

Réalisé par : Mr BOUZAYANI Malek

**Conception et développement d'une solution
web de gestion d'un parc machine.**

Table de matières

Introduction générale.....	1
Chapitre 1. Présentation générale du projet.....	3
Introduction	4
I. Présentation du projet.....	4
1. Problématique.....	4
1.1 La non-maitrise du hardware et du software.....	4
1.2 Le manque de traçabilité et de suivi	4
2. Etude de l'existant.....	5
2.1 Clarilog.....	5
2.2 H-inventory.....	5
3. Solution proposée	5
II. Méthodologie et modélisation adoptée.....	6
1. Méthodologie de conception	6
2. Méthodologie de travail.....	7
III. Framework Scrum	8
1. Équipe et Rôle.....	8
2. Fixation du Sprint time Box	9
3. Les cérémonies.....	9
3.1 Sprint planning	9
3.2 Daily meeting.....	9
3.3 Sprint Review.....	10
3.4 Sprint Rétrospective	10
Conclusion.....	10
Chapitre 2. Spécification des besoins.....	11
Introduction	12
I. Spécification des besoins	12
1. Identification des acteurs.....	12
2. Exigences fonctionnelles	13

Table de matières

3.	Exigences non fonctionnelles	14
II.	Modélisation des besoins fonctionnels	14
1.	Diagrammes de cas d'utilisation	14
1.1	Diagramme de cas d'utilisation de l'administrateur	15
1.2	Diagramme de cas d'utilisation du responsable	15
1.3	Diagramme de cas d'utilisation du technicien	16
2.	Description textuelle	16
2.1	Description de l'item S'authentifier	16
2.1.1	Diagramme de cas d'utilisation S'authentifier	16
2.1.2	Description textuelle du diagramme de cas d'utilisation S'authentifier	17
2.2	Description de l'item gérer compte technicien.....	17
2.2.1	Diagramme de cas d'utilisation Gérer compte technicien	17
2.2.2	Description textuelle du diagramme de cas d'utilisation Gérer compte technicien 18	
2.3	Description de l'item passer réclamations	19
2.3.1	Diagramme de cas d'utilisation Passé réclamations	19
2.3.2	Description textuelle de diagramme de cas d'utilisation Passé réclamations	19
2.4	Description de l'item Suivre l'état de réclamation.....	20
2.4.1	Diagramme de cas utilisation Suivre l'état de réclamations	20
2.4.2	Description textuelle de cas d'utilisations Suivre l'état de réclamations	20
2.5	Description de l'item gérer profil	21
2.5.1	Diagramme de cas utilisation gérer profil	21
2.5.2	Description textuelle de diagramme de cas d'utilisation Gérer profile	21
2.6	Description de l'item changer état réclamation	22
2.6.1	Diagramme de cas d'utilisation Changer état réclamation	22
2.6.2	Description textuelle de diagramme de cas d'utilisation Changer état réclamations.....	22
2.7	Description de l'item gérer interventions	24
2.7.1	Diagramme de cas d'utilisation Gérer interventions	24
2.7.2	Description textuelle de diagramme de cas d'utilisation Gérer interventions	24
III.	Le product backlog	26
	Conclusion.....	27
	Chapitre 3. Etude Conceptuelle.....	28

Introduction	29
I. Architecture du système	29
1. L'architecture REST API	29
2. Model repository pattern.....	29
II. Conception du sprint 1.....	30
1. Sprint Goal.....	30
2. Sprint Backlog.....	30
3. Diagramme de séquence « S'authentifier »	31
III. Conception du sprint 2.....	32
1. Sprint Goal.....	32
2. Sprint backlog.....	32
3. Diagramme de séquence «Gestion des machines»	33
4. Diagramme de séquence «Suivie l'état de réclamations»	34
IV. Conception du sprint 3.....	35
1. Sprint Goal.....	35
2. Sprint Backlog.....	35
2.1 Diagramme de séquence «Passer réclamation»	36
V. Conception du sprint 4.....	36
1. Sprint Goal.....	36
2. Sprint Backlog.....	37
3. Diagramme de séquence «Changer l'état d'une réclamation»	37
VI. Diagramme de classes	38
1. Diagramme de classes	38
2. Correspondance entre modèles objets et relationnels.....	39
Conclusion.....	39
Chapitre 4. Réalisation de l'application web.....	40
Introduction	41
I. Choix Techniques.....	41
1. Les outils de développement	41
2. Les Frameworks.....	42
2.1 JAVA SPRING BOOT	42
2.2 ANGULAR JS.....	42
3. Environnement logiciel.....	43
II. Les interfaces	44

Table de matières

1.	Partie administrateur	44
1.1	Interface d'authentification	44
1.2	Interface de tableau de bord.....	45
1.3	Interface gestion intervention.....	45
1.4	Interface gestion utilisateur	47
1.5	Interface gestion machine.....	49
1.6	Interface réclamation	50
2.	Interface réclamation	52
2.1	Partie responsable.....	52
2.2	Partie technicien.....	53
	Conclusion	53
	Conclusion Générale	54
	Perspective	55
	Bibliographie	56

Listes des figures

Figure 1 Localisation du Thor Syetem	Erreur ! Signet non défini.
Figure 2 Logo du Thor System	Erreur ! Signet non défini.
Figure 3 Organigramme du Thor System	Erreur ! Signet non défini.
Figure 4 Logo UML.....	7
Figure 5 Le processus Scrum	8
Figure 6 Equipe Scrum.....	9
Figure 7 Diagramme de cas d'utilisation de l'administrateur	15
Figure 8 Diagramme de cas d'utilisation du responsable	15
Figure 9 Diagramme de cas d'utilisation du technicien	16
Figure 10 Diagramme de cas d'utilisation S'authentifier	16
Figure 11 Diagramme de cas d'utilisation Gérer compte technicien.....	17
Figure 12 Diagramme de cas d'utilisation Passé réclamations	19
Figure 13 Diagramme de cas utilisation Suivre l'état de réclamations	20
Figure 14 Diagramme de cas utilisation gérer profil	21
Figure 15 Diagramme de cas d'utilisation Changer état réclamation.....	22
Figure 16 Diagramme de cas d'utilisation Gérer interventions	24
Figure 17 Architecture REST API.....	29
Figure 18 Model repository pattern.....	30
Figure 19 Digramme de séquence « s'authentifier »	31
Figure 20 Digramme de séquence « Gestion des machines ».....	33
Figure 21 Digramme de séquence « Suivi l'état de réclamation ».....	34
Figure 22 Digramme de séquence « passer réclamation »	36
Figure 23 Digramme de séquence « réclamation »	37
Figure 24 Digramme de classes	38
Figure 25 JAVA SPRING BOOT	42
Figure 26 ANGULAR JS.....	43
Figure 27 Page d'authentification	44
Figure 28 Page de tableau de bord	45
Figure 29 Page des interventions	45
Figure 30 Page d'ajouter une intervention	46
Figure 31 Page de modifier une intervention	46

Liste des figures

Figure 32 Notification de modifier	47
Figure 33 Liste utilisateur	47
Figure 34 Ajouter utilisateur	48
Figure 35 Modifier de modifier	48
Figure 36 Liste machine.....	49
Figure 37 Modifier machine	49
Figure 38 Supprimer machine	50
Figure 39 Liste Réclamation	50
Figure 40 Attribuer réclamation.....	51
Figure 41 Modifier état	51
Figure 42 Liste réclamation par responsable	52
Figure 43 Passé réclamation.....	52
Figure 44 Liste réclamation attribué au technicien.....	53
Figure 45 Changer l'état réclamation.....	53

Liste des Tableaux

Tableau 1 Les acteurs	12
Tableau 2 Description textuelle de cas d'utilisation « S'authentifier»	17
Tableau 3 Description textuelle de cas d'utilisation « Gérer les personnels»	19
Tableau 4 Description textuelle du cas d'utilisation Passer réclamation.....	20
Tableau 5 Description textuelle de cas d'utilisation «Suivre les états de l'offre»	20
Tableau 6 Description textuelle de cas d'utilisation «Editer profil»	21
Tableau 7 Description textuelle de diagramme de cas d'utilisation gérer les machines.....	23
Tableau 8 Description textuelle de cas d'utilisation «Gérer les rôles»	26
Tableau 9 Product backlog	27
Tableau 10 Backlog sprint 1	31
Tableau 11 Backlog sprint 2	33
Tableau 12 Backlog sprint 3.....	35
Tableau 13 Backlog sprint 4	37
Tableau 14 Les outils de développement.....	41
Tableau 15 L'environnement logiciel	44

Introduction générale

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprises ouvrent leur parc machines à leurs partenaires ou à leurs fournisseurs. Le parc machines est un ensemble de ressources matérielles et logicielles dont dispose une entreprise.

Pour assurer la survie et la pérennité de ses ressources, il est important d'avoir une gestion efficiente du parc de l'entreprise. La gestion du parc consiste donc d'une part à lister et à localiser les équipements de l'entreprise, d'autre part à effectuer des tâches de maintenance, d'assistance aux utilisateurs. Ces opérations peuvent être effectuées par une personne qualifiée, mais bien souvent ce travail dépasse ses compétences.

Dans ce cadre, nous nous sommes intéressés à la conception et l'implémentation d'une application web. Cette application qui est une solution web doit :

1. Gérer les réclamations.
2. Gérer les machines du parc.
3. Gérer les techniciens.
4. Suivre l'état d'une machine.

Notre travail se traduit dans ce rapport qui développe les différentes phases de développement de notre projet et qui sont organisées en quatre chapitres de la manière suivante :

Le premier chapitre sera dédié à la présentation de l'état de l'art. Nous allons tout d'abord présenter l'étude et à la critique de l'existant pour enfin proposer une solution adéquate. La méthodologie utilisée y sera également définie pour nous permettre de réaliser convenablement notre travail.

Le second chapitre sera consacré sur l'analyse des besoins fonctionnels et non fonctionnels. Nous modéliserons les besoins des utilisateurs via les diagrammes de cas d'utilisation ainsi que le backlog du projet.

Le troisième chapitre est dédié à l'étude conceptuelle de notre système. Dans ce chapitre, nous présenterons la modélisation des différents diagrammes nécessaires facilitant la compréhension du système.

Le quatrième chapitre présente l'environnement de travail, les outils logiciels que nous avons utilisés pour la réalisation de notre projet ainsi que quelques imprimés écrans.

Pour finir, une conclusion générale de tout le rapport sera nécessaire où nous proposerons les éventuelles améliorations susceptibles d'être ajoutées ultérieurement.

Chapitre 1. Présentation générale du projet

Introduction

- I- Présentation du projet
- II- Méthodologie et modélisation adoptée
- III- Framework Scrum

Conclusion

Introduction

La réalisation de ce projet nécessite une étude détaillée sur certaines notions qui touchent non seulement le cadre général du projet, mais aussi à sa réalisation. Ce chapitre sera destiné, en premier lieu, à la présentation de l'organisme d'accueil, ensuite nous présentons notre cadre du projet. Nous allons présenter les problèmes et les enjeux de l'organisme d'accueil dans le but de dévoiler les objectifs de notre travail.

I. Présentation du projet

1.Problématique

L'idée générale du projet consiste à concevoir un outil applicatif qui pourra de façon concrète permettre à un utilisateur de circonscrire un incident à travers les réclamations. L'administrateur utilisera le même système pour gérer la partie administrative des matériaux du parc.

Avant de plonger dans l'étude proprement dite de la solution, il est indispensable de prendre du recul et de faire un résumé des problèmes concrets existants que rencontrent au jour le jour nos différents acteurs.

C'est donc dans cette optique qu'une petite enquête a été menée auprès de ces personnes et la plupart des problèmes recensés sont les suivants :

1.1 La non-maitrise du hardware et du software

En tête de liste, ce problème est le plus récurrent chez la plupart des administrateurs réseaux et systèmes. 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.

1.2 Le manque de traçabilité et de suivi

Dans le réseau d'une entreprise, lorsqu'un équipement tombe en panne, nous avons du mal à le remplacer par un équipement adéquat (même système, même modèle, même périphérique, ...) qui remplirait exactement les mêmes fonctions. Ceci est dû au fait que le stock des ressources de l'entreprise n'est pas inventorié.

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

2. Etude de l'existant

Parmi les produits existants sur le marché, nous retrouvons :

2.1 Clarilog

Cette application a été créée par l'entreprise Clarilog France et permet entre autres :

- L'audit du parc informatique en utilisant le module Clarilog Fast Inventory qui permet de récolter les données sans déploiement d'agent.
- Une cartographie complète des équipements du parc.

2.2 H-inventory

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

- Gérer les incidents
- Effectuer du monitoring sur les services (alertes, mail...)

3. Solution proposée

Après une étude comparative sur les différentes solutions existantes, il est donc primordial au regard des inconvénients recensés de proposer une solution qui pourra répondre à nos besoins.

Nous proposons une solution (une application orientée web) qui proposant un ensemble de fonctionnalités. Cette solution permet aux responsables de passer une réclamation concernant un problème à provoquer une machine.

Notre plateforme est composée de 3 parties :

- **Une partie administrative :** C'est la partie qui permet la gestion de l'application web. Elle permet à l'administrateur de gérer les techniciens, gérer les interventions et affecter les réclamations.
- **Une partie pour responsable :** Cette partie est destinée au responsable permettant de passer une réclamation sur une machine qui est en panne et suivre leur état.
- **Une partie pour techniciens :** Cette partie est destinée au technicien permettant de consulter les réclamations et suivre leur état ainsi de créer une intervention.

Notre plateforme permet :

- **À l'administrateur :**

- Gérer les machines.
- Gérer les techniciens.
- Gérer les responsables.
- Affecter les réclamations.
- Suivre l'état de la réclamation.
- Gérer les interventions.
- Consulter le tableau de bord.
- S'authentifier.

➤ **À responsable :**

- Consulter les réclamations.
- Suivre l'état de la réclamation.
- Passer réclamations.
- S'authentifier.

➤ **À technicien :**

- Consulter les réclamations.
- Suivre l'état de la réclamation.
- Modifier l'état réclamations.
- S'authentifier.
- Gérer les interventions.

II. Méthodologie et modélisation adoptée

1.Méthodologie de conception

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie. [1]



Figure 1 Logo UML

UML permet grâce à sa représentation graphique, d'exprimer visuellement une solution objet, de faciliter la compréhension de l'évolution de la solution.

Cette étude se déroule en deux phases. Durant la première phase nous analyserons les besoins attendus de l'application par le biais des diagrammes de cas d'utilisations et les diagrammes de séquences pour approfondir la compréhension de l'exécution des fonctionnalités de l'application à développer. La deuxième phase s'intéresse au point de vue données à travers les diagrammes de classes.

2.Méthodologie de travail

Utiliser UML tout seul n'est pas suffisant pour modéliser correctement et convenablement un système logiciel. Nous avons besoin d'une pratique, d'un guide qui nous donne des étapes et des manières pour exploiter les diagrammes UML. Dans ce cadre viennent les méthodes agiles pour proposer les bonnes pratiques aidant le concepteur à avoir une méthode de conception. Pour cela, nous avons choisi de travailler avec la méthode SCRUM.

Scrum est une méthode de développement agile orientée projet informatique dont les ressources sont régulièrement actualisées. La méthode Scrum tire son nom du monde du rugby, Scrum = mêlée.

Le principe de base étant d'être toujours prêt à réorienter le projet au fil de son avancement. C'est une approche dynamique et participative de la conduite du projet. La mêlée est une phase de jeu essentielle au rugby. Elle permet au jeu de repartir sur d'autres bases. La réunion dans la méthode Scrum relaie la métaphore. [2]

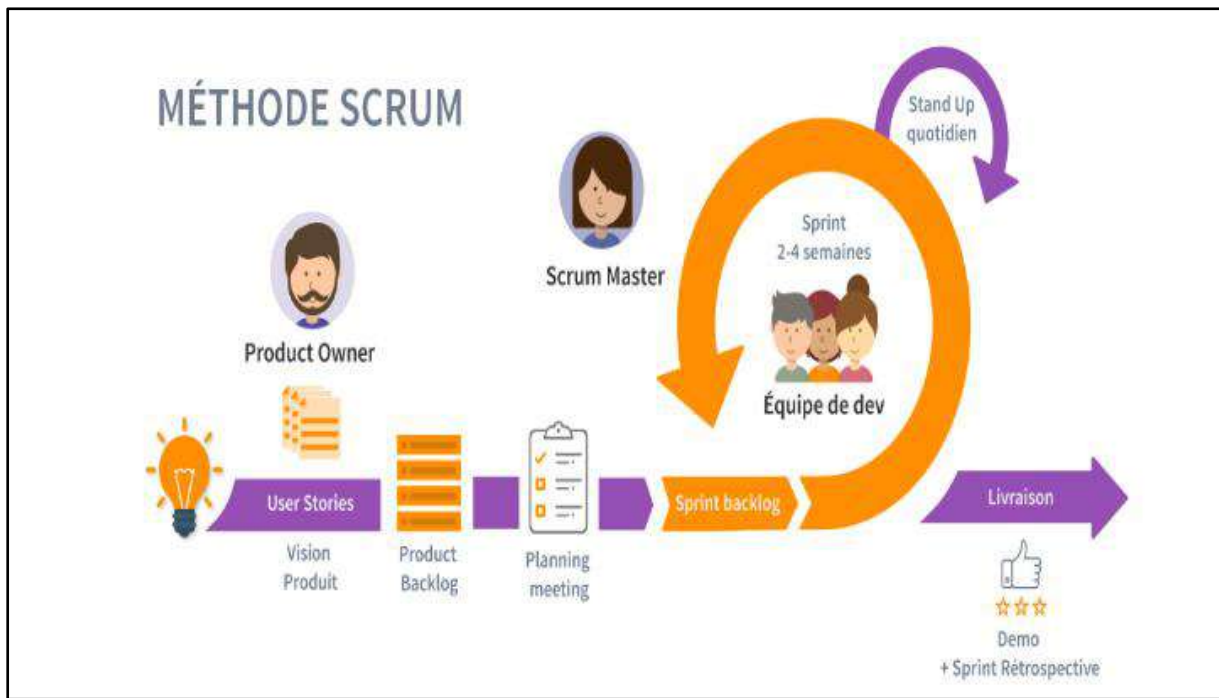


Figure 2 Le processus Scrum

III. Framework Scrum

Nous ne prétendons pas que notre projet répond aux exigences de SCRUM à cent pour cent mais nous nous en inspirons.

1.Équipe et Rôle

L'Équipe Scrum comprend un propriétaire de produit (Product Owner), une Équipe de Développement (Développement Team) et un Scrum Master.

- **Développement Team** : Ce sont les personnes chargées de la réalisation des sprints.
- **Product Owner** : C'est l'acteur clé du projet. C'est une partie des responsabilités du propriétaire par l'entreprise qui a une vision de ses besoins et le transmettre à l'équipe Scrum.
- **Scrum Master** : Il assure globalement le bon déroulement des programmes et protège l'équipe de tout problème extérieur. Il assure globalement l'organisation des réunions et la bonne application de la méthode agile.

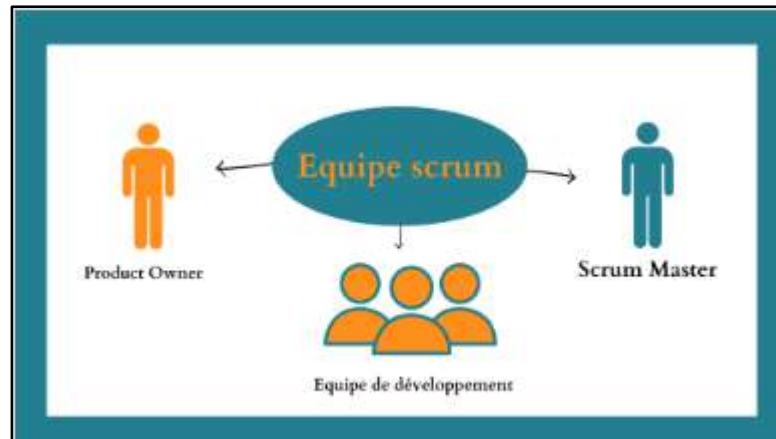


Figure 3 Equipe Scrum

2.Fixation du Sprint time Box

Avant d’entamer notre travail, nous fixons la durée des sprints à savoir les <sprints Time-Box>.

Pour notre projet, nous avons choisi que le sprint dur entre 7 et 20 jours.

3.Les cérémonies

3.1 Sprint planning

Nous organisons, avant chaque sprint, une réunion de planification, le sprint planning meeting. Ce planning sélectionne dans le Product backlog les exigences les plus prioritaires pour le client. Elles seront développées, testées et livrées à la fin du sprint. Elles constituent le sprint backlog, un sous-ensemble du Product backlog.

3.2 Daily meeting

L’objectif du Daily meeting est de s’assurer ensemble que nous avons bien atteint notre objectif chaque jours le Daily meeting se base sur trois questions :

- Qu’est-ce que tu as fait hier ?
- Quels sont les problèmes que tu as rencontrés ?
- Qu’est-ce que tu as prévu aujourd’hui ?

3.3 **Sprint Review**

Le sprint review, ou revue de sprint est la réunion destinée à évaluer les résultats obtenus par l'équipe scrum après un sprint. Elle permet d'analyser les progrès réalisés par le développement afin d'atteindre l'objectif fixé.

3.4 **Sprint Rétrospective**

Toute l'équipe Scrum participe à la réunion. La rétrospective a lieu juste après la revue de sprint et les intervenant qui sont venus y assister peuvent rester pour la rétrospective, à titre d'observateurs. Cependant, la confiance est nécessaire pour le succès d'une rétrospective et la présence de certaines personnes peut être animée par le Scrum Master. Mais, notamment dans les environnements difficiles, il est préférable que ce soit une personne extérieure à l'équipe qui joue le rôle de facilitateur de cette réunion le but de la réunion est d'améliorer le processus pour le prochain sprint.

Conclusion

Au cours de ce chapitre, nous avons présenté le cadre de notre projet de fin d'études et nous avons identifié l'objectif à atteindre et le travail à faire. Compte tenu de cela, nous présenterons la spécification des besoins fonctionnels et non-fonctionnels du projet dans le chapitre suivant.

Chapitre 2. Spécification des besoins

Introduction

- I- Spécification des besoins
- II- Modélisation des besoins fonctionnels
- III- Le product backlog

Conclusion

Introduction

Après avoir présenté le projet au niveau du chapitre précédent et avant de commencer la phase de développement, nous élaborons à ce niveau la spécification des besoins qui sert à bien organiser et clarifier les fonctionnalités du projet.

Ce chapitre consiste donc à déterminer l'ensemble des besoins de notre système et modéliser ses interactions, suivant la spécification de la méthodologie UML, tout en respectant certaines contraintes.

I. Spécification des besoins

Dans cette partie, nous détaillons les exigences spécifiques qui se manifestent en des besoins fonctionnels et des besoins non fonctionnels dont le but est d'identifier les services que doit fournir notre application.

1. Identification des acteurs

« Un acteur » représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Tout simplement un acteur est une entité physique (personne) ou abstraite (logiciel) capable d'utiliser le système afin de répondre à un besoin bien défini.

Les acteurs de notre application sont :

Acteur	Rôles
Administrateur	C'est l'administrateur de l'application web. Il effectue les réclamations aux techniciens, gère les techniciens, les utilisateurs (admin && responsables), les interventions, les machine, gère les codes QR de la machine et effectue le suivi des états des réclamations et les états de la machine.
Technicien	C'est utilisateur dans notre système. Il consulte leur réclamation concernant une machine qui a lui affecté. Et traiter une réclamation ainsi que changer leur état.
Responsable	C'est le responsable du parc il déclenche une réclamation concernant une machine et suivre a qui a été affecter ainsi que voir leurs états.

Tableau 1 Les acteurs

2.Exigences fonctionnelles

Les principaux besoins fonctionnels de notre application se résument autour des points suivants :

BF1 : Le système doit permettre à l'administrateur de s'authentifier et de se déconnecter.

BF2 : Le système doit permettre au responsable de s'authentifier et de se déconnecter.

BF3 : Le système doit permettre au technicien de s'authentifier et de se déconnecter.

BF4 : L'application Web doit permettre à l'administrateur de :

- BF4.1 : Gérer les machines.
- BF4.2 : Gérer les utilisateurs (admin responsables).
- BF4.3 : Gérer les techniciens (externe interne).
- BF4.4 : Gérer les interventions.
- BF4.5 : Suive les réclamations.
- BF4.6 : Affecter les réclamations.
- BF4.7 : Consulte tableau de bord.

BF5 : L'application Web doit permettre aux responsables de :

- BF5.1 :S'authentifier.
- BF5.2 : Passer une réclamation.
- BF5.3 : Gérer son compte.
- BF5.4 : Consulter et suivre l'état de leurs réclamations.

BF6 : L'application Web doit permettre aux techniciens de :

- BF6.1 :S'authentifier.
- BF6.2 : Consulte et traiter une réclamation.
- BF6.3 : Gérer son compte.
- BF7.4 : gérer les interventions.

3.Exigences non fonctionnelles

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur la performance de la plateforme, ce qui fait qu'ils ne doivent pas être négligés. Pour cela, il faut répondre aux exigences suivantes :

BNF1 : Sécurité : Les applications doivent être sécurisées et garantissent la protection et la confidentialité des données.

BNF2 : Performance : les applications doivent être performantes à travers leurs fonctionnalités et répondent d'une manière optimale et non-ambiguë à toutes les exigences d'utilisation.

BNF3 : Compatibilité

- Compatibilité sur différents systèmes d'exploitation.
- Compatibilité sur différentes plateformes

II. Modélisation des besoins fonctionnels

Une étude approfondie des besoins fonctionnels est indispensable avant d'entamer le développement pour obtenir, d'une manière plus formelle, une vue globale sur les exigences de notre application.

Cette partie présente alors une modélisation des besoins en faisant recours aux concepts fondamentaux d'UML, à savoir le diagramme de cas d'utilisation.

1.Diagrammes de cas d'utilisation

Un cas d'utilisation « use case » représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

1.1 Diagramme de cas d'utilisation de l'administrateur

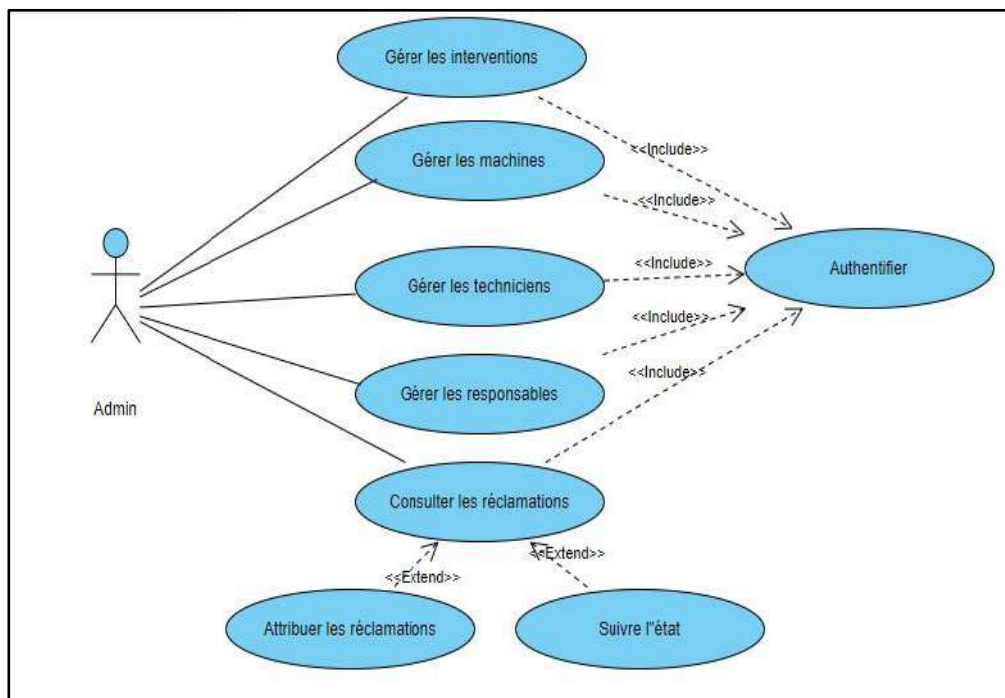


Figure 4 Diagramme de cas d'utilisation de l'administrateur

1.2 Diagramme de cas d'utilisation du responsable

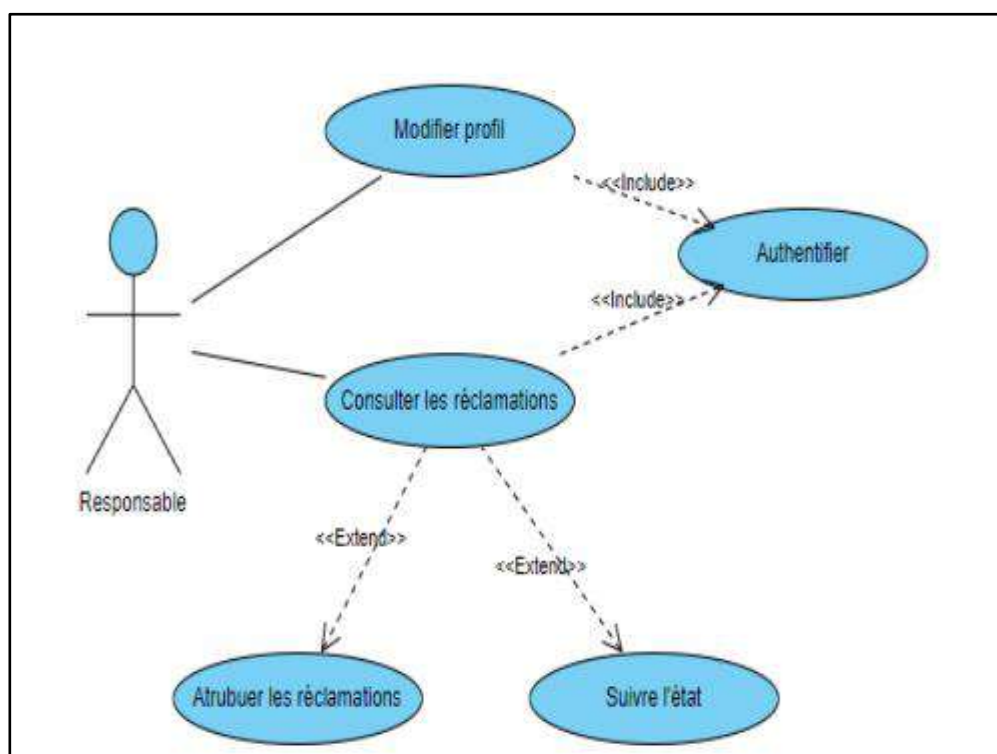


Figure 5 Diagramme de cas d'utilisation du responsable

1.3 Diagramme de cas d'utilisation du technicien

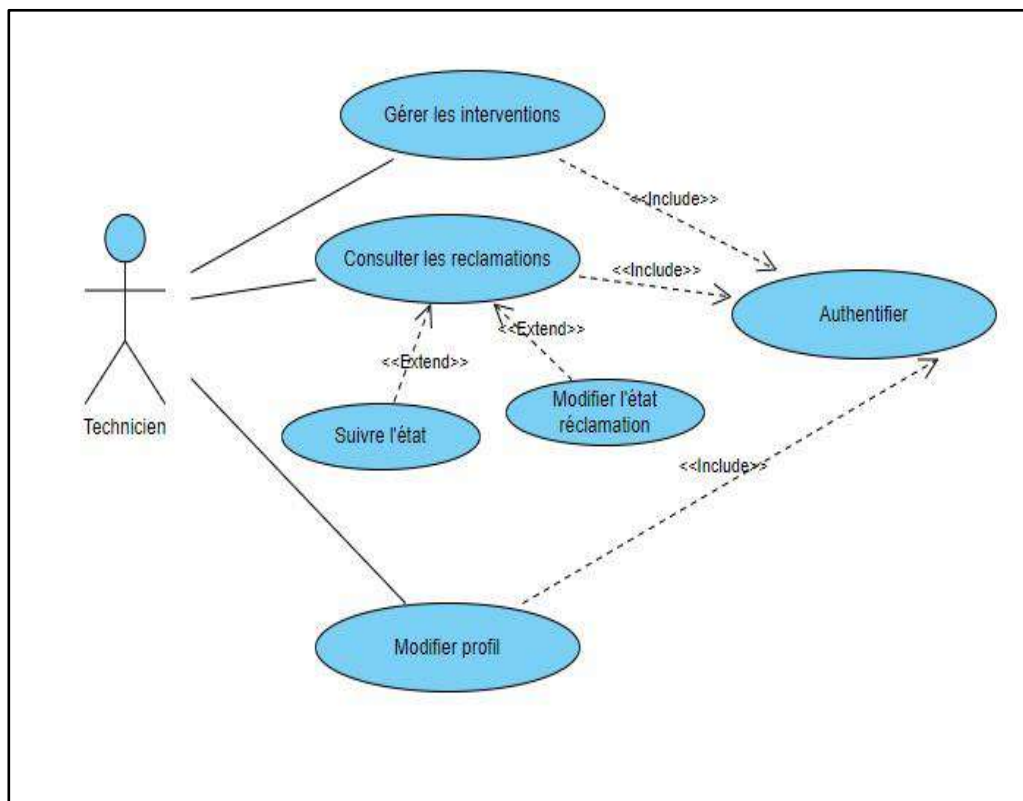


Figure 6 Diagramme de cas d'utilisation du technicien

2.Description textuelle

2.1 Description de l'item S'authentifier

2.1.1 Diagramme de cas d'utilisation S'authentifier

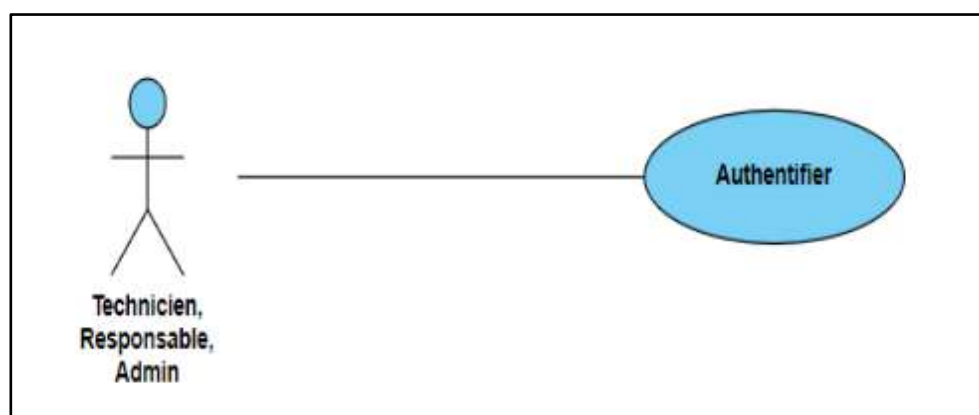


Figure 7 Diagramme de cas d'utilisation S'authentifier

2.1.2 Description textuelle du diagramme de cas d'utilisation

S'authentifier

Acteur	Administrateur, Responsable, Technicien
Objectif	Permet à l'acteur de s'authentifier
Pré-conditions	-Le système fonctionne -L'utilisateur saisit son login et son mot de passe
Post conditions	Ouverture de l'espace de l'acteur
Enchaînement nominal	1- Le système vérifie si les champs ne sont pas vides 2- Il vérifie ensuite si les informations sont valides 3- Le système redirige l'acteur vers son espace
Exception	- En (1) : message d'erreur « un champ est vide » - En (2) : message d'erreur « login ou mot de passe incorrect »

Tableau 2 Description textuelle de cas d'utilisation « S'authentifier »

2.2 Description de l'item gérer compte technicien

2.2.1 Diagramme de cas d'utilisation Gérer compte technicien

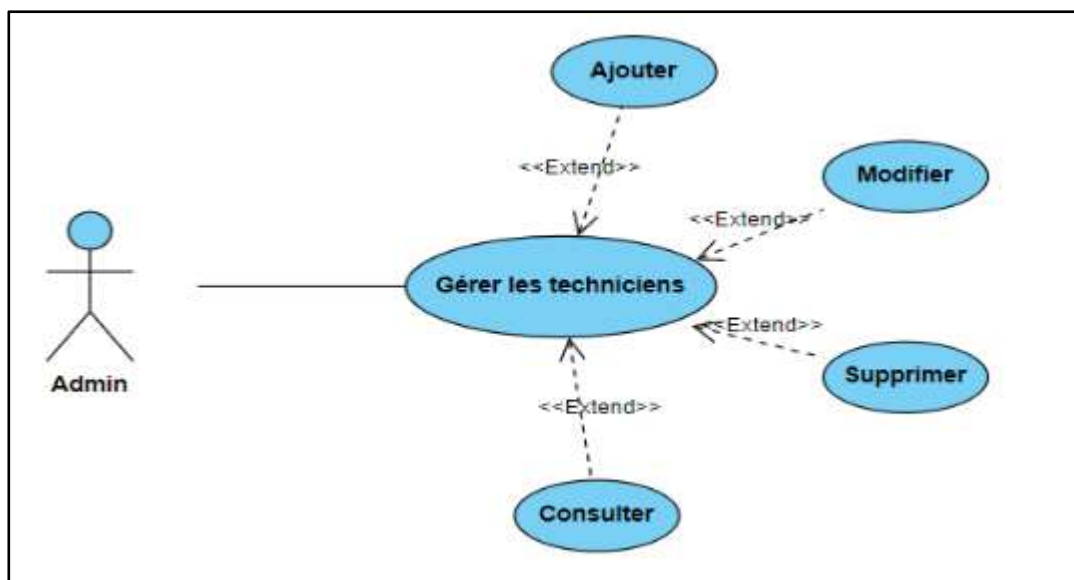


Figure 8 Diagramme de cas d'utilisation Gérer compte technicien

2.2.2 Description textuelle du diagramme de cas d'utilisation Gérer compte technicien

Acteur	Administrateur
Objectif	Permet à l'administrateur d'ajouter, modifier, supprimer et consulter un technicien.
Pré-conditions	-Le système fonctionne -S'authentifier
Post conditions	-Le technicien est bien ajouté -Le technicien est bien modifié -Le technicien est bien supprimé
Enchaînement nominal	<p>1-Le système affiche la page de la liste des techniciens</p> <p>Ajouter un technicien :</p> <p>1-L' administrateur clique sur le bouton «Ajouter»</p> <p>2-Le système renvoie le formulaire</p> <p>3-L' administrateur remplit les différents champs</p> <p>4-L' administrateur clique sur le bouton «Ajouter»</p> <p>5-Le système vérifie si les champs ne sont pas vides</p> <p>6- Il vérifie ensuite si les informations sont valides</p> <p>7- Le système ajoute un technicien et affiche un message de succès</p> <p>Modifier un technicien :</p> <p>1-L' administrateur clique sur l'icône modifié</p> <p>2-Le système renvoie le formulaire</p> <p>3-L' administrateur remplit les différents champs</p> <p>4-L' administrateur clique sur le bouton « Modifier»</p> <p>5-Le système vérifie si les champs ne sont pas vides</p> <p>6- Il vérifie ensuite si les informations sont valides</p> <p>7- Le système modifie un technicien et affiche un message de succès</p> <p>Supprimer un technicien :</p> <p>1- L'administrateur clique sur l'icône supprimé</p> <p>2- Le système demande la confirmation de suppression</p>

	3- L'administrateur confirme la suppression 4- Le système supprime le technicien et affiche un message de succès
Exception	Ajouter un technicien : - En (5) : message d'erreur « un champ est vide » - En (6) : message d'erreur « format est incorrecte » Modifier un technicien : - En (5) : message d'erreur « un champ est vide » - En (6) : message d'erreur « format est incorrecte »

Tableau 3 Description textuelle de cas d'utilisation « Gérer les personnels »

2.3 Description de l'item passer réclamations

2.3.1 Diagramme de cas d'utilisation Passé réclamations

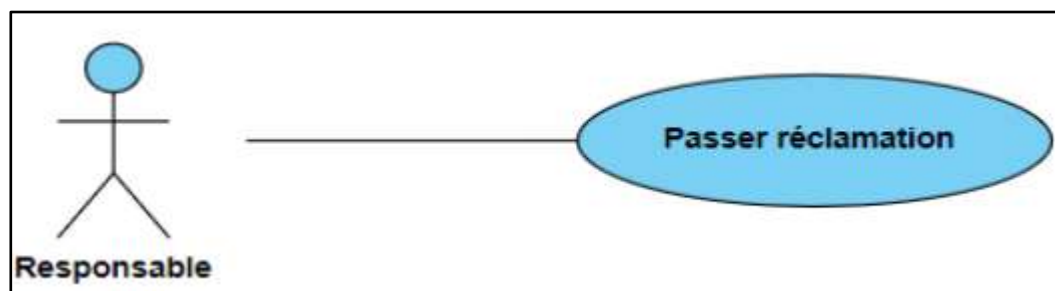


Figure 9 Diagramme de cas d'utilisation Passé réclamations

2.3.2 Description textuelle de diagramme de cas d'utilisation Passé réclamations

Acteur	Responsable
Précondition	Connexion internet Machine en panne Responsable demande d'ajouter une réclamation
Post-condition	Responsable est authentifié
Scenario nominal	1. Responsable ouvre le formulaire d'ajouter une réclamation 2. Responsable entre les détails ... 3. Le système vérifie la validité des informations entrées

	4. Le système ajoute la réclamation à la base de données et lui confirme de la création.
Exception	3. a Si les informations entrées par responsable ne sont pas valides ou vides, le système lui affiche un message d'erreur lui informant qu'il doit vérifier ses informations.

Tableau 4 Description textuelle du cas d'utilisation Passer réclamation

2.4 Description de l'item Suivre l'état de réclamation

2.4.1 Diagramme de cas utilisation Suivre l'état de réclamations

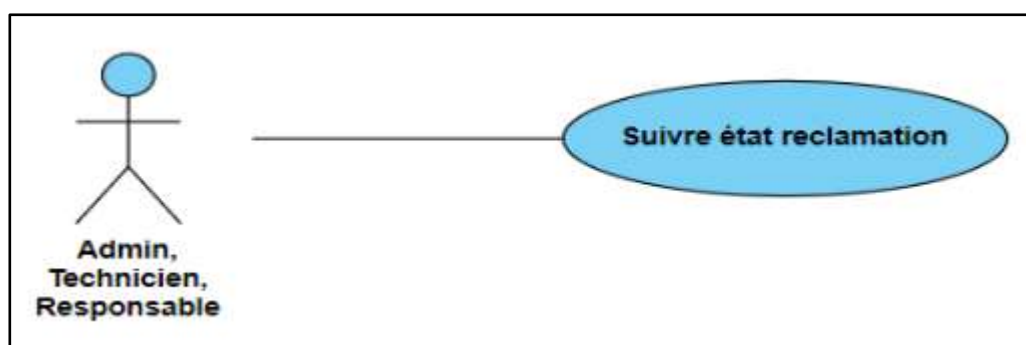


Figure 10 Diagramme de cas utilisation Suivre l'état de réclamations

2.4.2 Description textuelle de cas d'utilisations Suivre l'état de réclamations

Acteur	Administrateur, Responsable, Technicien
Objectif	Permet l'acteur de voir l'état des réclamations.
Préconditions	-Le système fonctionne -S'authentifier
Enchaînement nominal	1-L'acteur clique sur le bouton « les réclamations » 1-Le système affiche l'état des réclamations

Tableau 5 Description textuelle de cas d'utilisation «Suivre les états de l'offre»

2.5 Description de l'item gérer profil

2.5.1 Diagramme de cas utilisation gérer profil

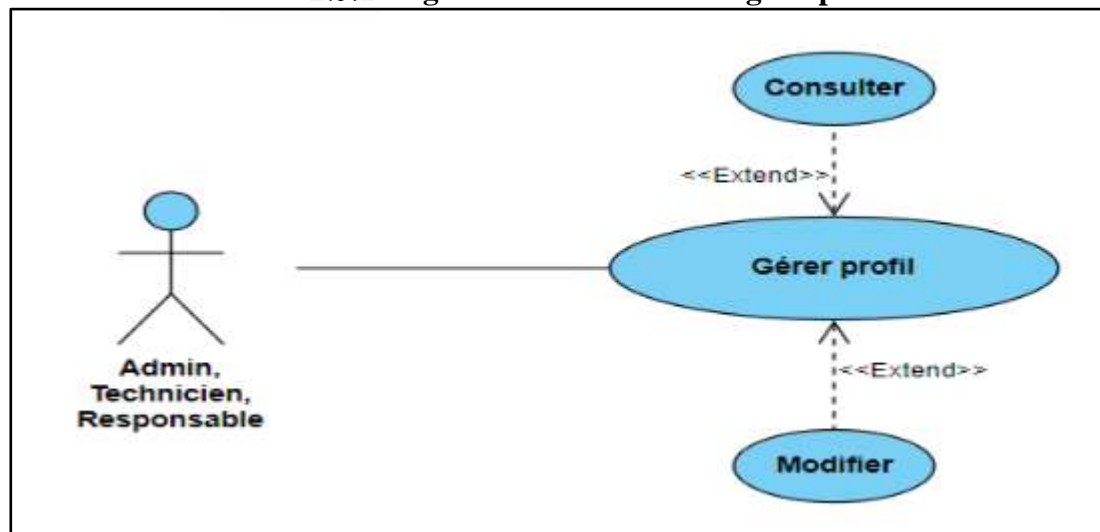


Figure 11 Diagramme de cas utilisation gérer profil

2.5.2 Description textuelle de diagramme de cas d'utilisation Gérer

profile	
Acteur	Administrateur, Responsable, Technicien
Objectif	Permet l'acteur de modifier son profil.
Pré-conditions	-Le système fonctionne -S'authentifier
Post conditions	Profil modifié.
Enchaînement nominal	1-l'acteur clique sur « Profil » 2-Le système affiche la page de profil 3-l'acteur remplit les champs de profil à modifier 4-l'acteur clique sur le bouton « Modifier » 5-Le système vérifie si le champ n'est pas vide 6-Le système modifie le profil et affiche un message de succès
Exception	- En (4) : message d'erreur « un champ est vide » - En (5) : message d'erreur « format est incorrecte »

Tableau 6 Description textuelle de cas d'utilisation «Editer profil»

2.6 Description de l'item changer état réclamation

2.6.1 Diagramme de cas d'utilisation Changer état réclamation

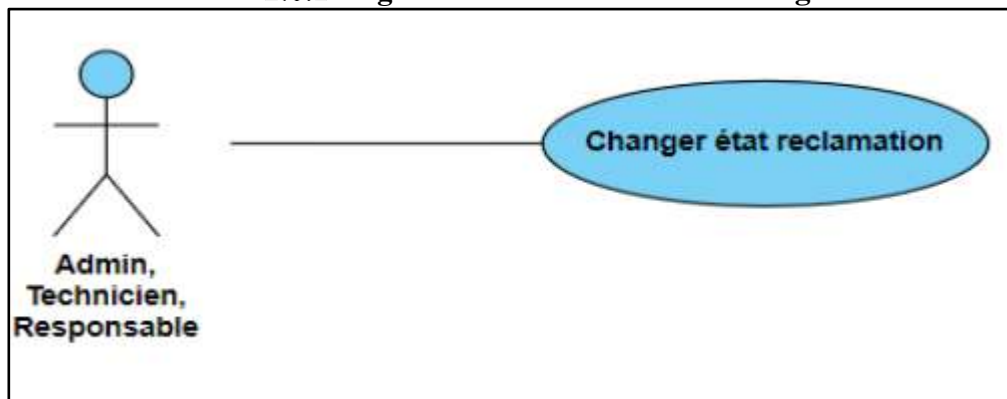


Figure 12 Diagramme de cas d'utilisation Changer état réclamation

2.6.2 Description textuelle de diagramme de cas d'utilisation Changer état réclamations

Acteur	Administrateur
Objectif	Permet à l'acteur d'ajouter, modifier et consulter les machines.
Pré-conditions	-Le système fonctionne -S'authentifier
Post conditions	- Machine est bien ajoutée - Machine est bien modifiée - Machine est bien supprimée
Enchaînement nominal	<p>1-Le système affiche la page de la liste des machines</p> <p><u>Ajouter une machine :</u></p> <p>1-L'acteur clique sur le bouton « Ajouter »</p> <p>2-Le système renvoie le formulaire</p> <p>3-L'acteur remplit les différents champs</p> <p>4-L'acteur clique sur le bouton « Ajouter»</p> <p>5-Le système vérifie si les champs ne sont pas vides</p>

	<p>6- Il vérifie ensuite si les informations sont valides</p> <p>7- Le système ajoute une machine et affiche un message de succès</p> <p><u>Modifier une machine :</u></p> <p>1-L'acteur clique sur l'icône modifier</p> <p>2-Le système renvoie le formulaire</p> <p>3-L'acteur remplit les différents champs</p> <p>4-L'acteur clique sur le bouton « Modifier »</p> <p>5-Le système vérifie si les champs ne sont pas vides</p> <p>6- Il vérifie ensuite si les informations sont valides</p> <p>7- Le système modifie une machine et affiche un message de succès</p> <p><u>Supprimer une machine :</u></p> <p>1- L'acteur clique sur l'icône supprimé</p> <p>2- Le système demande la confirmation de suppression</p> <p>3- L'acteur confirme la suppression</p> <p>4- Le système supprime machine et affiche un message de succès</p>
Exception	<p><u>Ajouter une machine :</u></p> <ul style="list-style-type: none"> - En (5) : message d'erreur « un champ est vide » - En (6) : message d'erreur « format est incorrect » <p><u>Modifier une machine :</u></p> <ul style="list-style-type: none"> - En (5) : message d'erreur « un champ est vide » - En (6) : message d'erreur « format est incorrect »

Tableau 7 Description textuelle de diagramme de cas d'utilisation gérer les machines

2.7 Description de l'item gérer interventions

2.7.1 Diagramme de cas d'utilisation Gérer interventions

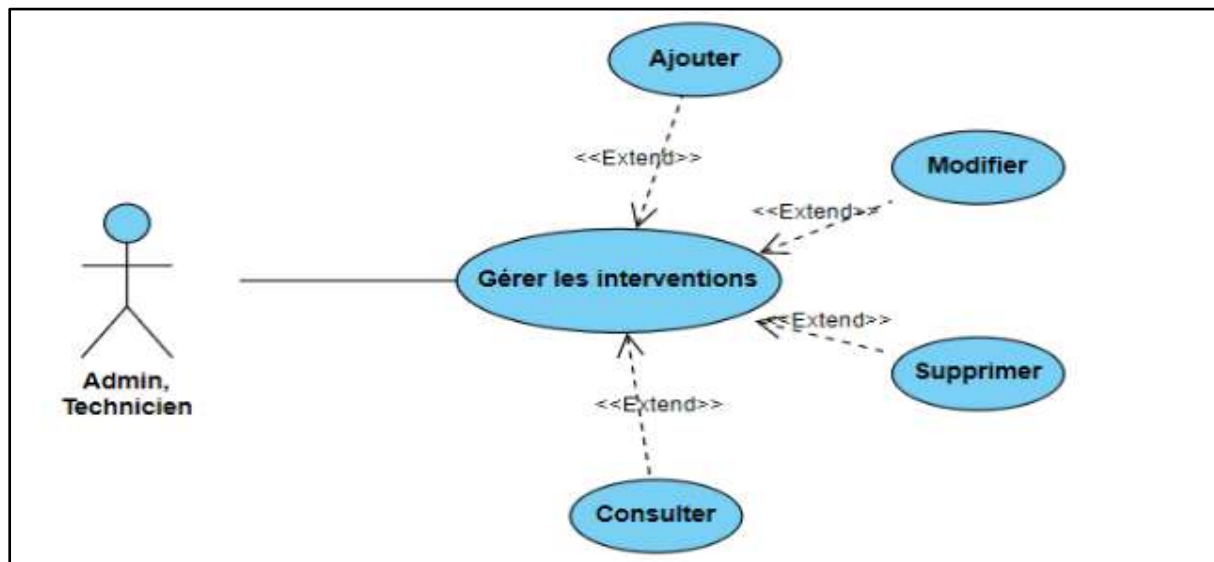


Figure 13 Diagramme de cas d'utilisation Gérer interventions

2.7.2 Description textuelle de diagramme de cas d'utilisation Gérer interventions

Acteur	Technicien, Administrateur
Objectif	Permet à l'acteur d'ajouter, modifier et consulter les interventions.
Pré-conditions	-Le système fonctionne -S'authentifier
Post conditions	- L'intervention est bien ajoutée - L'intervention est bien modifiée - L'intervention est bien supprimée
Enchaînement nominal	1-Le système affiche la page de la liste des rôles <u>Ajouter une intervention :</u> 1-L'acteur clique sur le bouton «Ajouter» 2-Le système renvoie le formulaire

	<p>3-L'acteur remplit les différents champs</p> <p>4-L'acteur clique sur le bouton «Ajouter»</p> <p>5-Le système vérifie si les champs ne sont pas vides</p> <p>6- Il vérifie ensuite si les informations sont valides</p> <p>7- Le système ajoute une intervention et affiche un message de succès</p> <p><u>Modifier une intervention :</u></p> <p>1-L'acteur clique sur l'icône modifier</p> <p>2-Le système renvoie le formulaire</p> <p>3-L'acteur remplit les différents champs</p> <p>4-L'acteur clique sur le bouton « Modifier»</p> <p>5-Le système vérifie si les champs ne sont pas vides</p> <p>6- Il vérifie ensuite si les informations sont valides</p> <p>7- Le système modifie une intervention et affiche un message de succès</p> <p><u>Supprimer une intervention :</u></p> <p>1- L'acteur clique sur l'icône supprimé</p> <p>2- Le système demande la confirmation de suppression</p> <p>3- L'acteur confirme la suppression</p> <p>4- Le système supprime l'intervention et affiche un message de succès</p>
Exception	<p><u>Ajouter une intervention :</u></p> <p>- En (5) : message d'erreur « un champ est vide »</p>

	<p>- En (6) : message d'erreur « format est incorrect»</p> <p><u>Modifier une intervention :</u></p> <p>- En (5) : message d'erreur « un champ est vide »</p> <p>- En (6) : message d'erreur « format est incorrect»</p>
--	--

Tableau 8 Description textuelle de cas d'utilisation «Gérer les rôles»

III. Le product backlog

ID	User Story	En tant que	Description	Priorité	Durée	Sprint
1	Interface Administrateur	Administrateur	L'admin consulte cette interface pour gérer plusieurs actions	Haut	5 jours	Sprint 1
2	Interface responsable	Responsable	Le responsable consulte cette interface pour gérer leurs actions	Haut	3 jours	
3	Interface technicien	Technicien	Le technicien consulte cette interface pour gérer leurs actions	Haut	3 jours	
4	Authentification	Responsable, admin, technicien	Authentification et accès à l'espace utilisateur	Moyenne	2 jours	
5	Gestion de profil	Responsable, admin, technicien	L'administrateur peut modifier son compte et les autres comptes dès les utilisateurs (technicien et responsable)	Haut	2 jours	
6	Gestion des utilisateurs	Administrateur	L'administrateur peut gérer les utilisateurs (admin et responsables)	Moyenne	3 jours	
7	Gestion des techniciens	Administrateur	L'administrateur peut gérer les techniciens	Moyenne	3 jours	
8	Gestion des machines	Administrateur	L'administrateur peut gérer les machines.	Moyenne	3 jours	Sprint2
9	Gestion des interventions	Administrateur	L'administrateur peut gérer les interventions.	Moyenne	3 jours	
10	Consulter la liste des réclamations	Administrateur	L'administrateur peut consulter la liste des réclamations.	Faible	2 jours	
11	Affecter des réclamations et	Administrateur	L'administrateur peut affecter les réclamations et suivre leur état.	Haut	5 jours	

	suivre leur état					
12	Consulter les réclamations.	Responsable	Le responsable peut consulter les réclamations	Faible	2 jours	Sprint 3
13	Passer une réclamation et suivre l'état d'une machine	Responsable	Le responsable peut passer une réclamation et suivre l'état d'une machine	Haut	5 jours	
	Suivre réclamation	Responsable	Responsable peut suivre l'état de sa réclamation.	Haut	3 jours	
15	Gestion intervention	Technicien	Le technicien peut gérer les interventions.	Moyenne	3 jours	Sprint 4
16	Consulter les réclamations	Technicien	Le technicien peut consulter les réclamations	Faible	2 jours	
17	Modifier l'état d'une réclamation	Technicien	Le technicien peut modifier l'état d'une réclamation.	Faible	2 jours	

Tableau 9 Product backlog

Conclusion

Dans ce chapitre, nous avons présenté les exigences fonctionnelles et non fonctionnelles de notre système ainsi que le raffinement des diagrammes des cas d'utilisation pour montrer les interactions entre les différents acteurs et le système. Ensuite, nous avons présenté les différents sprints à travers le Product backlog.

Le chapitre prochain sera consacré à l'étude conceptuelle.

Chapitre 3. Etude Conceptuelle

Introduction

- I- Architecture du système
- II- Conception du sprint 1
- III- Conception du sprint 2
- IV- Conception du sprint 3
- V- Conception du sprint 4
- VI- Diagramme de classes

Conclusion

Introduction

La partie conception est la partie importante dans la réalisation de n'importe quel projet. Elle permet de décrire d'une manière non ambiguë le fonctionnement d'un système afin de simplifier sa réalisation. Ce chapitre sera consacré à la conception de notre application.

Dans ce chapitre, nous identifions en premier lieu l'architecture de notre système. Par la suite, nous présentons quelques diagrammes de séquences et le diagramme de classes de notre projet.

I. Architecture du système

1. L'architecture REST API

API (interface de programme d'application) est un ensemble de règles permettant à différents programmes de communiquer entre eux. Il décrit la manière appropriée pour un développeur de logiciels de composer un programme sur un serveur qui communique avec diverses applications clientes.

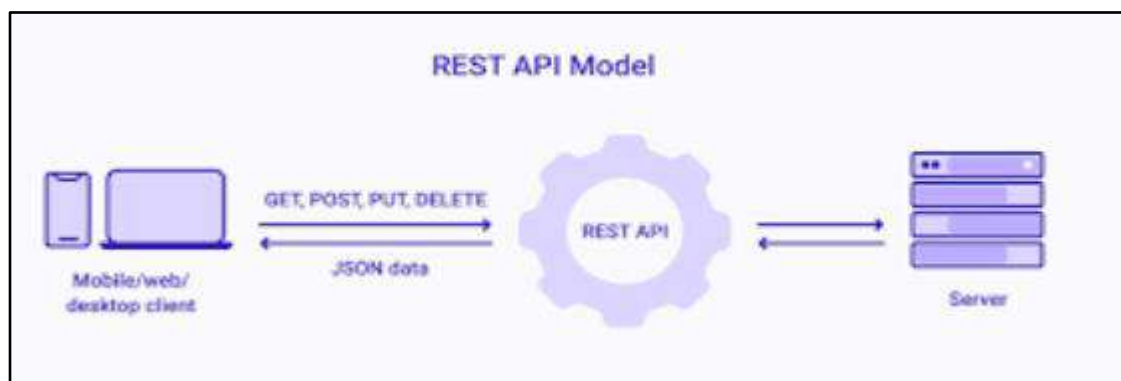


Figure 14 Architecture REST API

2. Model repository pattern

Model repository pattern est une abstraction de la couche d'accès aux données. Il masque les détails de la manière exacte dont les données sont enregistrées ou extraites de la source de données sous-jacente. Les détails de la façon dont les données sont stockées et récupérées se trouvent dans le référentiel respectif.

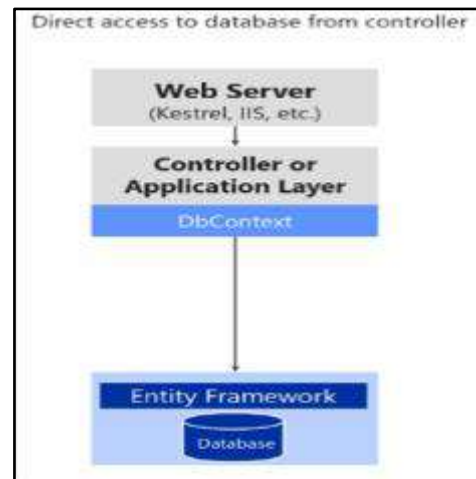


Figure 15 Model repository pattern

II. Conception du sprint 1

Dans le monde des méthodes Agiles, le sprint désigne le cycle de développement pendant lequel vont s'enchaîner un certain nombre de tâches pour, à terme, s'achever par la conception d'un produit final.

Cela se caractérise par un rassemblement de personnes impliquées dans un projet afin de se concentrer sur le développement de ce projet.

1. Sprint Goal

Est une brève explication de ce que l'équipe prévoit d'accomplir au cours d'un sprint Agile. Il s'agit d'un objectif tangible écrit ensemble par l'équipe et le Product Owner et limité dans le temps à la durée du sprint. En d'autres termes, les objectifs de sprint clarifient votre objectif lors d'un incrément de programme.

Notre but ici est de réaliser les différentes interfaces de l'application web.

2. Sprint Backlog

Le sprint backlog, c'est la liste de l'ensemble des éléments à réaliser tout au long du sprint afin d'être en mesure de livrer l'incrément à la fin de celui-ci. Ces éléments en question proviennent du Product backlog qui a été construit en amont, à la base du projet Scrum.

Une fois, nous avons défini le but du sprint, il est temps de décider quelles sont les fonctionnalités et leurs degrés d'importance incluent dans ce dernier. Plus précisément, quels noms de Backlog du produit seront inclus dans le Backlog du sprint.

User story	Taches	Degré de difficulté	Priorité
L'interface de technicien	Intégration des pages pour le technicien	Facile	Elevé
L'interface de l'administrateur	Intégration de la page pour l'administrateur	Facile	
L'interface de responsable	Intégration de la page pour le responsable.	Facile	
L'authentification	Intégration et testes nécessaires : Un formulaire login. Chaque personne est redirigée vers son interface grâce au middleware	Moyenne	

Tableau 10 Backlog sprint 1

3.Diagramme de séquence « S'authentifier »

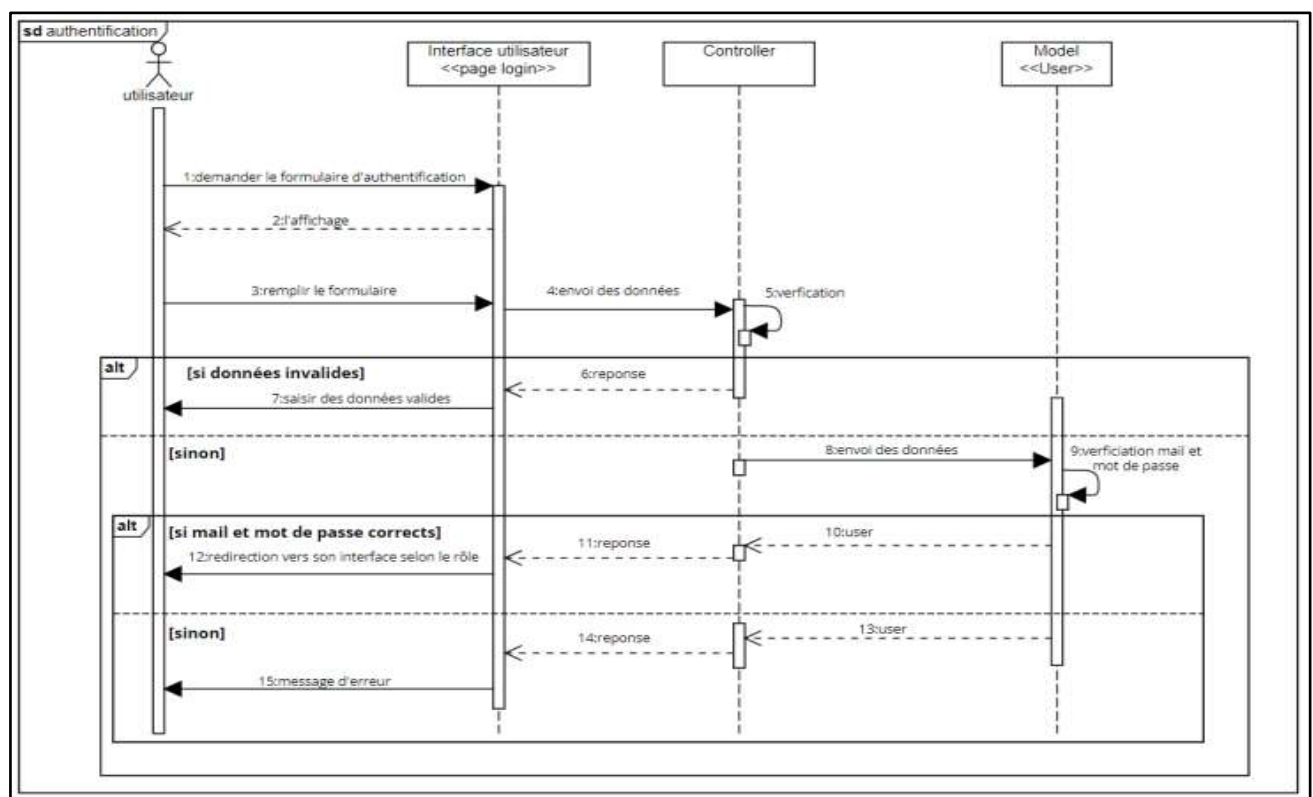


Figure 16 Diagramme de séquence « s'authentifier »

Lorsque l'utilisateur demande l'accès à l'application, il doit tout d'abord s'identifier par son email et son mot de passe via l'interface d'authentification. Ainsi les données seront envoyées au serveur d'application qui prend en charge la vérification et la consultation de la base de données. S'il est accepté, il y aura donc l'accès au système et aux interfaces de menu correspondant. Sinon, le serveur d'application lui affiche un message d'erreur afin de rectifier sa donnée.

III. Conception du sprint 2

Dans ce qui suit on va déterminer l'étude détaillée du deuxième sprint.

1. Sprint Goal

Le but de ce sprint est de développer les tâches de l'administrateur, la gestion des interventions, la gestion des machines, les techniciens, les responsables et suivie les réclamations.

2. Sprint backlog

Le tableau ci-dessous résume le Backlog de notre 2ème sprint.

User story	Tâches	Degré de difficulté	Priorité
Gestion des machines.	L'ajout, la modification et les suppressions des machines. Consultation de la liste des machines.	Moyenne	Elevé
Gestion des interventions.	L'ajout, la modification et les suppressions interventions. Consultation de la liste des interventions.	Moyenne	
Gestion techniciens.	L'ajout, la modification et la suppression des techniciens. Consultation de la liste des techniciens.	Moyenne	
Consulter la liste des réclamations	Consulter la liste des réclamations.	Facile	
Gestion des responsables	Consultation de la liste des responsables. L'ajout de la modification et la suppression responsables.	Moyenne	

Suivre les réclamations	Consultation de la liste des réclamations. Changer l'état d'une réclamation.	Facile	
--------------------------------	---	--------	--

Tableau 11 Backlog sprint 2

3. Diagramme de séquence «Gestion des machines»

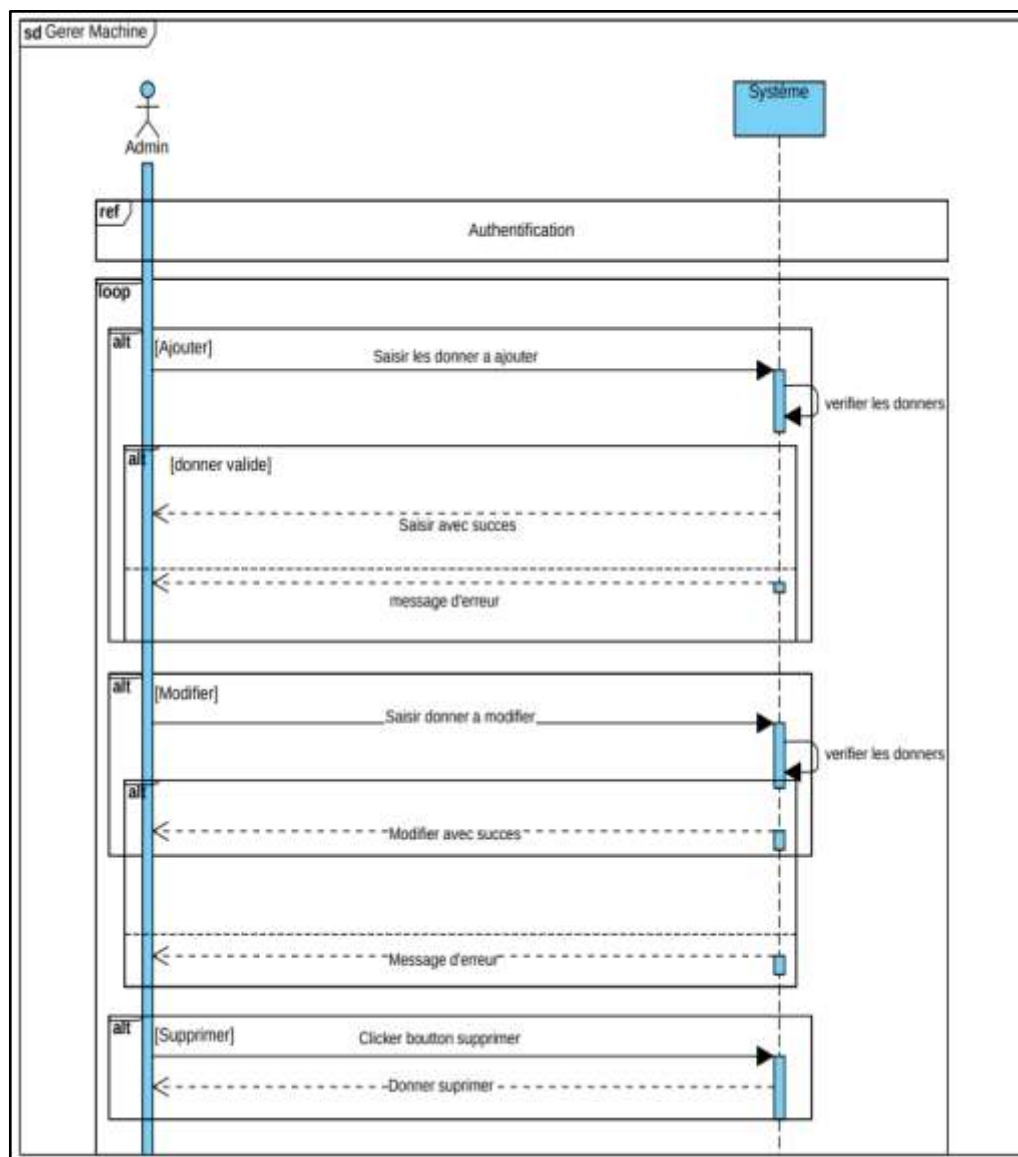


Figure 17 Diagramme de séquence « Gestion des machines »

Après l'authentification, l'utilisateur a le droit d'ajouter, modifier ou supprimer une machine. A propos de l'ajout, il suffit que l'utilisateur saisisse les informations nécessaires pour la création, par suite ces informations seront envoyées au serveur afin de les vérifier. Si les données sont valides, le système affichera un message de succès à l'utilisateur et la machine sera ajoutée, sinon il affichera un message d'erreur.

Pour la modification, l'utilisateur doit saisir les données à modifier, après ces données vont être vérifiées par le système, si les données sont valides alors un message de succès sera affiché à l'utilisateur et la machine sera modifiée, sinon un message d'erreur sera affiché par le système à l'utilisateur. En cas de suppression, l'utilisateur demande la suppression d'une machine, le système demande par conséquent la confirmation de la suppression, dès que l'utilisateur la confirme, la machine sera supprimée, et le système affichera un message de succès.

4. Diagramme de séquence «Suivi l'état de réclamations»

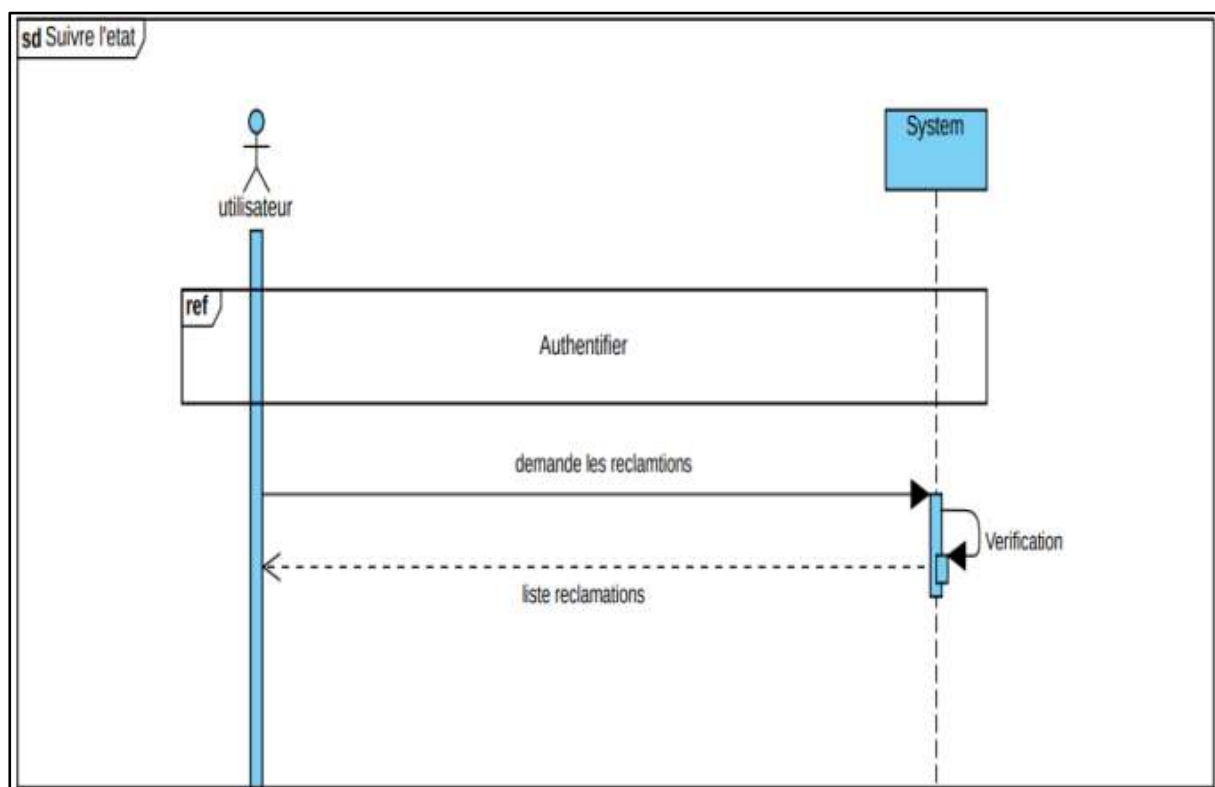


Figure 18 Diagramme de séquence « Suivi l'état de réclamation »

Après l'authentification, l'utilisateur clique sur le bouton des réclamations, par conséquent le système va se traiter et affichera la liste des réclamations.

IV. Conception du sprint 3

En partant sur le même principe que les sprints précédents, nous commençons par définir le but du sprint.

1. Sprint Goal

Le but de ce sprint est de réaliser la partie concernant les responsables et leurs tâches comme la passer une réclamation et suivre leur l'état.

2. Sprint Backlog

Le tableau ci-dessous résume le Backlog de notre quatrième sprint.

User story	Tâches	Degré de difficulté	Priorité
Passer une réclamation	Intégration d'un formulaire pour le passage de la réclamation	Moyenne	Elevé
	L'affichage la liste des réclamations.		
Suivre l'état de la réclamation	L'interface pour suivre l'état de la réclamation.		

Tableau 12 Backlog sprint 3

2.1 Diagramme de séquence «Passer réclamation»

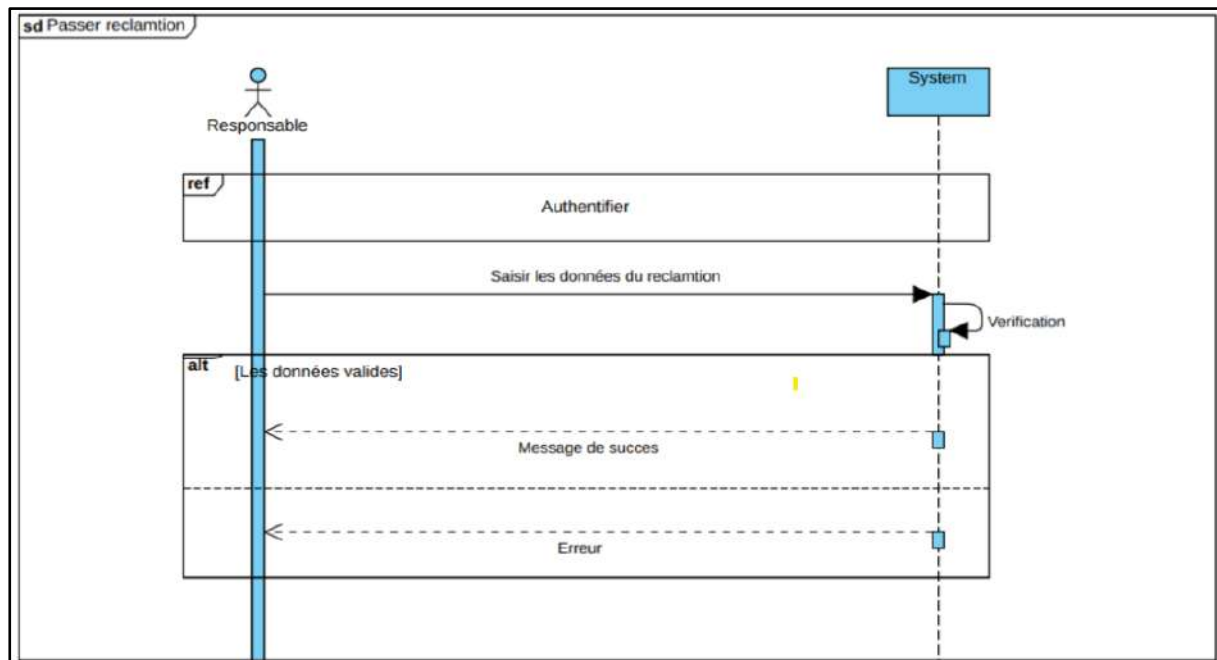


Figure 19 Diagramme de séquence « passer réclamation »

Le responsable saisie son réclamation, le système par conséquent va vérifier les réclamations : si elle est valide, il affichera un message de succès à l'utilisateur, et si elle est invalide, le système affichera un message d'erreur.

V. Conception du sprint 4

En partant sur le même principe que les sprints précédents, nous commençons par définir le but du dernier sprint.

1.Sprint Goal

Le but de ce sprint est de commencer la partie concernant les techniciens et leurs tâches comme la changer l'état d'une réclamation et gérer les interventions.

2. Sprint Backlog

Le tableau ci-dessous résume le Backlog de notre quatrième sprint.

User story	Tâches	Degré de difficulté	Priorité
Changer l'état d'une réclamation	Intégration d'un formulaire pour le passage de la réclamation	Moyenne	Elevé
	L'affichage la liste des réclamations.		
Consulter les réclamations	L'interface pour consulter les réclamations.		
Gestion des interventions	L'interface pour gérer les interventions.		

Tableau 13 Backlog sprint 4

3. Diagramme de séquence «Changer l'état d'une réclamation»

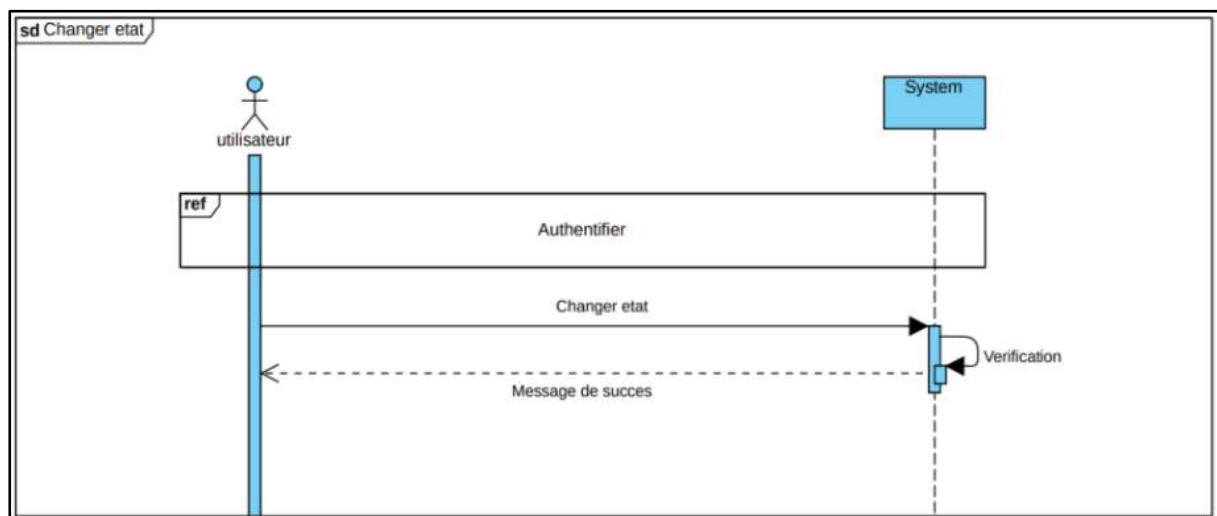


Figure 20 Diagramme de séquence « réclamation »

Après l'authentification, l'utilisateur clique sur le bouton de l'état de réclamation, par conséquent le système va se traiter et affichera un message de succès à l'utilisateur.

VI. Diagramme de classes

1. Diagramme de classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques.

La figure suivante représente le diagramme de classes de notre application :

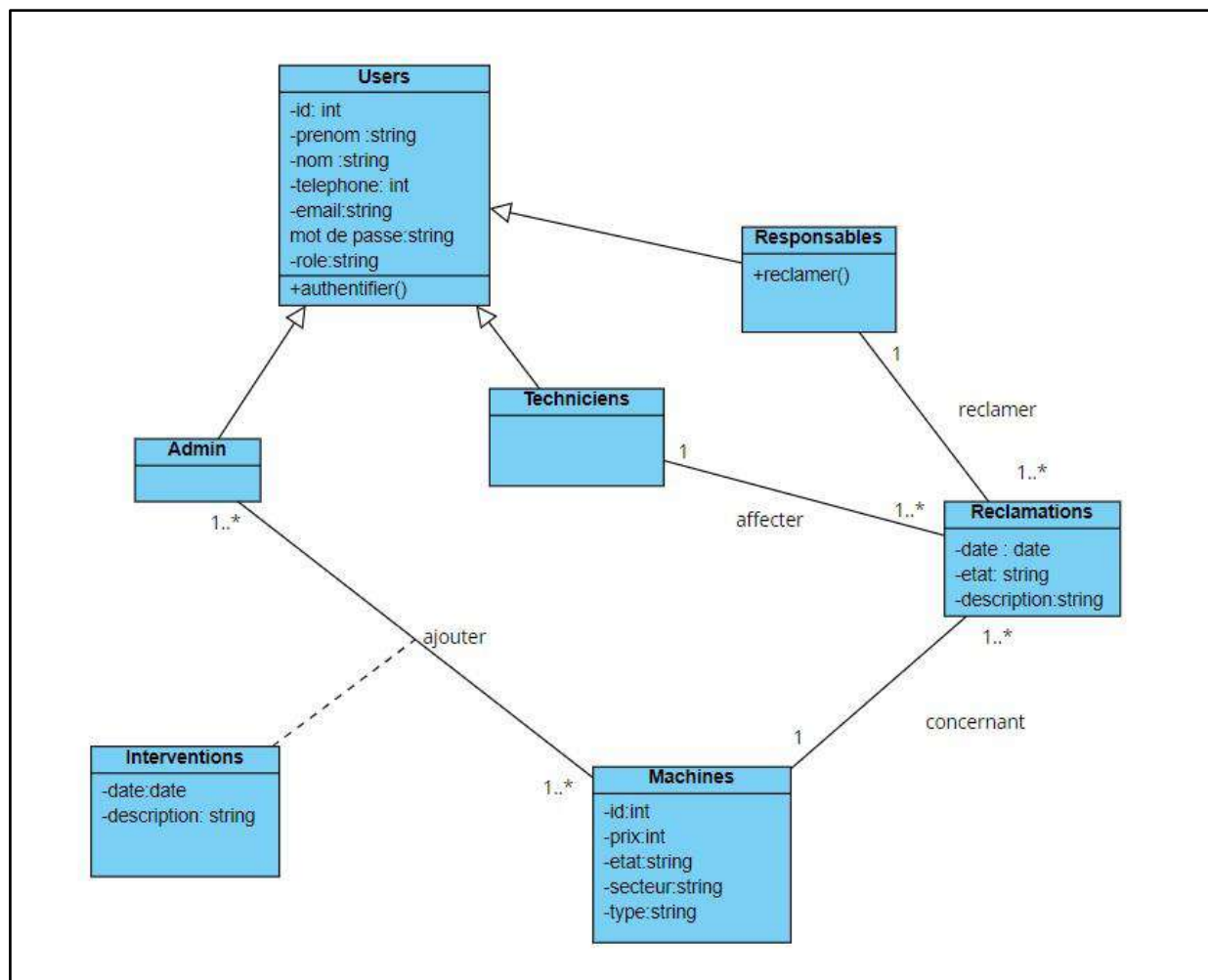


Figure 21 Diagramme de classes

2. Correspondance entre modèles objets et relationnels

- **Users** (id, nom, role, password, email, telephone);
- **Reclamations** (id, date, état, description, #IdMachine, #IdUser);
- **Machines** (id, code, prix, état, secteur, type);
- **Interventions** (id, date, description, #IdMachine, #IdUser);

Conclusion

Ce chapitre a été consacré à différentes étapes de la conception détaillée. En effet nous avons eu recours à différents diagrammes de conception diagramme de séquence et de classe afin de montrer les interactions entre les acteurs et le système et présenté la structure de notre application.

Cette étape nous prépare la phase de développement et de réalisation qui sera l'objectif de prochain chapitre.

Chapitre 4. Réalisation de l'application web

Introduction

I- Choix Techniques

II- Les interfaces

Conclusion

Introduction

Nous abordons dans ce chapitre le dernier volet de ce rapport, qui a pour objectif d'exposer la phase de réalisation. La phase de réalisation est considérée comme étant la concrétisation finale de toute la méthode de conception.

Ce chapitre, nous présentons les techniques utilisées, l'environnement logiciel de développement ainsi que les étapes de réalisation de notre solution en nous basant sur quelques captures d'écrans.

I. Choix Techniques

1. Les outils de développement





LOGO	Description
	JAVA est un langage de programmation largement utilisé pour coder des applications web. Il a été fréquemment choisi parmi les développeurs depuis plus de deux décennies, des millions d'applications Java étant utilisées aujourd'hui. [3]
	JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. [4]
	Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML. [5]
	Signifie « <i>HyperText Markup Language</i> » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. [6]

Tableau 14 Les outils de développement

2. Les Frameworks

Nous avons utilisées JAVA SPRING BOOT et ANGULAR JS comme des Framework pour le développement l'application.

2.1 JAVA SPRING BOOT

Un Framework développé par Pivotal en 2012 qui connaît depuis environ 4 ans, une explosion du nombre d'utilisateurs.

Un Framework de développement applicatif Java open source. Il est particulièrement recommandé pour le développement d'API.

Grâce à une configuration par défaut, une application Spring Boot permet de développer plus rapidement en se focalisant essentiellement sur le code métier. On dira que Spring Boot a une « opinion ». Bien entendu, ces fonctionnalités par défaut peuvent être modifiées pour correspondre aux besoins. Cela implique de respecter les conventions et règles imposées par Spring afin d'éviter de perdre trop de temps sur de la configuration applicative. Cela nous permet notamment d'éviter d'écrire de multiples fichiers de configuration en XML. [7]



Figure 22 JAVA SPRING BOOT



2.2 ANGULAR JS

Framework côté client, open source, basé sur TypeScript, et co-dirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète d'AngularJS, cadriciel construit par la même équipe. Les applications Angular se chargent rapidement grâce au nouveau Component Router, 33 qui permet de diviser automatiquement le code afin que les utilisateurs ne chargent que le code nécessaire au rendu de la vue qu'ils demandent.[8]



Figure 23 ANGULAR JS

3. Environnement logiciel

Nom de Logiciel	Logo	Utilisation
Visual Studio Code		Visual Paradigm Online ("VP Online") est Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto complétion, la coloration syntaxique, le débogage, et les commandes git. un outil de création de diagrammes basé sur le Web qui prend en charge un grand nombre de diagrammes commerciaux et techniques.
Eclipse		Eclipse est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français). [9]
Visual Paradigm		Visual Paradigm Online ("VP Online") est un outil de création de diagrammes basé sur le Web qui prend en charge un grand nombre de diagrammes commerciaux et techniques. [10]




MySQL		MySQL est un serveur de bases de données, bases de données relationnelles Open Source.[11]
Xampp		XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique.[12]
Word		Le logiciel Word permet de faire un traitement de texte et aussi de réaliser un tableau et de la mettre en forme. [13]

Tableau 15 L'environnement logiciel

II. Les interfaces

1.Partie administrateur

Dans cette partie, nous illustrons quelques interfaces de nos applications Web partie administrateur.

1.1 Interface d'authentification

La figure suivante présente l'interface d'authentification permettant à chaque acteur de se connecter.



Figure 24 Page d'authentification

1.2 Interface de tableau de bord

La figure suivante présente l'interface de tableau de boards.

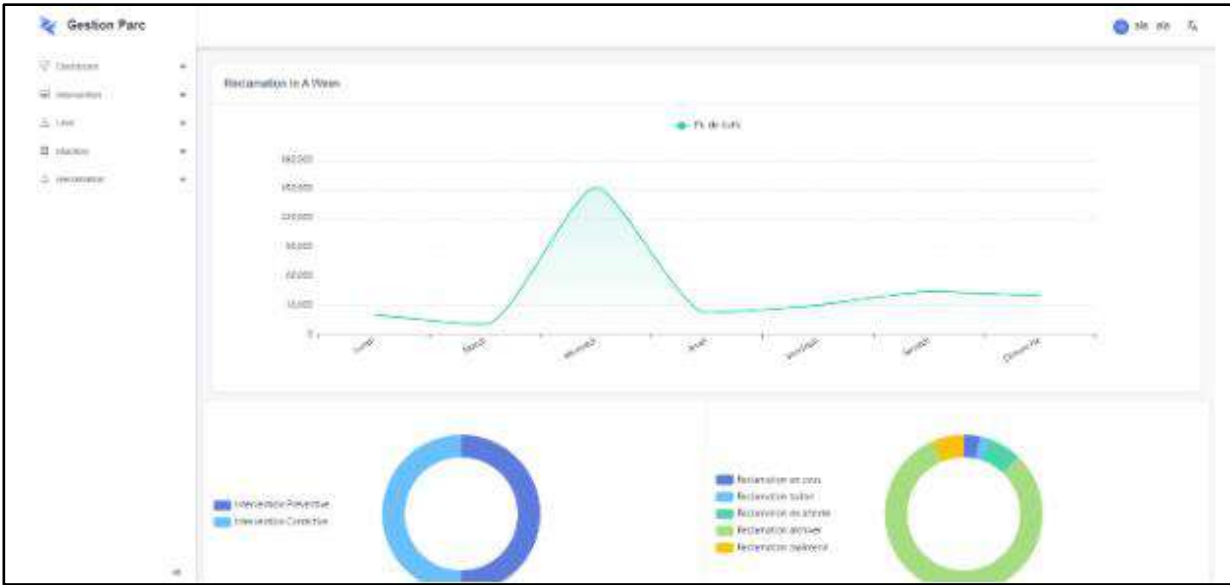


Figure 25 Page de tableau de bord

1.3 Interface gestion intervention

La figure suivante représente la liste des interventions.

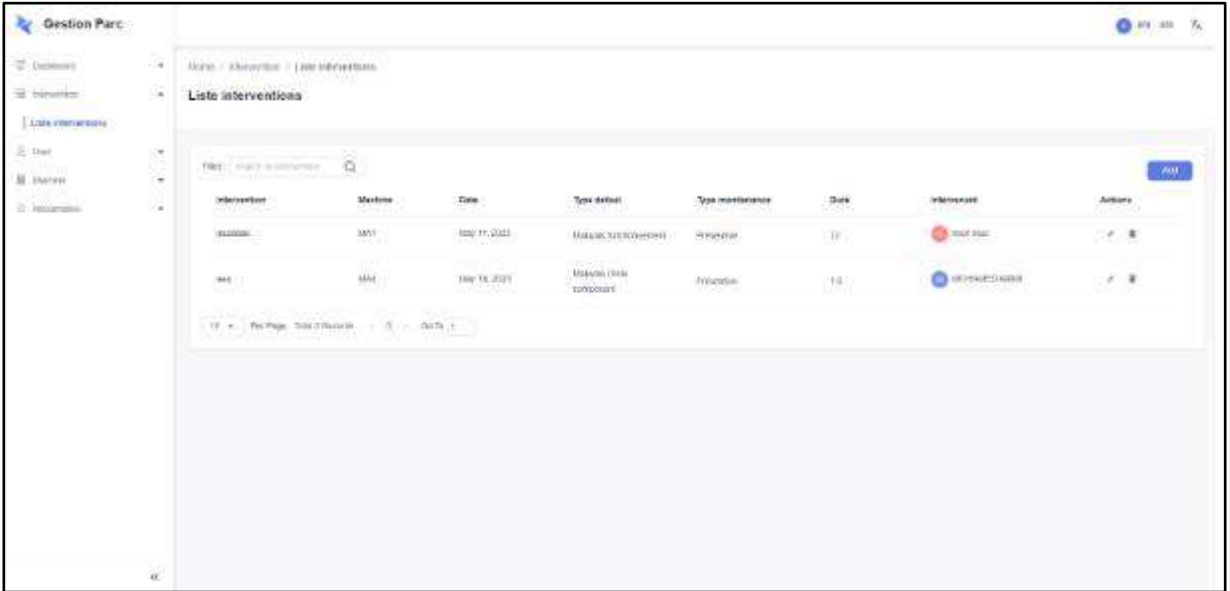


Figure 26 Page des interventions

La figure suivante représente comment je peux ajouter des nouvelles interventions.

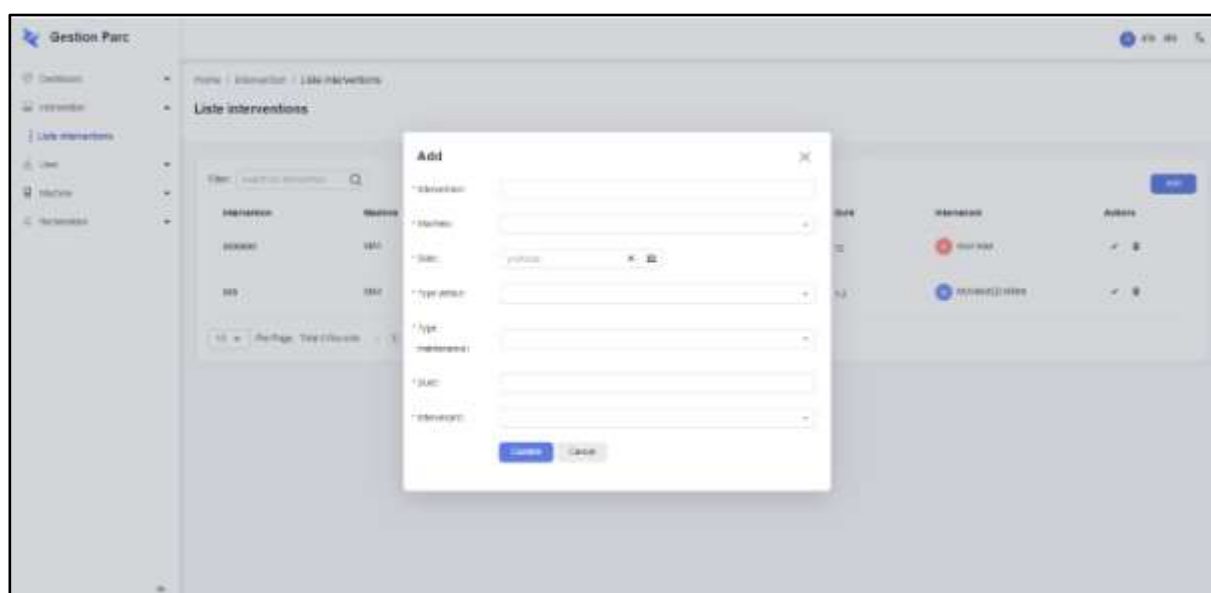


Figure 27 Page d'ajouter une intervention

La figure suivante représente comment je peux modifier une intervention.

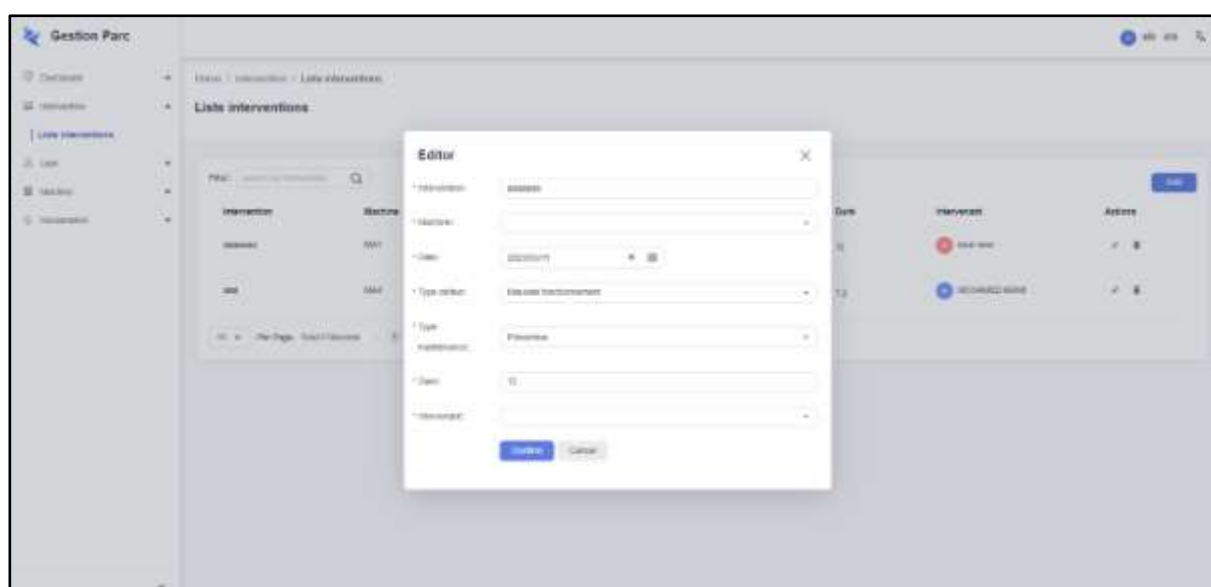


Figure 28 Page de modifier une intervention

Cette figure représente la confirmation de la modification de l’intervention.

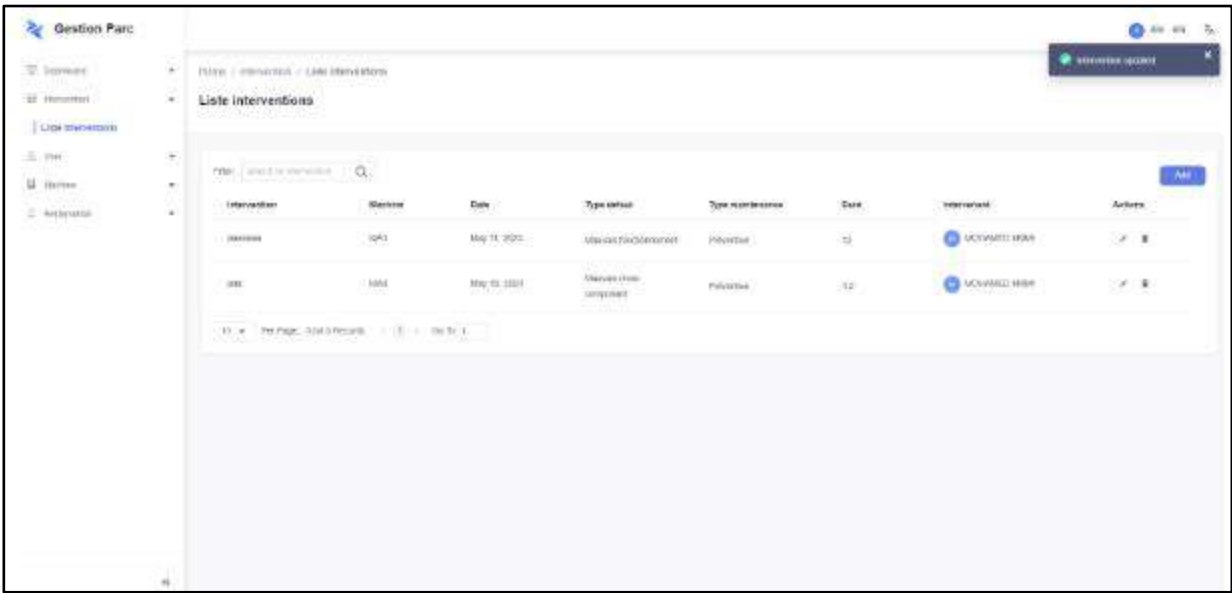


Figure 29 Notification de modifier

1.4 Interface gestion utilisateur

La figure suivante présente la liste des utilisateurs (technicien – responsable – admin).

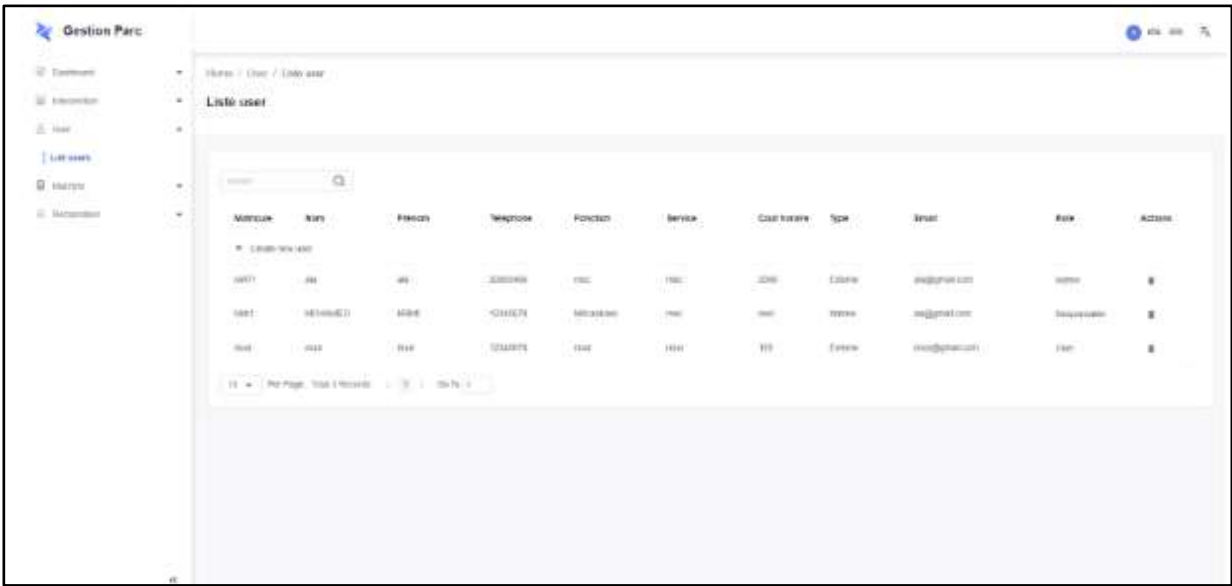


Figure 30 Liste utilisateur

La figure suivante représente l'interface de l'ajouter un nouvel utilisateur.

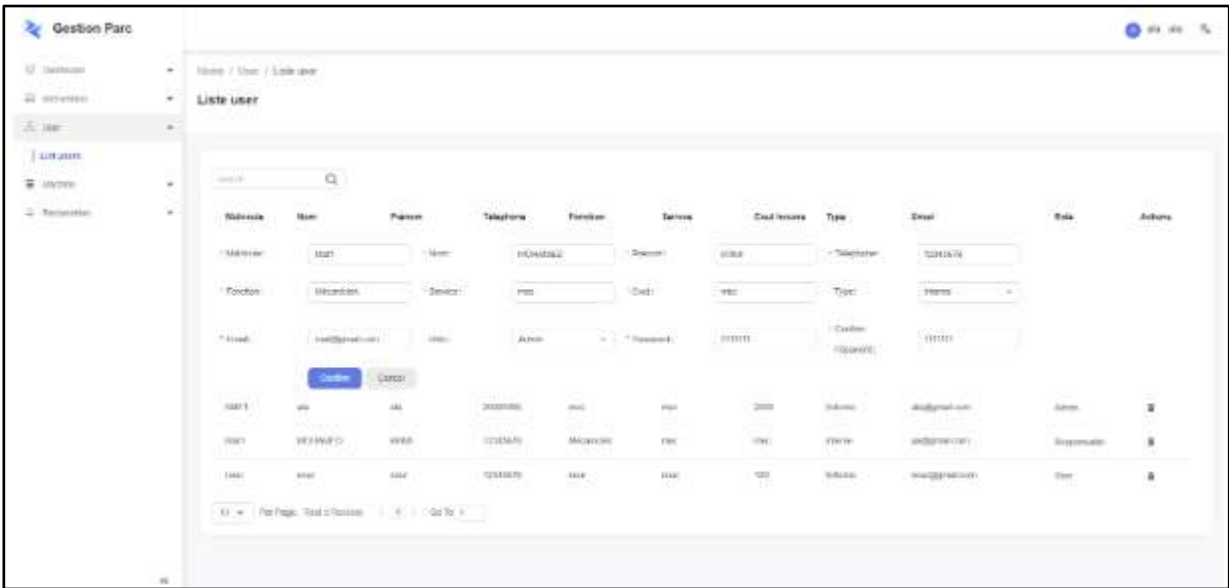


Figure 31 Ajouter utilisateur

La figure représente comment modifier un utilisateur.

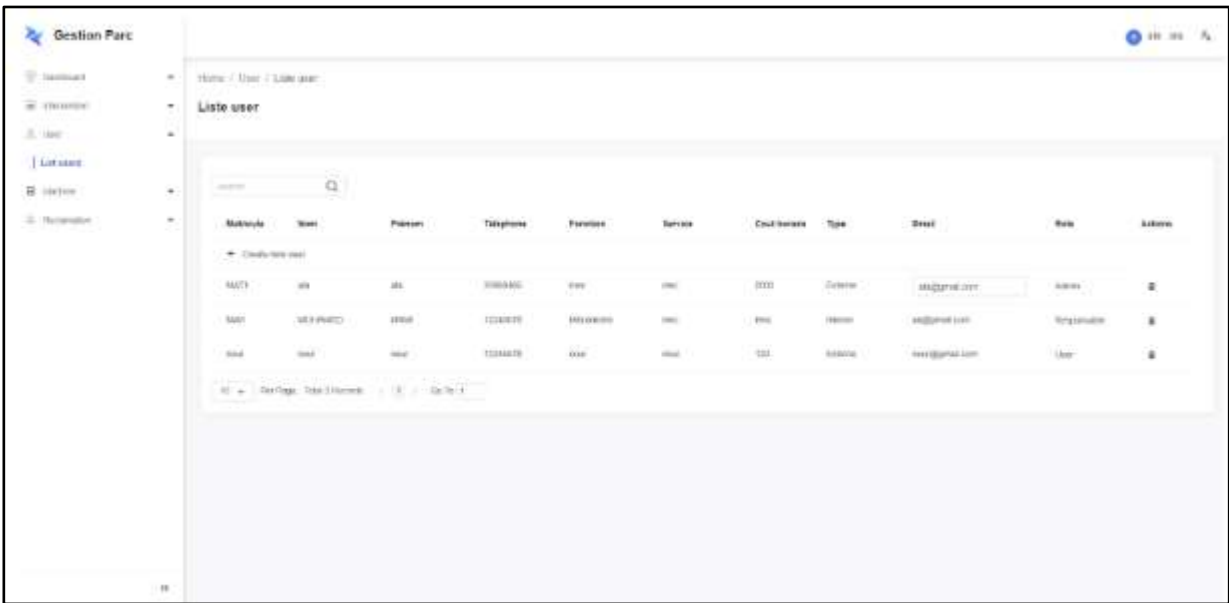


Figure 32 Modifier de modifier

1.5 Interface gestion machine

La figure suivante représente comment gérer la partie machine et Qr-Code.

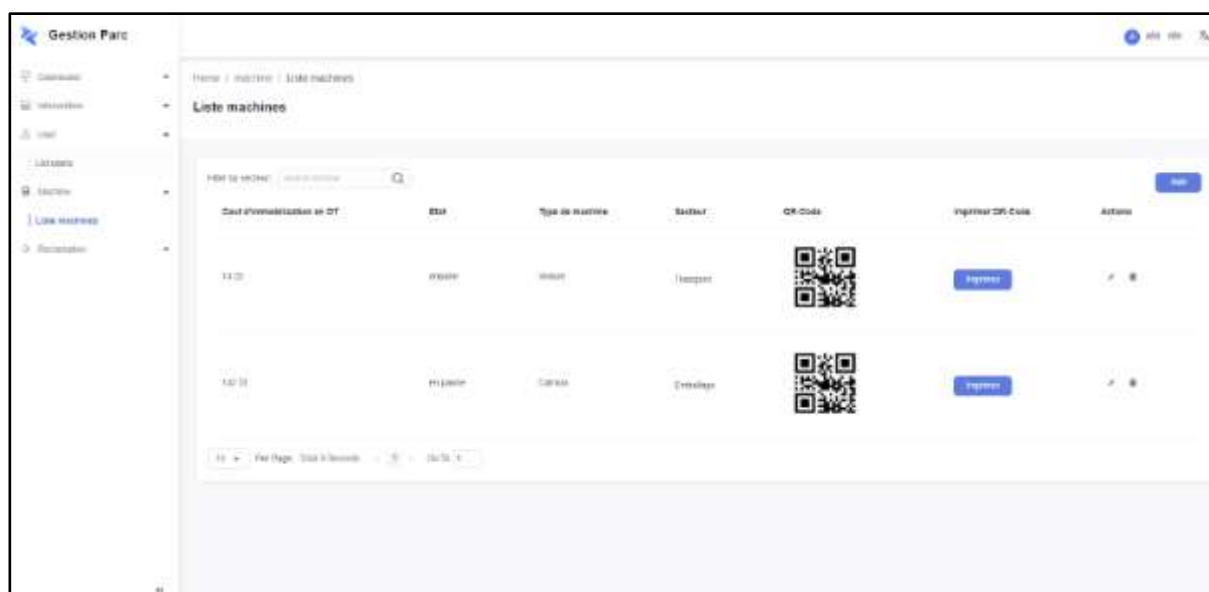


Figure 33 Liste machine

Cette partie représente comment je peux modifier une machine.

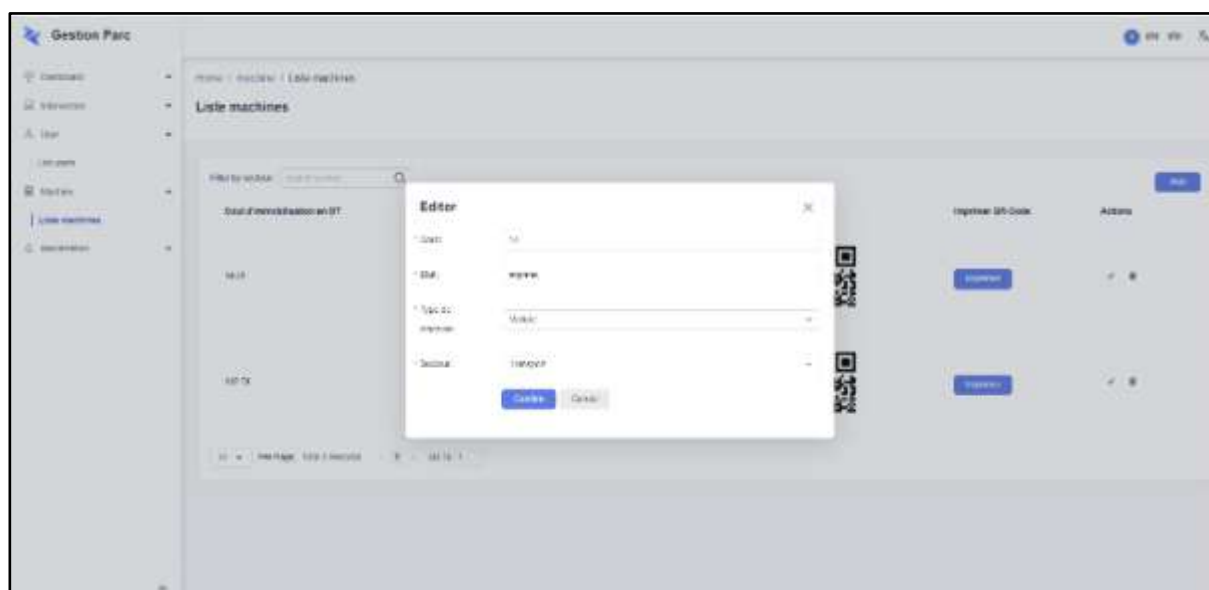


Figure 34 Modifier machine

Et cette partie représente la suppression d'une machine.

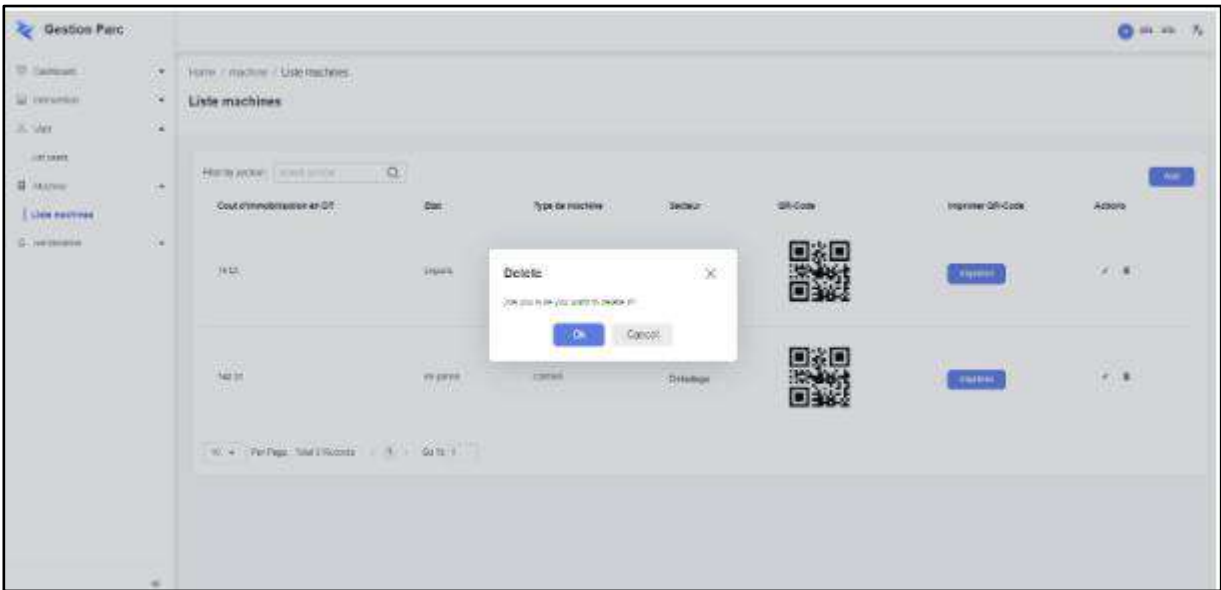


Figure 35 Supprimer machine

1.6 Interface réclamation

La figure représente la liste des réclamations

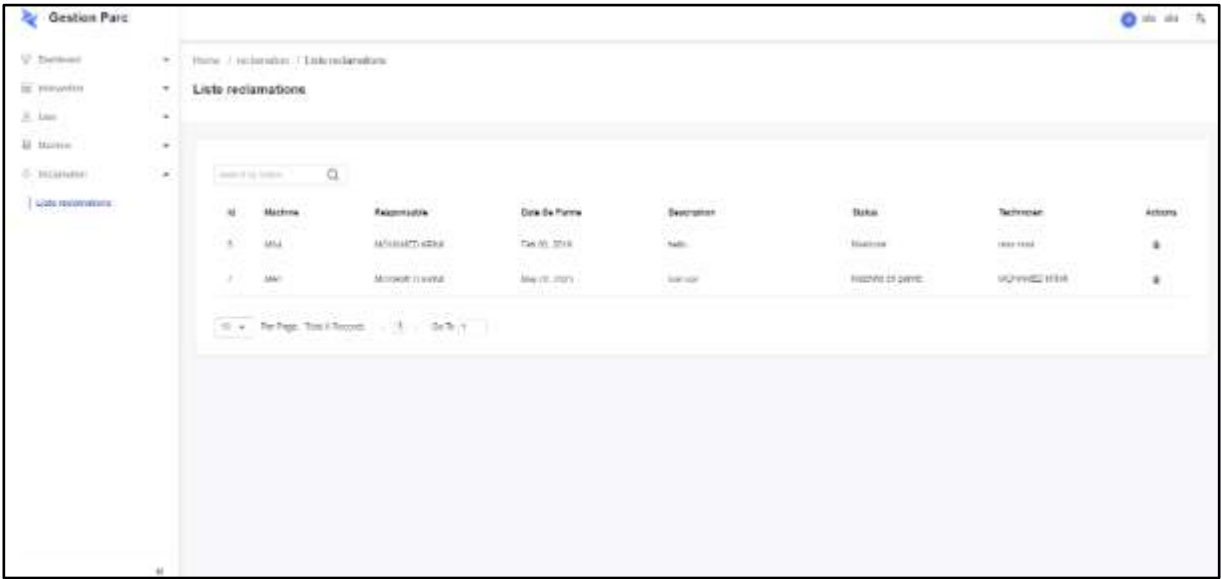


Figure 36 Liste Réclamation

Représente cette image comment je peux attribuer une réclamation (machine en pannes) au technicien.

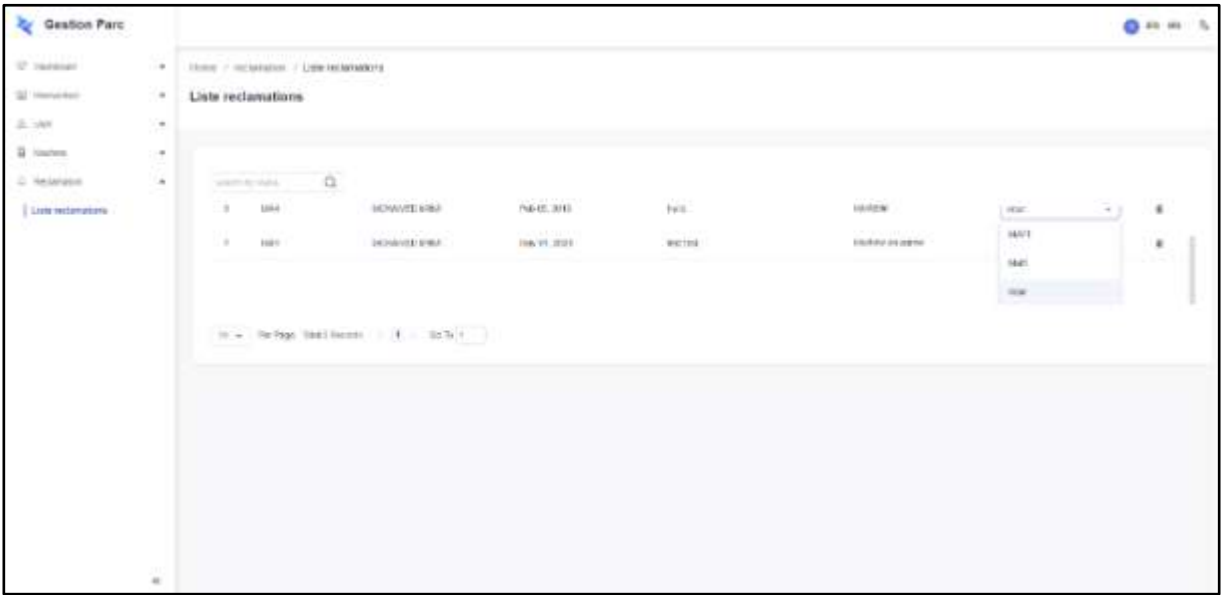


Figure 37 Attribuer réclamation

A l'aide de cette interface je peux modifier l'état de la réclamation étant qu'administrateur, responsable ou bien technicien.

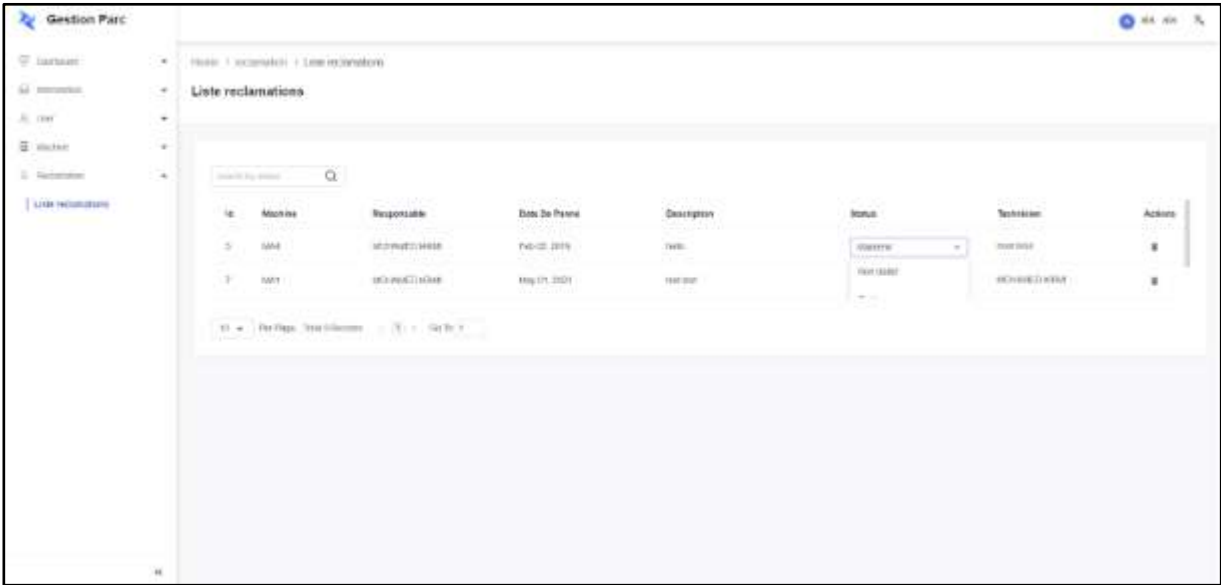


Figure 38 Modifier état

2.Interface réclamation

2.1 Partie responsable

C'est la partie du responsable qui peut voir à l'aide leur réclamation qui lui fait est suivre leur état.

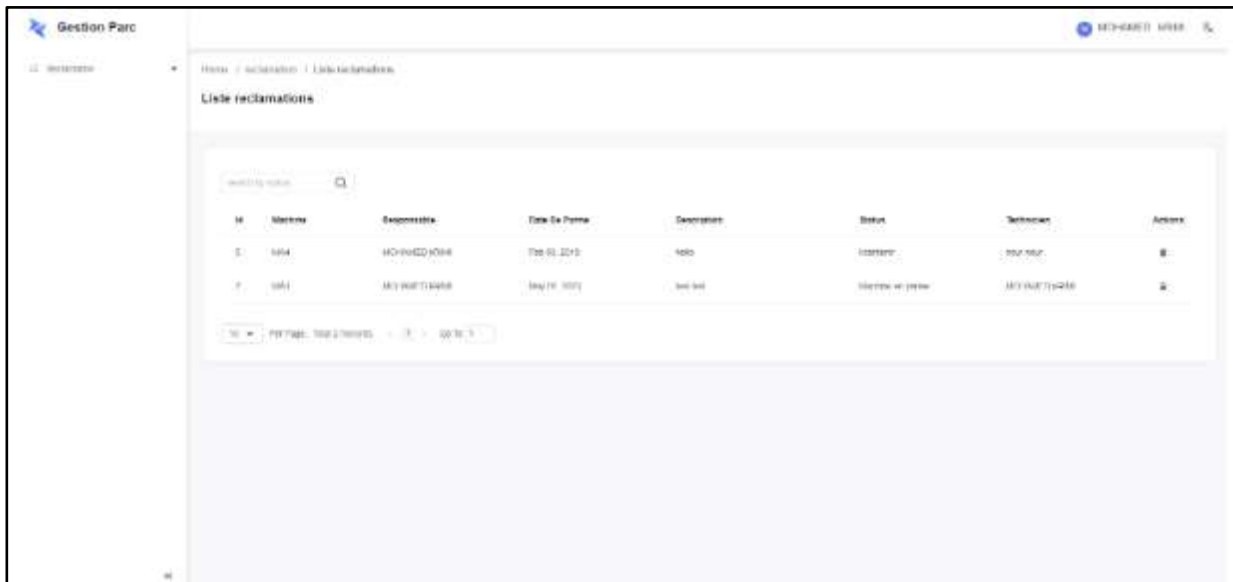


Figure 39 Liste réclamation par responsable

L'interface représente comment je peux passer une réclamation étant que responsable.

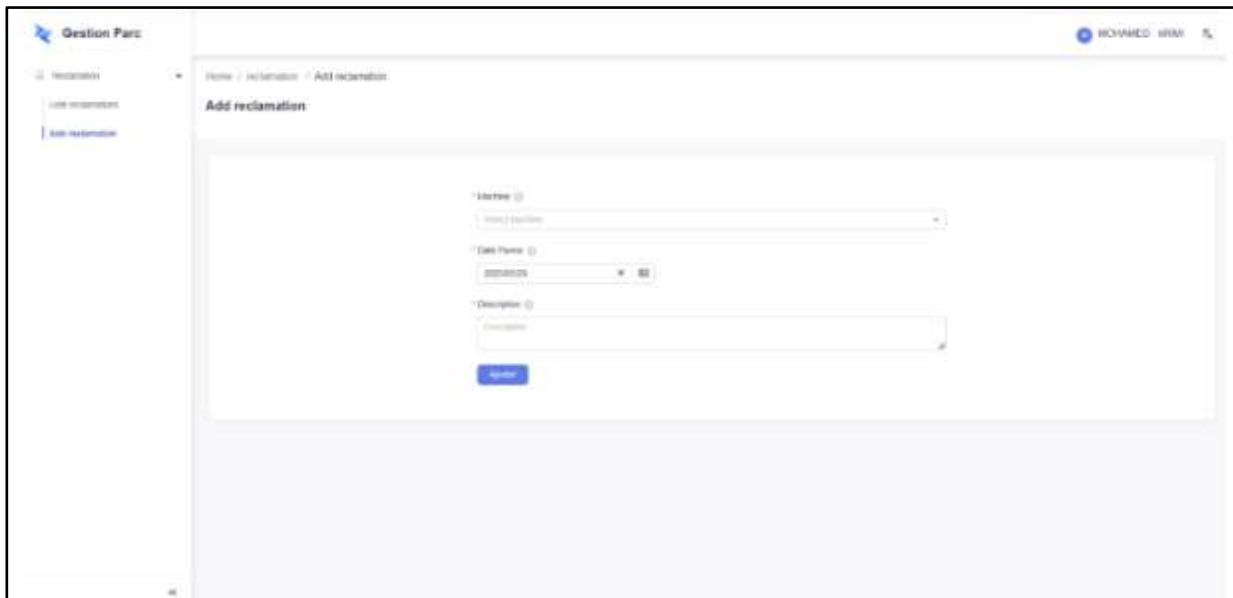


Figure 40 Passé réclamation

2.2 Partie technicien

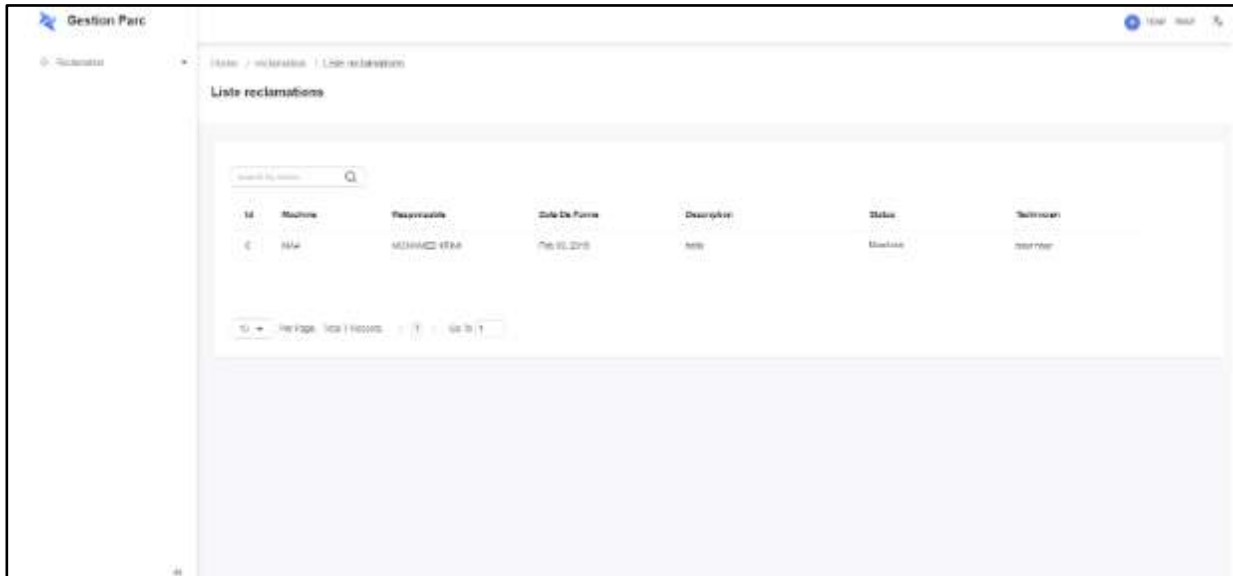


Figure 41 Liste réclamation attribué au technicien

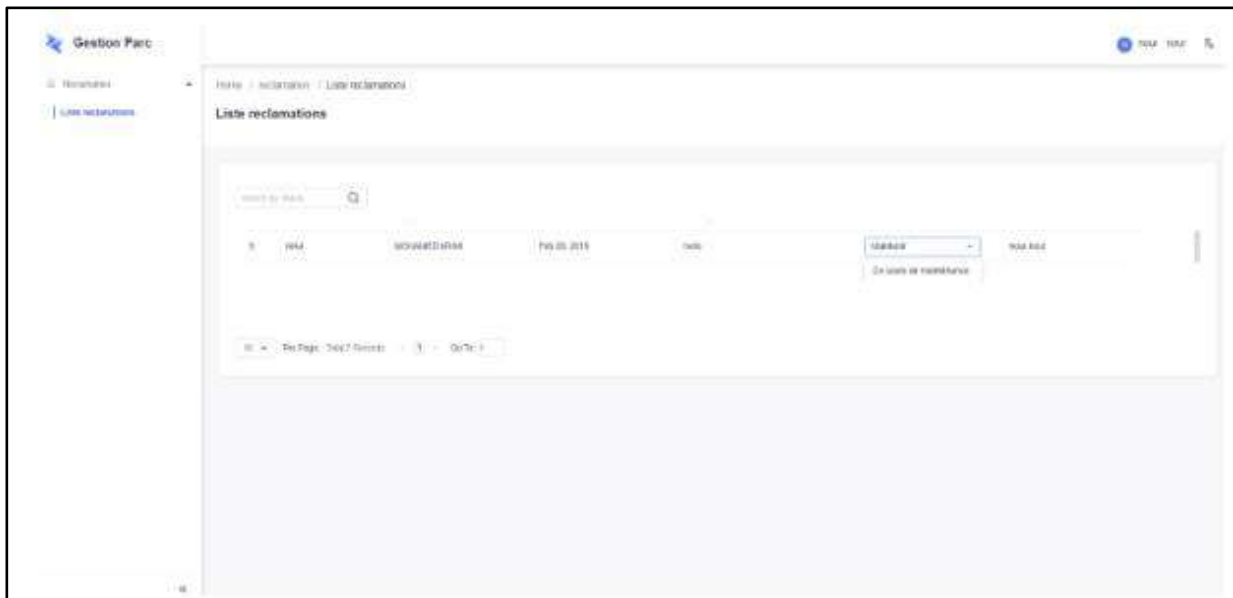


Figure 42 Changer l'état réclamation

Conclusion

Tout au long de ce chapitre, nous avons présenté l'environnement de travail logiciel avec lequel nous avons développé notre application web. Ensuite, nous avons présenté les différentes interfaces des applications développées.

Conclusion Générale

Ce document a été rédigé au terme de développer une application web de «**Gestion de parc**», qui devrait booster la gestion des ressources en pannes des entreprises.

La première phase de ce rapport était consacrée à la présentation de l'état de l'art, et nous avons présenté l'organisme d'accueil, ainsi que le projet proprement dit. Ces deux parties ont été suivies d'une analyse sur les applications existantes et leurs limites, ce qui nous a permis de poser la première pierre de l'édifice en proposant notre propre solution.

La deuxième phase quant à elle consistait à dégager les besoins fonctionnels, non fonctionnels de l'application, et par la même occasion les besoins techniques suivant la méthodologie de développement Scrum. Cela nous permettait par la suite de réaliser les diagrammes de cas d'utilisation qui nous serviraient dans la phase de conception.

La phase de conception nous a permis d'entrer plus en profondeur dans l'analyse et de parler de l'architecture de l'application.

Enfin dans la phase de réalisation nous avons présenté les outils et technologies utilisés pour réaliser ce travail. Puis nous nous sommes présenté les interfaces de notre système. Au final donc, il est important de souligner que ce projet a atteint les objectifs fixés au départ, et au-delà du sentiment de satisfaction qui s'en suit, il nous a permis de bénéficier de nouvelles connaissances venues compléter celles que nous avons acquises tout au long de notre formation.

Perspective

Comme tout autre travail, notre projet ne peut pas prétendre la perfection, il y a quelques améliorations que nous puissions les ajouter comme une application mobile contient un système de messagerie permet au responsable et les technicien de ce communiqué entre eux en cas d'une panne de machine, le technicien vas envoyer un message au responsable directement pour l'informer.

Ainsi cette application permet à l'administration et responsable de suivre les taches réalisées par le technicien pendant les journées et aussi nous n'avons pas assuré de compléter la partie modifier profil dans notre application.

Bibliographie

- [1] <https://www.lucidchart.com/pages/fr/langage-uml>
- [2] <https://www.piloter.org/projet/methode/scrum.htm>
- [3] <https://aws.amazon.com/fr/what-is/java/>
- [4] <https://developer.mozilla.org/fr/docs/Learn/JavaScript/>
- [5] <https://www.futura-sciences.com/tech/definitions/internet-css-4050/>
- [6] <https://developer.mozilla.org/fr/docs/Web/HTML>
- [7] <https://arcreane.com/moodle>
- [8] <https://fr.wikipedia.org/wiki/Angular>
- [9] <https://dept-info.labri.fr/ENSEIGNEMENT/programmation2/intro-eclipse/>
- [10] <https://online.visual-paradigm.com/fr/about-us/press-kit/>
- [11] <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-mysql/>
- [12] <https://fr.wikipedia.org/wiki/XAMPP>
- [13] <https://support.microsoft.com/>