à développer une application web qui permet de gérer les 3 modules principaux de notre projet en se basant sur les Framework JAVA JEE, avec une interface web simple à utiliser tout en gérant les rôles des différents acteurs de l'application.

3.4 Objectif

M2aSoft gère le service SAV actuellement à travers une application web M2aSav, et une application Bureau de gestion des contrats.

En étudiant l'existant et la spécification des besoins, nous voulons offrir un meilleur service à l'aide d'un véritable outil de gestion des contrats des maintenances et analyse de performance des équipes SAV.

Le premier objectif repose sur une nouvelle analyse des besoins à partir de la conception et la modélisation à la réalisation de projet tout en combinant les fonctionnalités des solutions existantes avec la résolution des lacunes actuelles. et l'ajout d'autres fonctionnalités qui sont mentionnées dans la partie précédente afin d'évaluer la performance des équipes SAV et la productivité des individus.

Le deuxième objectif de l'application doit atteindre un bon taux de satisfaction au niveau de la relation clientèle et ainsi pour les processus des opérations des équipes SAV.

3.5 Livrables

phase	Livrable	Responsable
Étude des besoins	Cahier de charges	Développeur du projet
Analyse et Conception	Diagrammes UML	Développeur du projet
Développement	Application web	Développeur du projet
Testes et recette	Maintenance du code	Développeur du projet
Documentation	Rapport du projet	Développeur du projet

4. Démarche de projet

Dans cette partie nous allons voir la méthode suivie lors de la réalisation de ce projet ainsi qu'une présentation du diagramme de Gant qui illustre l'enchaînement des tâches réalisés.

4.1 Gestion de projet Agile

Les méthodes agiles visent à fluidifier la relation entre utilisateurs et informaticiens. Elles ont souvent été mises en œuvre de façon dogmatique. Désormais, la réussite passe par certaines précautions et compromis. Lorsqu'il s'agit de développer une application, de paramétrer un progiciel ou de monter un site web un tant soit peu complexe, les interactions entre informaticiens et utilisateurs sont souvent difficiles à organiser.

La démarche traditionnelle consiste à demander aux utilisateurs de réaliser un cahier des charges qui permet aux développeurs de se mettre au travail. Ces développeurs livrent le résultat plusieurs mois plus tard et sont alors confrontés à la réaction d'utilisateurs estimant avoir été

mal compris ou dont les besoins ont évolué. C'est pour combattre cette relation rigide que sont nées les méthodes dites agiles, comme Scrum (la plus connue), EXtreme Programming ou Unified Process. Elles préconisent une relation permanente avec les utilisateurs, une livraison régulière de versions de l'application, ainsi qu'un mode de travail horizontal, avec une responsabilisation de chaque développeur.

4.2 Agilité pratique

Dans notre projet on a fait la bonne pratique.

4.3 Planning du projet

Le planning de projet est illustré dans le diagramme de Gant suivant dont on a chaque tâche et sa date de début et de fin.



Nom	Date de...	Date de fin
● Prise de contact	06/02/17	13/02/17
● Cahier de charges	14/02/17	28/02/17
● Présentation de cahier de charges	01/03/17	01/03/17
● Analyse du projet	02/03/17	30/03/17
● Présentation d'etat d'avancement	03/04/17	03/04/17
● initiation d'architecture	04/04/17	10/04/17
● Réalisation	11/04/17	26/05/17
● Test et Recettes	26/05/17	31/05/17

Figure 4: Planning du projet

On remarque que les tâches qui ont dominées le planning sont : l'analyse du projet et la réalisation.

4.4 Equipe de réalisation de projet

Nom et Prénom	Fonction/Rôle dans le projet
Zakaria haouazim	Stagiaire/Développeur
Abdeslam Essahraoui	Chef du projet

❖ **Développeur :**

- Réalisation de cahier de charges.
- Analyse des besoins.
- Conception et modélisation.
- Codage de l'application
- Effectuation des tests unitaire.
- Réalisation des livrables.

❖ **Chef du projet :**

Contrôle et validation de toutes les étapes effectuées par le développeur.

5. Conclusion

A travers ce chapitre, j'ai présenté la société M2aSoft et l'équipe de développement spécifique, puis nous avons étudié le projet et ses principaux besoins en présentant l'ensemble des fonctionnalités du futur système et finalement j'ai présenté l'organisation et la démarche du projet.

Chapitre 2 : Analyse et Conception

1. Introduction

Dans ce chapitre nous allons effectuer une analyse du projet en passant par l'étude et la critique de l'existant, a la présentation de la nouvelle solution M2aSav, puis nous aborderons l'étape de la conception en présentant des diagrammes d'UML.

2. Analyse de projet

2.1 Etude d'existant

Pour la gestion des contrats de maintenance M2aSoft utilise **une application Bureau** développé en **visuel basic**.

Le processus SAV, les réclamations des tiers sont gérés sous forme des tickets créés dans une **application web** développé en **asp.net**.

2.1.1 Application Bureau

Cette application permet de gérer les tiers et leurs contrats.

Fonctionnalités :

- gestion des tiers.
- enregistrement des contrats.
- enregistrement d'inventaire des contrats de maintenance.

Architecture technique :

- Technologie de développement : Visuel basic
- Base de données : SQL server.

Etat actuel : non opérationnelle.

2.1.2 Application : M2aSAV

Cette application couvre une sous partie de la gestion des équipes et de processus SAV c'est la partie des réclamations par les tiers.

Une réclamation est déclarée par le tiers sous forme d'un ticket (1) puis l'administration affecte ce ticket à un consultant pour intervention (2) comme illustre la figure suivante :

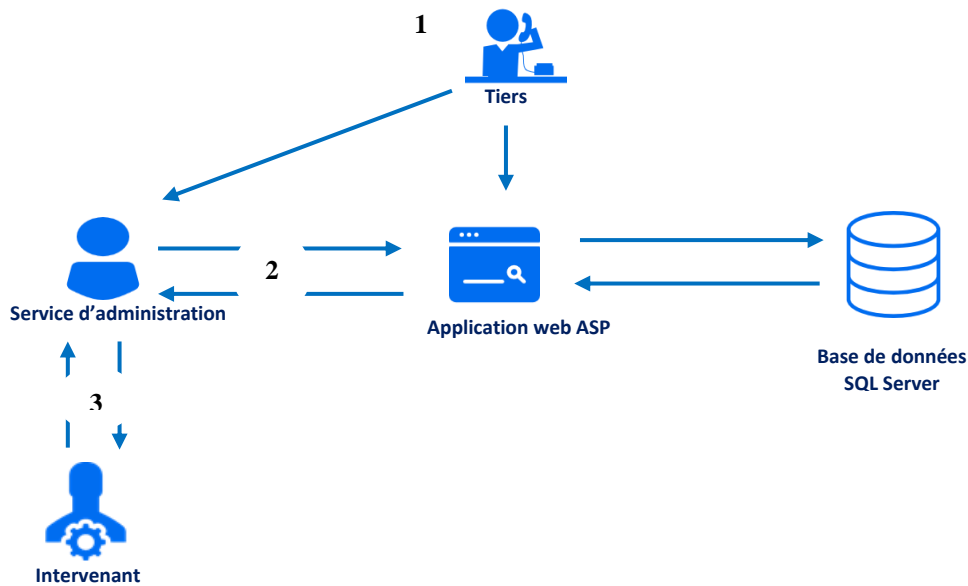


Figure 5: scénario de gestion des tickets et des interventions

Fonctionnalités :

- Réclamation des tickets de maintenance par les tiers.
- Réclamation d'un ticket par l'administration pour un tiers.
- Affectation des tickets aux consultants par l'administration.
- Suivi d'intervention des tickets par l'administration et les tiers.

Architecture technique :

- Technologie de développement : Asp.net.
- Base de données : SQL server.

Etat actuel : **opérationnelle.**

2.2 Critique de l'existant

D'après le cahier de charges et l'étude d'existant, voici un bilan qui permet d'évaluer les fonctionnalités du système actuel :

- ❖ A priori chacune des deux solutions possède sa propre base de données.
 - Redondance des données.
- ❖ L'application Bureau (gestion des contrats) : solution morte
 - Pas gestion des contrats et leurs produits associés.
 - On ne peut pas contrôler l'objet d'une réclamation d'un tiers (produit associé au contrat ou non).
- ❖ L'application Web (M2aSAV) :
 - Absence d'organisation hiérarchique des équipes de service SAV.
 - On ne peut pas gérer des équipes.
 - On ne peut pas gérer des profils (individus).
 - On ne peut pas suivre et évaluer la performance d'une équipe ou d'un individu.
 - La gestion des réclamations des tickets est assurée par le service d'administration.
 - L'affectation se fait directement à un individu de services SAV.
 - Pas de traçabilité sur les tickets en des points de temps ou par équipe.
 - Pas d'accès au système par les équipes (individus).
 - Gestion insuffisante des interventions.
 - Pas de gestion des planifications pour les interventions.
 - Pas de gestion des notes de frais des intervenants.

En résumé le system actuel ne permet pas d'évaluer la productivité des individus, cette productivité permet d'avoir les types de problématiques ou un individu est plus productif.

2.3 Solution : M2aSAV

M2aSAV est une nouvelle version de la solution existante M2aSAV. Cette version va gérer toutes les 3 volets entourant le service SAV de M2aSoft, en combinant les fonctionnalités des solutions existantes (Application Bureau, M2aSAV) avec la résolution des lacunes actuels.

Les processus métiers de cette solution sont décrits dans les parties suivantes.

3. UML

3.1 Choix d'UML



UML (Unified Modeling Language, que l'on peut traduire par langage de modélisation unifié) est une notation permettant de modéliser un problème de façon standard.

Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

3.2 Les besoins

Concevoir une application est un processus qui peut être relativement complexe. En effet, lors de l'élaboration d'un projet, de nombreuses données, concept, processus métiers, acteurs et autres se mélangent et peuvent très vite donner le tournis. Afin d'y voir plus clair, on peut regrouper ces différents termes par familles et les organiser de façon à former un ensemble cohérent, fonctionnel et conforme aux parties en présence.

3.3 La maîtrise d'œuvre

Pour les personnes qui conçoivent et réalisent un logiciel, maîtriser ce qu'il faut faire, ce que l'on a déjà fait et ce qu'il reste à faire est primordial.

Cela regroupe la conception, la communication ainsi que la planification. La modélisation UML fournit un formalisme relativement simple à appréhender mais qui s'avère très puissant à l'utilisation. Il sert notamment de support à la réflexion en permettant de modéliser les

interactions entre les différents utilisateurs et la solution logicielle. La plupart des documentations techniques et fonctionnelles peuvent ainsi voir leur lisibilité grandement améliorée grâce aux différents types de diagrammes supportés par UML.

3.4 Visual Pradigm

Visual Paradigm for UML est, comme son nom la laisse supposer, un logiciel permettant aux programmeurs de mettre en place des diagrammes UML. Disposant d'un outil créant des rapports personnalisables aux formats PDF, Word ou HTML afin de les partager et les publier sur Internet, cette application est compatible avec de nombreuses applications, standards et environnements. Ainsi, vous pourrez générer notamment des diagrammes de séquences ou de cas d'utilisation et ainsi produire du code source dans de nombreux langages comme le Java ou encore le C++, ou bien faire l'inverse, générer des diagrammes à partir de code déjà existant.



3.5 Identifications des acteurs.


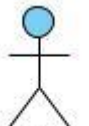
Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif Matériel ou autre système) qui interagit directement avec le système étudié.

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

Dans ce qui suit nous allons présenter les acteurs de notre application et leurs rôles brièvement avant d'aborder les limites et les détails d'utilisation respectivement dans les diagrammes de cas d'utilisation et de packages.

Les acteurs principaux sont :

Acteur	Rôle
 <p>Administrateur</p>	<p>L'administrateur est un responsable de service d'administratif de m2aSoft, il peut :</p> <ul style="list-style-type: none"> • Gérer les contrats de maintenance. • Gérer les équipes SAV. • Gérer et suivre le processus SAV.
 <p>Chef d'équipe</p>	<p>Chef d'équipe est un individu qui gère un ensemble d'intervenants dans équipes SAV, il peut :</p> <ul style="list-style-type: none"> • Gérer son équipe en suivant le processus SAV

	l'intervenant est un individu d'une équipe SAV il peut : <ul style="list-style-type: none"> Gérer et suivre le processus SAV.
	Tiers est un client qui possède un contrat que ce soit de maintenance ou sage alors il peut : <ul style="list-style-type: none"> Réclamer des tickets de maintenance et interagir avec le processus SAV

3.6 Diagramme d'activité

Le Diagramme d'activité est la représentation du processus telle qu'elle a été élaborée lors du travail qui a préparé la modélisation : il montre l'enchaînement des activités qui concourent au processus.

Dans cette partie nous allons utiliser le diagramme d'activité pour présenter les processus des 3 modules de notre application

3.6.1 Gestion des contrats :

Chaque tiers qui a un contrat sera enregistré dans le système et par suite ces contrats seront liés à un ensemble de produits comme illustre la figure suivante :

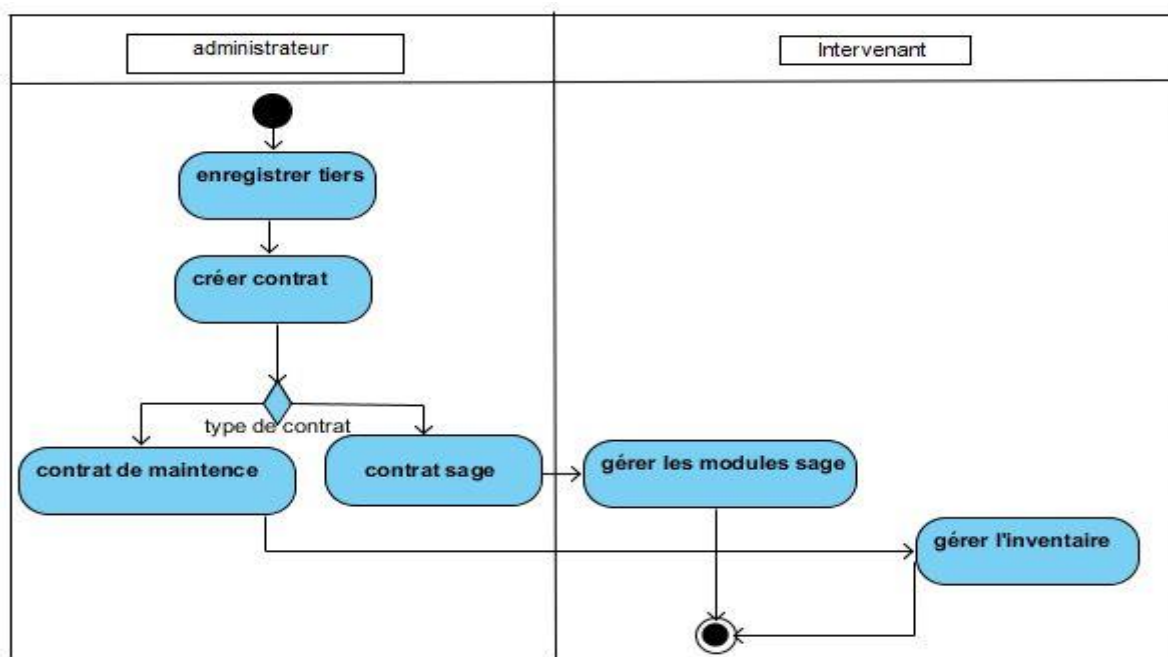


Figure 6: diagramme d'activité de gestion des contrats

3.6.2 Gestion des équipes SAV

La figure suivante décrit l'activité principale de la gestion des équipes SAV qui consiste à créer des équipes enregistré des individus avec leurs compétences puis, on peut affecter ou désaffecter un individu aux équipes.

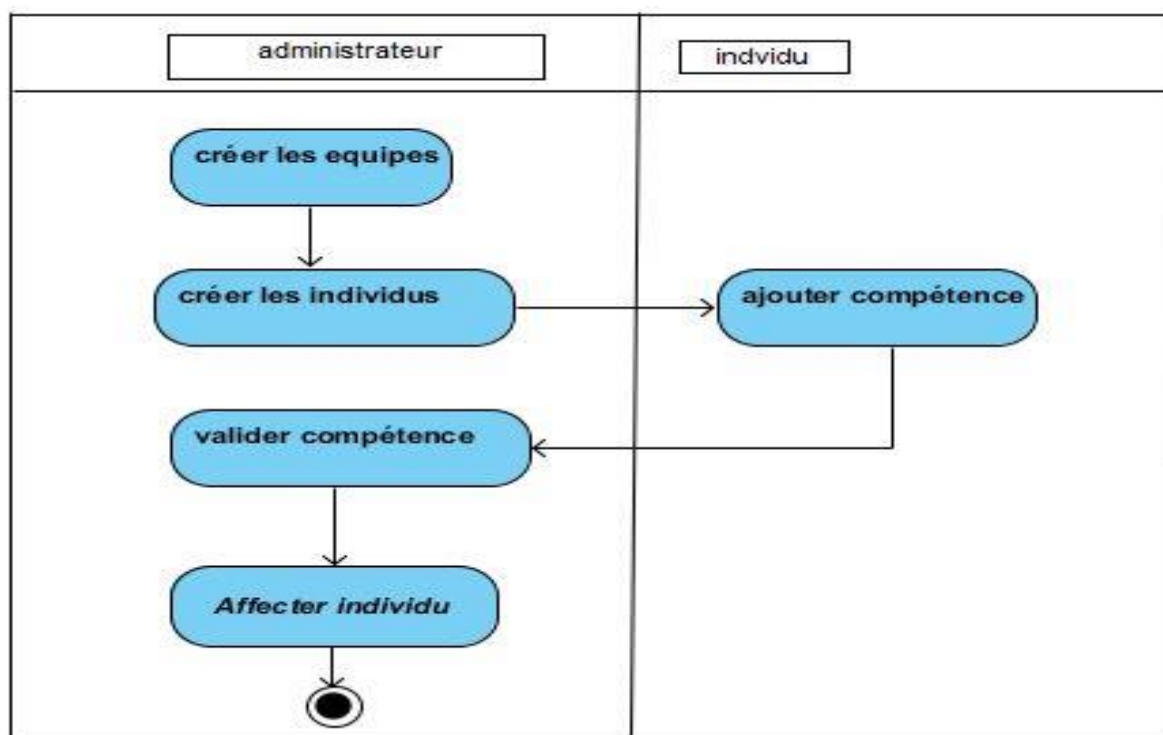


Figure 7: diagramme d'activité gestion des équipes SAV

3.6.3 Gestion de processus SAV

Le processus SAV commence par la création d'un ticket par un tiers, puis le service administratif affecte le ticket un une équipe (intervenant), Finalement l'intervenant doit gérer l'intervention pour ce tiers en finissant par la signature de fiche d'intervention.

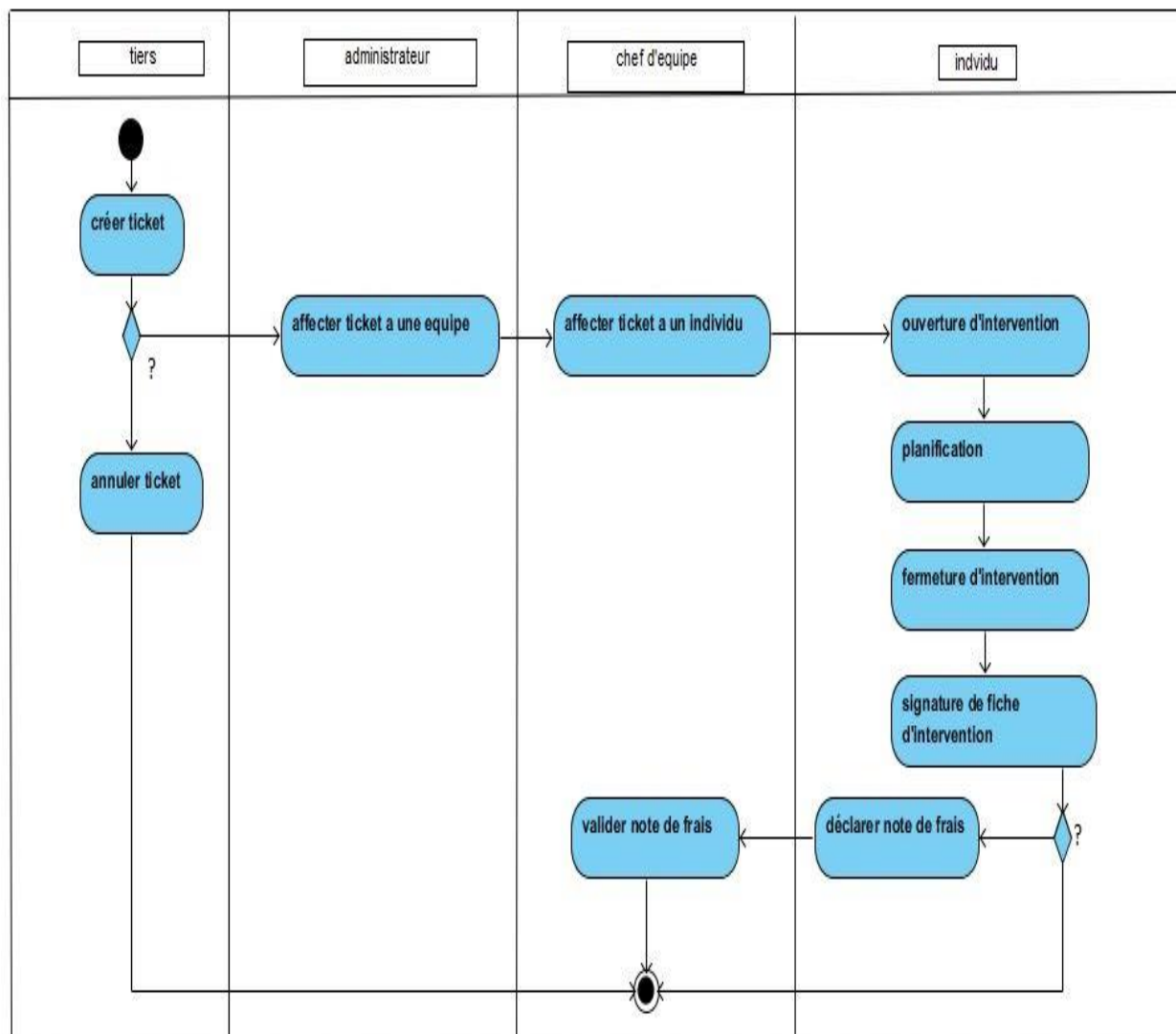


Figure 8: diagramme d'activité gestion de processus SAV

3.7 Diagramme de packages

Un package en UML (ou *paquetage* en français) est un groupe d'éléments, dans le but de les grouper dans des ensembles cohérents. Un package peut contenir la plupart des éléments UML : classes, objets, cas d'utilisations, composants, etc. Il peut également contenir des packages, créant une hiérarchie complète.

L'avantage des packages est qu'ils permettent de structurer les diagrammes et donnent une vision globale plus claire.

Nous utiliserons ce diagramme pour les diagrammes de cas d'utilisation et le diagramme de classes.

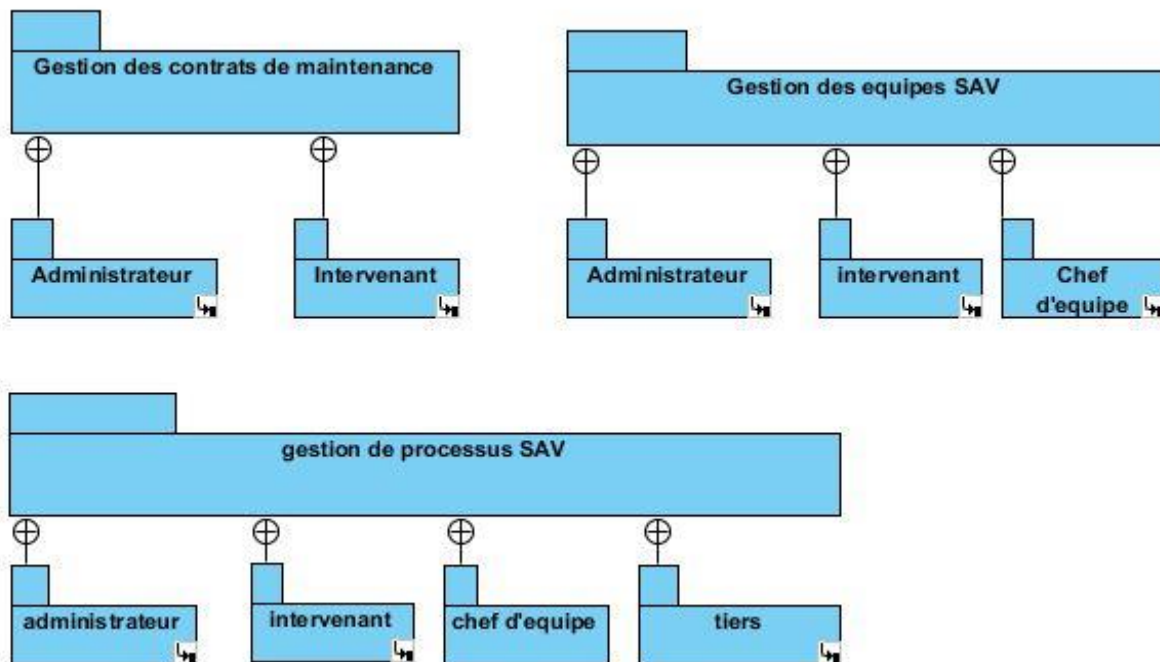


Figure 9: digramme de packages générale

Le diagramme de packages ci-dessus a pour objectif de donner une vision générale sur les cas d'utilisation et les acteurs et les relations entre eux.

3.8 Diagramme de 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. Les cas d'utilisation « use cases » permettent de structurer les besoins des utilisateurs aussi bien qu'ils expriment les exigences du système vis-à-vis des utilisateurs en partant du principe que les objectifs du système sont tous motivés. Les cas d'utilisation permettent ainsi de classer les acteurs et structurer les objectifs du système.

Dans ce qui suit on présentera les cas d'utilisation de chaque package et pour chaque acteur.

3.8.1 Cas d'utilisation Authentification.

Avant d'utiliser l'application chaque acteur doit être authentifié

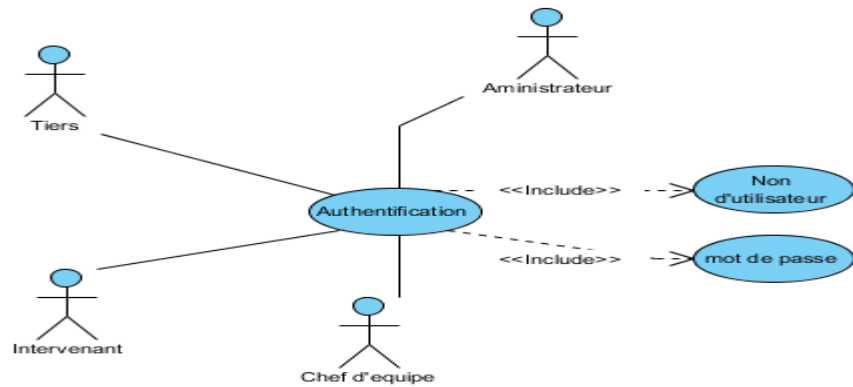


Figure 10: cas d'utilisation authentification

3.8.2 Package de gestion des contrats de maintenance

Cas d'utilisation	Gestion des contrats de la maintenance
But	Avoir une référence des tiers et organisé la relation SAV avec les tiers
Acteur	Administrateur, Intervenant
Pré condition	L'acteur doit être authentifié
Description	Ce module gère les contrats des tiers après l'enregistrement de ces derniers et le produits sous inscrits (inventaire, sage).

3.8.2.1 Package de gestion des contrats de maintenance-Administrateur

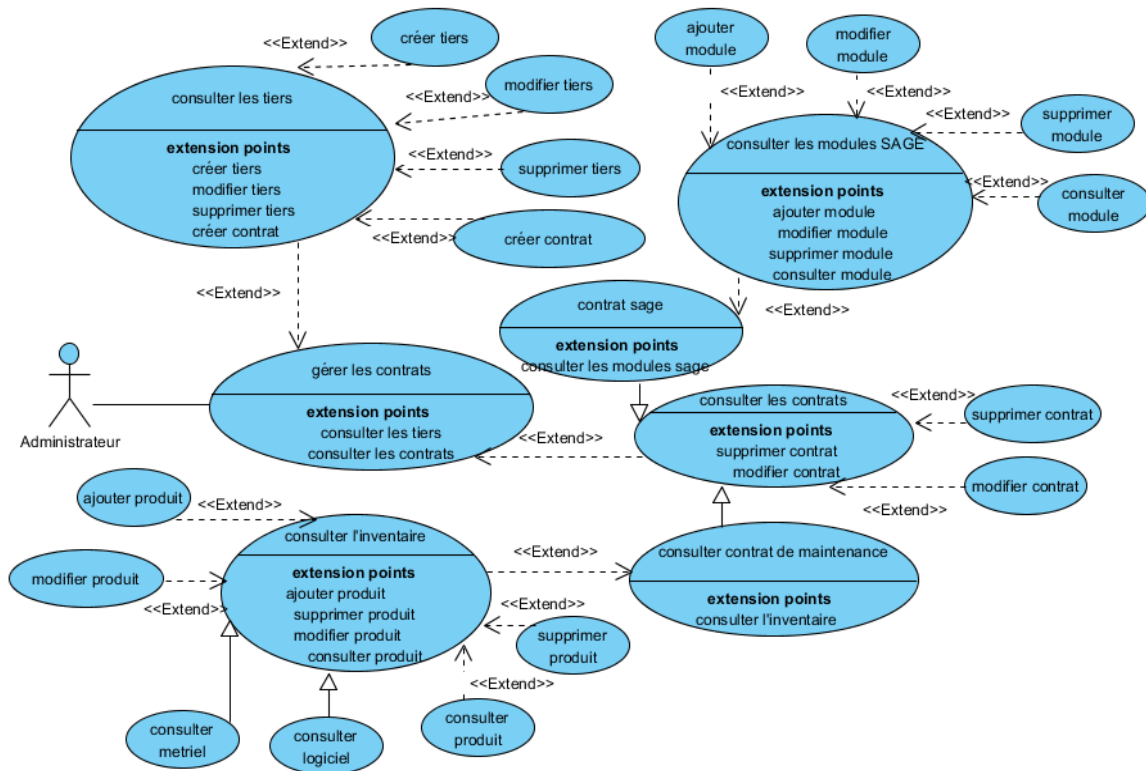


Figure 11: cas d'utilisation package de gestion des contrats de maintenance-Administrateur

3.8.2.2 Package de gestion des contrats de maintenance-Intervenant

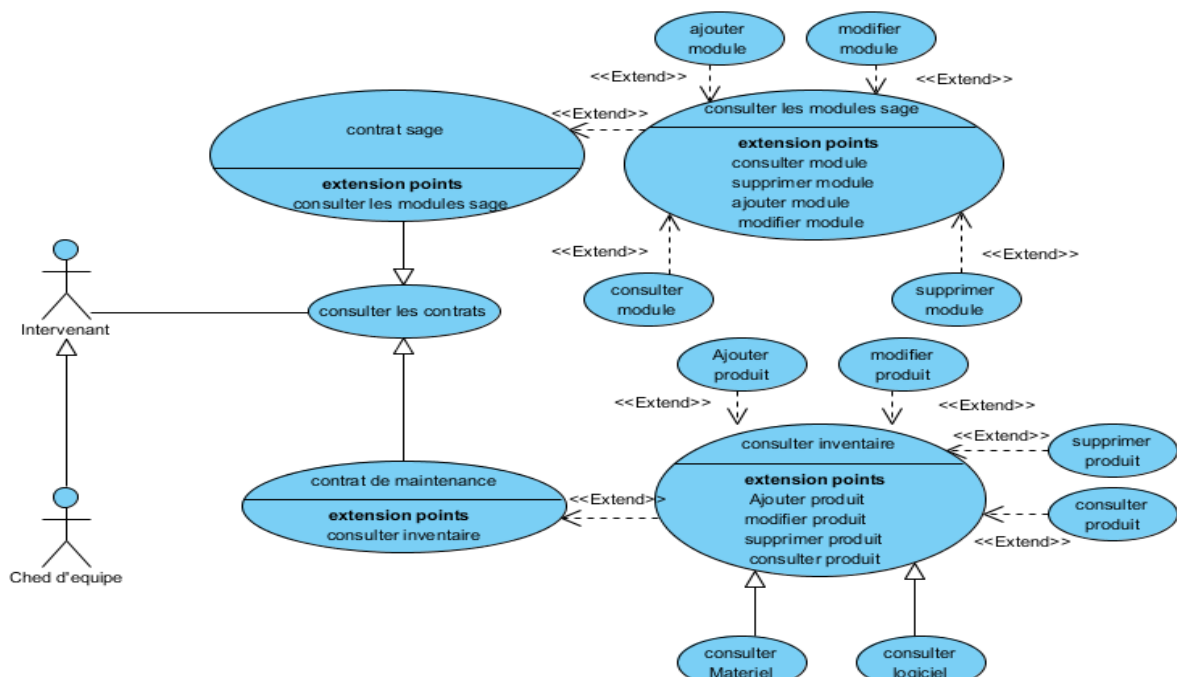


Figure 12: package de gestion des contrats de maintenance-Intervenant

3.8.3 Package de gestion des équipes SAV

Cas d'utilisation	Gestion des équipes SAV
But	Évalué la performance des équipes SAV et la productivité des individus
Acteur principale	Administrateur, Chef d'équipe
Pré condition	L'acteur doit être authentifié
Description	Ce cas gère l'ensemble des individus d'une équipe et les affectations des tickets et par suite on peut avoir les statistiques sur les tickets ce soit par équipe ou par individu

3.8.3.1 Package de gestion des équipes SAV-Administrateur

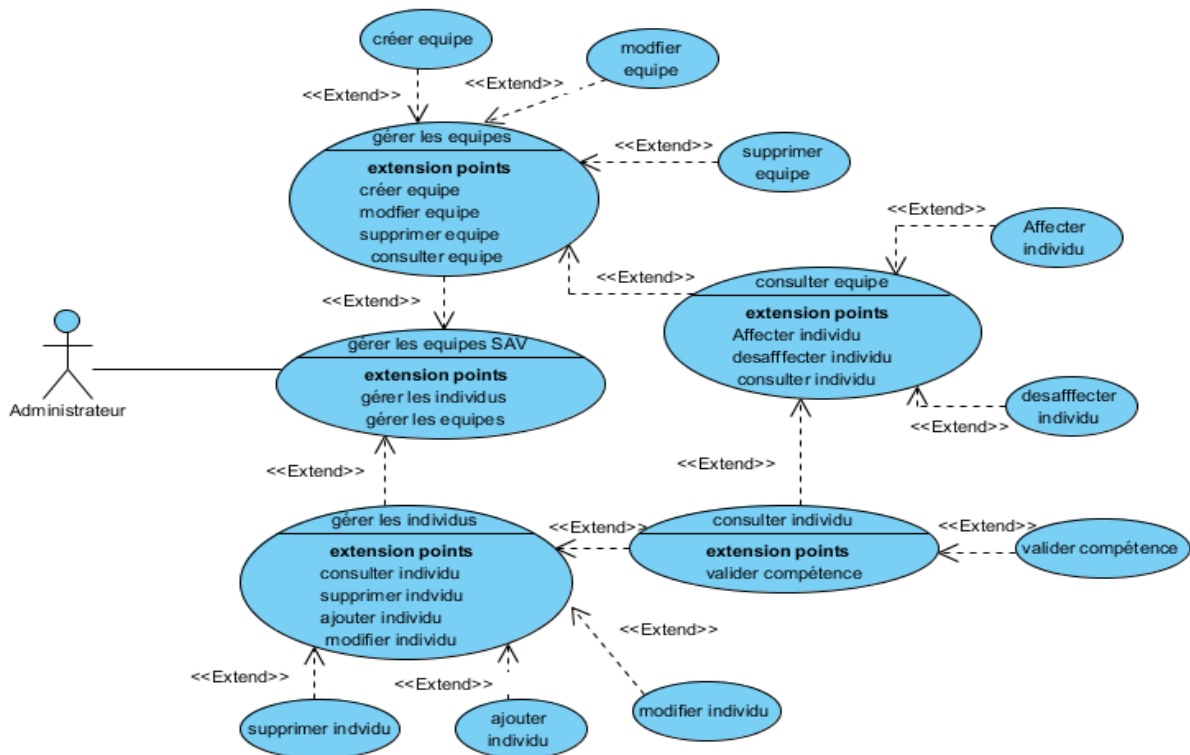


Figure 13: package de gestion des équipes SAV-Administrateur

3.8.3.2 Package de gestion des équipes SAV-Chef d'équipe

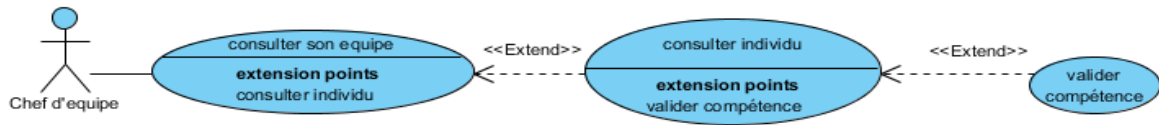


Figure 14: package de gestion des équipes SAV-Chef d'équipe

3.8.4 Package de gestion de processus SAV

3.8.4.1 Package de gestion de processus SAV-Tiers

Cas d'utilisation	Gestion de processus SAV
But	Orienté et organisé les réclamations des maintenances
Acteur principale	Tiers
Pré condition	L'acteur doit être authentifié
Description	Ce cas permet au tiers de créer un ticket de réclamation puis la possibilité de suivre ce ticket.

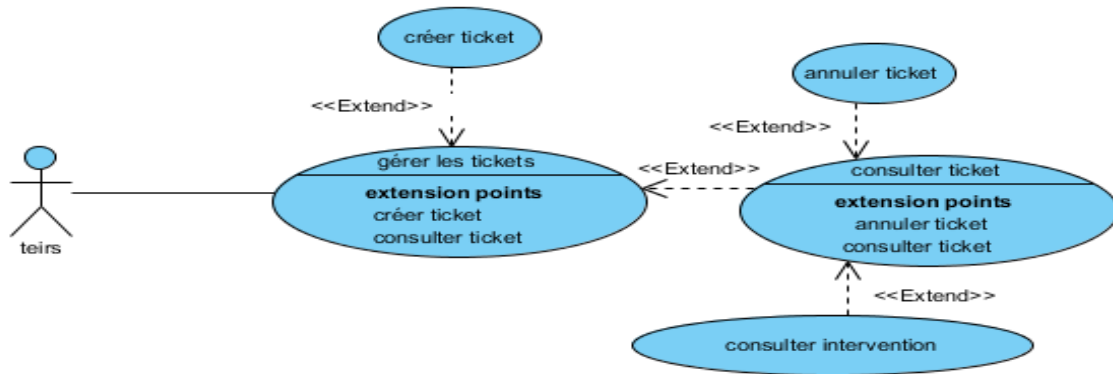


Figure 15: package gestion de processus SAV-Tiers

3.8.4.2 Package de gestion de processus SAV-Administration.

Cas d'utilisation	Gestion de processus SAV
But	Gérer les tickets selon la disponibilité des intervenants et la priorité de ticket

Acteur principale	Administrateur
Pré condition	L'acteur doit être authentifié
Description	Ce cas permet au service d'administration de bien gérer l'affectation et la suivie des tickets de l'affectation à la clôture.

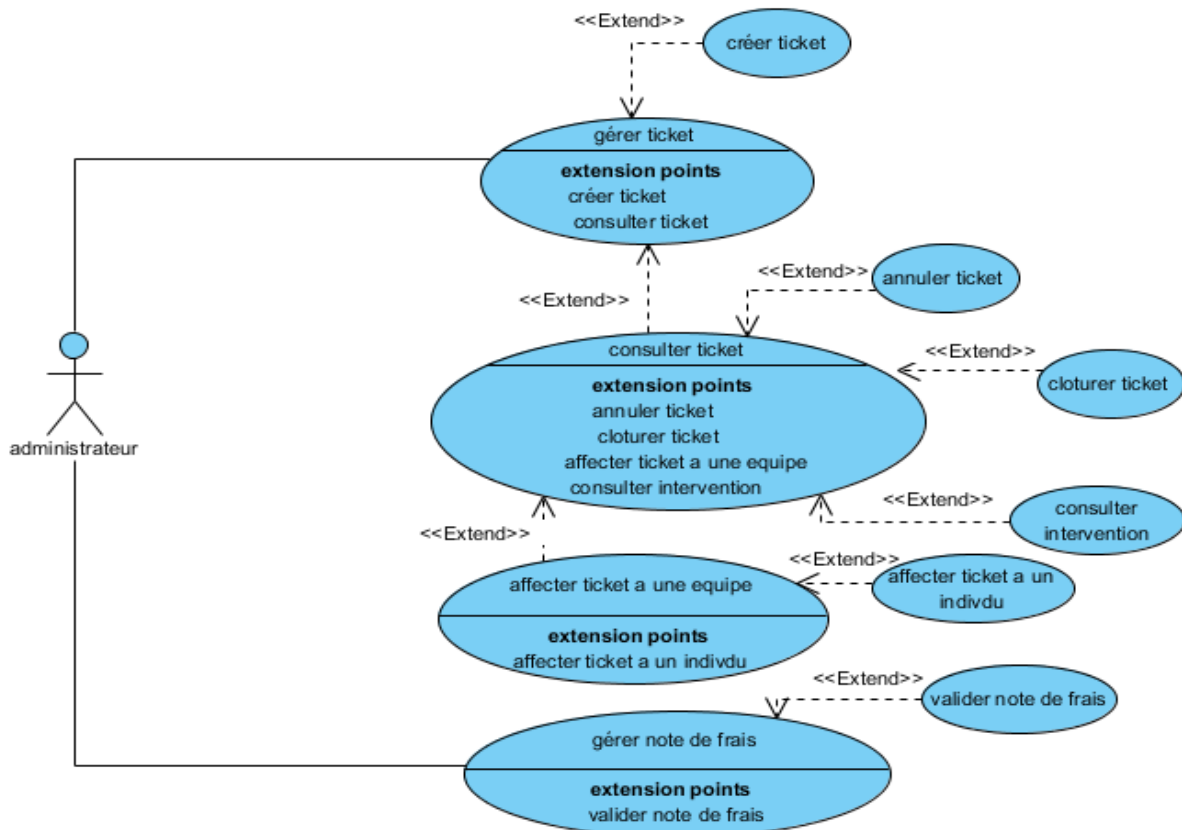


Figure 16: cas d'utilisation package de gestion de processus SAV-Administrateur

3.8.4.3 Package de gestion de processus SAV - Chef d'équipe, Intervenant

Cas d'utilisation	Gestion de processus SAV
But	Organisé le processus de résolution des réclamations.
Acteur principale	Chef d'équipe, Intervenant
Pré condition	L'acteur doit être authentifié
Description	Ce cas d'utilisation permet au chef d'équipe de gérer son équipe en affectant les tickets aux intervenants, l'intervenant par suite commence son intervention.

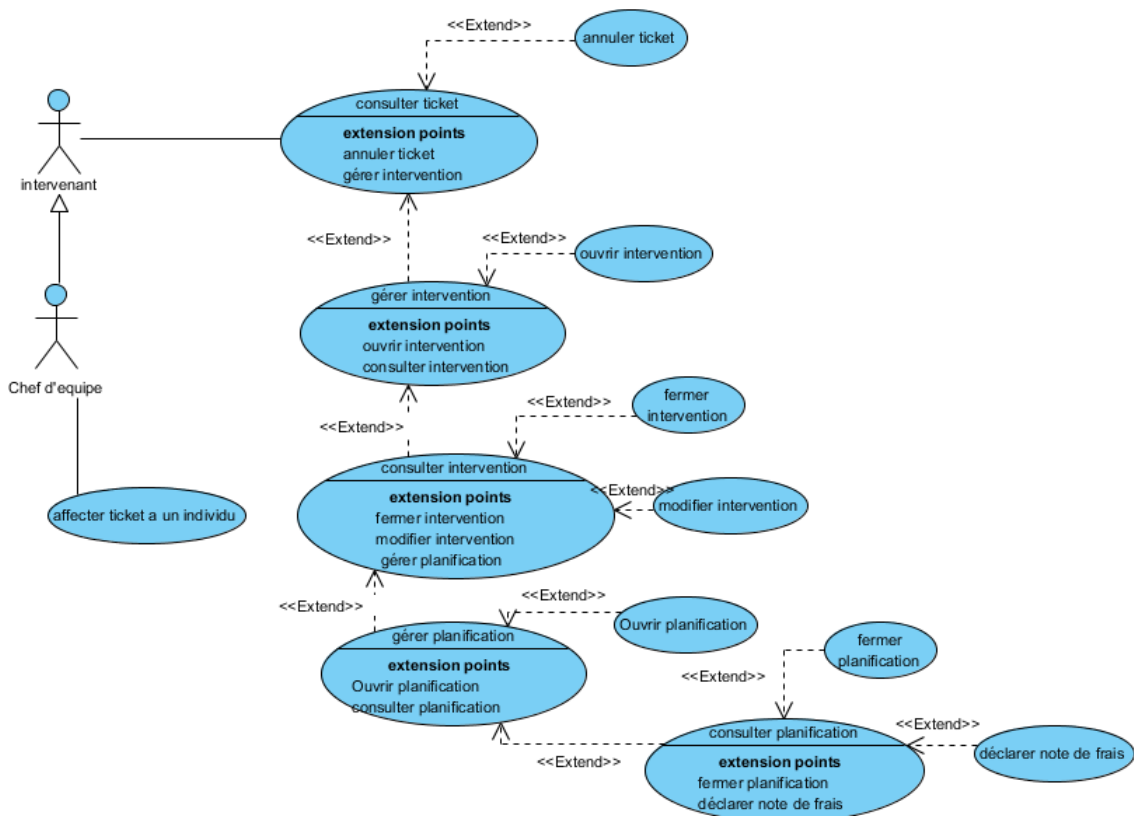


Figure 17: package de gestion de processus SAV-Chef d'équipe, Intervenant

3.9 Diagramme de classe

Un diagramme de classes est une collection d'éléments de modélisation (classes, paquetages...), qui montre la structure d'un modèle d'un point de vue statique tout en faisant une abstraction des aspects dynamiques et temporels.

Un diagramme de classe modélise l'ensemble d'informations collectées sur le système sous forme d'entités appelées « classe » ainsi que les différentes relations entre celles-ci.

En se basant sur ce diagramme, nous pouvons déduire le modèle logique des données.

3.9.1 Les packages du diagramme de classes

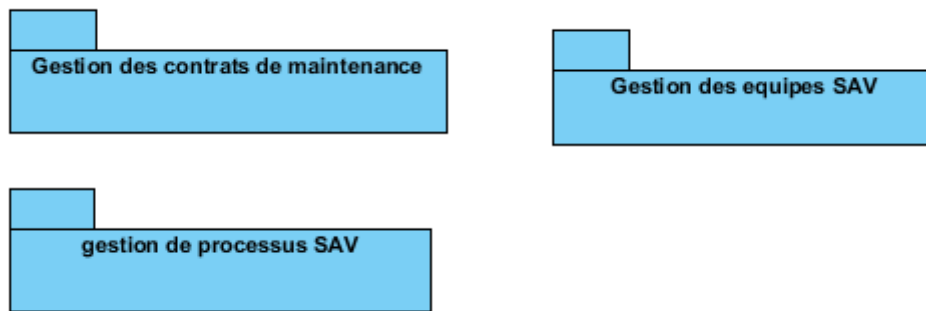


Figure 18: packages du diagramme de classes

3.9.2 Diagramme de classes globale de l'application.

Voici le diagramme de classes globale de notre application.

Package	Classes	Description
Gestion des contrats de maintenance	Tiers	contient les informations des clients de M2aSoft.
	Contrat	cette classe contient les informations d'un contrat. Elle peut être contrat de maintenance ou d'assistance SAGE
	ContratMaintenant	C'est une classe qui présente un contrat de maintenance.
	ContratSage	C'est une classe qui présente un contrat d'assistance SAGE.
	Responsable	Cette classe fait référence à un responsable d'un contrat.
	Inventaire	C'est la classe d'inventaire d'un contrat de maintenance qui est constitué de plusieurs produits matériels ou logiciels.
	Produit (Matériel, Logiciel)	ces classes contiennent les informations d'un produit de l'inventaire.
	Module	cette classe présente les produits liés à contrat d'assistance SAGE.
Gestion des équipes SAV	Equipe	C'est la classe qui présente les informations d'une équipe SAV.
	Individu	C'est la classe qui présente les informations d'un individu.
	Compétence	C'est la classe qui décrit une compétence d'un individu.
	IndividuEquipe	C'est une classe d'association entre équipe et individu, elle contient les informations

		d'affectation d'un individu à une équipe
Gestion de processus SAV	Ticket	c'est la classe qui présente la réclamation d'un tiers.
	Demandeur	C'est la classe qui fait référence au demandeur du ticket.
	AutreProduit	C'est la classe qui contient les informations d'un objet de ticket qui ne fait pas partie d'un contrat.
	TicketEquipe	C'est une classe d'association entre ticket et équipe, elle contient les informations d'affectation d'un ticket à une équipe.
	TicketIndividu	C'est une classe d'association entre ticket et équipe, elle contient les informations d'affectation d'un ticket à une équipe.
	Intervention	C'est la classe qui contient les informations d'une intervention d'un ticket.
	Prestation	C'est une intervention particulière.
	Planification	C'est la classe qui collecte les informations d'une planification.
	Note de frais	C'est la classe qui présente une note de frais pour un déplacement.
	Frais	C'est la classe qui présente un type de frais.

3.10 Diagramme de séquence

Les diagrammes de séquence représentent une succession particulière d'enchaînements, s'exécutant du début à la fin du cas d'utilisation, un enchaînement étant l'unité de description

de séquences d'actions. Un cas d'utilisation contient en général un scénario nominal et plusieurs scénarios alternatifs (qui se terminent de façon normale) ou d'erreur (qui se terminent en échec).

Dans ce qui suit on présente quelques scénarios de cas d'utilisation.

3.10.1 Diagramme de séquence-crédation de contrat

Voici le diagramme de séquence qui décrit le cas d'utilisation : créer contrat.

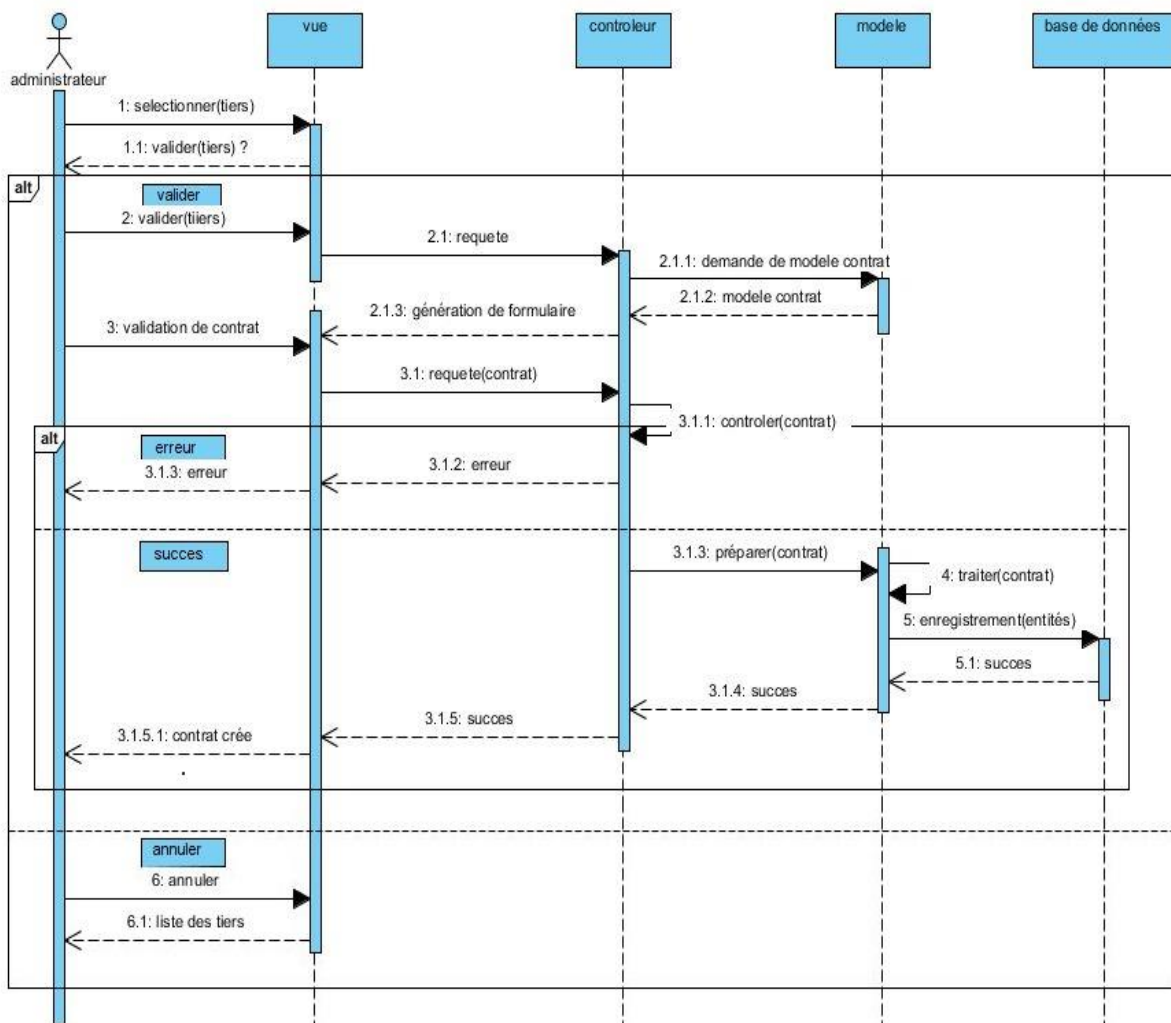


Figure 20: Diagramme de séquence créer contrat

3.10.2 Diagramme de séquence-crédation de ticket

Voici le diagramme de séquence qui décrit le cas d'utilisation : créer contrat

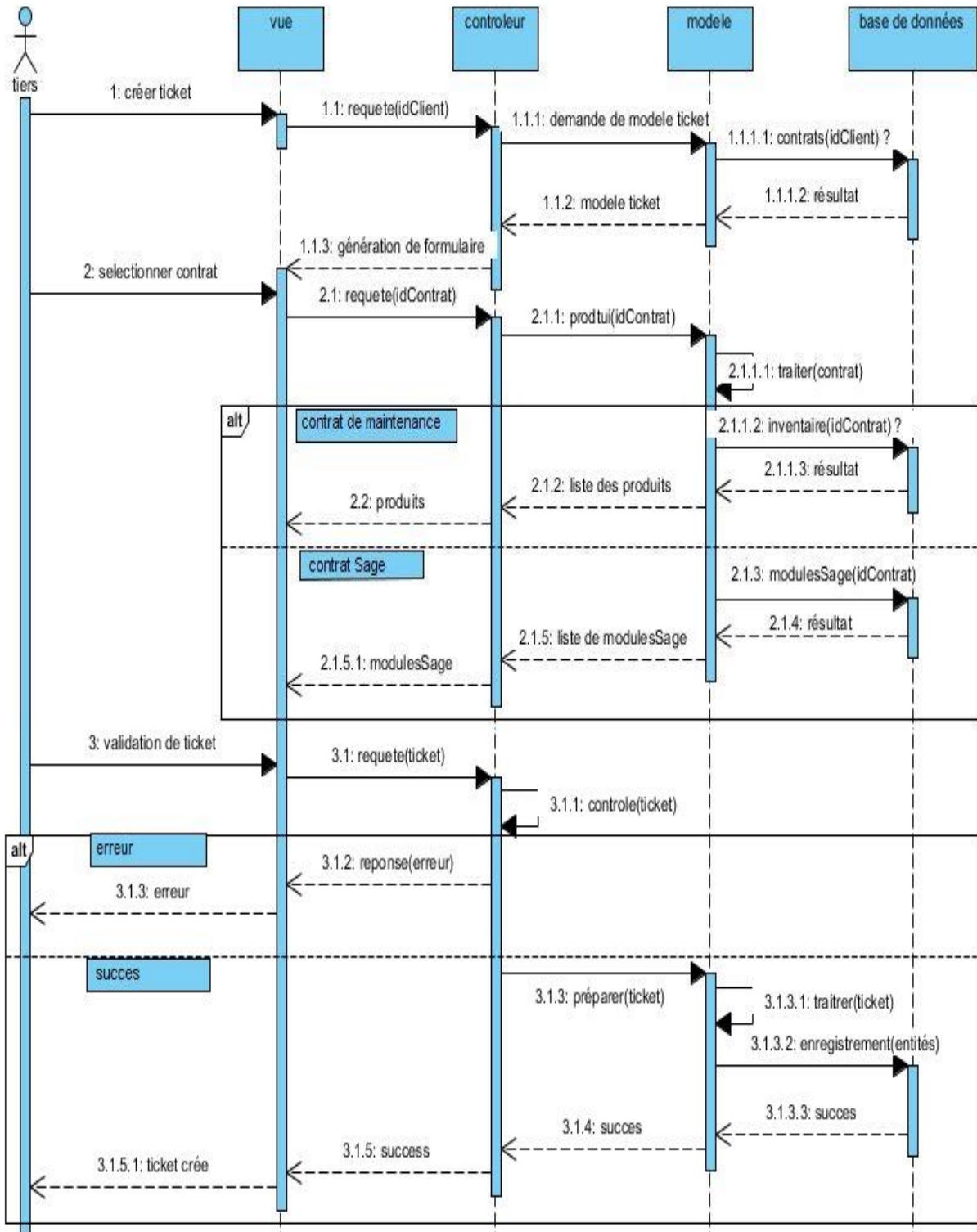


Figure 21: diagramme de séquence créer ticket

3.11 Diagramme de composants

Le diagramme de composants décrit l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les composants (qui utilisent quoi).

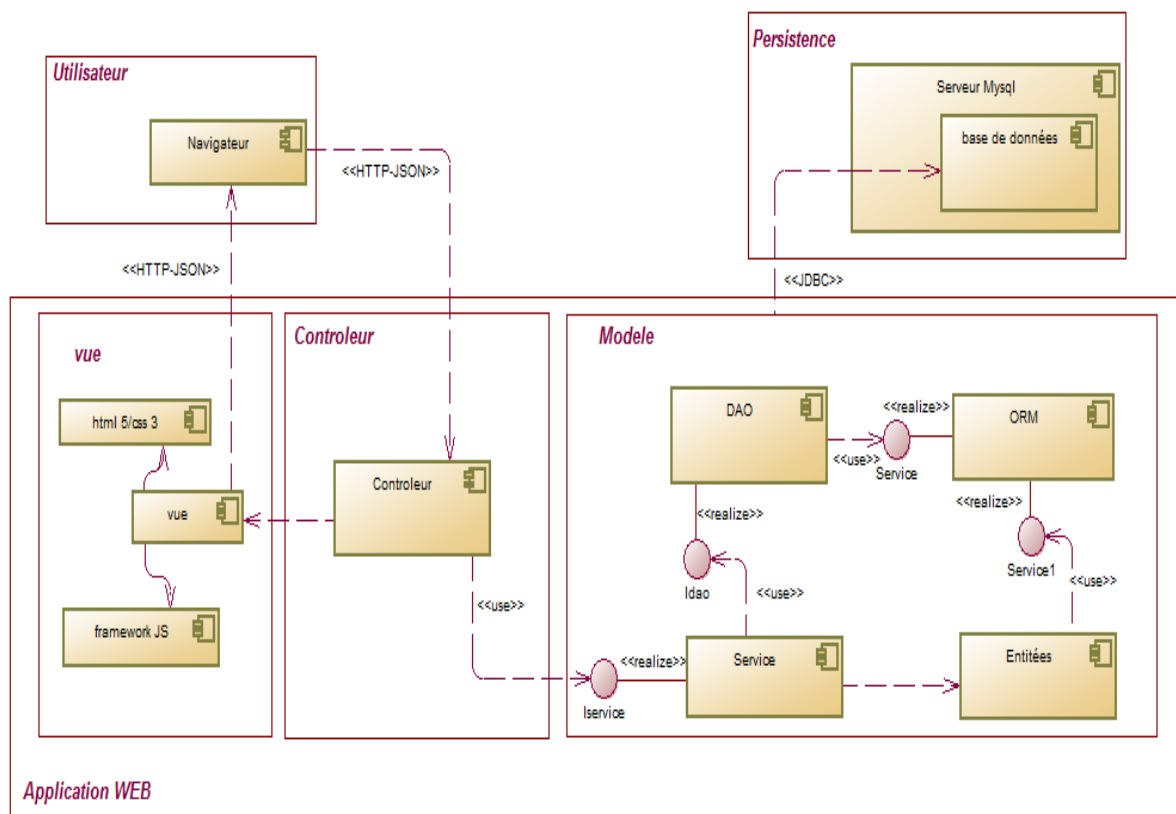


Figure 22: diagramme de composants

Notre projet est composé principalement de 3 composants principaux :

- **Navigateur** : ce composant communique avec l'application en utilisant le client web navigateur via le protocole http et format de données json.
- **Application web** : est organisé sous forme design pattern MVC
 - **Modèle** : contient toute les composants métier (service, entités, dao), qui communique entre eux via des interfaces.
 - **Contrôleur** : c'est le cœur de l'application utilise le modèle pour présenter les résultats à la couche vue

- **Vue** : C'est la couche de présentation des données.
- **Persistence** : c'est le composant qui permet de sauvegarder les données dans la base de données.

3.12 Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

Voici le diagramme de déploiement de notre application :

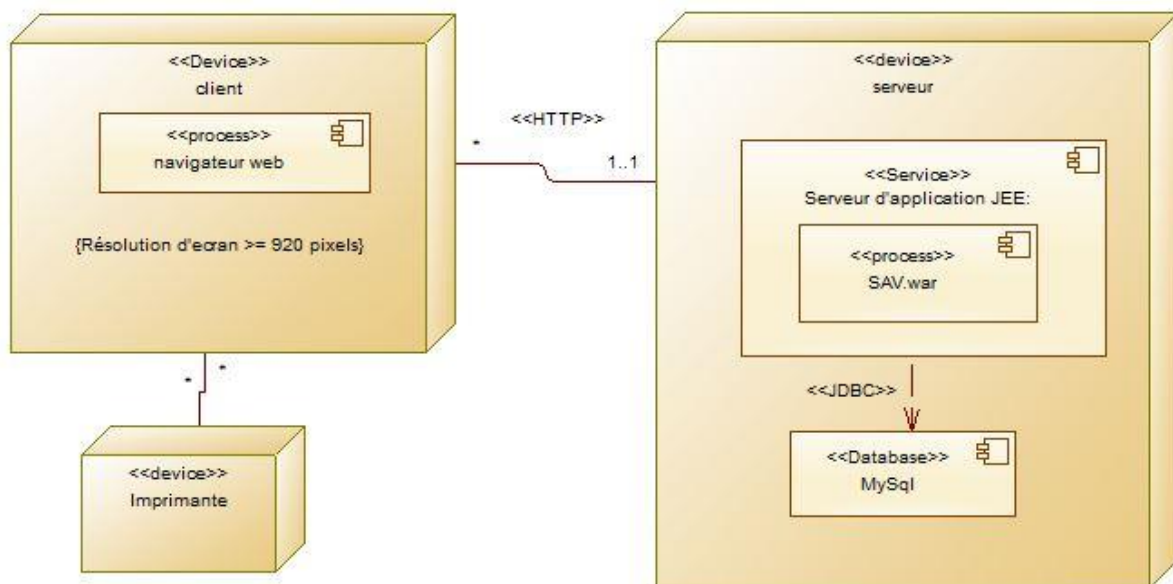


Figure 23: diagramme de déploiement

Notre application sera déployer dans un serveur qui contient un serveur d'application JEE, et un serveur de base de données MySQL.

Plusieurs clients peuvent se connecter à l'application via le protocole http.

4. Conclusion

Dans ce chapitre nous avons présenté une analyse globale du projet ainsi que la modélisation de la nouvelle solution M2aSav.

Chapitre 3 : Études technique et Réalisation

1. Introduction

Dans ce chapitre nous allons effectuer dans un premier temps une étude technique du projet en présentant l'architecture générale du projet et les technologies et Framework utilisés.

Dans la deuxième partie on présentera une série de captures d'écran et finalement nous abordons les différents types de test effectué pour cette solution.

2. L'architecture générale de l'application

Parmi Les exigences techniques d'un projet informatique (développement d'application) on peut citer les critères suivants :

- **Les performances** : temps de réponse, Haute disponibilité, tolérance aux pannes.
- **La maintenance** : Une application doit évoluer dans le temps, Doit être fermée à la modification et ouverte à l'extension
- **Sécurité** : la gestion des rôles et des droits.
- **Portabilité**

Pour ces raisons et d'autres l'architecture générale choisie pour le développement de ce projet est une architecture multicouche. Ce Modèle préconise le découpage d'une application en plusieurs couches et conteneur spécialisées.

L'intérêt d'un modèle d'architecture en couches réside dans l'indépendance des couches entre elles. Cela permet de disposer d'un système évolutif (changement de base de données ou d'outil de mapping objet / relationnel sans impacter les autres couches par exemple), et de pouvoir développer et tester séparément les couches.

Le schéma de la figure 22 décrit l'ensemble des composants constituant l'architecture générale du projet.

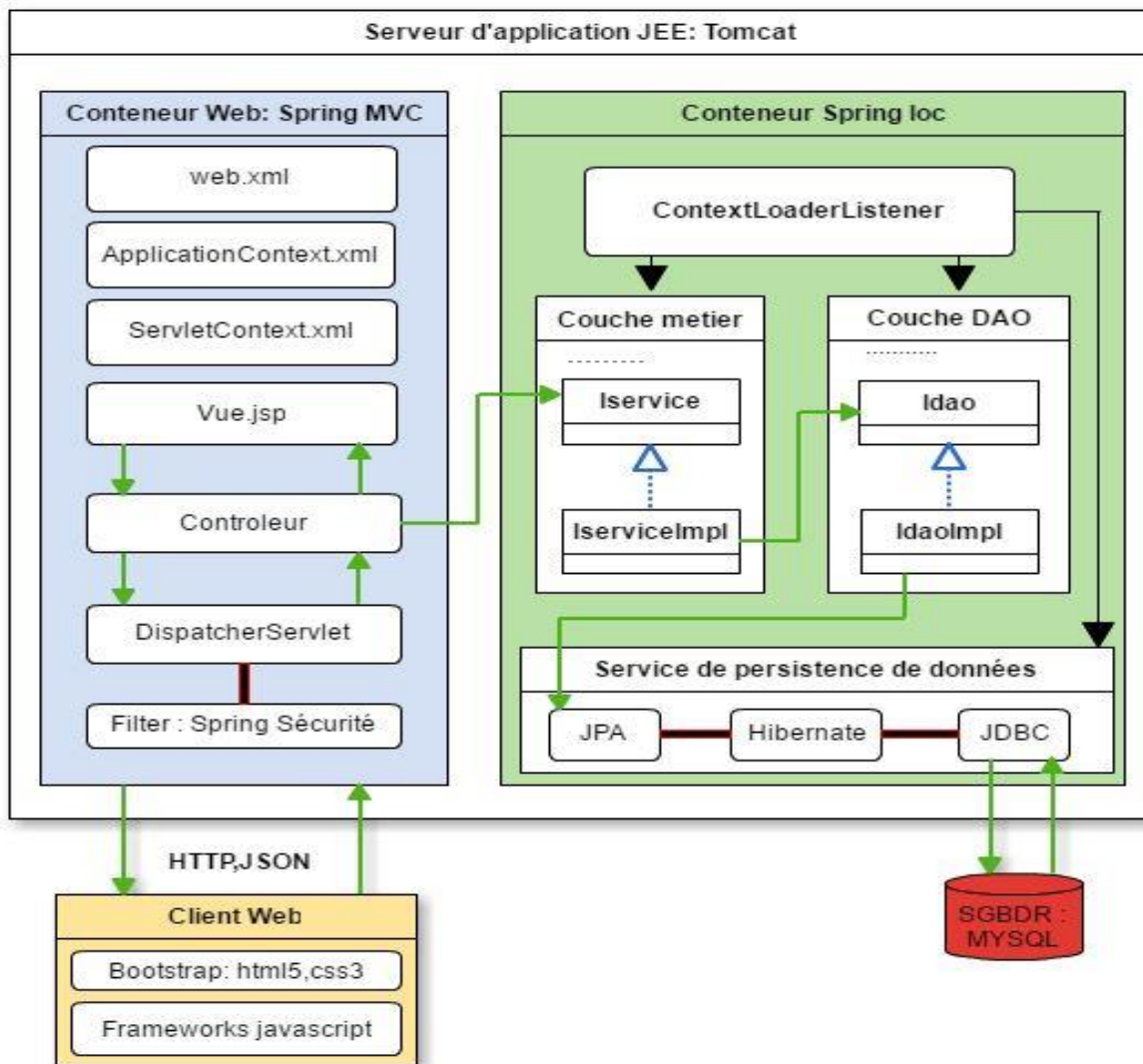


Figure 24: architecture générale de l'application

En résumé et de point de vue technique comme illustre la figure le projet est constitué principalement de 2 parties.

2.1 Back-end

Le Back-end est la partie de production des résultats fonctionnels du projet, tourne dans un serveur d'application JEE **Tomcat**.

Cette partie est constituée principalement de 2 conteneurs :

❖ **Spring Ioc** : c'est le cœur de l'application (en anglais **core container**). Ce conteneur prend en charge la création, la gestion du cycle de vie et les dépendances des objets qu'il gère.

C'est à ce niveau ou les traitements fonctionnels et la préparation des résultats sont faits en utilisant des couches spécialisées.

- **Couche métier** : c'est la couche responsable de servir toutes les services fonctionnels de l'application en communiquant avec la couche dao.
- **Couche DAO** : c'est la couche qui permet de faire des interactions avec la base de données à l'aide des services de persistance de données (**JPA/Hibernate**).

❖ **Web** : ce conteneur est réalisé grâce au Framework **Spring MVC**.

Ce module utilise les services fournis par la couche métier afin de les présenter au utilisateur final.

La sécurité de ces services et la gestion des rôles des utilisateurs est assurée grâce au Framework **Spring Sécurité**.

2.2 Front-end

Le Back-end est la partie responsable de la présentation au côté client des résultats fonctionnels issus du serveur.

Cette partie est réalisée par les technologies web suivantes :

- **Framework Bootstrap** : html5-css3.
- **Framework JavaScript** : JQuery/Ajax.

2.3 Fonctionnement

Dans cette architecture après le démarrage de serveur d'application Tomcat ce dernier lit le fichier **web.xml** dont nous avons les informations de configuration et de déploiement pour les composants de notre application web.

Ensuite le **ContextLoaderListener** est instancié, cet instance est utilisée pour charger le ou les fichiers de configuration de l'application ces fichiers de configuration sont :

- **ApplicationContext.xml** : contient les paramètres d'accès aux ressources, la définition des beans, la configuration de la sécurité...
- **ServletContext.xml** : est un fichier de configuration qui est créé lorsque l'application Web est démarrée. Il contient différents paramètres d'initialisation qui peut être configurés dans web.xml.

Après le démarrage de l'application le client web peut envoyer des requêtes vers l'application en utilisant le protocole web HTTP, ce client utilise les technologies déjà mentionner dans la figure 22.

Les requêtes envoyés par le client sont interceptés par une chaîne de filtre de Spring Sécurité, ces filtres permettent de vérifier l'identité d'utilisateur ainsi que s'il a le droit d'accéder au lien en question ou non.

Si le client a le droit d'accéder alors la requête est envoyée au DispatcherServlet qui est le point d'entrée de l'application qui permet de chercher le contrôleur adéquat et retourner une vue contenant la réponse, nous reviendrons dans ce rapport au fonctionnement de DispatcherServlet.

Avant de générer la vue au client le contrôleur consulte la couche métier qui est responsable de traitement métier de l'application à travers l'interface des services **IService**

3. Technologies et Framework de développement

3.1 Java

Java est un langage développé par Sun Microsystems depuis la fin des années 1980. La première version de Java a été mise au point en 1991. Java a connu depuis un essor considérable, notamment dans le domaine des applications distribuées via Internet, grâce au soutien de Netscape depuis 1995.

L'aspect purement Objet de Java permet une meilleure répartition du travail entre les programmeurs qui peuvent développer ou utiliser différents objets sans se préoccuper des traitements réalisés à l'intérieur. De plus, il est plus facile d'adapter et de faire évoluer les spécificités d'un objet à partir du moment où, vu de l'extérieur, cet objet réalise les mêmes

actions. Ainsi, les objets mis en œuvre en Java sont réutilisables à souhait. Pour résumer, Java est portable, sûr, orienté objet et indépendant de toute plateforme.

3.2 La technologie JEE

L'élaboration du portail s'appuie sur la plateforme JEE (Java Enterprise Edition) qui est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants.

JEE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées.

Ce choix est justifié par plusieurs facteurs à savoir :

- La maturité et la richesse de cette technologie
- La possibilité de la réutilisation des différents composants qui en font partie
- La séparation forte qu'offre la plupart des Framework relevant de cette architecture
- La maintenabilité des applications.

3.3 XML

XML (Extensible Markup Language, «langage de balisage extensible») est un langage informatique de balisage générique qui dérive du SGML.

Cette syntaxe est dite extensible car elle permet de définir différents espaces de noms, c'est à dire des langages avec chacun leur vocabulaire et leur grammaire.

Nous avons utilisé ce langage pour définir les fichiers de configuration de l'application.

3.4 Spring

Spring est un framework créé et supporté par l'entreprise **SpringSource** qui fait maintenant partie de VMware. Il permet de simplifier le développement d'applications Java et est devenu un standard dans l'industrie du développement logiciel basé sur la plateforme Java, surtout dans le développement Java EE.

Spring est un conteneur léger qui facilite le développement avec des pojos, c'est-à-dire avec des classes Java qui n'ont pas besoin de s'exécuter dans un conteneur spécifique ou dans un serveur d'applications. Il se compose d'un noyau (**core**) et de plusieurs modules qui ajoutent des fonctionnalités.

3.4.1 Spring Ioc

Permet de faire L'injection de dépendances (Dependency Injection) est un mécanisme qui permet d'implémenter le principe de l'inversion de contrôle. Il consiste à créer dynamiquement (injecter) les dépendances entre les différentes classes en s'appuyant sur une description (fichier de configuration ou métadonnées) ou de manière programmatique. Ainsi les dépendances entre composants logiciels ne sont plus exprimées dans le code de manière statique mais déterminées dynamiquement à l'exécution.

3.4.2 Spring MVC

Spring MVC est un framwork qui permet d'implémenter des applications selon le design pattern MVC. Donc, comme tous autre MVC framework, Spring MVC se base sur le principe (Modèle-Vue-Contrôleur).

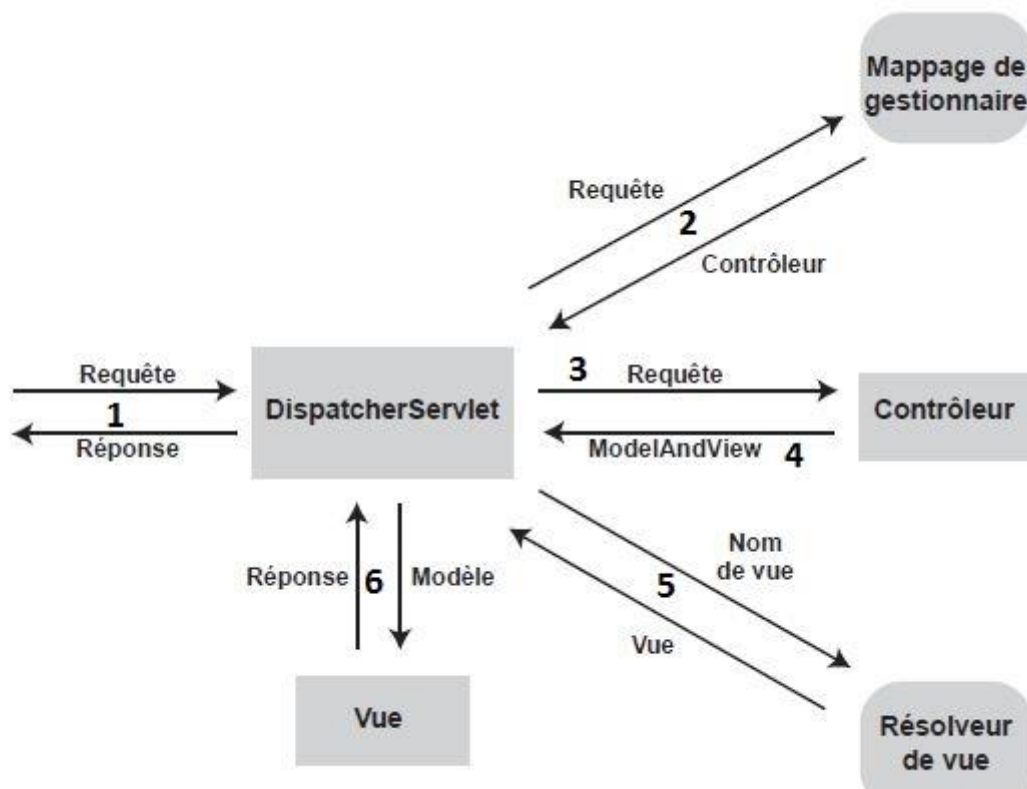


Figure 25: schéma Spring MVC

Spring MVC est l'un des plus importants module de Spring.il se fonde sur le conteneur Spring Ioc, dont il exploite les fonctionnalités pour simplifier sa configuration.

3.4.3 Spring Sécurité

Spring Security est un module incontournable d'une application développée en Spring. Il apporte tout le nécessaire pour sécuriser une application et il a l'avantage d'être vraiment personnalisable. La notion de sécurité informatique n'est pas une mince affaire et sa mise en place est parfois longue et demande d'être constamment adaptée au niveau réseau, serveurs, application... Spring Security n'intervient que sur le domaine applicatif.

Nous avons utilisé Spring sécurité pour la sécurité applicative de note application :

- ❖ **l'authentification** : Vérification de l'identité de la personne.
- ❖ **les autorisations** : Gestion des droits d'accès aux ressources.

3.5 JPA/Hibernate

Travailler dans les deux univers qui sont, l'orienté objet et le relationnel peut être lourd et consommateur en temps. Hibernate est un outil de mapping objet/relationnel pour le monde Java. Il s'occupe du transfert des classes Java dans les tables de la base de données (et des types de données Java dans les types de données SQL). Il permet aussi de requêter les données et propose des moyens de les récupérer. Il peut donc réduire de manière significative le temps de développement qui aurait été autrement perdu dans une manipulation manuelle des données via SQL et JDBC.

3.6 Jackson

Jackson est une bibliothèque basée sur Java très populaire et efficace pour sérialiser ou la carte des objets Java à JSON, et vice versa.

3.7 Bootstrap 3

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'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.

3.8 JQuery / Ajax

JQuery est une bibliothèque JavaScript permettant de se simplifier la vie à moindre coût pour tout ce qui est manipulation d'une page HTML.

Nous avons profité de L'architecture informatique **Ajax** (acronyme d'Asynchronous Javascript and Xml) on utilisant la fonction \$.ajax().

Les échanges de données entre client et serveur sont effectués en utilisant le format de données **JSON**.

3.9 Environnement de développement

3.9.1 Eclipse



Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

3.9.2 Maven



Maven est un outil « open source » d'Apache Jakarta. Il permet de faciliter et d'automatiser la gestion et la construction d'un projet java. Son but est la production d'un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.

Maven utilise un paradigme connu sous le nom de Project Object Model (POM) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production.

En résumé Maven permet :

- D'automatiser la compilation, les tests unitaires et le déploiement des applications du projet (jar, war).
- De gérer les dépendances des bibliothèques nécessaires au projet
- De générer des documentations du projet : rapport de compilation et des tests unitaires, javadoc
- De générer un site web complet du projet.

3.9.3 Apache Tomcat



Apache Tomcat est un conteneur de servlet J2EE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des servlets et des JSP de Sun Microsystems.

Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP.

3.9.4 MySql

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde³, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

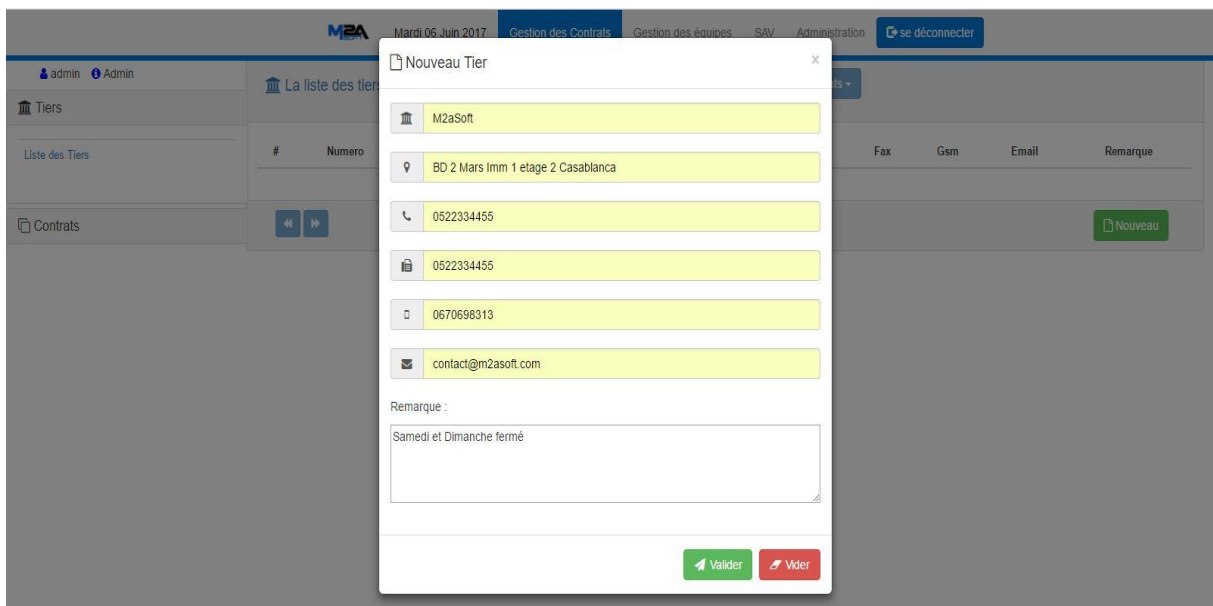
4. Réalisation

4.1 Fruit du travail

Dans cette partie on va présenter quelques captures d'écran en a un circuit qui part de la création d'un tiers à la fin d'une intervention :

4.1.1 Création de tiers

4.1.1.1 Pop-up nouveau tiers



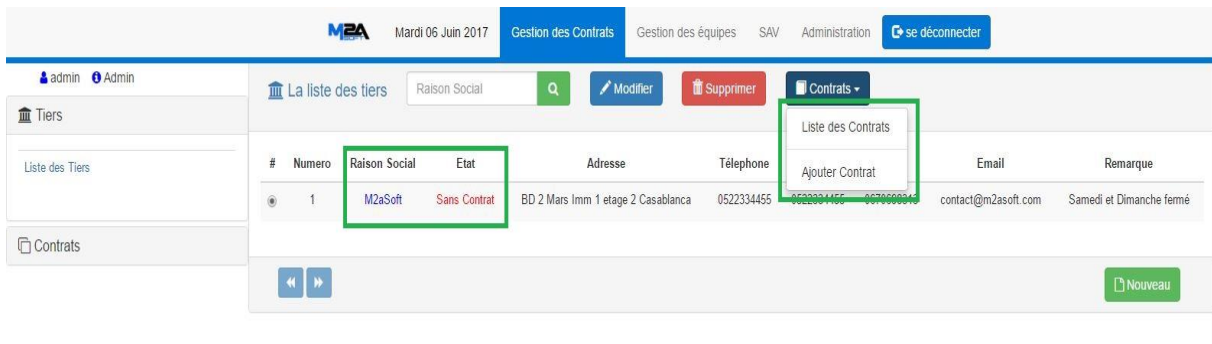
The screenshot displays a web interface with a 'Nouveau Tier' modal form. The form fields are as follows:

- Name: M2aSoft
- Address: BD 2 Mars Imm 1 etage 2 Casablanca
- Phone: 0522334455
- Fax: 0522334455
- Mobile: 0670698313
- Email: contact@m2asoft.com
- Remarque: Samedi et Dimanche fermé

Buttons at the bottom of the form: Valider (green), Vider (red).

La figure ci-dessus permet de créer nouveau tiers en spécifiant les cordonnées nécessaire.

4.1.1.2 Vérification de la création



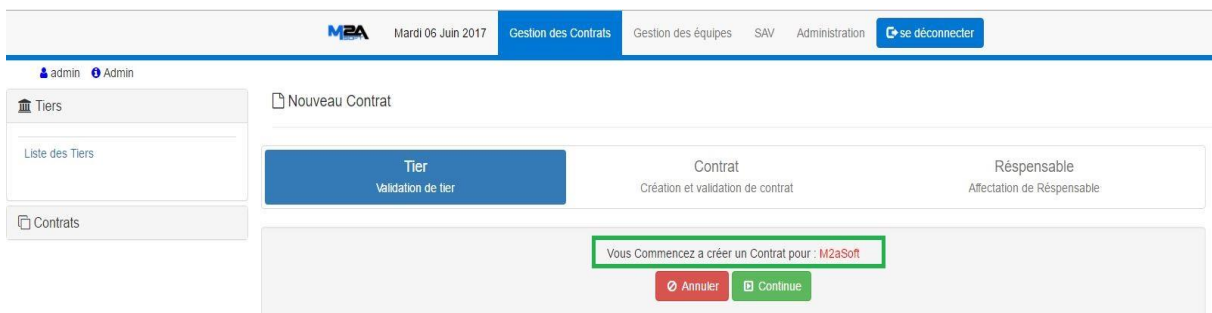
Le tiers est créé avec succès et par suite on peut :

- Ajouter un contrat.
- Voir la liste des contrats.

4.1.2 Création de contrat

Pour un tiers la procédure de la création d'un contrat est le suivant :

4.1.2.1 Vérification de tiers



D'après cette fenêtre on peut continuer ou annuler la création de contrat.

4.1.2.2 Nouveau contrat

La figure suivante permet de créer nouveau contrat pour un tiers en spécifiant :

- Le type de contrat :
 - Contrat de maintenance (matériel ou logiciel).
 - Contrat d'assistance sage.
- Durée de contrat : date début, durée par année ou mois.
- Tranche de contrat : durée de contrat, montant de tranche.

MBA Mardi 06 Juin 2017 Gestion des Contrats Gestion des équipes SAV Administration se déconnecter

admin Admin

Tiers
Liste des Tiers

Contrats

Nouveau Contrat

Tier
Tier validé avec succès

Contrat
Création et validation de contrat

Résponsable
Affectation de Responsable

Numéro de Contrat

Type : Matériel Logiciel

date Début 04/10/2017

Durée: 1 Ans 0 Mois

Durée de Tranche: Trimestrielle

Montant de Tranche 10000 DH

Contact contact@m2asoft.com

Validation: Résiliation:

Continue

4.1.2.3 Affectation de responsable

Cette étape de la création de contrat est facultative.

MBA Mardi 06 Juin 2017 Gestion des Contrats Gestion des équipes SAV Administration se déconnecter

admin Admin

Tiers
Liste des Tiers

Contrats

Nouveau Contrat

Tier
Tier validé avec succès

Contrat
Création et validation de contrat

Résponsable
Affectation de Responsable

Nom Complet Ahmed Ahmed

Email Ahmed@m2asoft.com

Gsm 0670698313

Valider

4.1.2.4 Vérification de la création de contrat.

MBA Mardi 06 Juin 2017 Gestion des Contrats Gestion des équipes SAV Administration se déconnecter

admin Admin

Tiers
Liste des Tiers

Contrats
La liste des Contrats

Raison Social Recherche Avancée Modifier Supprimer Action

#	id	Tier	Type	Date de debut	Date de Fin	Durée de Tranche	Montant de Tranche	Contact	Validation	Résiliation
1	M2aSoft	Materiel-Logiciel	2017-10-04	2018-10-04	Trimestrielle	10000.0	contact@m2asoft.co	Produits Sage	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Inventaire Materiel
Inventaire Logiciel
Produits Sage
Nouveau Produit

Le contrat est bien créé et par suite on peut gérer l'inventaire ou les modules sage selon le type de contrat dans notre cas on va gérer l'inventaire de contrat de maintenance (matériel-logiciel).

4.1.3 Gestion d'inventaire (contrat de maintenance).

Cette partie permet d'enregistrer un ensemble de produit qui sont associés à un contrat et qui feront l'objet d'une réclamation de maintenance.

4.1.3.1 Nouveau produit

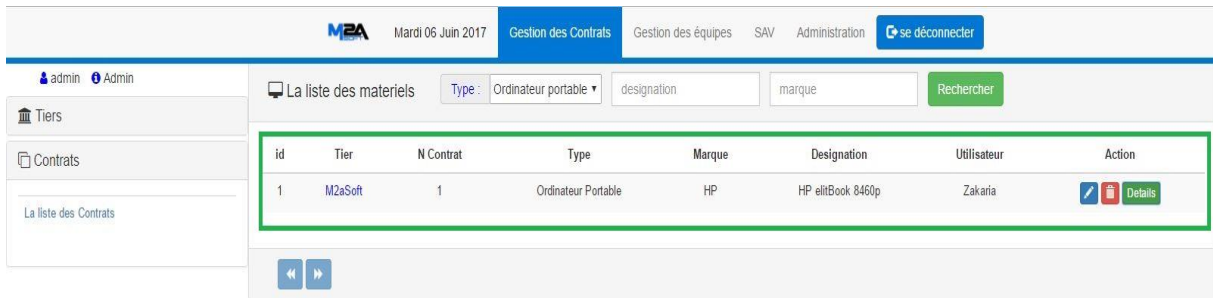
Dans cette étape on va ajouter un nouveau produit à l'inventaire matériel du contrat créé précédemment

The screenshot shows the 'Nouveau Produit' form in the M2A system. The form is for adding a new material product. It includes fields for Type (Ordinateur Portable), Designation (HP eliteBook 8460p), Marque (HP), SN (2345677REZ), Systeme (Windows 7), Configuration (i5 6gb ram), Etat initiale (Bon), Active (checked), Utilisateur (Zakaria), Fournisseur (M2aSoft), Prix d'achat (6000), Prix de Vente (7000), and date de Livraison (02/06/2017). A 'Valider' button is at the bottom.

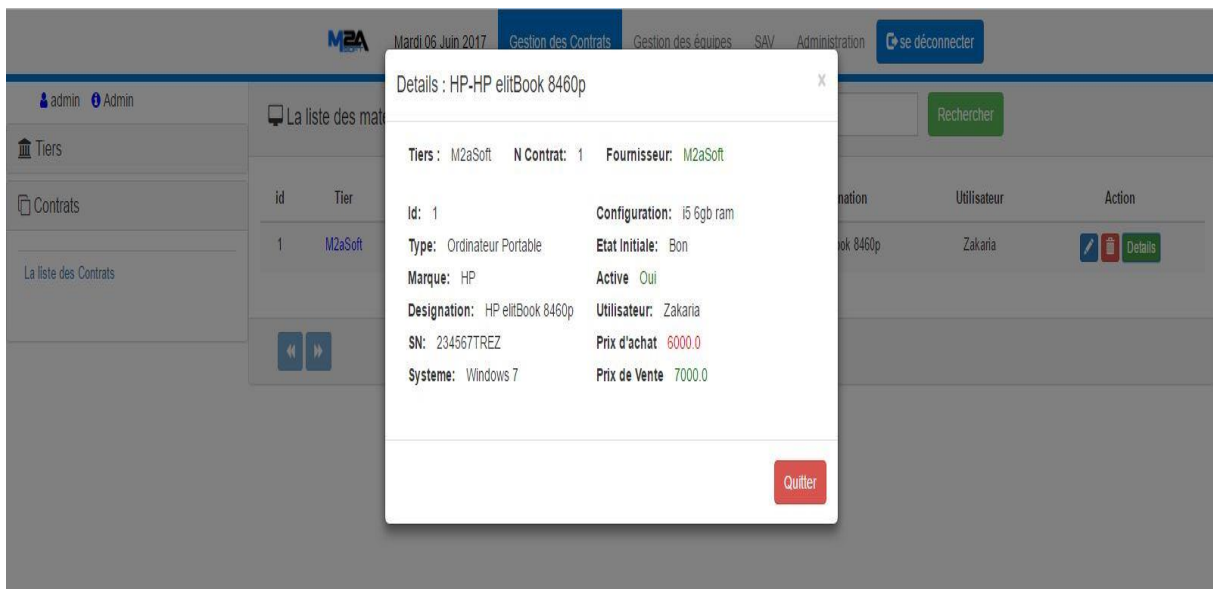
Dans la fenêtre ci-dessus permet d'ajouter un nouveau produit à l'inventaire, son contenu varié selon le type de contrat.

Si le produit est fourni par m2asoft un collapse est affiché afin de remplir : le prix d'achat, prix de vente et la date de livraison.

4.1.3.2 Vérification de produit

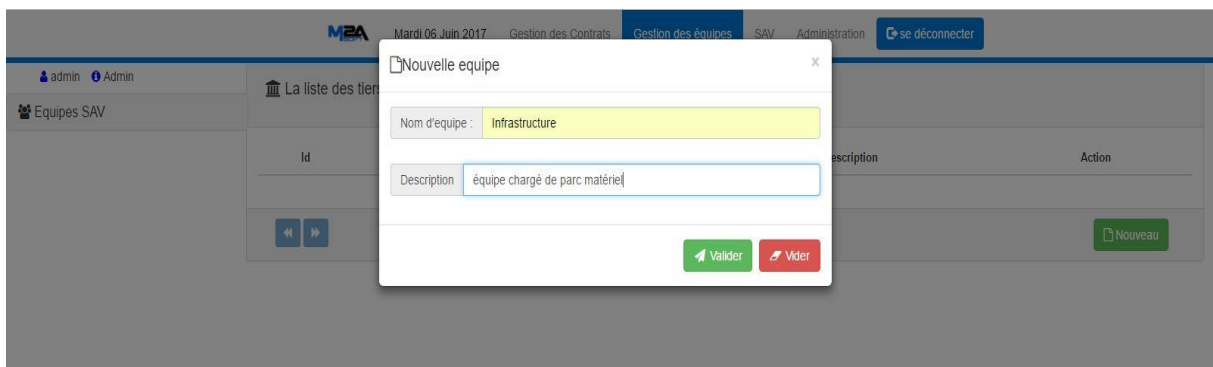


Le produit est ajouté avec succès et on peut voir les détails en Pop-up.



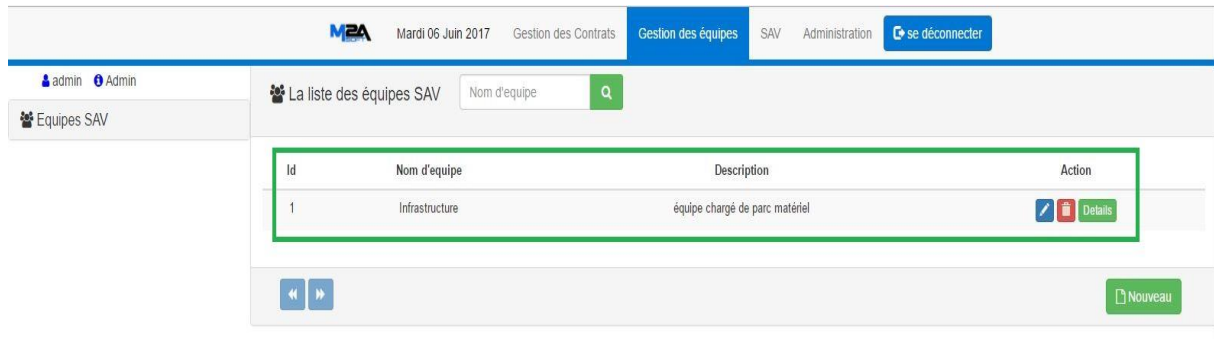
4.1.4 Gestion des équipes SAV

4.1.4.1 Création d'une équipe SAV



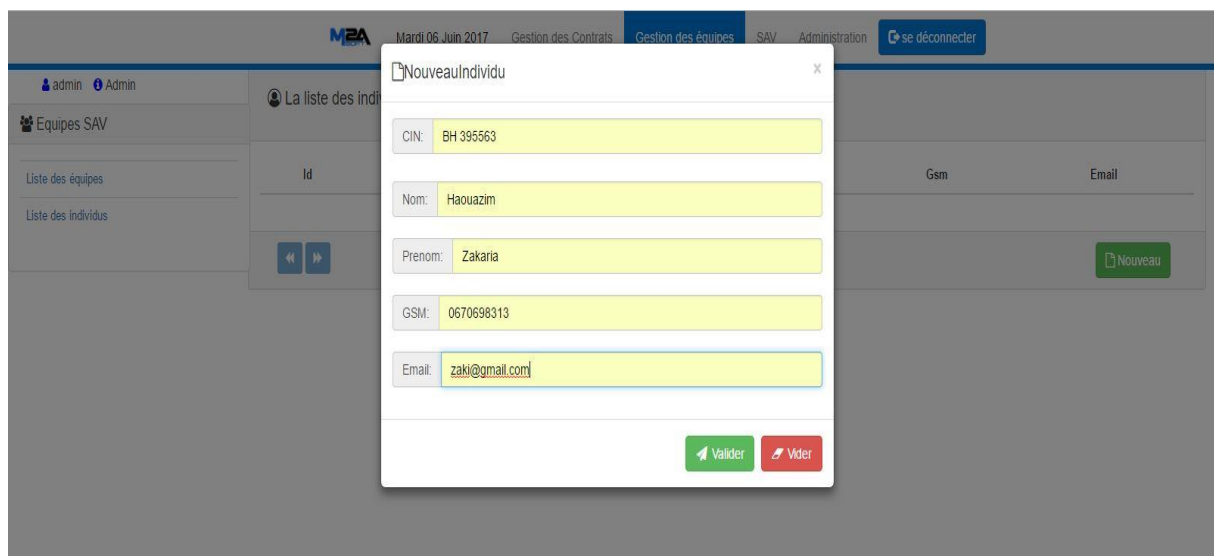
La fenêtre précédente permet de créer une équipe en spécifiant : le nom et la description.

4.1.4.2 Vérification de la création de l'équipe



L'équipe est créée avec succès.

4.1.4.3 Enregistrement d'un individu



Ce Pop-up permet d'enregistrer un individu.

4.1.4.4 Vérification de l'enregistrement d'un individu.

L'individu est bien ajouter au système.

MBA Mardi 06 Juin 2017 Gestion des Contrats **Gestion des équipes** SAV Administration [se déconnecter](#)

admin Admin

Equipes SAV

La liste des individus

Id	Cin	Nom	Prenom	Gsm	Email	
3	BH 395563	Haouazim	Zakaria	0670698313	zaki@gmail.com	Details
2	BH 108990	sanaa	sanaa	0670698313	contact@m2asoft.com	Details
1	BH 395569	youabed	mohamed	0670698313	contact@m2asoft.com	Details

[Nouveau](#)

4.1.4.5 Consultation d'une équipe.

On consultant une équipe on peut :

- Voir l'ensemble des individus affecter à l'équipe (intervenants, chef d'équipe).
- Affecter nouveau un nouveau individu
- Désaffecter un individu.

MBA Mardi 06 Juin 2017 Gestion des Contrats **Gestion des équipes** SAV Administration [se déconnecter](#)

admin Admin

Equipes SAV

Détails d'équipe :

Nom d'équipe : Infrastructure
Description : équipe chargé de parc matériel
Chef d'équipe : youabed mohamed

Affectation d'individu :

[+ Affecter Individu](#)

ID	Nom Complet	Role	Date Affectation	Tickets	Action
1	youabed mohamed	Chef d'équipe	2017-06-11 12:21:52.0	tickets	Désaffecter

Les individus à affecté sont libres (n'appartient à aucune équipe).

Résultat :

The screenshot shows the 'Gestion des équipes' page in the M2A application. The team 'Infrastructure' is selected, with details: Nom d'équipe: Infrastructure, Description: équipe chargé de parc matériel, Chef d'équipe: youabed mohamed. A table lists team members:

ID	Nom Complet	Role	Date Affectation	Tickets	Action
1	youabed mohamed	Chef d'équipe	2017-06-11 12:21:52.0	tickets	✖ Désaffecter
2	sanaa sanaa	Intervenant	2017-06-11 12:22:21.0	tickets	✖ Désaffecter

A '+ Affecter Individu' button is visible in the top right of the table area.

L'individu est bien ajouté à l'équipe.

Désaffecter un individu :

Pour désaffecter un individu il suffit de cliquer sur le bouton désaffecter :

The screenshot shows the same team details page, but with a confirmation dialog box open. The dialog text reads: 'Attention Voulez vous vraiment désaffecter : sanaa sanaa'. There are 'Valider' and 'Annuler' buttons at the bottom of the dialog. The table from the previous screenshot is visible in the background, with the 'Désaffecter' button for 'sanaa sanaa' highlighted.

Ensuite il faut valider le pop-up

4.1.4.6 Consultation d'individu

En consultant un individu on peut :

- Voir les informations de l'individu.
- Voir les traces de l'individu par équipe et pour chaque équipe nous avons un lien vers les tickets qui sont affectées à cet individu.
- Ajouter et valider une compétence pour l'individu.

MBA Mardi 06 Juin 2017 Gestion des Contrats **Gestion des équipes** SAV Administration [se déconnecter](#)

admin Admin

Equipes SAV

Details Individu :

Cin : BH 108990
 Nom : sanaa
 Prénom : sanaa
 Email : contact@m2asoft.com
 Gsm : 0670698313

[Compétences](#) [Nouvelle Compétence](#)

ID	Nom Complet	Equipe	Date Affectation	Date Fin	Tickets
5	sanaa sanaa	Développement spécifique	2017-06-11 12:59:51.0	En cours	Tickets
4	sanaa sanaa	Infrastructure	2017-06-11 12:55:45.0	2017-06-11 12:59:34.0	Tickets
3	sanaa sanaa	Infrastructure	2017-06-11 12:53:56.0	2017-06-11 12:54:21.0	Tickets
2	sanaa sanaa	Infrastructure	2017-06-11 12:22:21.0	2017-06-11 12:32:58.0	Tickets

Nouvel compétence :

Ce pop-up permet d'ajouter une compétence pour l'individu en spécifiant un mot clé et la description.

MBA Mardi 06 Juin 2017 Gestion des Contrats **Gestion des équipes** SAV Administration [se déconnecter](#)

Details Individu

Cin : BH 108990
 Nom : sanaa
 Prénom : sanaa
 Email : contact@m2a
 Gsm : 0670698313

[Compétences](#) [Nouvelle Compétence](#)

Nouvelle Compétences

mot clé : .NET

Description : Visual basic,C#,ASP.NET MVC 6, Hibernete ASP

[Valider](#) [Vider](#)

ID	Nom Complet	Equipe	Date Affectation	Date Fin	Tickets
5	sanaa sanaa	Développement spécifique	2017-06-11 12:59:51.0	En cours	Tickets
4	sanaa sanaa	Infrastructure	2017-06-11 12:55:45.0	2017-06-11 12:59:34.0	Tickets
3	sanaa sanaa	Infrastructure	2017-06-11 12:53:56.0	2017-06-11 12:54:21.0	Tickets
2	sanaa sanaa	Infrastructure	2017-06-11 12:22:21.0	2017-06-11 12:32:58.0	Tickets

Résultat :

On voit bien dans la fenêtre suivante que la compétence a été bien ajoutée et en attente de la validation.

Details Individu :

Cin : BH 108990
 Nom : sanaa
 Prénom : sanaa
 Email : contact@m2asoft.com
 Gsm : 0670698313

Compétences Nouvelle Compétence

ID	Clé	Description	Etat	Action
1	.NET	Visual basic, C#, ASP.NET MVC 6, Hibermete ASP	non Valide	Valider

ID	Nom Complet	Equipe	Date Affectation	Date Fin	Tickets
5	sanaa sanaa	Développement spécifique	2017-06-11 12:59:51.0	En cours	Tickets
4	sanaa sanaa	Infrastructure	2017-06-11 12:55:45.0	2017-06-11 12:59:34.0	Tickets
3	sanaa sanaa	Infrastructure	2017-06-11 12:53:56.0	2017-06-11 12:54:21.0	Tickets
2	sanaa sanaa	Infrastructure	2017-06-11 12:22:21.0	2017-06-11 12:32:58.0	Tickets

4.1.5 Gestion de processus SAV

4.1.5.1 Création de ticket

Nouveau Tickets

Numero Tickets :

Tiers : M2aSoft

Contrat : 1 materiel-logiciel

Matériel Logiciel

Produit : Ordinateur Portable--HP--HP ellitBook 8460p--Zakaria

Autre Produit. Nom Produit :

Objet : Plantage continue

Description : apres l'ouverture de session l'ordinateur reste planté

Priorité : Urgent

Information de Demandeur:

Demandeur : Zakaria

Email : zakaria.haouazim@gmail.com

[Valider](#)

La création d'un ticket demande la validation d'un formulaire constitué de 3 parties :

- **Produit à maintenir** : pour renseigner un produit il faut choisir le contrat et suivant le type de contrat on aura un type de produit bien déterminé à réclamer. En suite le demandeur peut choisir entre un produit qui fait partie de contrat ou un autre.
- **Détails de la réclamation** : dans cette partie le demandeur doit spécifier (l'objet, description et la priorité).
- **Information de demandeur** : demandeur, email.

Résultat : le ticket a été bien ajouté comme montre la fenêtre suivante.

id	Tier	N Contrat	Objet	Inventaire	Date de Réclamation	Etat	Date de Résolution	Date de Annulation	Action
1	M2aSoft	1	Plantage continue	Non	2017-06-11 15:40:01.0	En Attente			Details

4.1.5.2 Consultation de ticket

En consultant un ticket on peut :

- Savoir si l'objet de la réclamation fait partie de l'inventaire du contrat ou non.
- Voir les détails de ticket.
- Annuler le ticket.
- Affecter le ticket.
- Ouvrir intervention pour le ticket
- Avoir l'état de ticket (annulé, clôturé, en cours en attente).

MBA Mardi 06 Juin 2017 Gestion des Contrats Gestion des équipes SAV Administration se déconnecter

Détails Tickets

Tiers : M2aSoft
Contrat : 1

Demandeur : Zakaria
Email : zakaria.haouazim@gmail.com

Ticket : 1
Etat : En Attente

Inventaire	Produit	Objet	Description	Date de Réclamation	Date de Résolution	Date d'annulation	Cause d'annulation
Non	HP elitBook 8460p	Plantage continue	apres l'ouverture de session l'ordinateur reste planté	2017-06-11 15:40:01.0			

Ouvrir Intervention Affecter Annuler

Id	Equipe	Date d'affectation	Date Fin	Résolu	Détails
----	--------	--------------------	----------	--------	---------

4.1.5.3 Affectation de ticket

Ce pop-up permet d'affecter un ticket a une équipes SAV ou individu.

MBA Mardi 06 Juin 2017 Gestion des Contrats Gestion des équipes SAV Administration se déconnecter

Détails Tickets

Tiers : M2aSoft
Contrat : 1

Ticket : 1
Etat : En Attente

Affectation de Ticket Numero : 1

Equipe : Infrastructure

Individu : youabed mohamed--Chef d'equipe

Valider Annuler

Inventaire	Produit	Objet	Description	Date de Réclamation	Date de Résolution	Date d'annulation	Cause d'annulation
Non	HP elitBook 8460p	Plantage continue	apres l'ouverture de session l'ordinateur reste planté	2017-06-11 15:40:01.0			

Ouvrir Intervention Affecter Annuler

4.1.5.4 Suivre de ticket

Le ticket est bien affecté, et on peut suivre les équipes (les intervenants) dont elle est passée, et savoir dans quelle équipe et avec quel individu est résolu ...

La fenêtre suivant montre le suivie de ticket qu'on vient de créer.

Inventaire	Produit	Objet	Description	Date de Réclamation	Date de Résolution	Date d'annulation	Cause d'annulation
Non	HP eliteBook 8460p	Plantage continue	apres l'ouverture de session l'ordinateur reste planté	2017-06-11 15:40:01.0			

Ouvrir Intervention Affecter Annuler

Id	Equipe	Date d'affectation	Date Fin	Résolu	Détails
2	Infrastructure	2017-06-11 17:13:59.0	-----	En Cours	Détails Désaffecter

id	Nom Complet	Date d'affectation	Date de Fin	Résolu	Action
2	youabed + mohamed	2017-06-11 17:13:59.0	-----	En Cours	Désaffecter
1	youabed + mohamed	2017-06-11 16:39:31.0	2017-06-11 17:11:47.0	Non	

4.2 Test

En informatique, un test désigne une procédure de vérification partielle d'un système. Son objectif principal est d'identifier un nombre maximum de comportements problématiques du logiciel afin d'en augmenter la qualité.

On distingue plusieurs types de test dans ce qui suit on cite les tests que nous avons fait sur notre application.

4.2.1 Test unitaire

En programmation informatique, le **test unitaire** (ou « **T.U.** », ou « **U.T.** » en anglais) est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »).

4.2.2 Test fonctionnel

Ce a pour but de vérifier la conformité de l'application développée avec le cahier des charges initial. Ils sont donc basés sur les spécifications fonctionnelles et techniques.

5. Conclusion

Ce chapitre a présenté l'architecture de l'application ainsi que les technologies et outils qui sont utilisés pour mener ce projet au bien.

Finalement et avant de présenter les tests effectués, nous avons données le fruit du travail sous forme des captures d'écrans.

Conclusion Générale

Mon projet de fin d'études consistait à la conception et la réalisation d'une application de gestion des contrats de maintenance et d'analyse de performance des équipes SAV, dont l'objectif principale est d'avoir un service SAV productif.

Ma mission principale dans ce projet était de développer les trois modules de base qui entourent le service SAV : (gestion de contrats de maintenance gestion des équipes SAV, gestion de processus SAV).

J'étais amené dans un premier lieu à établir un cahier de charge en spécifiant les besoins de notre application. Ensuite nous avons abordé une analyse approfondie du projet en étudiant l'existant afin de proposer une meilleure solution M2aSAV. Les diagrammes d'UML ont fait l'objet de la conception pour dégager les différents modules de l'application et finalement l'étude des outils et des techniques susceptibles de convenir à la réalisation de ce projet.

Ce rapport a présenté le projet en 3 parties chapitre indispensables (Contexte du projet, Analyse et Conception, Etude technique et réalisation).

Webliographie

- Livre : Spring par la pratique – Julian Dubois – Eyrolles 2012 PDF
- Livre : Java Persistence et Hibernate – Gary Mak – Pearson 2014 PDF
- <http://www.objectdb.com> – cours JPA – date de consultation : 04/04/2017
- <https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>-
Documentation officiel de Spring – consultation durant la réalisation du projet.
- <https://www.developpez.net> - cours et tutoriel Framework JAVA JEE consultation
durant la réalisation de ce projet.