

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



Université ABDERAHMANE MIRA de BÉJAIA

M É M O I R E

Pour l'obtention du diplôme de Master en Informatique

Spécialité : Génie logiciel

Sous thème

Conception et Réalisation d'une application web

de Gestion des pannes

Cas d'étude : SONATRACH

Réalisé par : **ADJED El mahdi** et **HADJI Nadia**

jury :

Président : Mr AMROUN Kamal

Examineur : Mr MEHAOUED Kamal

Encadrant : Mr OUZEGGANE Redouane

Année Universitaire :2021/2022

* *Remerciements* *

Nous désirons avant tout remercier le Dieu pour la chance et le courage qu'il nous a donné pour atteindre notre objectif . Ensuite, Nous tenons à exprimer toute notre reconnaissance à Monsieur *OUZEGGANE Redouane*, nous le remercions de nous avoir encadrés, orientés et conseillés. Nous voudrions aussi remercier le personnel de Sonatrach spécialement Monsieur *Louiba Moustapha* et *Faoudi Samir* ,pour leurs disponibilités ,leurs patiences et surtout leurs conseils, qui ont contribué à alimenter nos réflexions.

Au membres de jury ,nous présentons nos remerciements. Enfin ,nous tenons à témoigner toute notre gratitude à toute personne participée au succès de la réalisation de ce projet , pour leur confiance et leur soutien inestimable.

※ *Dédicaces* ※

A ma famille, elle qui m'a doté d'une éducation digne,
A la femme qui a souffert sans me laisser souffrir, A toi ma très chère mère Quoi que je fasse ou
que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta
bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour
affronter les différents obstacles. : mon adorable mère *Tassaadit*
A l'homme, mon précieux offre du dieu : mon cher père *Abdelmalek*, Que ce travail traduit ma
gratitude et mon affection.
A ma chère sœur *Rima* qui m'a soutenu et encouragé durant ces années d'études A mon
adorable frère *Md Akli*
A ma grande-mère, mes oncles et mes tantes. Que Dieu leur donne une longue et joyeuse vie.
A tous mes cousins et mes cousines spécialement *Loubna* .
A mes amis *Raouf Meghaoui, Mazigh, Adam*, et ma chère *Yassmin Tidadini* et a tous amis sans
exceptions, Merci pour vos encouragements.

HADJI Nadia

※ *Dédicaces* ※

Je tiens à dédier ce modeste travail à ceux qui me sont très chers D'abord à mes parents qui m'ont donné la vie et l'amour et qui m'ont soutenu tout au long de mon cursus étudiantin, je les remercie infiniment.

A mon cher frère : Boubekeur

A ma chère sœur : Katia

A toute ma famille paternelle et maternelle sans exception.

Sans oublier mes amis auxquels j'exprime une grande reconnaissance : Kakou, Yasmine, Ryma, Nadia, Wardia, Hamida, Hani, Islam, Massi.

A tous ceux qui m'aiment et me connaissent de près ou de loin à la réalisation de ce travail.

ADJED El mahdi

TABLE DES MATIÈRES

Table des Matières	i
Liste des tableaux	v
Liste des figures	vi
Liste des acronymes	viii
Introduction générale	1
1 présentation de l'organisme d'accueil et le recueil des besoins	3
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	3
1.2.1 Historique	3
1.2.2 Principales activités de la société	4
1.2.3 Présentation de la Région de Transport Centre (RTC)	4
1.2.4 Organigramme de la RTC	5
1.2.5 Rôle de Sonatrach	6
1.2.6 Présentation de la structure concernée par l'étude (Service STS)	6
1.2.7 Les équipements de SONATRACH	6
1.3 Etude de l'existant	7
1.4 Problématique et Objectifs	7
1.4.1 Problématique	7
1.4.2 L'objectif	7
1.5 Les Besoins Fonctionnels et Non Fonctionnels	8
1.5.1 Les Besoins Fonctionnels	8
1.5.2 Les Besoins non Fonctionnels	8
1.6 Langage et Processus de développement	9
1.7 Conclusion	9
2 Analyse des besoins	10
2.1 Introduction	10

2.2	Identification des acteurs	10
2.3	Diagramme de contexte	10
2.4	Identification des messages	11
2.5	La relation entre les acteurs	12
2.6	Identification des cas d'utilisation	13
2.7	Diagramme de cas d'utilisation	14
2.7.1	Diagramme de cas d'utilisation – Acteur : «Utilisateur »	15
2.7.2	Diagramme de cas d'utilisation – Acteur : « Opérateur »	15
2.7.3	Diagramme de cas d'utilisation – Acteur : « Ingénieur »	15
2.7.4	Diagramme de cas d'utilisation – acteur « Administrateur »	16
2.8	Description Textuelle des cas d'utilisation	17
2.8.1	Description de Cas d'utilisation « S'authentifier »	17
2.8.2	Description de cas d'utilisation « Déclarer une panne »	18
2.8.3	Description de cas d'utilisation « Ajouter un utilisateur »	19
2.9	Diagramme de séquence système	19
2.9.1	Diagramme de séquence système cas d'utilisation « s'authentifier »	20
2.9.2	Diagramme de séquence système cas d'utilisation « Ajouter un utilisateur »	21
2.9.3	Diagramme de séquence système cas d'utilisation « Déclarer une panne »	22
2.10	Conclusion	22
3	Conception et Schéma Relationnel de données	23
3.1	Introduction	23
3.2	Diagrammes d'interactions	23
3.2.1	Les Diagrammes d'interaction des cas d'utilisation	24
	Diagramme d'interaction de cas d'utilisation «S'authentifier »	24
	Diagramme d'interaction de cas d'utilisation «Ajouter un utilisateur»	25
	Diagramme d'interaction de cas d'utilisation «Déclarer une panne»	26
3.3	Diagramme de classe de domaine	27
3.3.1	Description détaillée des attributs de classes	27
3.3.2	Description détaillée des attributs des classes d'associations	29
3.4	Schéma relationnel	29
3.4.1	Règles de passage au modèle relationnel	30
3.4.2	Le passage au modèle relationnel	30
3.5	Conclusion	30
4	La Réalisations	32
4.1	Introduction	32
4.2	Langage et environnement de développement	32
4.2.1	Les langages utilisés	32
4.2.2	Outils et bibliothèque	33
4.3	Modèle MVC	33
4.4	Structure du code source	34
4.5	Schéma physique de la base de données	35
4.6	Diagramme de Déploiement	35

4.7	Captures d'écrans	36
4.7.1	L'interface Homme-Machine « Authentification »	36
4.7.2	L'interface Homme-Machine « Accueil-Ingénieure »	37
4.7.3	L'interface Homme-Machine « Accueil-opérateur »	37
4.7.4	L'interface Homme-Machine « Gestion des Sites »	37
4.7.5	L'interface Homme-Machine « Gestion des Equipements »	38
4.7.6	L'interface Homme-Machine « Gestion des utilisateurs »	38
4.7.7	L'interface Homme-Machine « Gestion des pannes »	39
4.7.8	L'interface Homme-Machine « Gestion des Réparations »	39
4.8	Un scénario d'exécution	40
4.9	Conclusion	41
	Conclusion générale	42
	Bibliographie	44
	A La Méthode UPXp	45
A.1	Description	45
A.1.1	Présentation du processus unifié (UP)	45
	. Le processus unifié : un cadre général	45
	Le processus unifié est piloté par les cas d'utilisation	45
	Le processus unifié est centré sur l'architecture	46
	Le processus unifié est itératif et incrémental	46
A.1.2	Le cycle de vie du processus unifié	46
A.2	Présentation de l'eXtreme Programming (XP)	47
A.3	La démarche simplifiée	48
	B Description des cas d'utilisation et les diagrammes de séquence	49
B.1	Description textuelle des cas d'utilisation	49
B.1.1	Description de cas d'utilisation « Consulter le rapport mensuel »	49
B.1.2	Description de cas d'utilisation « Lister les équipements »	50
B.1.3	Description de cas d'utilisation « Modifier un équipement »	52
B.1.4	Description de cas d'utilisation «Ajouter un équipement »	52
B.1.5	Description de cas d'utilisation « Supprimer un équipement »	53
B.1.6	Description de cas d'utilisation « Modifier une panne déjà déclarée »	54
B.2	Diagrammes séquences systèmes	55
B.2.1	Diagramme de séquence système du cas d'utilisation «Consulter le rapport mensuel»	55
B.2.2	Diagramme de séquence système du cas d'utilisation «Réparer une panne»	56
B.2.3	Diagramme de séquence système du cas d'utilisation «Lister les équipements»	57
B.2.4	Diagramme de séquence système du cas d'utilisation «modifier un équipement»	58
B.2.5	Diagramme de séquence système du cas d'utilisation «Ajouter un équipement»	59

B.2.6	Diagramme de séquence système du cas d'utilisation «supprimer un équipement»	60
B.2.7	Diagramme de séquence système du cas d'utilisation «modifier une panne déjà déclarée»	61
C	Les diagrammes d'interactions	63
C.1	DIAGRAMMES D'INTERACTION	63
C.1.1	Diagramme d'interaction de cas d'utilisation «Ajouter un équipement» .	63
C.1.2	Diagramme d'interaction de cas d'utilisation «Modifier un équipement» .	64
C.1.3	Diagramme d'interaction de cas d'utilisation «supprimer un équipement»	65
C.1.4	Diagramme d'interaction de cas d'utilisation «Réparer une panne»	66
C.1.5	Diagramme d'interaction de cas d'utilisation «Lister un équipement» . .	67

LISTE DES TABLEAUX

2.1	Identification des messages	12
2.2	identification des cas d'utilisation	13
2.3	Description du cas «Authentification »	17
2.4	Description du cas «Déclarer une panne »	18
2.5	Description du cas « Ajouter un utilisateur»	19
3.1	Description des classes d'objets et leurs attributs.	28
3.2	Description des classes d'association et leurs attributs.	29
B.1	Description de cas d'utilisation « Consulter le rapport mensuel	49
B.2	Description du cas «Réparer une panne»	50
B.3	Description du cas «lister les équipements»	51
B.4	Description de cas d'utilisation « Modifier un équipement »	52
B.5	Description de cas d'utilisation «Ajouter un équipement»	53
B.6	Description du cas «Supprimer un équipement»	54
B.7	Description du cas «Modifier une panne déjà déclarée»	55

TABLE DES FIGURES

1.1	Activité de SONATRACH	4
1.2	Organigramme de SONATRACH	5
2.1	Diagramme de Contexte.	11
2.2	La relation entre les acteurs	13
2.3	Diagramme de cas d'utilisation	14
2.4	Diagramme de cas d'utilisation de l'utilisateur	15
2.5	Diagramme de cas d'utilistion de l'opérateur	15
2.6	Diagramme de cas d'utilistion de l'ingénieur	16
2.7	Diagramme de cas d'utilistion de l'administrateur	16
2.8	Diagramme de séquence système du cas d'utilisation « S'authentifier »	20
2.9	Diagramme de séquence système du cas d'utilisation «Ajouter un utilisateur»	21
2.10	Diagramme de séquence système du cas d'utilisation « Déclarer une panne »	22
3.1	Diagramme d'interaction de cas d'utilisation «S'authentifier»	25
3.2	Diagramme d'interaction de cas d'utilisation «Ajouter un utilisateur».	26
3.3	Diagramme d'interaction de cas d'utilisation «Déclarer une panne»	27
3.4	Diagramme de classe de domaine.	29
4.1	Schéma physique de la base de données	35
4.2	Diagramme de déploiement	36
4.3	Interface Homme-Machine de L'authentification	36
4.4	Interface Homme-Machine de Accueil-Ingénieure	37
4.5	Interface Homme-Machine de Accueil-opérateur	37
4.6	L'interface Homme-Machine « Gestion des sites ».	38
4.7	L'interface Homme-Machine « Gestion des Equipements ».	38
4.8	L'interface Homme-Machine « Gestion des utilisateurs ».	39
4.9	L'interface Homme-Machine « Panne ».	39
4.10	L'interface Homme-Machine « Gestion des Réparations »	40
A.1	Démarche simplifiée	48

B.1	Diagramme de séquence système du cas d'utilisation «consulter le rapport mensuel».	56
B.2	Diagramme de séquence système du cas d'utilisation «réparer une panne».	57
B.3	Diagramme de séquence système du cas d'utilisation «Lister les équipements». . .	58
B.4	Diagramme de séquence système du cas d'utilisation «modifier un équipement». . .	59
B.5	Diagramme de séquence système du cas d'utilisation «Ajouter un équipement». . .	60
B.6	Diagramme de séquence système du cas d'utilisation «Supprimer un équipement».	61
B.7	Diagramme de séquence système du cas d'utilisation «Diagramme de séquence système du cas d'utilisation «modifier une panne déjà déclarée».	62
C.1	Diagramme d'interaction de cas d'utilisation «Ajouter un équipement»	63
C.2	Diagramme d'interaction de cas d'utilisation «Modifier un équipement»	64
C.3	Diagramme d'interaction de cas d'utilisation «supprimer un équipement»	65
C.4	Diagramme d'interaction de cas d'utilisation «Réparer une panne»	66
C.5	Diagramme d'interaction de cas d'utilisation «Lister un équipement»	67

LISTE DES ACRONYMES

A	API	Application Programming Interface .
B	BDD	base de donnée
C	CSS	Cascading Style Sheet
D	DRGB	Direction Régionales de Béjaia .
E	ENAC ENIP	Entreprise Nationale de Canalisations Entreprise Nationale de Pétrochimie.
G	GNL GPL	Gaz Naturel Iquéfiés. Gaz de Pétrole Liquéfiés .
H	HTTP HTML	Hypertext Transfer Protocol. hyper Text markup language.
I	IHM	Interface Homme Machine
J	JS	JavaScript.
M	MVC MySQL	Modele Vue Controleur My Structured Query Language
N	N-TIC	Nouvelles Technologies de l'Information et de la Communication.
R	RTC	Région de transport Centre Béjaia

S	SGBD	Système de Gestion de Base de Donnée.
	SI	Système d'Information .
	SIE	Surté Interne Etablissement.
	SONATRACH	Société nationale pour la recherche, la production, le transport, la transformation, et la commercialisation des hydrocarbures
	SPA	Single page application.
	STH	Société de gestion des terminaux à hydrocarbures.
	STS	Système Technologique de Sécurité.
U	UML	Unified Modeling Lnaguage .
W	WAMP	Windows Apache MySQL PHP.

INTRODUCTION GÉNÉRALE

Nul ne peut négliger la présence et l'importance de l'informatique dans notre vie quotidienne, elle est partout, dans tous les domaines et dans tous les organismes et institutions. Il est clair que le système d'information joue un rôle primordial au sein de l'entreprise, ce qui donne une place privilégiée à l'informatique. La mise en œuvre de cette technologie moderne de traitement de l'information s'avère nécessaire, pour assurer une gestion plus fiable, plus rigoureuse, moins fastidieuse, ainsi pour minimiser les risques d'erreurs et d'assurer la disponibilité de l'information à toute éventuelle demande.

Pour tirer profit des nouvelles technologies d'information et de télécommunication (N-TIC), les entreprises sont tenues à informatiser, d'une façon complète et partielle leur système d'information. Après cet phase d'informatisation, vient l'étape de maintenance de ce S.I. informatisé afin de résoudre les différents problèmes non-traité initialement, de corriger des erreurs et des bugs des applicatifs et des progiciels de système d'information et, enfin d'étendre le champs et le rayon d'informatisation du système d'information de l'entreprise pour faire face aux différents changements internes et externes à l'entreprise et faire face aux différents enjeux stratégiques et aux évolution techniques.

Dans ce contexte d'amélioration continue des solutions informatiques aux seins des organismes, nous avons effectué un stage au niveau du service de la sûreté interne de SONATRACH, où nous avons constaté un ensemble de problèmes dont le traitement manuel des données, ce qui entraîne des risques d'erreur, l'absence des statistiques, due essentiellement à l'inexistence d'une base données mise à jour en temps réel, une perte de temps et d'informations qui empêche le bon fonctionnement du service. Pour remédier à ces lacunes, l'entreprise SONATRACH souhaite développer une application web permettant de gérer les équipements informatiques, et qui doit répondre à un ensemble des besoins notamment : la gestion des équipements (l'ajout des équipements - modification - suppression ...), et leurs maintenances, la gestion des utilisateurs et la gestion des sites. Et comme a été mentionné ci-dessus, dans le cadre de notre projet de fin de cycle master en génie logiciel, nous avons effectué un stage, d'une durée de 2 mois, au niveau du département S.I.E. (Sûreté Interne de l'Entreprise). Ceci nous a permis de comprendre l'environnement et l'organisation de travail pour la gestion de maintenance de matériels informatique (Ordinateurs, Imprimantes, les caméras, Projecteurs . . .) du service STS

(Service Technologies Système). Il nous a été demandé de concevoir et réaliser une solution informatique, sous forme d'une application web, pour la gestion des pannes qui se trouvent au niveau de ce service.

Pour bien présenter notre travail, nous avons structuré ce mémoire de la façon suivante : Après l'introduction générale, le premier chapitre s'articule sur la présentation générale de l'organisme d'accueil, ainsi que la description du processus de gestion des pannes informatiques et une liste des problèmes liés à cette gestion, et enfin la description de la solution proposée ainsi que le recueil des exigences fonctionnelles et non-fonctionnelles que cette dernière doit répondre. Le chapitre suivant sera consacré à la phase d'analyse des besoins, qui commence par la définition des acteurs de l'application, le contexte de l'application et les messages échangés avec ses acteurs, les différents cas d'utilisation par acteurs. Ces cas d'utilisation permettent de modéliser l'aspect fonctionnel de l'application. Par la suite, nous avons établi une description textuelle de chaque cas d'utilisation et aussi le diagramme de séquence correspondant.

Le troisième chapitre est dédié à la conception du système, avec l'élaboration des diagrammes d'interaction et le diagramme de classes du domaine. Ce dernier sera transformé au schéma relationnel, en appliquant les règles des passages. Le chapitre suivant, à savoir le chapitre de Réalisation, sera consacré à la présentation des outils et technologies utilisés pour concrétiser toutes les idées captées dans les deux chapitres précédents, et seront présentés aussi, les diagrammes de déploiement, le schéma physique de données ainsi que l'estimation de volume de données nécessaire, quelques captures d'écran de notre application avec des scénarios de tests et validation de chaque cas d'utilisation.

Enfin, ce mémoire sera terminé par une conclusion générale avec des propositions d'amélioration que nous n'avons pas pu concrétiser.

CHAPITRE 1

PRÉSENTATION DE L'ORGANISME D'ACCUEIL ET LE RECUEIL DES BESOINS

1.1 Introduction

Dans tout projet d'informatisation d'un organisme ou de l'un de ses services, l'étude de l'existant est une étape très importante pour bien comprendre et cerner le besoin. Ainsi, dans ce premier chapitre, nous présenterons l'organisme d'accueil, son historique, ses différentes activités, et nous nous focalisons sur le service technologies systèmes, nous décrirons la problématique existante dans ce service ainsi que la solution informatique à proposer.

1.2 Présentation de l'organisme d'accueil

SONATRACH est parmi les grandes entreprises existantes dans l'Algérie, en vue de son lourd poids sur l'économie du pays, Qui est un point intermédiaire assurant le contrôle d'acheminement des hydrocarbures de Haoud El Hamra, Situé à 25 km de Hassi Messaoud vers le port pétrolier de Bejaia pour la commercialisation.

1.2.1 Historique

SONATRACH est une société nationale de transport et de commercialisation des hydrocarbures créée le 31 décembre 1963, pour être un instrument d'intervention de l'état dans le secteur pétrolier aux côtés des compagnies françaises. Au début des années 80, la restructuration de SONATRACH se met en œuvre, cette étape figure parmi les plus importantes dans le développement de la société. Elle devient une société nationale pour la recherche, la production, le transport, la transformation et la commercialisation des Hydrocarbures. Le réseau de transport qui appartient à SONATRACH s'étend sur tout le territoire national. Avec 28 canalisations, ces pipelines relient le Sud au Nord et acheminent les d'hydrocarbures non raffinés. Aujourd'hui, SONATRACH est la première entreprise du continent africain. Elle est classée 12ème parmi les compagnies pétrolières mondiales, 2ème exportateur de GNL et de GPL et 3ème exportateur de GAZ. Dans le

but d'assurer une meilleure efficacité organisationnelle et économique, SONATRACH fut divisée en 18 entreprises nationales, parmi lesquelles on cite : NAFTAL, ENIP, ENAC, NAFTEC, Les activités de base de la SONATRACH ont été fixées en 1992, afin d'atteindre ses objectifs nationaux et consolider ces positions internationales.

1.2.2 Principales activités de la société

Les quatres activités essentielles de La SONATRACH sont schématisés comme suit :

1. **L'Activité Amont** : Recouvre les métiers de recherche, d'exploration, de développement et de production des hydrocarbures .
2. **L'Activité Transport** : Assure l'acheminement des différents hydrocarbures par canalisation ;

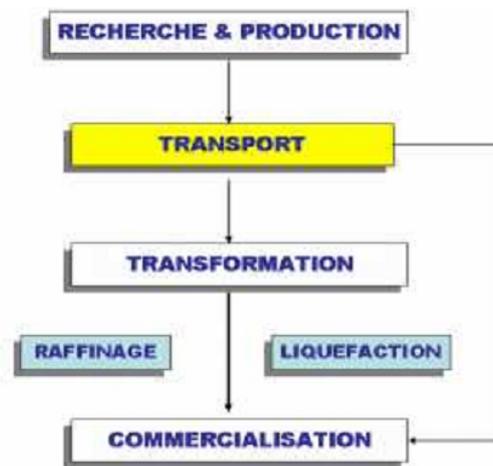


FIGURE 1.1 – Activité de SONATRACH

3. **L'Activité Aval** : Prend en charge le développement et l'exploitation des complexes hydrocarbures (liquéfaction, raffinerie,...).
4. **L'Activité Commercialisation** : A pour mission l'élaboration et l'application de la stratégie commerciales SONATRACH sur le marché national et international.

1.2.3 Présentation de la Région de Transport Centre (RTC)

RTC représente la direction régionale de Bejaïa, qui est une direction parmi les 7 directions opérationnelles composant l'activité de transport par canalisation avec les régions d'ARZEW, SKIKDA, HAOUD ELHAMRA, GPDF, et GME ces deux derniers sont 2 gazoducs traversant l'Italie et l'Espagne. La vocation de la DRGB est le transport des hydrocarbures par canalisation, stockage et chargement des navires. Le terminal marin de Bejaïa est divisé en deux parties sont :

- Terminal Nord
- Terminal Sud

1.2.4 Organigramme de la RTC

C'est l'une des sept directions opérationnelles qui comporte l'activité de Transport par Canalisations des hydrocarbures. La RTC est chargée du transport par canalisations, stockage et livraison des hydrocarbures par l'exploitation d'un port pétrolier et d'un poste de chargement en haute mer, mis à la disposition de la Société de gestion des Terminaux à Hydrocarbures (STH) depuis 2005. La région transport centre de Bejaia est composée de trois sous-directions divisées chacune en plusieurs départements. L'organigramme suivant donne la structuration hiérarchique fonctionnelle de la région transport centre de Bejaia. La figure ci-dessous représente l'Organigramme de la RTC-Bejaia.

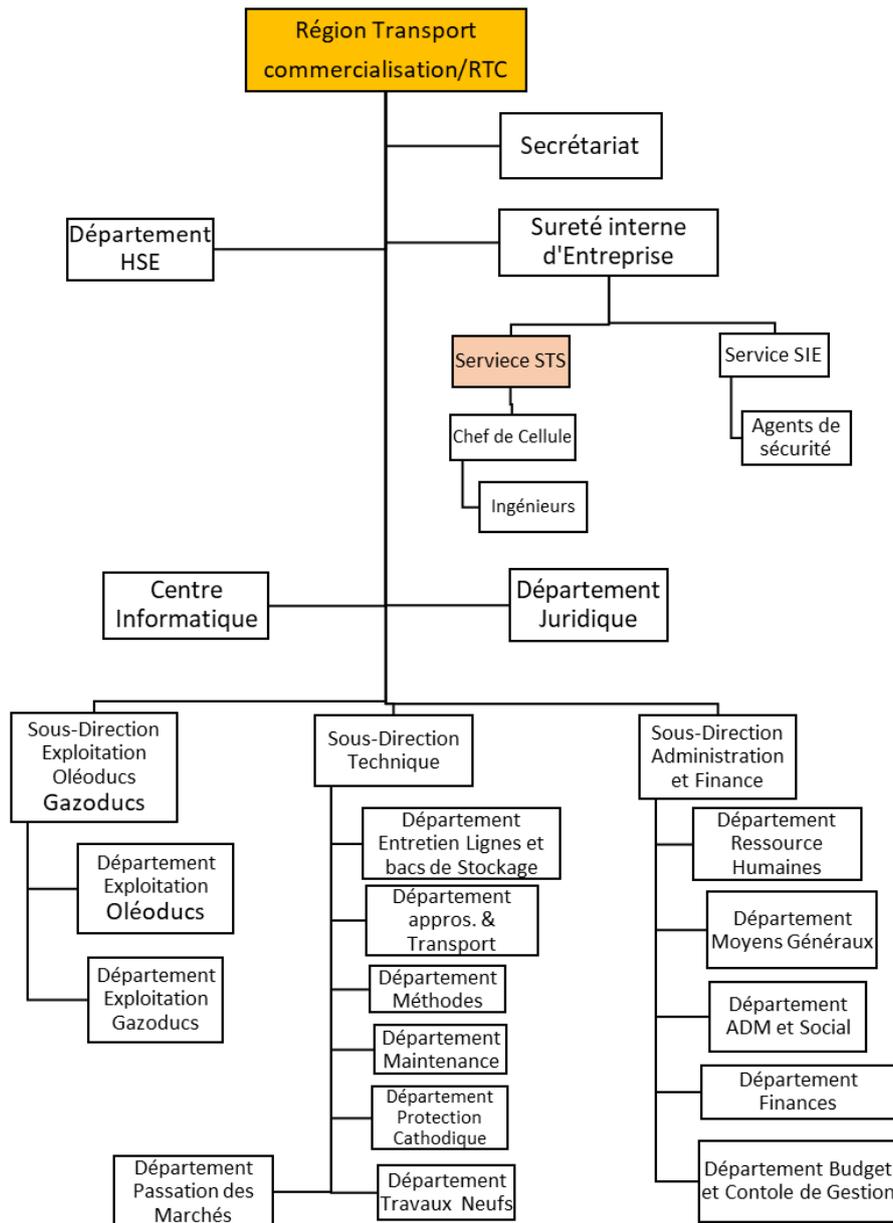


FIGURE 1.2 – Organigramme de SONATRACH

1.2.5 Rôle de Sonatrach

SONATRACH est le premier groupe d'hydrocarbures en Afrique ,elle joue un rôle important dans divers secteurs :

- Prospection, exploration et exploitation.
- Développement, gestion et management de transport, stockage et moyens de chargement « Transformation et raffinage ».
- Marketing.
- Liquéfaction du gaz naturel, traitement et valorisation des hydrocarbures gazeux.
- Mise en place de toutes formes d'activités en joint-venture à l'intérieur et à l'extérieur du territoire algérien, en collaboration avec des compagnies étrangères.
- Une fourniture constante d'hydrocarbures à usage domestique.
- La recherche pour la promotion et la valorisation de toute forme d'énergie et sourc.
- Le développement de toute activité étant directement ou indirectement liée avec l'industrie des hydrocarbures.

1.2.6 Présentation de la structure concernée par l'étude (Service STS)

Le département SIE (Sûreté Interne Etablissement) se divise en deux services, service SIE et service STS (Service Technologie Système).

Service SIE :Il comprend les agents de sécurité.

Service (STS) : Service Technologique Système, a pour but de gérer les pannes des systèmes télésurveillances, il contient la Central, le contrôle d'accès, la télésurveillance, et l'anti-intrusion

- Central : gère tous les équipements matériels et logicielles .
- Contrôle d'accès : gère le contrôle d'accès à l'entreprise, on trouve les tripodes, tourniquets, porte pour accès à badges.
- Télésurveillance : on trouve la différente caméra Dôme, caméra fixes, caméra mobile (tourrelle, caméra atex).
- Anti Intrusion : fil micro point.

1.2.7 Les équipements de SONATRACH

On va citer les différents équipements matérielles et logicielles.

1. **Les équipements matériels** : L'ensemble des équipements matériels sont :

Les équipements du système central

- Serveur Informatique
- Unité Centrale Informatique
- Switch et Routeur
- Encodeur et Enregistreur Vidéo

Les équipements du système télésurveillance :

- Caméra Fixe .
- Caméra Mobile (simple et ATEX) .
- Caméra DOME .

Les équipements du système contrôl d'accès :

- Tripode .
- Tourniquet .
- Porte à accès à badge

2. Les équipements logiciels : L'ensemble des équipements logiciels sont :

- L'application ISIS qui est dans « Central », elle sert à Gérer les équipements matériels.
- La base de données SQL Serveur 2000
- Windows Server (L'Active Directory).

1.3 Etude de l'existant

Durant la période de notre stage au niveau de SONATRACH de Béjaia, dans le service STS nous nous sommes basés sur la procédure de la gestion des pannes des équipements. Cette procédure consiste à :

- Lors d'une panne d'un équipement, l'utilisateur va déclarer la panne soit avec un appel téléphonique soit avec un rapport bien rédigé.
- Un ingénieur ou l'administrateur sera chargé de la réparation de la panne déclarée

1.4 Problématique et Objectifs

Après l'analyse de la situation du service STS, nous avons recensé les problèmes fréquents et les plus rencontrés afin de proposer des solutions.

1.4.1 Problématique

Le service STS souffre de plusieurs problèmes, La gestion des pannes est faite d'une manière manuelle, dans des fichiers Word, qui provoque plusieurs soucis :

- Utilisation de plusieurs documents (rédaction des rapports).
- La difficulté de la maintenance, au niveau de l'affectation d'un ingénieur pour la réparation.
- Une perte de temps dans la recherche des documents des équipements déjà réparés.
- Absence des statistiques liées à ses pannes.

1.4.2 L'objectif

L'objectif est de développer une application, avec une base de données derrière, qui répond aux différents problèmes ci-dessus cités. Donc, l'application à développer pour SONATRACH doit répondre aux objectifs suivants :

- Offrir une bonne gestion et suivi des équipements (ajouter un équipement, modifier un équipement, supprimer un équipement).

- Offrir, aussi, une bonne gestion des utilisateurs et les sites (ajout, modification et suppression).
- Automatiser l'ensemble des documents utilisés.
- Offrir un suivi de la maintenance des équipements, avec leurs historiques.
- Offrir la possibilité d'avoir l'état des équipements.
- Faciliter la déclaration des pannes ainsi que leurs réparations.

1.5 Les Besoins Fonctionnels et Non Fonctionnels

Après avoir cerné la problématique du service STS, et fixer les objectifs de l'application à concevoir et à réaliser, nous passons maintenant à l'élaboration du recueil des besoins, ou exigences, auxquels notre applications doit répondre. Ces besoins sont de deux types, à savoir les besoins fonctionnels et les besoins non-fonctionnels ou techniques.

1.5.1 Les Besoins Fonctionnels

Les besoins fonctionnels représentent les fonctionnalités que l'application développée doit réaliser, et qui sont :

Gestion des utilisateurs : l'administrateur doit être capable d'effectuer les opérations de gestion telle que l'ajout, la suppression, la modification et la consultation des informations caractérisant chacun des utilisateurs.

Gestion des équipements : Il s'agit d'un outil permettant d'effectuer les opérations de gestion tel que l'ajout, la suppression, la modification et la consultation des informations caractérisant chacun des équipements, qui sera une tâche réservée pour l'administrateur aussi.

Consultation de l'état des équipements : Le technicien aura la possibilité d'avoir un suivi sur l'état des équipements, qui leur permet de connaître les pannes qui se sont produites, la date de la panne, ainsi que les détails de cette panne, et en cas d'une panne déjà réparé il aura l'historique, la date de réparation, le réparateur, et les détails de réparation.

Gestion des pannes : l'opérateur ainsi que le technicien auront la possibilité de déclarer une panne (Ajouter une panne), lorsqu'il s'agit d'une panne déclarée par un opérateur le technicien aura la possibilité de bien spécifier la panne (Modifier la panne), de réparer la panne (Supprimer la panne).

1.5.2 Les Besoins non Fonctionnels

Représentent les exigences techniques que le système doit vérifier :

L'extensibilité : l'application développée devra être extensible, c'est-à-dire elle aura la possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

La sécurité : L'application devra être sécurisée, les informations ne seront pas accessibles à tout le monde, mais avec un identifiant et mot de passe attribués aux utilisateurs.

L'interface homme machine et l'ergonomie : L'application doit respecter les principes des interfaces Homme/Machine (IHM) tels que l'ergonomie et la fiabilité, et le thème adopté par l'application doit être inspiré des couleurs et du logo type de l'entreprise d'accueil.

1.6 Langage et Processus de développement

Concernant les phases d'analyse et la conception de notre application, nous allons utiliser le langage UML (Unified Modeling Language), avec quelques diagrammes essentiels, pour modéliser différents aspect du système à mettre en place, et pour la démarche et les étapes à suivre, nous avons utilisé un processus allégé d'UP (Unified Process) décrit dans [Annexe A].

1.7 Conclusion

Dans ce premier chapitre, l'organisme d'accueil, à savoir l'entreprise SONATRACH, ainsi que le service STS, ont été présenté. Par la suite, le processus de maintenance et de suivis de matériel informatique a été décrit. Dans ce processus nous avons détecté des problèmes lié principalement à la gestion manuelle de l'équipement informatique et leurs maintenances, pour cela nous avons proposé une application qui automatise cette gestion et doit respecter les besoins fonctionnels et techniques énuméré à la fin de ce chapitre. Avant de passer au second chapitre, qui sera dédié à l'analyse des besoins, nous avons aussi fixer la méthode d'analyse et de conception basée sur le langage UML avec un processus proche de UP.

CHAPITRE 2

ANALYSE DES BESOINS

2.1 Introduction

Dans ce chapitre, dédié à la spécification et l'analyse des besoins, nous allons identifier les acteurs et leurs interactions avec le système, en utilisant le diagramme de contexte. Ceci nous aidera à identifier, pour chaque acteur, les différents cas d'utilisation qui doivent être adéquats avec les fonctionnalités attendues de notre application.

Par la suite, chaque cas d'utilisation sera décrit textuellement, avec ses différents scénarios, en utilisant un format tabulaire, et il sera aussi décrit par un diagramme séquence système. Dans ce dernier diagramme, le système est vu comme une boîte noire qui répond aux différents messages des différents acteurs.

2.2 Identification des acteurs

Un acteur représente un rôle joué par une entité externe, 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 porteur de données [2]. En ce qui concerne notre système, nous avons pu identifier les acteurs suivants :

L'opérateur : Son rôle est la signalisation de l'état de l'équipement en cas d'une panne.

L'ingénieur : Son rôle est la réparation des pannes, il peut aussi faire une déclaration d'une panne.

L'administrateur : Son rôle est la gestion des équipements informatiques et l'affectation des pannes à des ingénieurs, il peut aussi déclarer une panne ainsi que sa réparation .

2.3 Diagramme de contexte

Nous allons représenter l'interaction de futur système (le système est considéré comme une boîte noire) avec son environnement extérieur, en identifiant les différents messages échangés entre les différents acteurs identifiés précédemment.

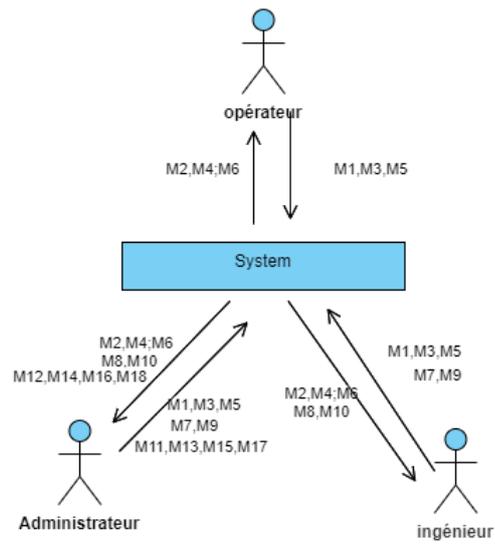


FIGURE 2.1 – Diagramme de Contexte.

2.4 Identification des messages

résumons les différents messages circulés entre le système et les acteurs :

Acteur	Message acteur vers Système		Message Système vers Acteur	
Utilisateur (Opérateur, ingénieur ou Administrateur)	M1	L'authentification (identifiant et mot de passe).	M2	Réponse du système, l'affichage de la page d'accueil en cas du succès d'authentification et la page d'authentification en cas d'échec d'authentification.
	M3	Demande de déclaration d'une panne.	M4	Affichage de l'interface «Déclarer une panne ».
	M5	Confirmation de la panne	M6	Notification d'une panne.
Ingénieur, Administrateur	M7	Demande de l'interface « Réparer la panne ».	M8	Affichage de l'interface «Réparer la panne ».
	M9	Confirmation de la réparation.	M10	Notification renvoyée
Administrateur	M11	Accès à la gestion des utilisateurs.	M12	Affichage de la liste des utilisateurs avec possibilité d'ajout, de la suppression et de modification.
	M13	Accès à la gestion des sites.	M14	Affichage de la liste des sites avec possibilité d'ajout, de la suppression et de modification.
	M15	Accès à la gestion des équipements.	M16	Affichage de la liste des équipements avec possibilité d'ajout, de la suppression et de modification.
	M17	Demande de consultation du rapport mensuel.	M18	Affichage de consultation.

TABLE 2.1 – Identification des messages

2.5 La relation entre les acteurs

D'après [9] (page 81), nous avons ajouté un acteur abstrait (acteur généralisé) appelé utilisateur, pour permettre la représentation de tout acteur non encore identifié par le système. Un héritage, en termes de fonctionnalités, a été défini dans la figure ci-dessous :

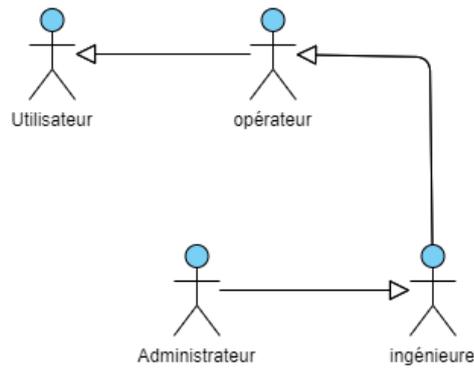


FIGURE 2.2 – La relation entre les acteurs

2.6 Identification des cas d'utilisation

Dans le tableau ci-dessous, nous avons énuméré les différents cas d'utilisation par acteur.

N°	Cas d'utilisation	Acteur
1	Authentification	Opérateur Administrateur Ingénieur
2	Déclaration d'une panne	Opérateur Administrateur Ingénieur
3	Afficher la liste des équipements	Ajouter un équipement
		Modifier un équipement
		Supprimer un équipement
4	Afficher la liste des sites	Ajouter un site
		Modifier un site
		Supprimer un site
5	Afficher la liste des utilisateurs	Ajouter un utilisateur
		Modifier un utilisateur
		Supprimer un utilisateur
6	Modifier la panne déclarée	Administrateur L'ingénieur
7	Réparer la panne	Administrateur L'ingénieur
8	Consultation de rapport mensuel.	Administrateur L'ingénieur

TABLE 2.2 – identification des cas d'utilisation

2.7 Diagramme de cas d'utilisation

En langage UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système. Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne.

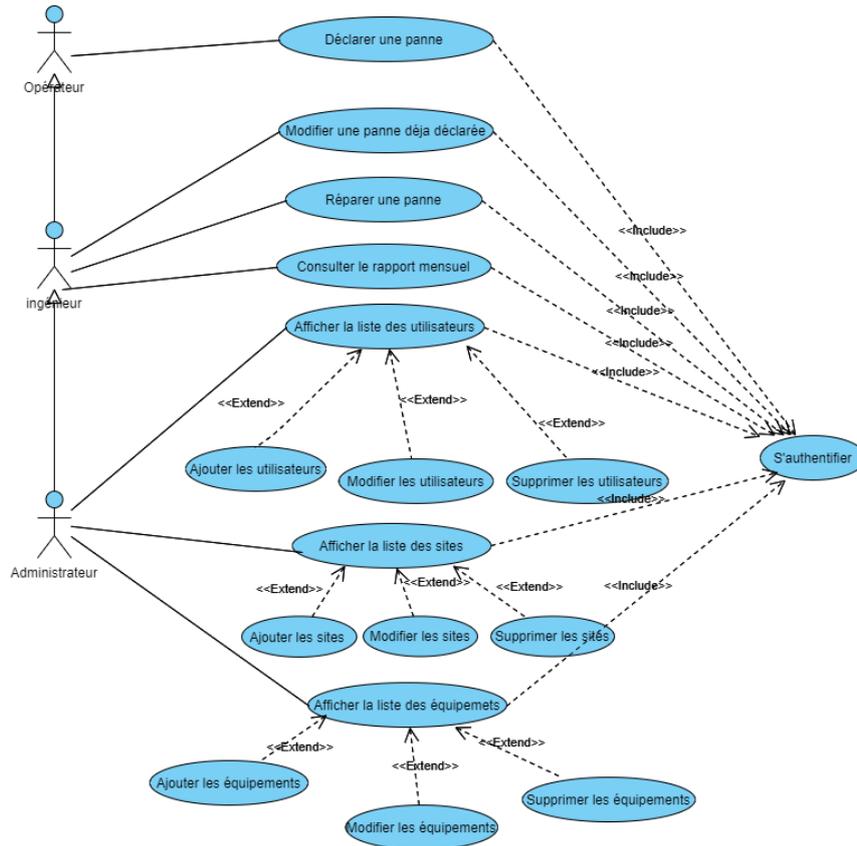


FIGURE 2.3 – Diagramme de cas d'utilisation

2.7.1 Diagramme de cas d'utilisation – Acteur : «Utilisateur »

Le diagramme de cas d'utilisation pour l'acteur «Utilisateur » représente Uniquement l'authentification ,comme la figure ci -dessous l'indique .

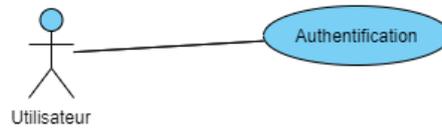


FIGURE 2.4 – Diagramme de cas d'utilisation de l'utilisateur

2.7.2 Diagramme de cas d'utilisation – Acteur : « Opérateur »

Le diagramme suivant représente les cas d'utilisation associé à l'opérateur, ce dernier peut réaliser qu'une seule tâche représentée dans la Figure ci-dessous.

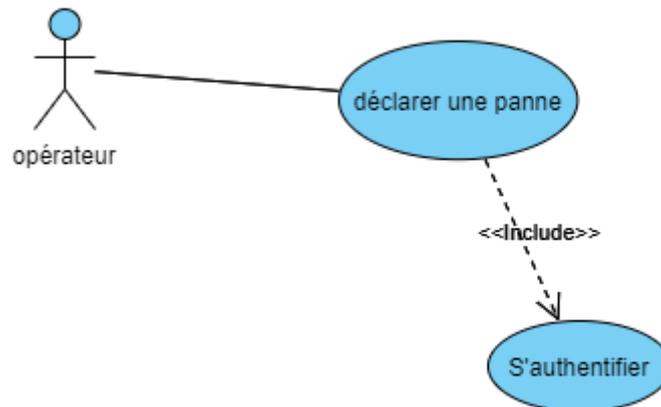


FIGURE 2.5 – Diagramme de cas d'utilistion de l'opérateur

2.7.3 Diagramme de cas d'utilisation – Acteur : « Ingénieur »

Le diagramme ci-dessous représente les tâches concernées par l'ingénieur :

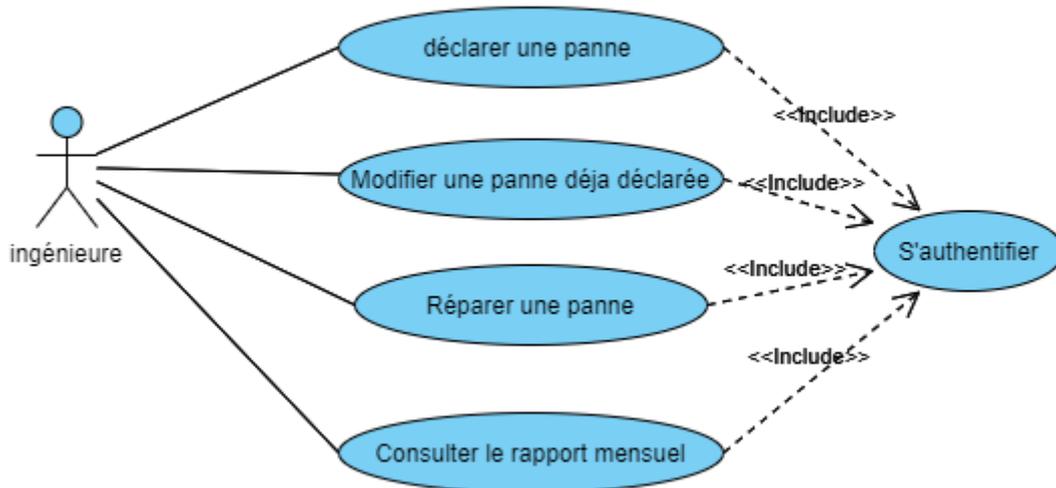


FIGURE 2.6 – Diagramme de cas d’utilisation de l’ingénieur

2.7.4 Diagramme de cas d’utilisation – acteur « Administrateur »

Dans le diagramme suivant, nous allons présenter les tâches liées à l’administrateur :

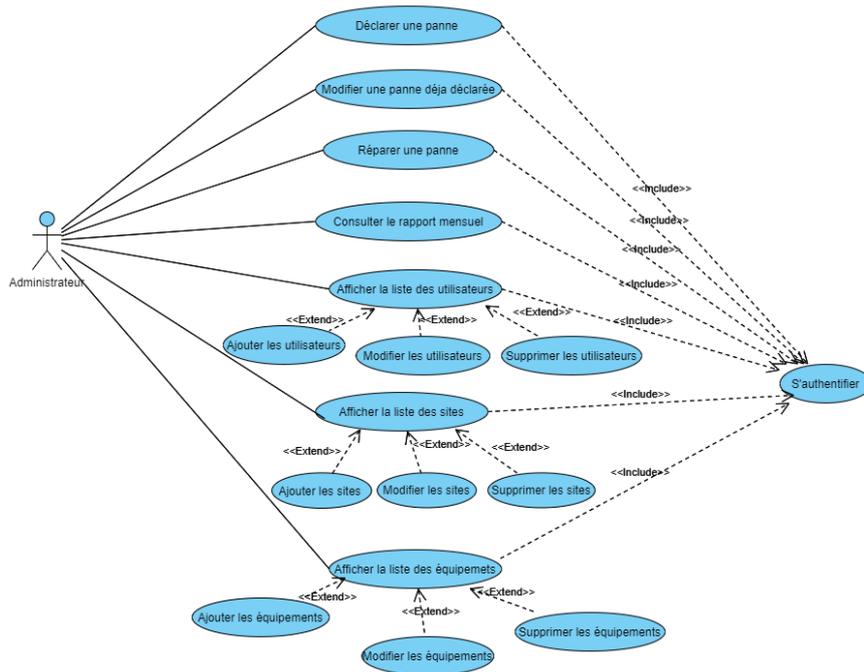


FIGURE 2.7 – Diagramme de cas d’utilisation de l’administrateur

2.8 Description Textuelle des cas d'utilisation

Cette étape consiste à définir textuellement quelque cas d'utilisation, et cela par :

- Identification du cas : son objectif ainsi que les acteurs concernés.
- Description d'un scénario nominal : c'est la description du déroulement des actions selon certaines conditions.
- Un scénario alternatif : présente les actions qui s'exécutent lorsque les conditions sont invalides .
- Un scénario d'exception : le scénario d'exception sert lorsqu'une étape du déroulement pourrait être perturbée à cause d'un événement anormal .

Dans ce qui suit, Nous allons donner une description textuelle pour chaque cas d'utilisation

2.8.1 Description de Cas d'utilisation « S'authentifier »

Le tableau suivant représente la description des cas d'utilisation « s'authentifier » :

Cas d'utilisation « Authentification »	
objectif	Attribuer le rôle à l'utilisateur authentifié après l'identification.
Acteur	L'utilisateur
Description	L'utilisateur doit saisir l'email et un mot de passe puis valide le formulaire d'authentification. le système utilise les informations saisis pour vérifier la cohérence, en cas du succès de l'authentification ,le système affiche l'interface adéquate pour l'utilisation sinon un message d'erreur sera affiché avec la page d'authentification pour ré-authentifier .
précondition	/
Scénario nominal	c'est le premier cas d'utilisation déclenché pour tous les utilisateurs ce cas est pour le but d'accéder au système ,peut le résumer dans les lignes suivantes : 1-Le système affiche l'interface de l'authentification. 2-l'utilisateur saisi son email et son mot de passe et valide l'opération. 3- le système vérifie les informations saisis : -si l'utilisateur a oublié de saisie son email ou son mot de passe "Exception 1" sera généré. -si le mot de passe ou l'email sont incorrect "Exception 2" sera généré . 4- dans le cas du succès le système affiche l'interface d'accueil adéquat à chaque utilisateur.
Scénario d'exception	Exception 1 : Le système affiche une erreur à l'utilisateur lui indiquant qu'il a oublié de saisir l'identifiant ou mot de passe. Exception 2 : Le système indique une erreur liée à la cohérence des informations saisis.

TABLE 2.3 – Description du cas «Authentification ».

2.8.2 Description de cas d'utilisation « Déclarer une panne »

Le tableau ci-dessous représente la description du cas d'utilisation «Déclarer une panne» :

Cas d'utilisation « Déclarer une panne »	
objectif	La déclaration de la panne.
Acteur	L'administrateur - L'ingénieur -l'opérateur
Description	L'utilisateur doit d'abord s'authentifier puis accéder à sa page d'accueil, il choisit l'équipement correspond au type sélectionné. L'utilisateur remplit les champs nécessaires puis il valide , une notification de système sera renvoyée indique l'enregistrement de la panne saisie.
précondition	L'authentification
Scénario nominal	Ce cas est pour le but de déclaration d'une panne , on peut le résumer dans les lignes suivantes : 1-l'utilisateur demande la page de déclaration d'une panne. 2- Le système renvoie la page demandée. 3- l'utilisateur remplit les champs nécessaires (type équipement, date de la panne, description de la panne). 4- le système vérifie les informations saisies : -Si l'utilisateur a oublié de remplir un des champs « Exception1» sera générée. Dans le cas de succès, le système sauvegarde les informations saisies, et affiche un message de confirmation.
Scénario d'exception	Exception 1 : Le système affiche une erreur à l'utilisateur lui indiquant l'obligation de la saisie du champ oublié.

TABLE 2.4 – Description du cas «Déclarer une panne ».

2.8.3 Description de cas d'utilisation « Ajouter un utilisateur »

Le tableau ci-dessous représente la description des cas d'utilisation «Ajouter un utilisateur » :

Cas d'utilisation « Ajouter un utilisateur »	
objectif	Ce cas permet à l'administrateur d'ajouter les différents types des utilisateurs.
Acteur	L'administrateur
Description	L'administrateur doit d'abord s'authentifier puis accéder à la page « gestion des utilisateurs » Il peut consulter la liste des utilisateurs comme il peut choisir l'opération « ajouter un utilisateur ». .
précondition	L'authentification
Scénario nominal	Ce cas est pour le but de la gestion des utilisateurs, on peut le résumer dans les lignes suivantes : 1-l'administrateur demande la page de gestion des utilisateurs. 2- Le système renvoie la page demandée. 3-l'administrateur choisit la page « ajouter un utilisateur ». 4- le système renvoie la page demandée. 5-l'administrateur remplit les champs nécessaires. 6- le système vérifie les informations saisies : -Si l'administrateur a oublié de remplir un des champs « Exception1 » sera générée. -Dans le cas de succès, le système sauvegarde les informations saisies , et affiche un message de confirmation. .
Scénario d'exception	Exception 1 : Le système affiche une erreur à l'utilisateur lui indiquant l'obligation de la saisie du champ oublié.

TABLE 2.5 – Description du cas « Ajouter un utilisateur».

2.9 Diagramme de séquence système

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Nous allons présenter les diagrammes de séquences pour les cas d'utilisation déjà détailler.

2.9.1 Diagramme de séquence système cas d'utilisation « s'authentifier »

Chaque utilisateur doit s'authentifier pour accéder à son espace de travail, par la saisie de son nom d'utilisateur et un mot de passe.

Le système vérifie les informations saisies, dans le cas d'échec ou l'un des champs est incorrect un message d'erreur sera affiché sinon l'utilisateur aura l'accès a son interface.

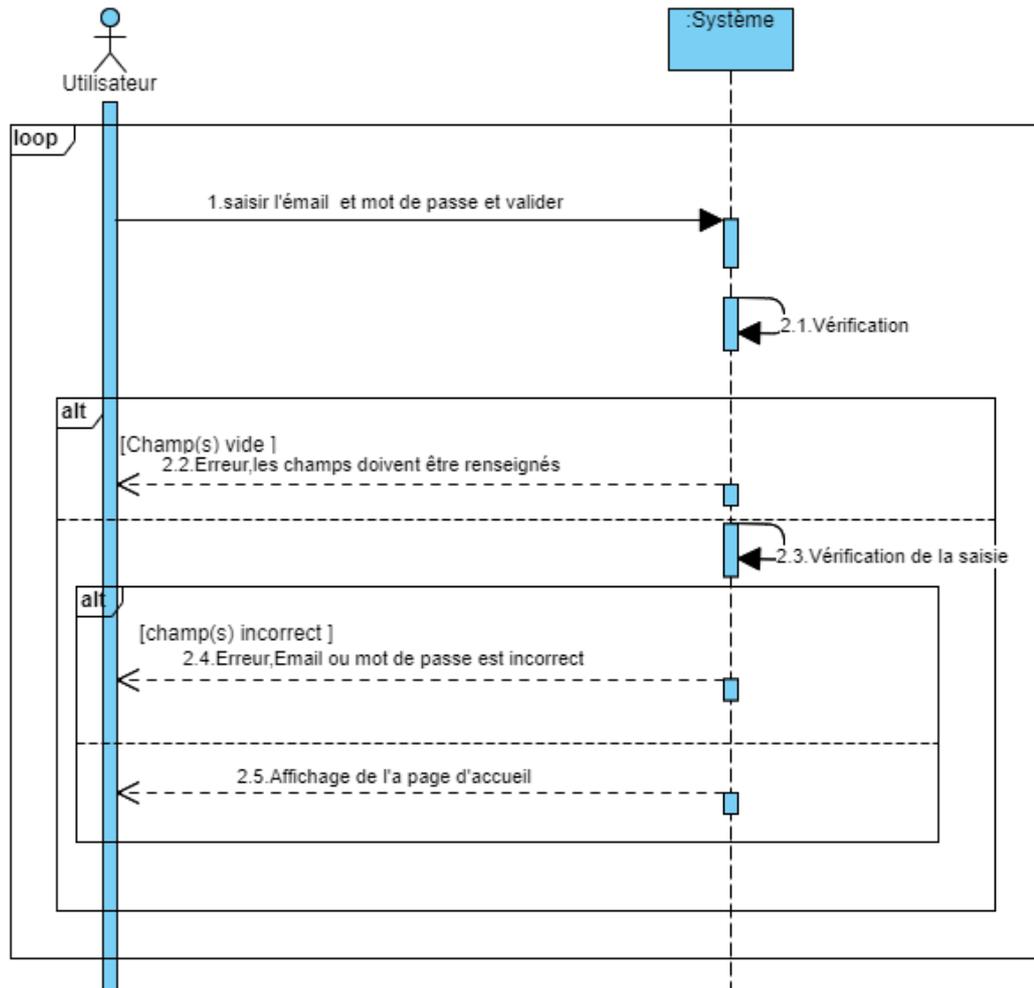


FIGURE 2.8 – Diagramme de séquence système du cas d'utilisation « S'authentifier »

2.9.2 Diagramme de séquence système cas d'utilisation « Ajouter un utilisateur »

Pour qu'un administrateur puisse ajouter un utilisateur, il faut qu'il authentifie , puis il accède à la page « gestion des utilisateurs » et demande la page de l'ajout d'un utilisateur , remplir tous les champs qui sont obligatoires comme le nom , le prénom, le site,....., Sinon une erreur sera envoyer .

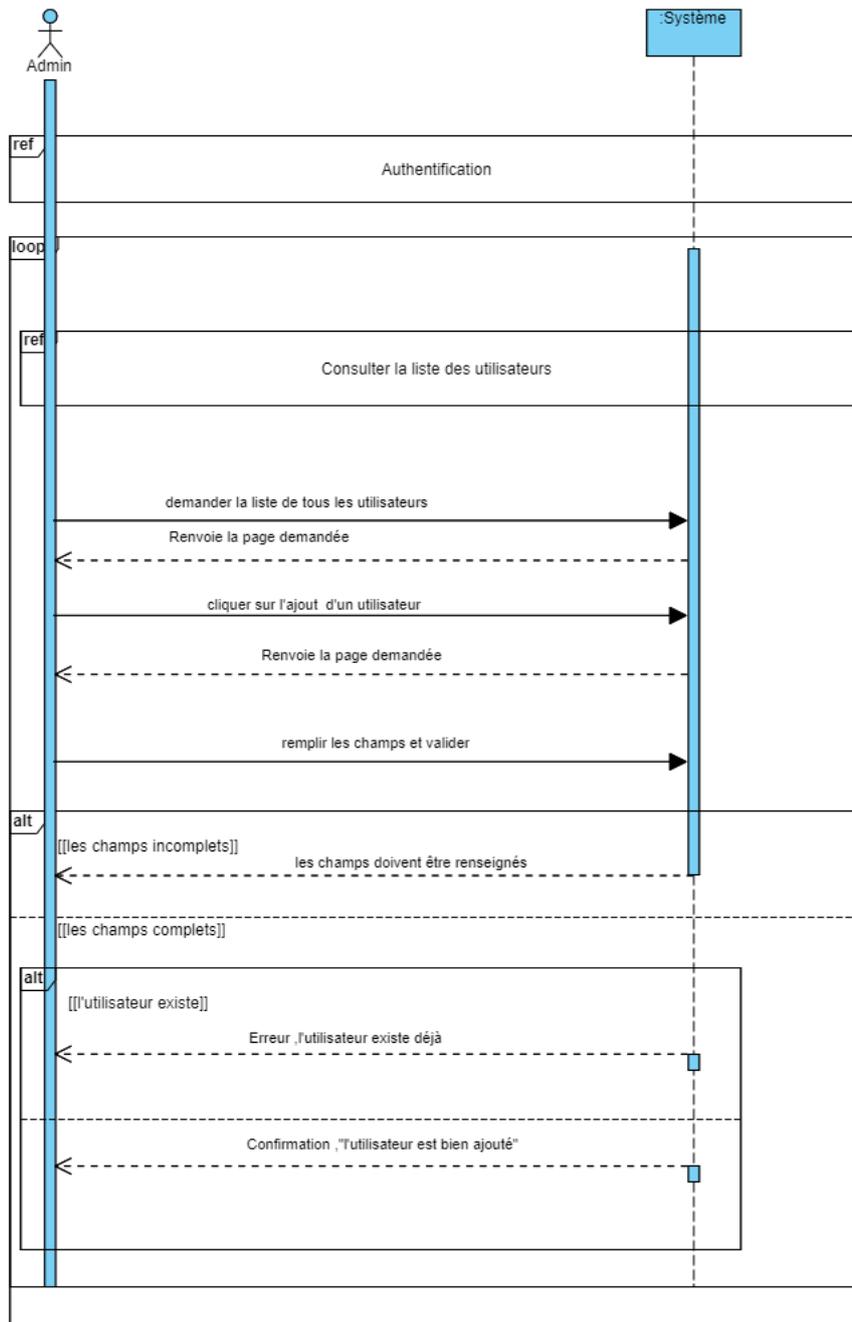


FIGURE 2.9 – Diagramme de séquence système du cas d'utilisation «Ajouter un utilisateur »

2.9.3 Diagramme de séquence système cas d'utilisation « Déclarer une panne »

Dans une déclaration d'une panne ,utilisateur doit d'abord s'authentifier puis il demande la page « déclarer une panne » ,il doit remplir les champs qui sont obligatoires sinon une erreur sera envoyer **Remarque**

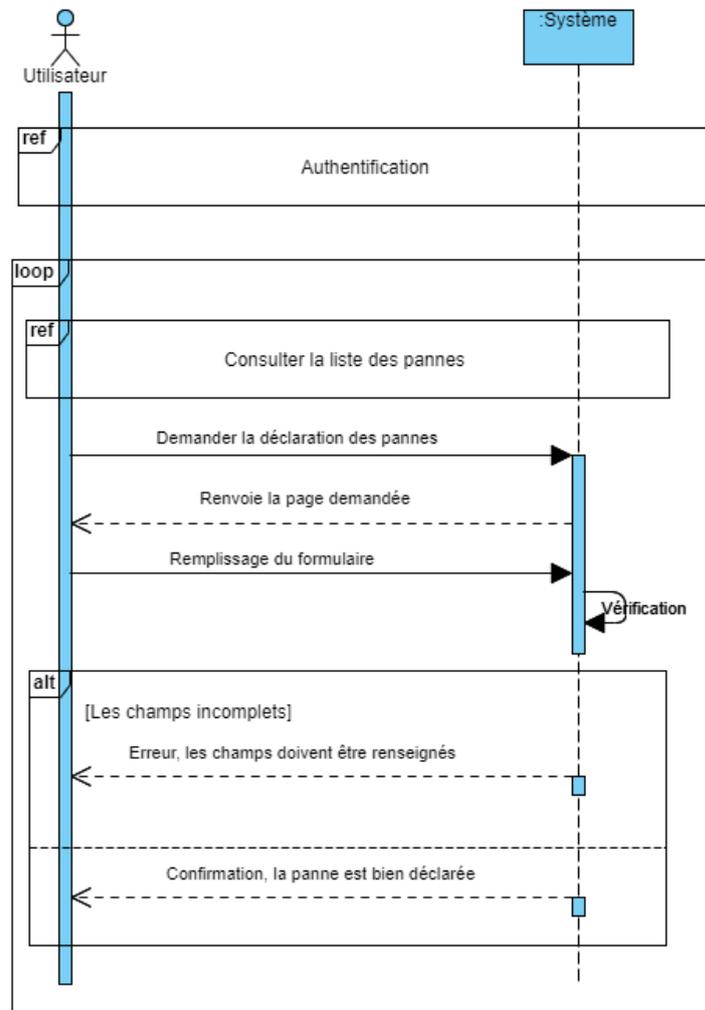


FIGURE 2.10 – Diagramme de séquence système du cas d'utilisation « Déclarer une panne »

- pour plus de description textuelle des autres cas d'utilisations ainsi que leurs diagrammes de séquences système voir [Annexe B]

2.10 Conclusion

Dans ce chapitre nous avons effectué l'analyse des besoins de notre application et de comprendre ce que doit faire l'application qui est vu comme une boîte noire. Ceci, nous a permis de répondre à la question « QUOI ». Le chapitre suivant, dédié à la conception et l'élaboration du schéma relationnel, dans lequel le système sera détaillé en termes d'objets intervenant à la réalisation de chaque cas d'utilisation, nous permettra de répondre à la question « Comment ? ».

CHAPITRE 3

CONCEPTION ET SCHÉMA RELATIONNEL DE DONNÉES

3.1 Introduction

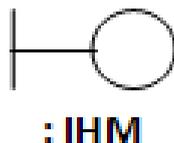
La conception du projet correspond à l'une des premières phases dans le cycle de vie elle permet de structurer, organiser, planifier le projet. Dans ce chapitre, nous allons présenter en détails la conception du projet à travers les diagrammes d'interaction, le diagramme de classes et aussi le modèle relationnel.

3.2 Diagrammes d'interactions

Un diagramme d'interaction est un type de diagramme UML utilisé pour représenter le comportement interactif d'un système. Les diagrammes d'interaction se concentrent sur la description du flux de messages au sein d'un système, qui est décomposé en objets en interaction afin de réaliser un cas d'utilisation.

Les objets utilisés dans le diagramme d'interaction sont de trois types de classes :

- **Classes d'interface (boundary)** : Ces classes sont utilisées pour modéliser l'interaction entre le système et ses acteurs c'est-à-dire les utilisateurs et les systèmes externes. Ce type de classe est schématisé comme suit :



- **Classes de Contrôle (Control)** : Elles contiennent les traitements et la cinématique de l'application. Elles assurent aussi la transition entre les classes d'interface et les classes

entités. Le stéréotype utilisé est :



- **Classes entités (entity)**) :Elles représentent la persistance des données, c'est-à-dire qui vont garder leurs informations (données) après l'exécution d'un cas d'utilisation particulier. Ces classes représentent les objets métiers et ils sont schématisés comme suit :



3.2.1 Les Diagrammes d'interaction des cas d'utilisation

Diagramme d'interaction de cas d'utilisation «S'authentifier »

L'utilisateur de l'application doit s'authentifier pour accéder à l'application, et d'exploiter de ces fonctionnalités, après la saisie de l'identifiant et le mot de passe .

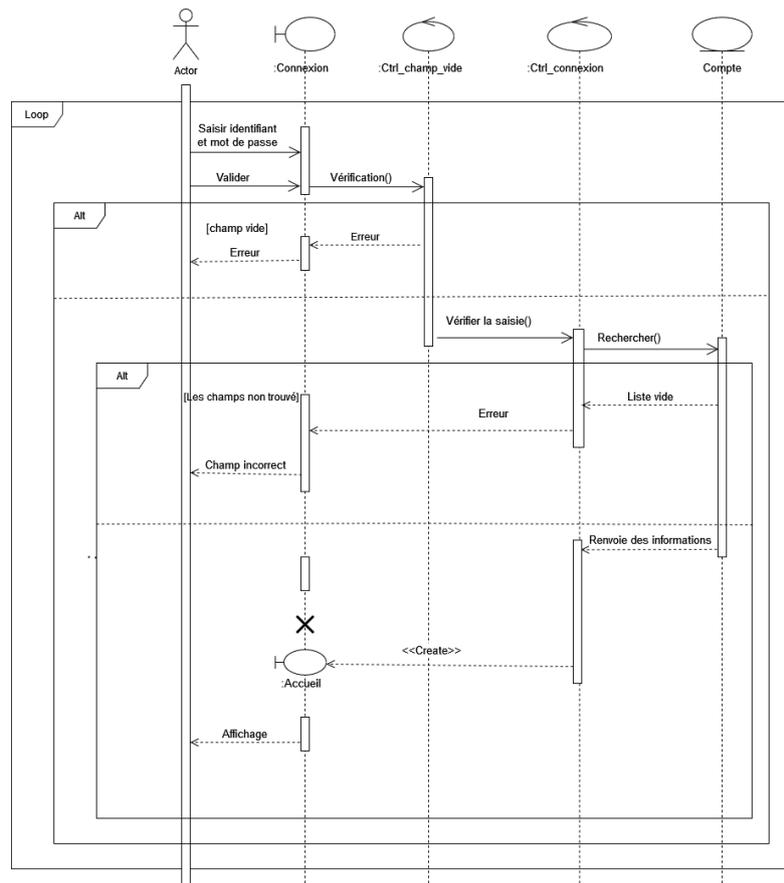


FIGURE 3.1 – Diagramme d’interaction de cas d’utilisation «S’authentifier ».

Diagramme d’interaction de cas d’utilisation «Ajouter un utilisateur»

L’administrateur doit s’authentifier pour pouvoir ajouter un utilisateur.

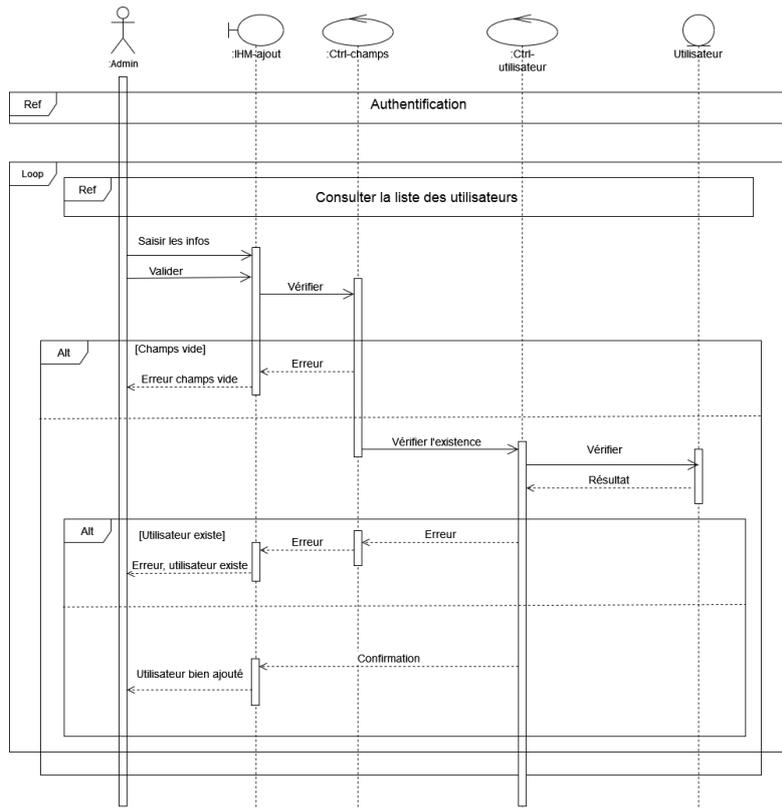


FIGURE 3.2 – Diagramme d’interaction de cas d’utilisation «Ajouter un utilisateur».

Diagramme d’interaction de cas d’utilisation «Déclarer une panne»

L’administrateur doit d’abord s’authentifier pour accéder à la page de déclaration des pannes.

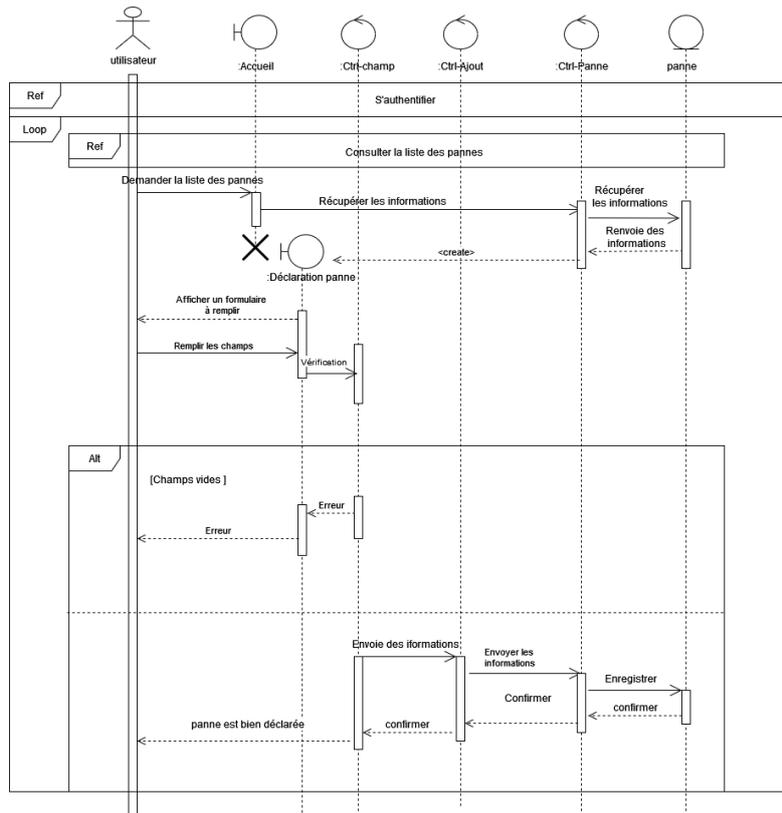


FIGURE 3.3 – Diagramme d’interaction de cas d’utilisation «Déclarer une panne»

3.3 Diagramme de classe de domaine

Un diagramme de classes fournit une vue globale d’un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l’interaction.

La figure suivante, représente le diagramme de classe de domaine qui représente les données de l’application.

Remarque :

L’entité « Compte » n’a pas été représentée dans le diagramme de classe du domaine ci-dessus, puisqu’elle est équivalente à l’entité « Utilisateur ». Et aussi un utilisateur possède un seul compte et un compte est affecté à un seul utilisateur (deux classes équivalente).

3.3.1 Description détaillée des attributs de classes

Nous allons décrire les différentes classes schématisées dans la figure précédente. Cette description sera présentée sous forme d’un tableau, comme présenté dans la table ci-dessous :

Classe	Responsabilité	Attributs		
		signification	Définition	Type
Utilisateur	Classe qui enregistre les informations des utilisateurs	idUtilisateur	Identifiant	Numérique
		email	l'email de l'utilisateur	Chaîne de caractère
		Mot de passe	Mot de passe de l'utilisateur	Chaîne de caractère
		nom	Nom de la personne	Alphabétique
		prénom	Prénom de la personne	Alphabétique
		estActivé	L'état de la personne	Boolean
Administrateur	Classe qui enregistre les informations des administrateur	idAdmin	Identifiant	Numérique
Opérateur	Classe qui enregistre les informations des opérateurs	idOpérateur	Identifiant	Numérique
Ingénieur	Classe qui enregistre les informations des ingénieurs	idIngénieur	Identifiant	Numérique
Equipement	Classe qui enregistre les informations des équipements	idEquipement	Identifiant	Numérique
		Num serie	Numéro de série de l'équipement	Chaîne de caractère
		type	Le type de l'équipement	Alphabétique
		designation	La description de l'équipement	Alphabétique
		date debut service	La date de début de service	Date
Panne	Classe qui enregistre les informations des pannes	idPanne	Identifiant	Numérique
		description	La description sur la panne	Alphabétique
		Commentaire	un commentaire sur la panne	Alphabétique
		dateP	La date de la panne	Date
Détail	Classe qui stoke les détails de la panne	idDetail	Identifiant	Numérique
		commentaire	les détails ajoutés	Alphabétique
Notification	Classe qui stoke les notifications	idNotification	Identifiant	Numérique
		Message	le message de la notification	Alphabétique
DétailsN	Classe qui stoke les détails des notification	idDetailN	Identifiant	Numérique
		estLu	l'etat de la notification	Boolean

TABLE 3.1 – Description des classes d'objets et leurs attributs.

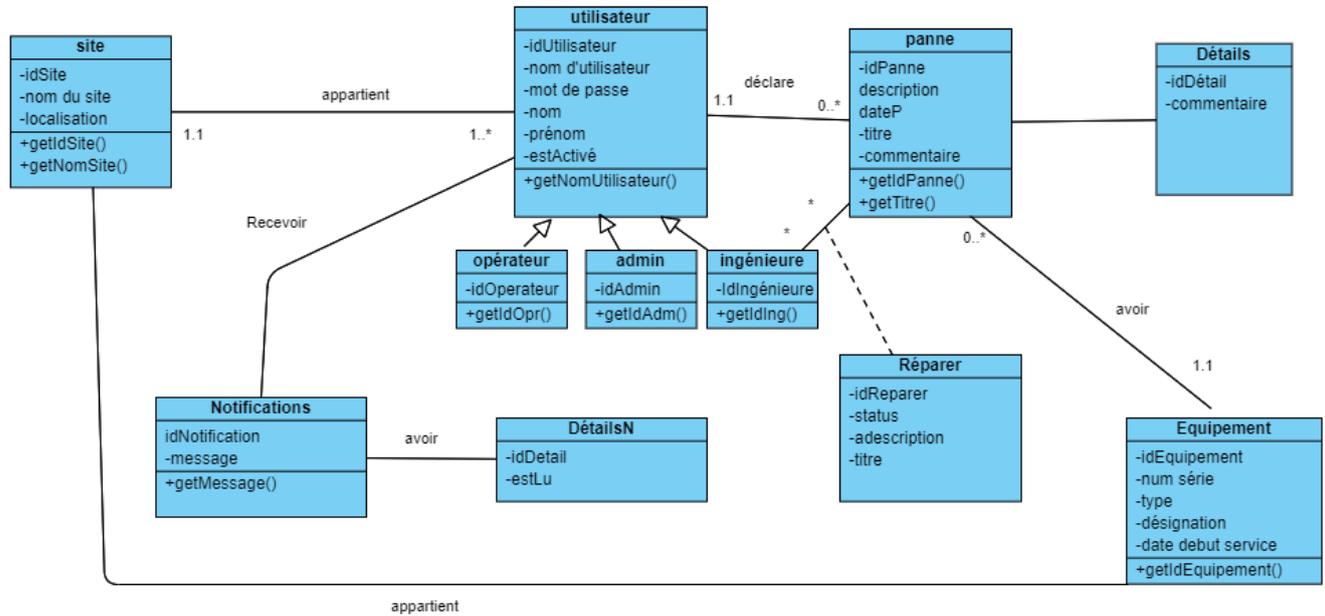


FIGURE 3.4 – Diagramme de classe de domaine.

3.3.2 Description détaillée des attributs des classes d’associations

La classe d’association et ses attributs sont représentés par le tableau ci-dessous :

Classe d’association	Attributs		
	Signification	Définition	Type
Réparer	IdRéparer	Identifiant	Numérique
	description	La description de la panne réparée	Chaîne de caractère
	date réparation	La date de la réparation de la panne	Date

TABLE 3.2 – Description des classes d’association et leurs attributs.

3.4 Schéma relationnel

Nous allons réaliser le modèle relationnel à partir du diagramme de classes du domaine, qui est le modèle logique de données qui décrit de manière abstraite comment les données sont représentées et structurées dans la base de données. Pour décrire une relation, nous allons simplement la nommer, suivie de son nom d’attribut entre parenthèses. L’identifiant d’une relation est constitué d’un ou plusieurs attributs qui constituent la clé primaire. Une relation peut référencer une autre relation à l’aide d’une clé étrangère, qui est la clé primaire de la relation référencée.

3.4.1 Règles de passage au modèle relationnel

Les règles de passage au modèle relationnel sont :

- **Classes** : Chaque classe devient une relation .
- **Association unique-unique** : on supprime l'une des relations, et ses attributs sont migrés vers l'autre relation. Dans notre cas, nous avons supprimé la relation issue de la classe ayant moins d'association.
- **Association multiple-unique** : On ajoute la clé primaire de la relation coté cardinalité multiple comme clé étrangère dans la relation coté cardinalité unique.
- **Association multiple-multiple** : Créer une nouvelle relation pour l'association qui a comme clé primaire la concaténation des deux clés primaires des deux relations issues des deux classes associées.
- **Cas d'héritage** : Il existe trois décompositions pour traduire une association d'héritage en fonction des contraintes existantes :
 - **Décomposition par distinction** : garder toutes les relations issues des classes mère et filles.
 - **Décomposition descendante (push-down)** : Supprimer la relation issue de la classe mère et transférer ces attributs vers les relations issues des classes filles .
 - **Décomposition ascendante (push up)** : Supprimer les relations issues de classes filles et transférer leurs attributs vers la relation issue de la classe mère. Souvent, nous ajoutons un attribut supplémentaire dans la relation issue de la classe mère qui représente son type par rapport aux classes fille.

3.4.2 Le passage au modèle relationnel

Après avoir appliqué tous les règles de passage au modèle relationnel, nous avons obtenu le schéma suivant :

Utilisateur(idUtilisateur,nom d'utilisateur,email,mot de passe,role,estActivé).

Equipement(idEquipement,mid,nom,description,type,#idSite).

Site(idSite,nomSite,locationSite).

Reparer(idReparer,titre,description,date de reparation,#idUser,#idpanne).

panne(idPanne,titre,description,status,idDetail,commentaire,#idUtilisateur,#idEquipement).

Notification(idNotification, #idUtilisateur,message,estLu).

Remarque

les diagrammes d'interactions des autres cas d'utilisations voir [Annexe C]

3.5 Conclusion

Dans ce chapitre nous avons présenté la phase de conception de notre projet à travers les diagrammes d'interactions, qui nous ont permis de décrire de manière globale et détaillée, le fonctionnement désiré du système. Par la suite nous avons décrit les règles de passage du diagramme de classe vers le modèle relationnel qui nous permet d'avoir le schéma de la base de

données de l'application à réaliser.

Le prochain chapitre, sera consacré à la phase de réalisation, dans laquelle nous implémentons le schéma relationnel, et les différents composants logiciels de notre application.

4.1 Introduction

Ce chapitre sera dédié à la présentation de la réalisation de notre projet, l'ensemble des outils et les langages de programmation utilisées, ainsi que le schéma physique de la base de données, et un diagramme de déploiement, et nous terminons par quelques interfaces homme-machine.

4.2 Langage et environnement de développement

Nous avons utilisé un ensemble de langage qu'on va énumérer dans cette section.

4.2.1 Les langages utilisés

l'ensemble des langages utilisés sont :

- **Html** (HyperText Markup Language) : est un langage informatique est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage[7].
- **CSS**(Cascading Style Sheets) : Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.[7]
- **JS**(JavaScript) : Le JavaScript est un langage informatique utilisé sur les pages web. Ce langage à la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activé côté serveur. L'exécution du code est effectué par votre navigateur internet tel que Firefox ou Internet Explorer.[7]
- **NodeJS** :Node.js sert à faire du Javascript server side, et peut être utilisé dans des

applications de bases de données, la plus populaire étant MySQL. Ce n'est ni un framework ni un serveur. Node.js est souvent confondu avec ce dernier, car c'est sa base : créer des applications en temps réel, où le serveur a la possibilité de transmettre de l'information au client.[14]

- **MySQL** : un système de gestion de base de données relationnelle open source. Il est basé sur le langage SQL (Structure Query Language), utilisé pour ajouter, supprimer et modifier des informations dans la base de données. Les commandes SQL standard, telles que ADD, DROP, INSERT et UPDATE, peuvent être utilisées avec MySQL.[3]

4.2.2 Outils et bibliothèque

les outils et les bibliothèques utilisés sont **WampServer** : (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP.

WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL .[10]

VueJS : est un framework évolutif pour construire des interfaces utilisateur. À la différence des autres frameworks monolithiques, Vue a été conçu et pensé pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque se concentre uniquement sur la partie vue, et il est vraiment simple de l'intégrer avec d'autres bibliothèques ou projets existants. D'un autre côté, Vue est tout à fait capable de faire tourner des applications web monopages quand il est couplé avec des outils modernes et des bibliothèques complémentaires.

Dans notre projet ,nous avons utilisés VueJs version 3.[6] **Sequelize** : Sequelize est un ORM pour node.js compatible avec différents moteurs de base de données comme Mysql, Sqlite... etc.[4]

Express : Express est une infrastructure d'applications Web Node.js minimaliste et flexible qui fournit un ensemble de fonctionnalités robuste pour les applications Web et mobiles.[15]

Bootstrap : Bootstrap est un Framework crée par deux développeurs du réseau social Twitter et mis en open source en 2012, il est destiné à faciliter la création d'application web. Il regroupe une collection d'outil fournis sous la forme de classes CSS et de librairies JavaScript et jQuery, permettant aux développeurs de gagner du temps et de réaliser simplement des codes complexes (animation, tableau, carousel,...) tout en réduisant la quantité de caractères requis, et donc le poids du site web.[1]

Visual Studio Code : 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.[5]

4.3 Modèle MVC

Dans notre projet nous avons utilisésLe modèle MVC (Model View Controller) qui représente une façon de découper un programme informatique en trois parties.

Le modèle :correspond à la gestion de notre base de données qui seront manipulées. Il concerne

l'accès à la base de données (CRUD soit Create,Read, Update et Delete).

Les vues : elles correspondent à l'affichage des données aux utilisateurs, Permettant ainsi de les créer ,les modifier, les mettre à jour ou les supprimer. Les vues seront des pages HTML qui permettront l'interaction avec les utilisateurs .

Les contrôleurs : ils permettent l'enchainements des vues entre elles, en fonction des actions des utilisateurs dans chacune d'elles .On appelle cet enchainement le « routage ».[15]

4.4 Structure du code source

Lors du développement de notre application, nous avons séparé la partie backend du frontend .

1. **Backend** :nous avons structuré la partie du backend comme suit :

Le répertoire config : comporte le fichier config.js qui sert à configurer les paramètres nécessaires, comme la connexion à la base de données.

Le répertoire contrôle :la logique de notre application et la colle entre modèles et vues. Ici, nous appelons nos modèles pour obtenir les données, puis nous mettons ces données sur nos vues à envoyer aux utilisateurs .

Le répertoire Modèle : la partie de notre application qui traitera de la base de données ou de toute fonctionnalité liée aux données.

Le répertoire Middleware : contient les scripts qui permet d'effectuer des vérifications comme l'authentification, l'autorisation (les privilèges) .

Le répertoire routes : contient le fichier index.js qui gère le routage de l'application .

Le script server.js :contient les informations nécessaires du lancement de serveur du backend comme le numéro de port choisi .

Le script package.js : c'est un fichier JS qui stocke les informations liée au projet, les dépendances ,les Devdependencies(les dépendances de développement du projet),la Licence du projet, les scripts de production , des tests ... et d'autres informations .

Node modules : c'est un fichier qui contient tous les modules utilisés lors de la création du notre projet.

2. **Frontend** : pour la partie Frontend, elle est structurée comme suit :

Public : comporte le fichier index.html .

index.html :c'est le fichier html qu'on vas appliquer sur app.vue.

Src : se compose des sous-dossiers :

Assets : regroupe le dossier css pour les styles ,et webfonts pour les fonts.

Components :regroupe les différents composants du l'application.

Router : contient le fichier index.js qui gère le routage de l'application .

Views : regroupe l'ensemble des vues (les interfaces graphiques).

Main.js :c'est le fichier principale de la partie fontend .

Store : c'est le state management de vue, permet de partager des données entre des différents composants au même temps, cela est garantie grâce à l'utilisation du Vuex .

App.vue : représente notre application .

Vue.config.js : c'est un fichier de configuration .

Node-modules : c'est un fichier qui contient tous les modules utilisés lors du développement de la partie front .

4.5 Schéma physique de la base de données

L'implémentation de notre base de données est faite avec l'utilisation d'un outil dans Wamp « PhpMyAdmin » qui nous permet de schématiser les tables avec les relations . Dans Le schéma physique, les tables sont représentés avec les types des attributs ,les clés primaires et les champs référencés.

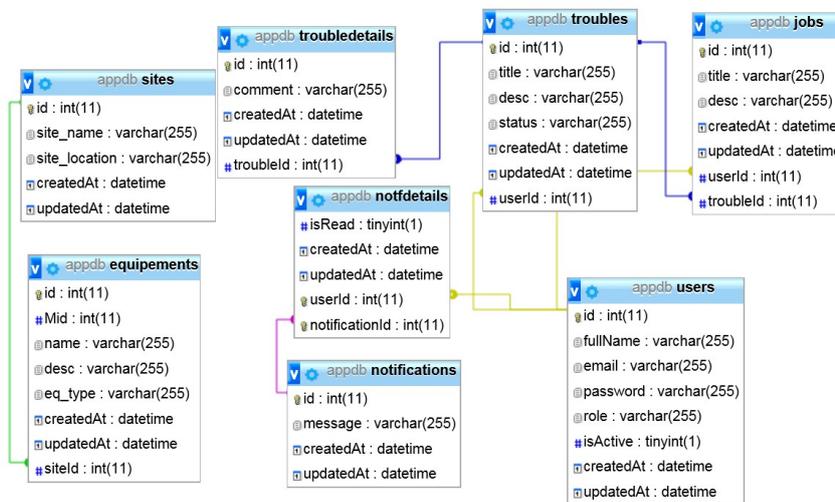


FIGURE 4.1 – Schéma physique de la base de données

4.6 Diagramme de Déploiement

la figure suivante présente le diagramme de déploiement de notre application.

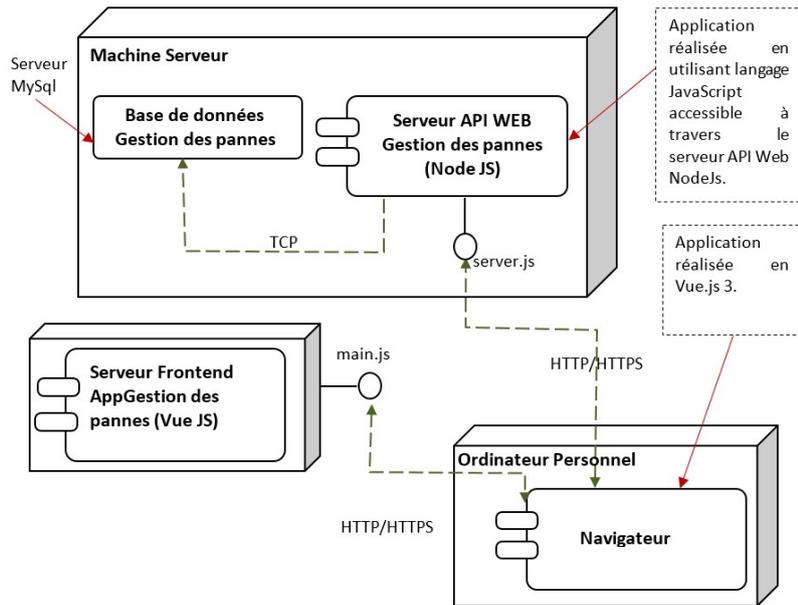


FIGURE 4.2 – Diagramme de déploiement

4.7 Captures d’écrans

Dans cette section, nous allons présenter quelques vues de l’interface Homme-Machine de notre application.

4.7.1 L’interface Homme-Machine « Authentification »

L’accès des utilisateurs (ingénieur, opérateur, Administrateur) à l’application nécessite l’authentification, en saisissons leur Emails et mot de passe.



FIGURE 4.3 – Interface Homme-Machine de L’authentification .

4.7.2 L'interface Homme-Machine « Accueil-Ingénieure »

Après l'authentification de l'ingénieur, il aura l'accès à la déclaration d'une panne ainsi que sa réparation.

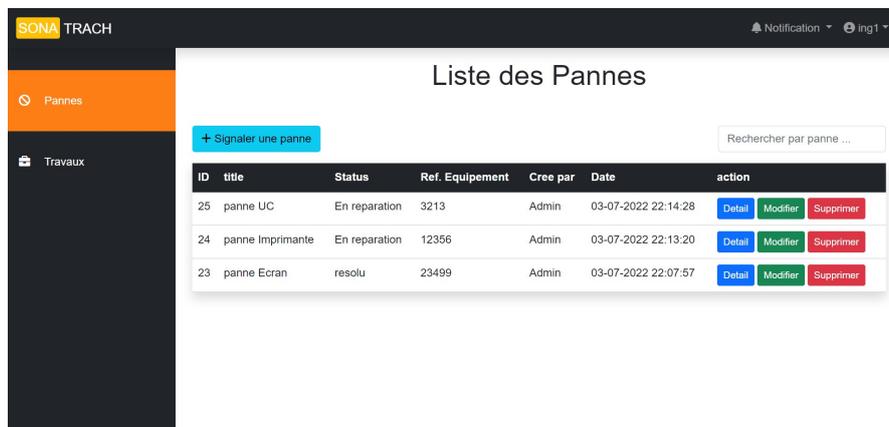


FIGURE 4.4 – Interface Homme-Machine de Accueil-Ingénieure .

4.7.3 L'interface Homme-Machine « Accueil-opérateur »

La page d'accueil correspond à l'opérateur, il peut déclarer une panne.

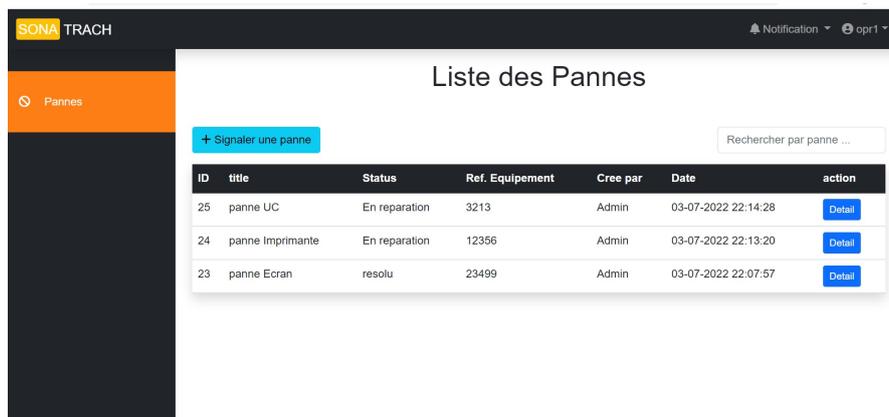


FIGURE 4.5 – Interface Homme-Machine de Accueil-opérateur .

4.7.4 L'interface Homme-Machine « Gestion des Sites »

Après l'authentification de l'administrateur, il aura l'accès à la gestion des Sites, l'ajout d'un site, la modification d'un site existant ou sa suppression, un site peut être recherché dans le champ de recherche, nous ne tapons que les premières lettres du nom du site .

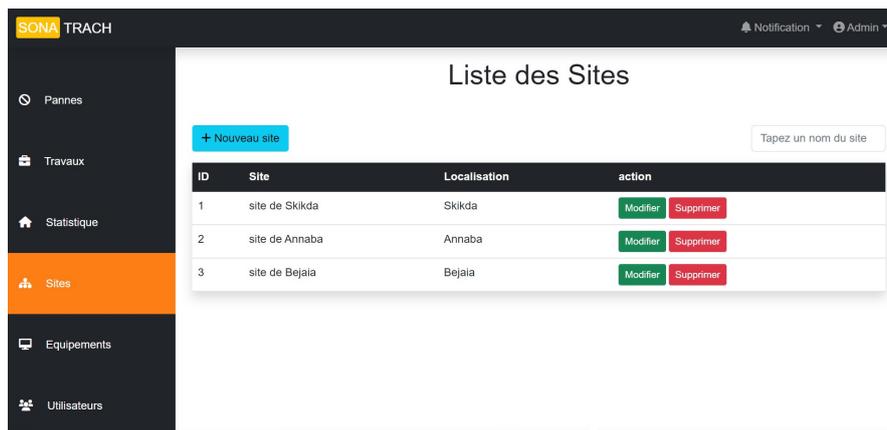


FIGURE 4.6 – L’interface Homme-Machine « Gestion des sites ».

4.7.5 L’interface Homme-Machine « Gestion des Equipements »

Après l’authentification de l’administrateur, il aura l’accès à la gestion des équipements, l’ajout d’un nouvel équipement, la modification d’un équipement existant ou sa suppression, un équipement peut être recherché dans le champ de recherche, nous ne tapons que les premières lettres du nom de l’équipement .

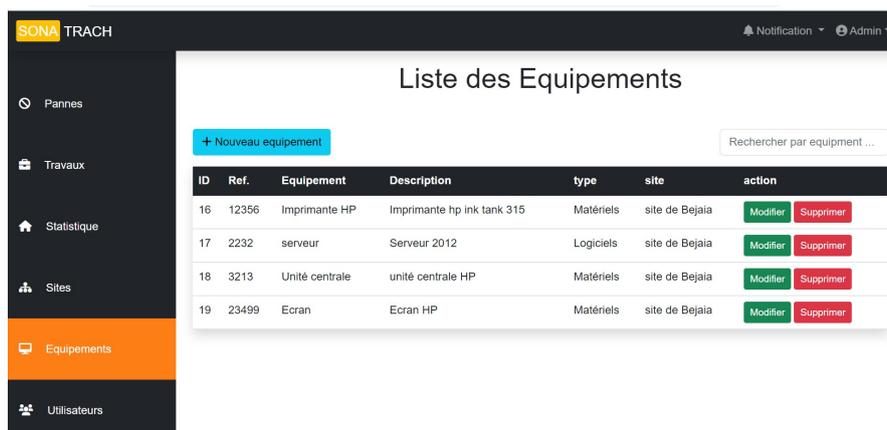


FIGURE 4.7 – L’interface Homme-Machine « Gestion des Equipements ».

4.7.6 L’interface Homme-Machine « Gestion des utilisateurs »

Après que l’administrateur sera authentifié, il aura l’accès à la gestion des utilisateurs, d’ajouter un nouvel utilisateur ou d’effectuer des modifications sur les informations personnelles ou modifier son rôle, ainsi que la désactivation de l’utilisateur qui signifie une suppression physique du compte d’un utilisateur. Il aura l’accès à rechercher un utilisateur en tapons sur la zone de recherche les premières lettres du nom de l’utilisateur .

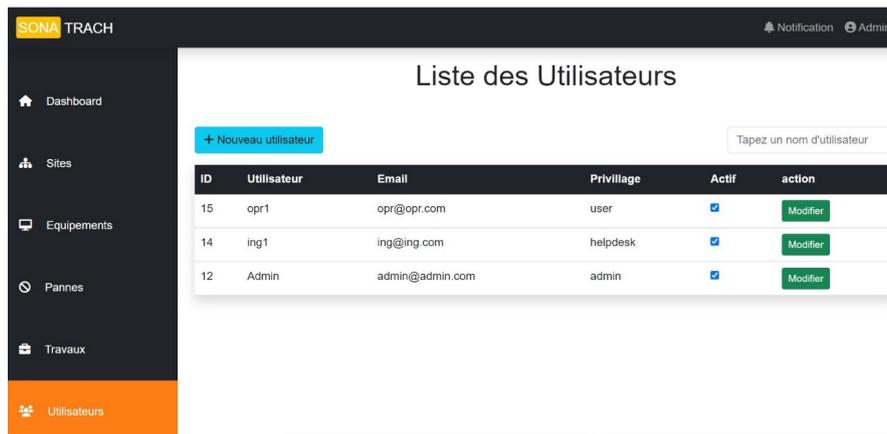


FIGURE 4.8 – L’interface Homme-Machine « Gestion des utilisateurs ».

4.7.7 L’interface Homme-Machine « Gestion des pannes »

Après l’authentification de l’utilisateur(ici c’est l’administrateur), Dans ca page d’accueil il aura l’accès à consulter la liste des pannes (avec la possibilité d’ajouter des détails, modification et la suppression de la panne) ainsi que la signalisation d’une nouvelle panne, comme il peut effectuer des recherches en tapons les premières lettres du titre de la panne sur la zone de recherche,comme il peut aussi ajouter des détails pour une panne déjà déclarée .

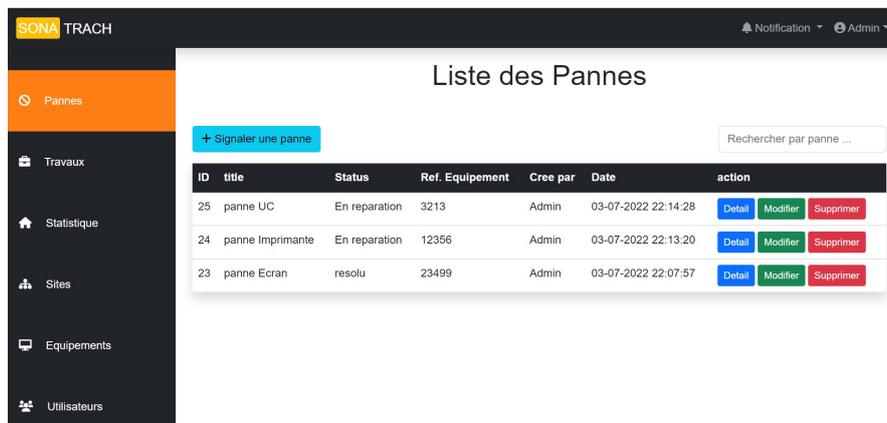


FIGURE 4.9 – L’interface Homme-Machine « Panne ».

4.7.8 L’interface Homme-Machine « Gestion des Réparations »

Après l’authentification de l’utilisateur(l’ingénieur ou l’administrateur), Dans sa page d’accueil il aura l’accès à consulter la liste des traveaux des pannes concernées (avec la possibilité de modification de l’état du travail,ou modification des informations des traveaux) ainsi qu’il peut effectuer des recherches en tapons les premières lettres du titre du travail dans la zone de recherche .

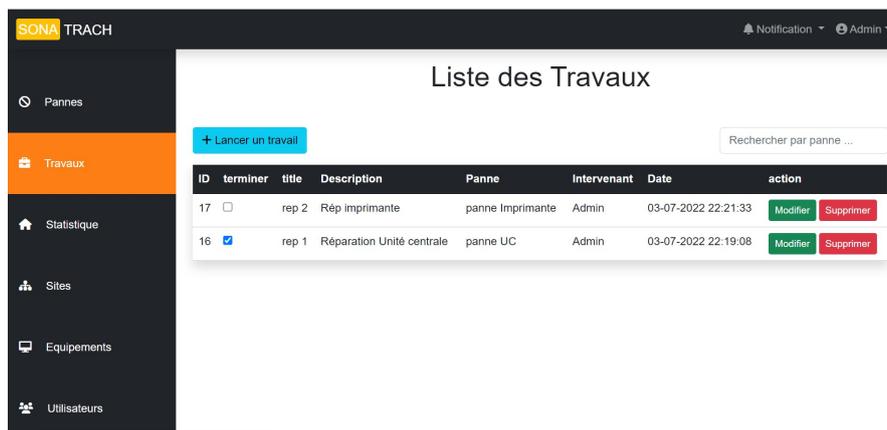


FIGURE 4.10 – L’interface Homme-Machine « Gestion des Réparations »

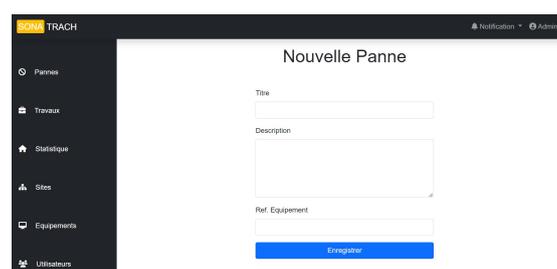
4.8 Un scénario d’exécution

Nous allons présenter un scénario d’exécution pour le cas d’utilisation « Déclarer une panne », et l’acteur choisi est L’administrateur

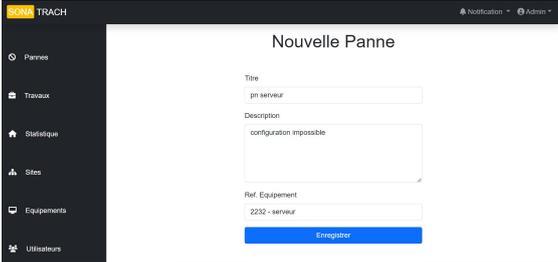
Après que L’administrateur sera authentifié , il aura l’accès à cette interface :



pour qu’il déclare une panne ,il suffit qu’il clique sur le bouton « signaler une panne »



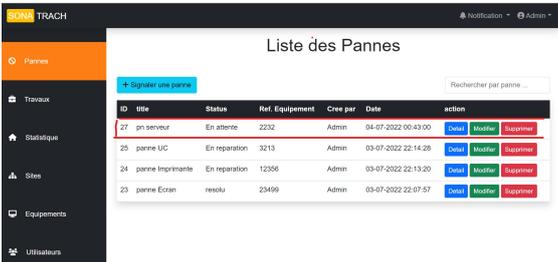
Après le remplissage des champs ,il doit cliquer sur le bouton Enregistrer pour la validation de la déclaration de la panne.



The screenshot shows a web interface for reporting a new fault. The title is "Nouvelle Panne". The form contains the following data:

Field	Value
Titre	pn serveur
Description	configuration impossible
Ref. Equipment	2232 - serveur

La panne est bien déclarée avec la possibilité d'ajouter des détails ou de modification de la déclaration en cas d'erreur ou de la suppression de la panne.



The screenshot shows a table listing reported faults. The first row is highlighted in red.

ID	Titre	Status	Ref. Equipment	Creé par	Date	action
27	pn serveur	En attente	2232	Admin	04-07-2022 00:43:00	Details Modifier Supprimer
25	panne UC	En reparation	3213	Admin	03-07-2022 22:14:28	Details Modifier Supprimer
24	panne Imprimante	En reparation	12356	Admin	03-07-2022 22:13:20	Details Modifier Supprimer
23	panne Ecran	resolu	23499	Admin	03-07-2022 22:47:57	Details Modifier Supprimer

4.9 Conclusion

Dans ce chapitre nous avons déclarés les aspects techniques liés à la réalisation de ce projet, la présentation des différents langages de programmation utilisés pour développer le projet, le diagramme de déploiement et enfin quelque interfaces Homme-machine.

CONCLUSION GÉNÉRALE

Aujourd'hui l'informatique est partout, Que ce soit dans les magasins, les distributeurs, en passant par les stations-services, les écoles, les bibliothèques, les aéroports, ou au travail, plus rien ne fonctionne sans elle.

L'environnement concurrentiel dans lequel évoluent les entreprises les oblige à se renouveler en permanence et à investir dans les nouvelles technologies. C'est pourquoi la technologie est de plus en plus présente dans les entreprises, Aujourd'hui, vous devez vous adapter à la technologie et être aussi à jour que possible. Pour cela, ces dernières doivent informatiser leurs systèmes d'informations et aussi le système informatique, et maintenir fonctionnel, continuellement, les différentes applications informatiques.

Dans ce cadre, la bonne gestion des pannes en général, et des pannes informatiques en particulier, contribue à la pérennité des solutions informatiques installées dans l'entreprise. Et ceci fait partie des objectifs de notre projet, qui consiste à concevoir et à réaliser une application de gestion des pannes pour l'entreprise SONATRACH.

Pour atteindre les objectifs fixés dans notre travail, nous avons suivi une démarche dans la réalisation qui a été comme suit :

Premièrement, c'était la collecte de maximum de données concernant la déclaration et la prise en charge des pannes informatiques du service STS de l'entreprise SONATRACH, comprendre la problématique de cette gestion qui est liée principalement au travail purement manuel, sauf quelques documents MS-Word, ce qui rend l'accès à l'information (Rechercher et filtre, traçabilité, état de sortie, . . .) très difficile, voire même impossible. Nous avons aussi décrit la solution proposée ainsi que le recueil des besoins fonctionnels et techniques que cette solution doit vérifier.

Ensuite, c'était le passage à la phase de la spécification et d'analyse des besoins qui nous a permis de distinguer les différents acteurs qui vont interagir avec l'application visée, leurs différents cas d'utilisations qui sont décrits textuellement sous format tabulaire ainsi que le diagramme de séquence système.

L'étape suivante est la conception détaillée de notre solution, dans laquelle les diagrammes de séquence issues de la phase précédente sont détaillés par des diagrammes d'interaction où le système est remplacé par les différents objets (Interface, Contrôle et Entité) en interaction pour réaliser le cas d'utilisation concerné. Les objets entités avec les règles de gestion nous ont permis d'établir le diagramme de classe, qui est la base de schéma relationnel en appliquant un ensemble de règles de passage.

En dernier lieu, c'est la phase de réalisation du projet, dans laquelle nous avons présenté les différents outils et langages utilisés dans notre projet, qui s'agit d'une application Web avec des technologies liées principalement au langage Javascript (Full-stack : Node-js et Vue Js). Par la suite, nous avons décrit la structure du code source coté serveur (Backend) et coté client (Frontend). Ceci, nous a permis de séparer la logique Client et celle du Serveur, et d'utiliser le rendu coté client (Gestion d'affichage est au niveau du navigateur). Ceci rend notre application facilement extensible à d'autres solutions de bureau et mobile. Enfin de cette phase, nous avons présenté quelques interfaces de notre application ainsi qu'un scénario d'exécution du cas d'utilisation « Déclarer une panne ».

Bien évidemment, et comme tout travail humain, des lacunes et manques de notre projet peuvent être remédiées dans des futures projets, pour cela, nous traçons les perspectives suivantes :

- Généraliser notre solution pour d'autres services de SONATRACH en prenant en considération d'autres types de matériel et leurs pannes.
- Ajouter l'étiquetage du matériel par des codes à bar ou QR-code.
- Etendre notre travail par une application mobile pour rendre efficace et rapide les notifications de pannes et leurs prises en charge.

BIBLIOGRAPHIE

- [1] Bootstrap. *site web url : JDN*<https://www.journaldunet.com>.*Date de consultation : Mars/2022.*
- [2] Le campus numérique des iut, iutenligne.net, uml. *url*<https://prive.iutenligne.net/>.*Date de consultation : Juin/2022.*
- [3] Mysql. *url :https://www.mysql.com/fr/*. *Date de consultation : Mars/2022.*
- [4] Nodejs :exemples d'utilisation avec l'orm sequelize. <https://numa-bord.com>.*Date de consultation : Mai/2022.*
- [5] Visual studio code. *site web url :https://framalibre.org/*. *Date de consultation : Mars/2022.*
- [6] Vuejs ,le site officiel du vuejs. *url : https://vuejs.org/*.*Date de consultation : Mars/2022.*
- [7] W3schools online web. *url :https://www.w3schools.com*.*Date de consultation : Mars/2022.*
- [8] Roza Anouche and Dihia Beggache. Conception et développement d'une application web basée sur les frameworks react js et nodejs pour le suivi du personnel dans un chantier. cas : Sarl gts développement. 2020.
- [9] Laurant AUDIBERT. *UML 2 - De l'apprentissage à la pratique*. <https://laurent-audibert.developpez.com/Cours-UML/?page=mise-en-oeuvreuml>. Publié le 31 octobre 2006, 12/01/2009.
- [10] Romain Bourdon. Site officiel du wampserver. *url :https://www.wampserver.com/*.*Date de consultation : Mars/2022.*
- [11] Hassan Djirdeh. *Fullstach vue 3, Express.js,MongoDB*. 2021.
- [12] Fernando Doglio. *Pro REST API Development with Node. js*. Apress, 2015.
- [13] KEZIZ KAHOUADJI. Conception et développement d'une application web de gestion des pannes, cas d'étude :naftal. 2019.
- [14] NodeJS. Express. *url : https://nodejs.org/*. *Date de consultation : Mars/2022.*
- [15] Eric Sarrion. *programmation avec Node. js, Express.js,MongoDB*. 09/2014.
- [16] Romain willmann. *Apprendre à créer des appliactions web avec javascript*. la tipografica vearese Srl -Italie, 2017.

A.1 Description

C'est une méthode simple et générique qui se situe à mi-chemin entre UP (Unified Process) qui constitue un cadre générale très complet de processus de développement, et XP (Xtreme Programming) qui est une approche minimaliste à la mode centrée sur le code.

A.1.1 Présentation du processus unifié (UP)

UP (Unified Process) est une méthode générique de développement de logiciels. Générique signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation (exemple : R.UP ou X.UP). C'est plus ou moins vrai pour toute méthode, qu'elle se définisse elle-même comme générique ou pas.

. Le processus unifié : un cadre général

Le processus unifié est une démarche de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. Ses principales caractéristiques sont :

- Il est à base de composants .
- Il utilise le langage UML (ensemble d'outils et de diagrammes).
- Il est centré sur l'architecture .
- Il est itératif et incrémental .

Le processus unifié est piloté par les cas d'utilisation

L'objectif principal d'un système logiciel est de rendre service à ses utilisateurs ; il faut par conséquent bien comprendre les désirs et les besoins des futurs utilisateurs. Le processus de développement sera donc centré sur l'utilisateur. Le terme utilisateur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. L'utilisateur représente donc une personne ou une chose dialoguant avec le système en cours de développement.

Les cas d'utilisation font apparaître les besoins fonctionnels et leur ensemble constitue le modèle des cas d'utilisation qui décrit les fonctionnalités complètes du système. Les cas d'utilisation ne sont pas un simple outil de spécification des besoins du système. Ils vont complètement guider le processus de développement à travers l'utilisation de modèles basés sur l'utilisation du langage UML.

Le processus unifié est centré sur l'architecture

Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place. L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit. L'architecture logicielle équivaut aux aspects statiques et dynamiques les plus significatifs du système. L'architecture émerge des besoins de l'entreprise, tels qu'ils sont exprimés par les utilisateurs et autres intervenants et tels qu'ils sont reflétés par les cas d'utilisation.

Le processus unifié est itératif et incrémental

Le développement d'un produit logiciel destiné à la commercialisation est une vaste entreprise qui peut s'étendre sur plusieurs mois. On ne va pas tout développer d'un coup. On peut découper le travail en plusieurs parties qui sont autant de mini projets, chacun d'entre eux représentant une itération qui donne lieu à un incrément.

Une itération désigne la succession des étapes de l'enchaînement d'activités, tandis qu'un incrément correspond à une avancée dans les différents stades de développement.

A.1.2 Le cycle de vie du processus unifié

Le processus unifié répète un certain nombre de fois une série de cycles. Tout cycle se conclut par la livraison d'une version du produit aux clients et s'articule en 4 phases : création, élaboration, construction et transition, chacune d'entre elles se subdivisant à son tour en itérations.

1. **Création** Première phase du cycle de vie du processus unifié, la création traduit une idée en vision de produit fini et présente l'étude de rentabilité pour ce produit. Elle essaie de répondre à un certain nombre de questions : Que va faire le système pour les utilisateurs ? A quoi peut ressembler l'architecture d'un tel système ? Quels sont l'organisation et les coûts du développement de ce produit ? C'est à ce niveau où les principaux cas d'utilisation seront spécifiés. L'identification des risques majeurs, la mise sur place d'une architecture provisoire du système à concevoir et la préparation de la phase d'élaboration seront les principales tâches à effectuer durant cette étape de la création.
2. **Elaboration** Elle permet de préciser la plupart des cas d'utilisation et de concevoir l'architecture du système. L'architecture doit être exprimée sous forme de vue de chacun des modèles. Lors de cette phase une architecture de référence sera conçue. Au terme de

cette étape, le chef de projet doit être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet.

3. Construction C'est le moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet, elle est maintenant stable. Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version. Celle-ci doit encore avoir des anomalies qui peuvent être en partie résolues lors de la phase de transition.
4. Transition Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la fabrication, la formation des utilisateurs clients, la mise en oeuvre d'un service d'assistance et la correction des anomalies constatées (ou le report de leur correction à la version suivante).

A.2 Présentation de l'eXtreme Programming (XP)

L'eXtreme Programming ou XP, est une méthode de développement de projet mise au point à la fin des années 90 par Kent Beck, Ward Cunningham et Ron Jeffries.

XP doit son nom au fait qu'elle place l'activité de programmation au centre du projet, et qu'elle obtient ses résultats en combinant et en poussant à l'extrême certaines pratiques de développement. XP est un ensemble de pratiques qui couvre une grande partie des activités de la réalisation d'un logiciel, de la programmation proprement dite à la planification du projet, en passant par l'organisation de l'équipe de développement et les échanges avec le client. Ces pratiques n'ont en soi rien de révolutionnaire : il s'agit simplement de pratiques de bon sens, mises en oeuvre par des développeurs ou des chefs de projet expérimentés, telles que :

- Un utilisateur à plein-temps dans la salle projet. Ceci permet une communication intensive et permanente entre les clients et les développeurs, aussi bien pour l'expression des besoins que pour la validation des livraisons.
- Écrire le test unitaire avant le code qu'il doit tester, afin d'être certain que le test sera systématiquement écrit et non pas négligé.
- Programmer en binôme, afin d'homogénéiser la connaissance du système au sein des développeurs, ainsi que de permettre aux débutants d'apprendre des experts. Le code devient ainsi une propriété collective et non individuelle que tous les développeurs ont le droit de modifier.
- Intégrer de façon continue, pour ne pas retarder à la fin du projet le risque majeur de l'intégration des modules logiciels écrits par des équipes ou des personnes différentes.

XP s'appuie sur les valeurs suivantes :

Communication : La communication est la composante fondamentale de tout travail en équipe. XP installe la communication dans le projet à tous les niveaux.

Simplicité : Communication et retour d'information : ces dispositions sont menacées, à mesure que le produit évolue et s'accroît, par la lourdeur et l'inertie du processus. Pour cette raison, XP érige la simplicité en véritable discipline, tant au niveau de la conception que du processus lui-même (scénarios utilisateur, séances de planifications, propriété collective du code, pas de

spécialisations techniques).

Feedback : Le retour d'information permet de tracer et d'ajuster le processus en vue d'améliorer la qualité, la maintenance et la productivité. A chacune des tâches de production, XP câble un mécanisme ou une pratique permettant de valider cette production de manière quasiment continue.

Courage : La force d'une équipe même dotée d'une approche efficace ne se mesure pas tant à ce qu'elle fait lorsque tout va bien, qu'à la manière dont elle affronte les difficultés. Pour cette raison XP valorise également le Courage.

Elle est bien adaptée pour des projets de taille moyenne où le contexte (besoins utilisateurs, technologies informatiques) évolue en permanence .

A.3 La démarche simplifiée

La problématique que pose la mise en oeuvre d'UML est simple : comment passer de l'expression des besoins au code de l'application ? Comment obtenir le plus efficacement possible un code informatique opérationnel, complet, testé, et qui réponde parfaitement aux besoins exprimés par les utilisateurs ?

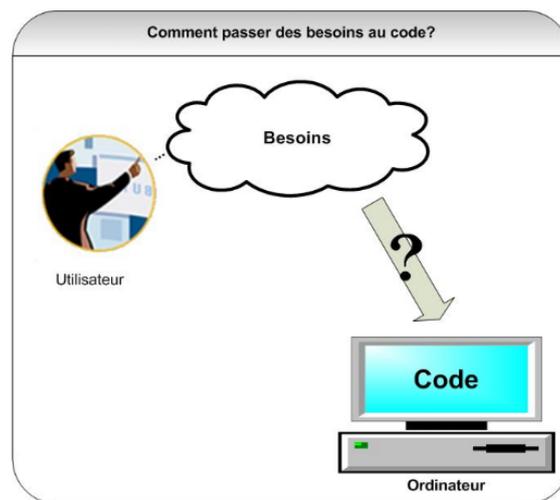


FIGURE A.1 – Démarche simplifiée

Elle a donc montré son efficacité dans la pratique et elle est :

- conduite par les cas d'utilisation, comme UP, mais bien plus simple .
- relativement légère et restreinte, comme XP, mais sans négliger les activités de modélisation en analyse et conception .
- fondée sur l'utilisation d'un sous-ensemble nécessaire et suffisant du langage UML (modéliser 80 percent des problèmes en utilisant 20 percent d'UML) .

Dans tous les cas, il faut garder à l'esprit qu'une méthode n'est pas une formule magique. Le fait de produire des diagrammes UML selon un ordre établi n'est en aucun cas une garantie de réussite. Une méthode ne sert qu'à canaliser et ordonner les étapes de la modélisation. La valeur n'est pas dans la méthode, mais dans les personnes qui la mettent en oeuvre.

ANNEXE B

DESCRIPTION DES CAS D'UTILISATION ET LES DIAGRAMMES DE SÉQUENCE

B.1 Description textuelle des cas d'utilisation

B.1.1 Description de cas d'utilisation « Consulter le rapport mensuel »

Le tableau ci-dessous représente la description des cas d'utilisation « Consulter le rapport mensuel » :

Cas d'utilisation « Consulter le rapport mensuel »	
objectif	Ce cas d'utilisation permet à l'administrateur et à l'ingénieur de consulter le rapport mensuel.
acteurs	Administrateur- Ingénieur
Description	L'utilisateur doit d'abord s'authentifier puis il demande l'affichage du rapport mensuel
Pré condition	L'authentification
Scénario nominal	Ce cas d'utilisation est déclenché lorsque l'Administrateur ou l'ingénieur veut voir le taux des pannes au fonction des équipements : 1-l'acteur demande au système de consulter le rapport mensuel. 2- le système affiche une fiche de consultation. 3-l'acteur sélectionne une période. 4-le système affiche les informations de la situation sélectionnée.
Scénario alternatif	/
Scénario d'exception	/

TABLE B.1 – Description de cas d'utilisation « Consulter le rapport mensuel

Description de cas d'utilisation « Réparer une panne »

Le tableau ci-dessous représente la description des cas d'utilisation « Réparer une panne » :

Cas d'utilisation «Réparer une panne»	
objectif	Ce cas d'utilisation a pour objectif de déclarer la réparation d'une panne.
acteurs	L'administrateur - L'ingénieur
Description	Ce cas d'utilisation permet de déclarer la réparation d'une panne lorsque la tâche est exécutée.
Pré-condition	L'authentification
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsque l'administrateur ou l'ingénieur demande la déclaration de la réparation de la panne :</p> <ol style="list-style-type: none"> 1-L'ingénieur ou l'administrateur demande la liste des pannes. 2- Le système renvoie la page demandée. 3- L'ingénieur sélectionne une panne et demande la déclaration de sa réparation. 4- Le système renvoie la page demandé. 5-L'utilisateur remplit les champs nécessaires. <p>4-Le système vérifie les informations saisies :</p> <ul style="list-style-type: none"> -Si les champs obligatoires ne sont pas renseignés « Exception 1 » sera générée. -Sinon le système sauvegarde les informations saisies et affiche la confirmation de la déclaration.
Scénario alternatif	/
Scénario d'exception	Exception 1 : Le système affiche une erreur « Les champs doivent être renseignés ».

TABLE B.2 – Description du cas «Réparer une panne»

B.1.2 Description de cas d'utilisation « Lister les équipements »

Le tableau ci-dessous représente la description des cas d'utilisation «Lister les équipements» :

Cas d'utilisation «Lister les équipement»	
objectif	Ce cas d'utilisation permet à l'administrateur d'ajouter, modifier ou supprimer un équipement.
acteurs	Administrateur
Description	L'administrateur gère les équipements
Précondition	L'authentification
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsque l'Administrateur veut consulter les équipements existants avec la possibilité de l'ajout, modification ou la suppression.</p> <p>Ajouter un équipement :</p> <p>1-L'administrateur demande au système d'ajouter un équipement. 2-Le système renvoie la page demandée. 3-L'administrateur saisie les informations nécessaires d'un équipement et valide la création. 4-Le système vérifie la saisie :</p> <ul style="list-style-type: none"> -Si les champs sont incomplets «Exception 1 » sera générée -Sinon l'équipement sera crée et le système renvoie un message de confirmation de la création. <p>Modifier un équipement :</p> <ul style="list-style-type: none"> -L'administrateur accède à la modification des informations d'un équipement. -L'administrateur modifie les informations de l'équipement. -Si l'administrateur confirme la modification, l'équipement sera modifié. -Sinon la modification sera annulée. <p>Supprimer un équipement :</p> <p>1-L'administrateur demande au système de supprimer un équipement. 2-Le système renvoie la page demandée. 3-L'administrateur sélectionne un équipement et confirme la suppression. 4-Le système renvoie un avertissement de suppression. 5-L'administrateur confirme la suppression. 6-L'équipement sera supprimé et un message de confirmation sera envoyé.</p>
Scénario alternatif	/
Scénario d'exception	Exception 1 : Le système renvoie un message d'erreur indiquant que le remplissage des champs est obligatoire.

TABLE B.3 – Description du cas «lister les équipements»

B.1.3 Description de cas d'utilisation « Modifier un équipement »

Le tableau ci-dessous représente la description des cas d'utilisation « Modifier un équipement » :

Cas d'utilisation « Modifier un équipement »	
objectif	Ce cas d'utilisation permet à l'administrateur de modifier un équipement.
acteurs	Administrateur
Description	Après l'authentification l'admin aura l'accès de modifier les informations d'un équipement.
Précondition	L'authentification
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsque l'Administrateur accède à la modification des équipements :</p> <ol style="list-style-type: none"> 1-L'administrateur demande au système la liste de tous les équipements. 2-Le système renvoie la page demandée. 3- L'administrateur clique sur la modification d'un équipement. 4- Le système renvoie les informations de l'équipement concerné avec la possibilité de modification. 5- L'administrateur modifie les champs concernés et enregistre la modification et valider. <p>-Si les champs sont erronés « Exception 1 » sera générée. -Sinon le système enregistre les modifications et renvoie un message de confirmation.</p>
Scénario alternatif	/
Scénario d'exception	Exception 1 : Le système affiche une erreur « les champs saisis sont incorrectes ».

TABLE B.4 – Description de cas d'utilisation « Modifier un équipement »

B.1.4 Description de cas d'utilisation «Ajouter un équipement »

Le tableau ci-dessous représente la description des cas d'utilisation « Ajouter un équipement » :

Cas d'utilisation «Ajouter un équipement»	
objectif	Ce cas permet à l'administrateur d'ajouter un équipement.
acteurs	L'administrateur
Description	L'administrateur doit d'abord s'authentifier puis accéder à la page « gestion des équipements » Il peut consulter la liste des sites comme il peut choisir l'opération « ajouter un équipement ».
Pré-condition	L'authentification
Scénario nominal	Ce cas est pour le but de la gestion des équipements, on peut le résumer dans les lignes suivantes : 1-l'administrateur demande la page de gestion des équipements. 2- Le système renvoie la page demandée. 3-l'administrateur choisit la page « ajouter un équipement ». 4- le système renvoie la page demandée. 5-l'administrateur remplit les champs nécessaires. 6- le système vérifie les informations saisies : -Si l'administrateur a oublié de remplir un des champs « Exception1 » sera générée. -Dans le cas de succès, le système sauvegarde les informations saisies, et affiche un message de confirmation. -Si l'administrateur déclare un équipement qui existe déjà « Exception 2 » sera généré.
Scénario alternatif	/
Scénario d'exception	Exception 1 : Le système affiche une erreur à l'utilisateur lui indiquant l'obligation de la saisie du champ oublié. Exception 2 : Le système affiche une erreur à l'administrateur lui indiquant l'existence de site.

TABLE B.5 – Description de cas d'utilisation «Ajouter un équipement»

B.1.5 Description de cas d'utilisation « Supprimer un équipement »

Le tableau ci-dessous représente la description des cas d'utilisation «Supprimer un équipement » :

Cas d'utilisation «Supprimer un équipement»	
objectif	Ce cas permet à l'administrateur de Supprimer un équipement.
acteurs	L'administrateur
Description	Après l'authentification l'administrateur accède à a page de« gestion des équipements » pour supprimer un équipement.
Pré-condition	L'authentification
Scénario nominal	<p>Le cas « supprimer un équipement » est déclenché lorsque l'administrateur demande la suppression d'un équipement bien spécifié.</p> <p>1-l'administrateur demande au système la liste de tout les équipements.</p> <p>2- Le système renvoie la page demandée.</p> <p>3-l'administrateur sélectionne un équipement et clique sur « supprimer et valider ».</p> <p>4-Le système demande la confirmation de la suppression.</p> <p>-Si l'administrateur confirme la suppression,l'équipement sera supprimé et le système renvoie une confirmation sinon la suppression sera annulée.</p>
Scénario alternatif	/
Scénario d'exception	/

TABLE B.6 – Description du cas «Supprimer un équipement»

B.1.6 Description de cas d'utilisation « Modifier une panne déjà déclarée »

Le tableau ci-dessous représente la description des cas d'utilisation « Modifier une panne déjà déclarée » :

Cas d'utilisation «Modifier une panne déjà déclarée»	
objectif	Ce cas permet à l'utilisateur de modifier la déclaration d'une panne.
acteurs	Opérateur - L'ingénieur – L'administrateur.
Description	Après l'authentification les utilisateurs peuvent modifier une panne qui est déjà déclarée.
Pré-condition	L'authentification
Scénario nominal	<p>Ce cas est déclenché lorsque l'utilisateur accède à la page « déclaration de la panne » pour modifier sa description.</p> <p>1-l'utilisateur demande au système la page « déclaration de la panne ».</p> <p>2- Le système renvoie la page demandée.</p> <p>3-L'utilisateur choisit « Modification de la description ».</p> <p>4-Le système renvoie la page demandée.</p> <p>5-L'utilisateur remplit les champs et enregistrer.</p> <p>6-Le système confirme la modification.</p>
Scénario alternatif	/
Scénario d'exception	/

TABLE B.7 – Description du cas «Modifier une panne déjà déclarée»

B.2 Diagrammes séquences systèmes

B.2.1 Diagramme de séquence système du cas d'utilisation «Consulter le rapport mensuel»

Lorsque l'administrateur ou l'ingénieur veut consulter le rapport mensuel, celui-ci accède à la rubrique « consulter le rapport mensuel » Le système affiche une fiche de consultation, L'ingénieur ou l'administrateur sélectionne une période, Le système affiche les informations de la situation sélectionnée.

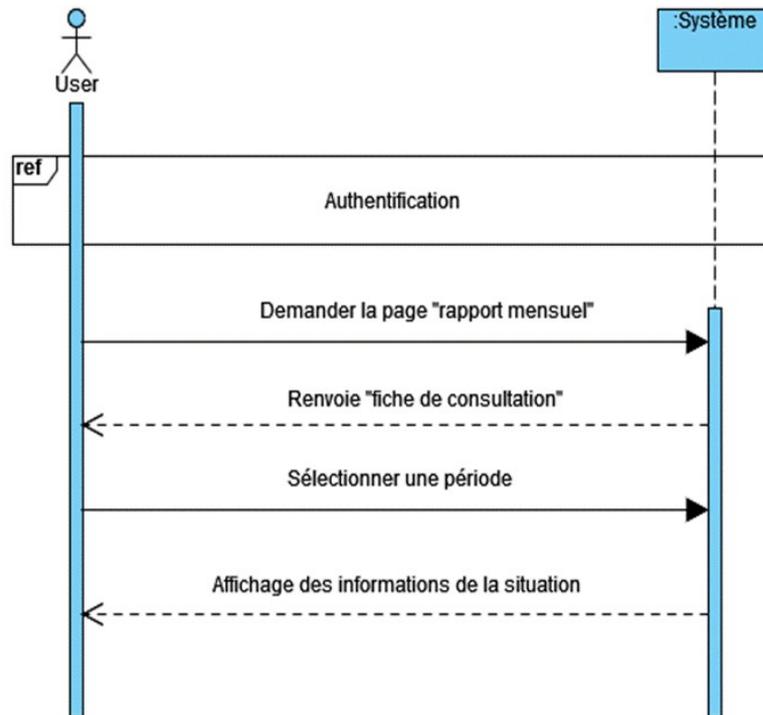


FIGURE B.1 – Diagramme de séquence système du cas d'utilisation «consulter le rapport mensuel».

B.2.2 Diagramme de séquence système du cas d'utilisation «Réparer une panne»

Lorsque l'ingénieur ou l'administrateur veut réparer une panne, celui-ci accède à la liste des pannes, il sélectionne la panne à réparer et clique sur « Réparer », puis remplit un formulaire et valide l'opération. Le système vérifie les champs saisis, dans le cas d'erreur, il le signale à l'ingénieur ou l'administrateur. Si non, les informations de la réparation seront enregistrées.

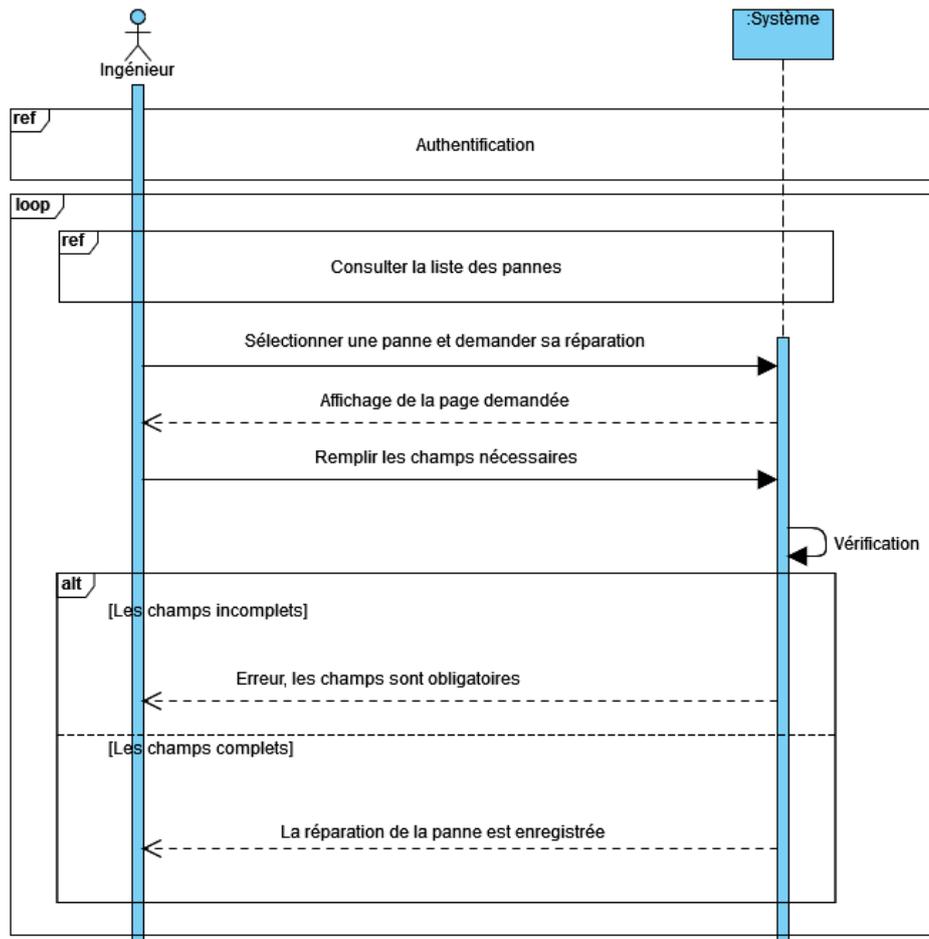


FIGURE B.2 – Diagramme de séquence système du cas d'utilisation «réparer une panne».

B.2.3 Diagramme de séquence système du cas d'utilisation «Lister les équipements»

L'administrateur consulte la liste des équipements, en ayant la possibilité de créer un équipement, de le modifier ou de le supprimer.

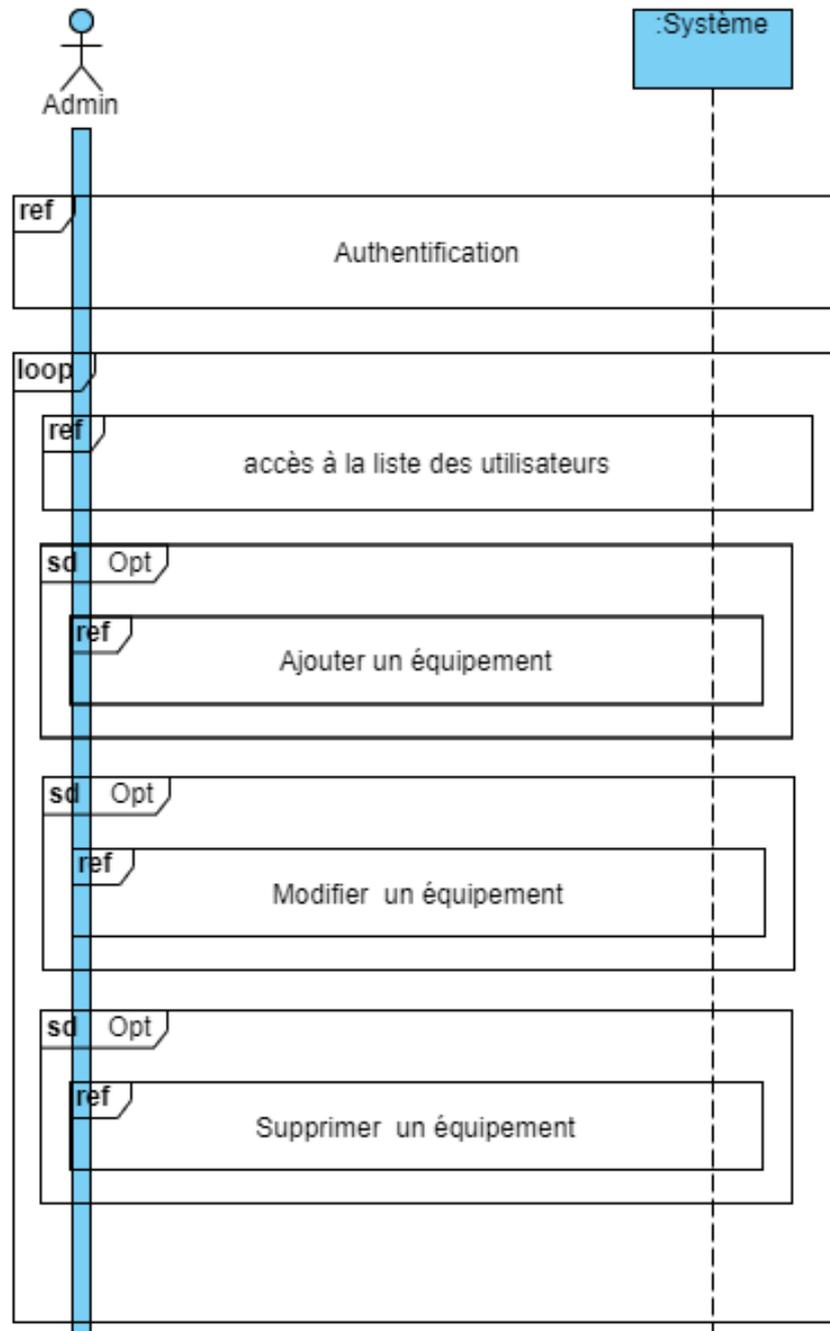


FIGURE B.3 – Diagramme de séquence système du cas d'utilisation «Lister les équipements».

B.2.4 Diagramme de séquence système du cas d'utilisation «modifier un équipement»

Lorsque l'administrateur veut modifier les informations d'un équipement, celui-ci accède à la liste des équipements, sélectionne l'équipement à modifier et clique sur « Modifier », puis remplit un formulaire de modification et valide l'opération. Le système vérifie les champs saisis, dans le cas d'erreur, il le signale à l'administrateur. Si non, les informations de l'équipement seront modifiées.

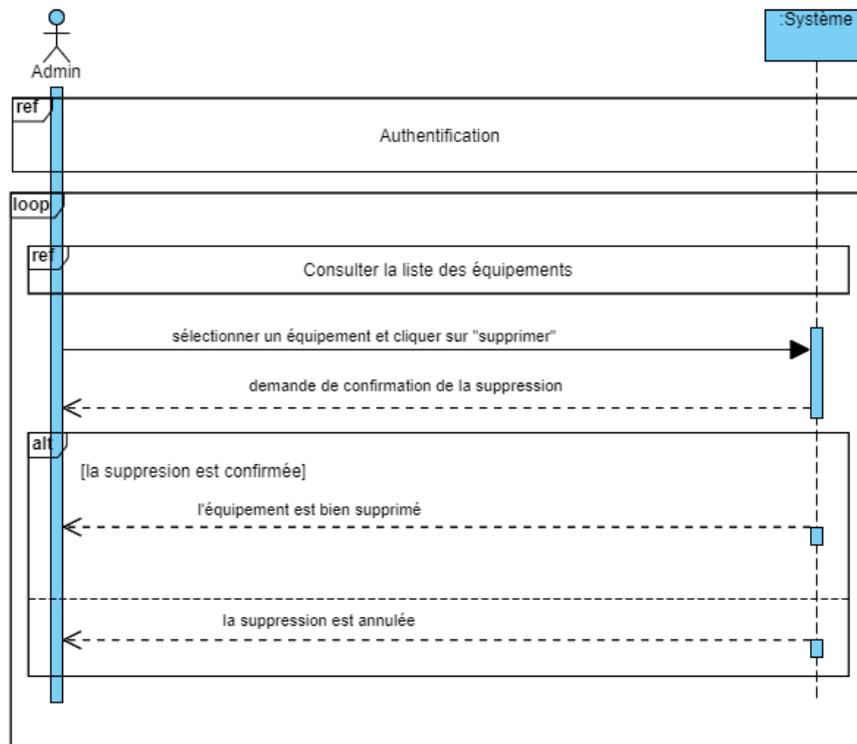


FIGURE B.4 – Diagramme de séquence système du cas d'utilisation «modifier un équipement».

B.2.5 Diagramme de séquence système du cas d'utilisation «Ajouter un équipement»

Pour ajouter un équipement, l'administrateur accède à la listes des équipements et clique sur « Ajouter » puis remplit un formulaire et valide l'opération. En cas d'erreur dans la saisie, le système lui signale une erreur. Sinon, il ajoute l'équipement.

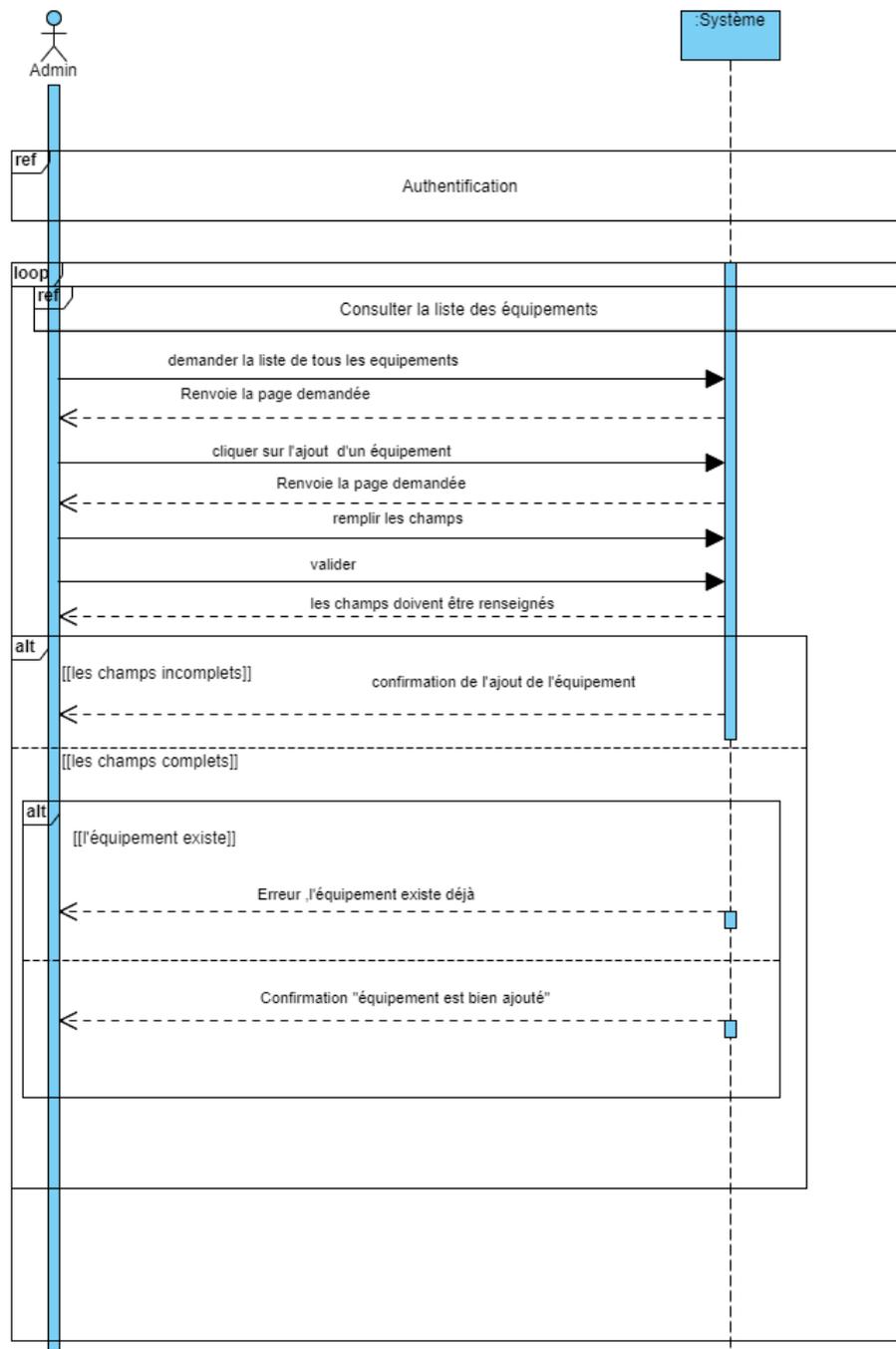


FIGURE B.5 – Diagramme de séquence système du cas d'utilisation «Ajouter un équipement».

B.2.6 Diagramme de séquence système du cas d'utilisation «supprimer un équipement»

Si l'administrateur décide de supprimer un équipement, il accède à la liste des équipements, puis il sélectionne l'équipement à supprimé et confirme la suppression.

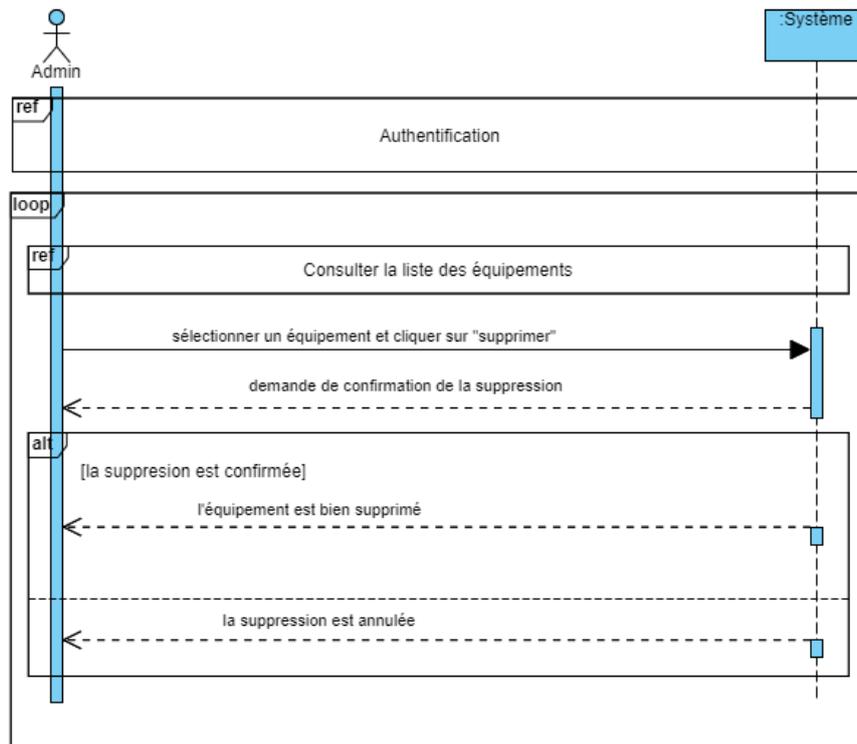


FIGURE B.6 – Diagramme de séquence système du cas d'utilisation «Supprimer un équipement».

B.2.7 Diagramme de séquence système du cas d'utilisation «modifier une panne déjà déclarée»

Lorsque l'administrateur veut modifier les informations d'une panne déjà déclarée, celui-ci accède à la liste des pannes, sélectionne la panne à modifier et clique sur « Modification de la description », puis il modifie la description et valide l'opération. Le système vérifie les champs saisis, dans le cas d'erreur, il le signale à l'administrateur. Si non, les informations de la description de la panne seront modifiées.

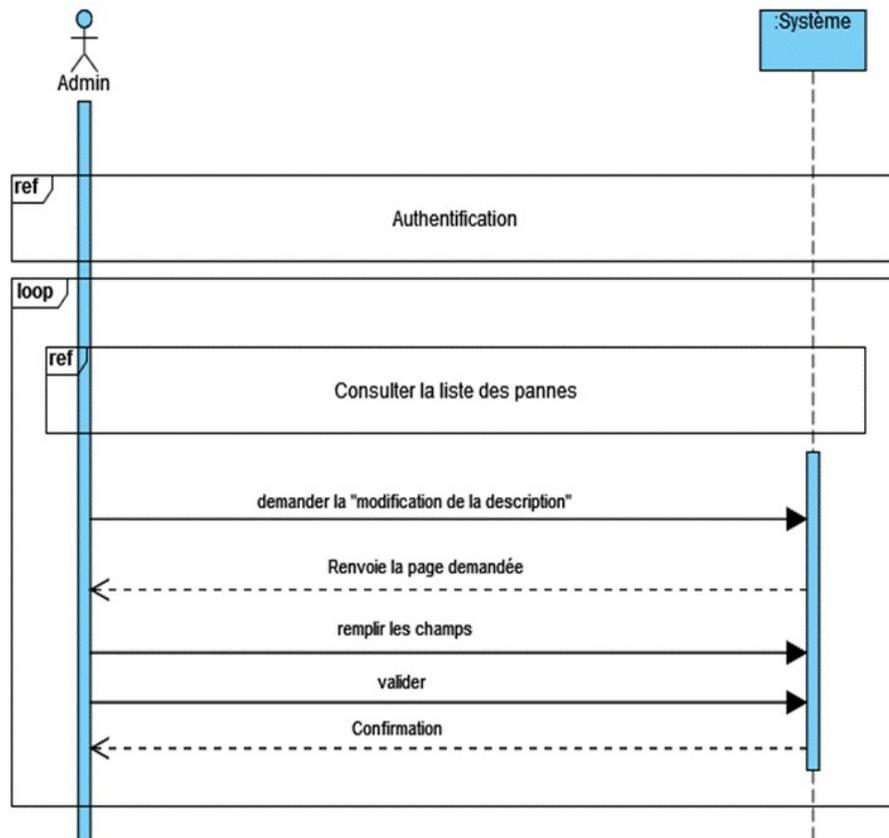


FIGURE B.7 – Diagramme de séquence système du cas d'utilisation «Diagramme de séquence système du cas d'utilisation «modifier une panne déjà déclarée».

ANNEXE C

LES DIAGRAMMES D'INTERACTIONS

C.1 DIAGRAMMES D'INTERACTION

C.1.1 Diagramme d'interaction de cas d'utilisation «Ajouter un équipement»

La figure suivante représente le diagramme d'interaction «ajouter un équipement» :

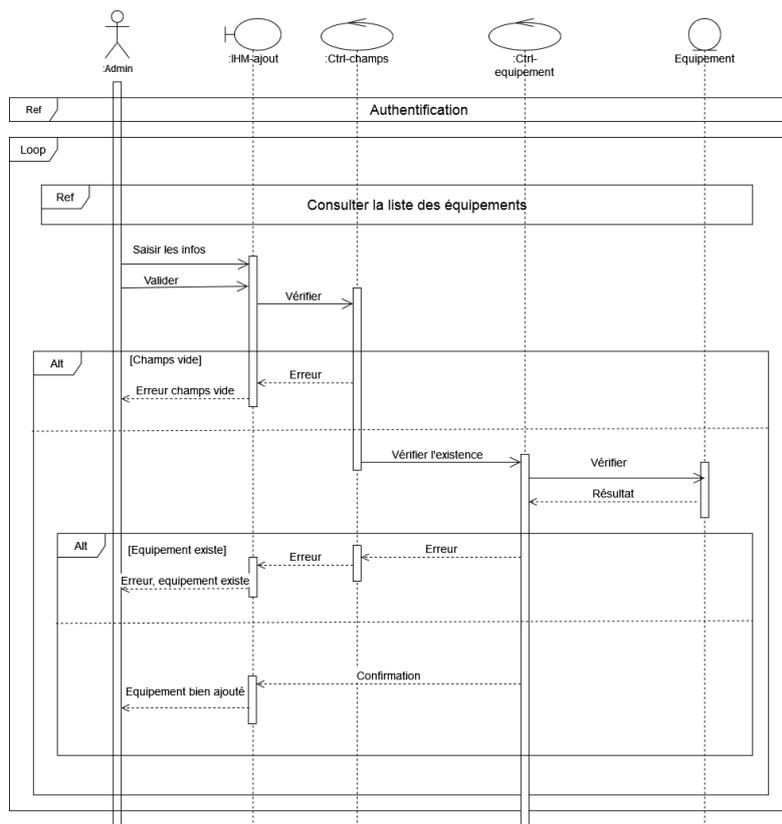


FIGURE C.1 – Diagramme d'interaction de cas d'utilisation «Ajouter un équipement»

C.1.2 Diagramme d'interaction de cas d'utilisation «Modifier un équipement»

La figure suivante représente le diagramme d'interaction «Modifier un équipement» :

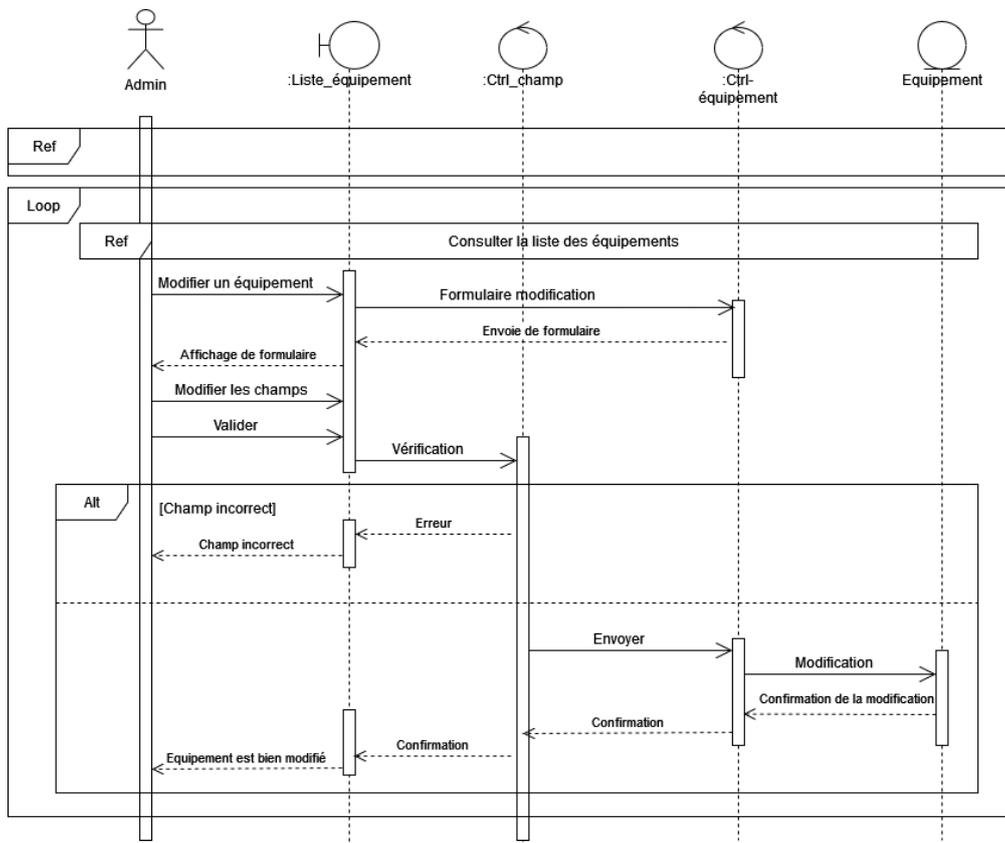


FIGURE C.2 – Diagramme d'interaction de cas d'utilisation «Modifier un équipement»

C.1.3 Diagramme d'interaction de cas d'utilisation «supprimer un équipement»

La figure suivante représente le diagramme d'interaction «supprimer un équipement» :

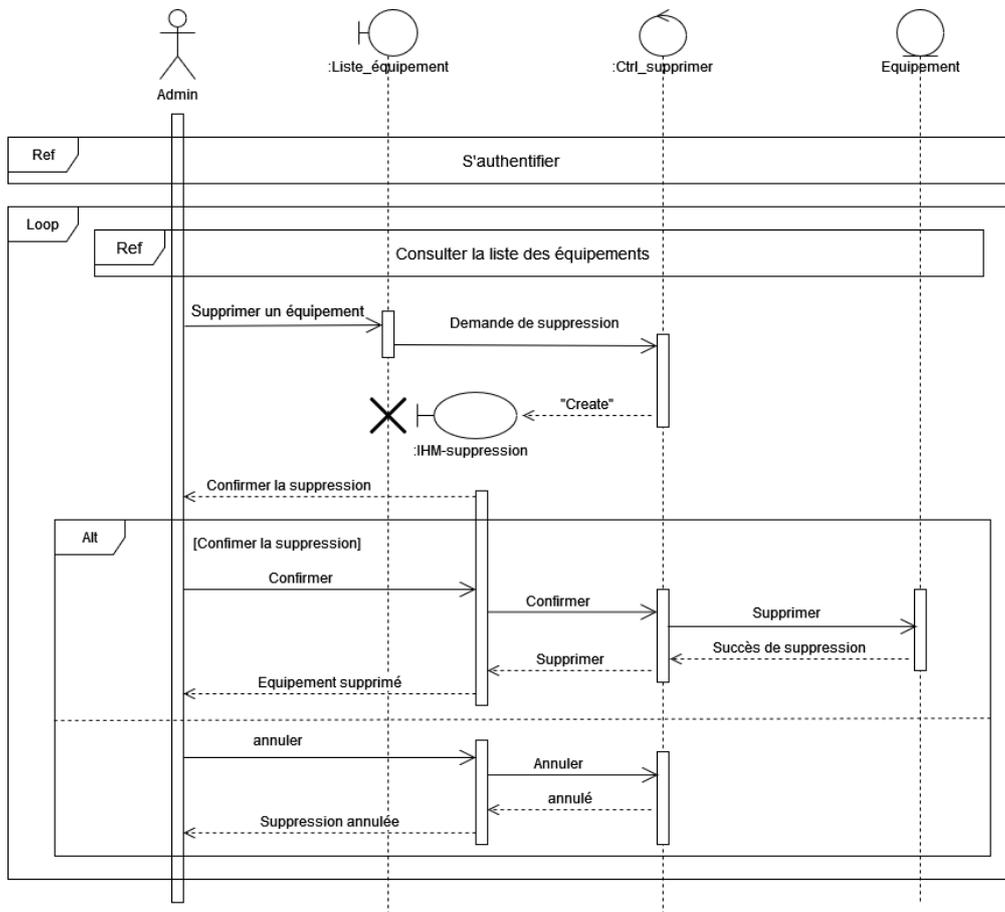


FIGURE C.3 – Diagramme d'interaction de cas d'utilisation «supprimer un équipement»

C.1.4 Diagramme d'interaction de cas d'utilisation «Réparer une panne»

La figure suivante représente le diagramme d'interaction «Réparer une panne» :

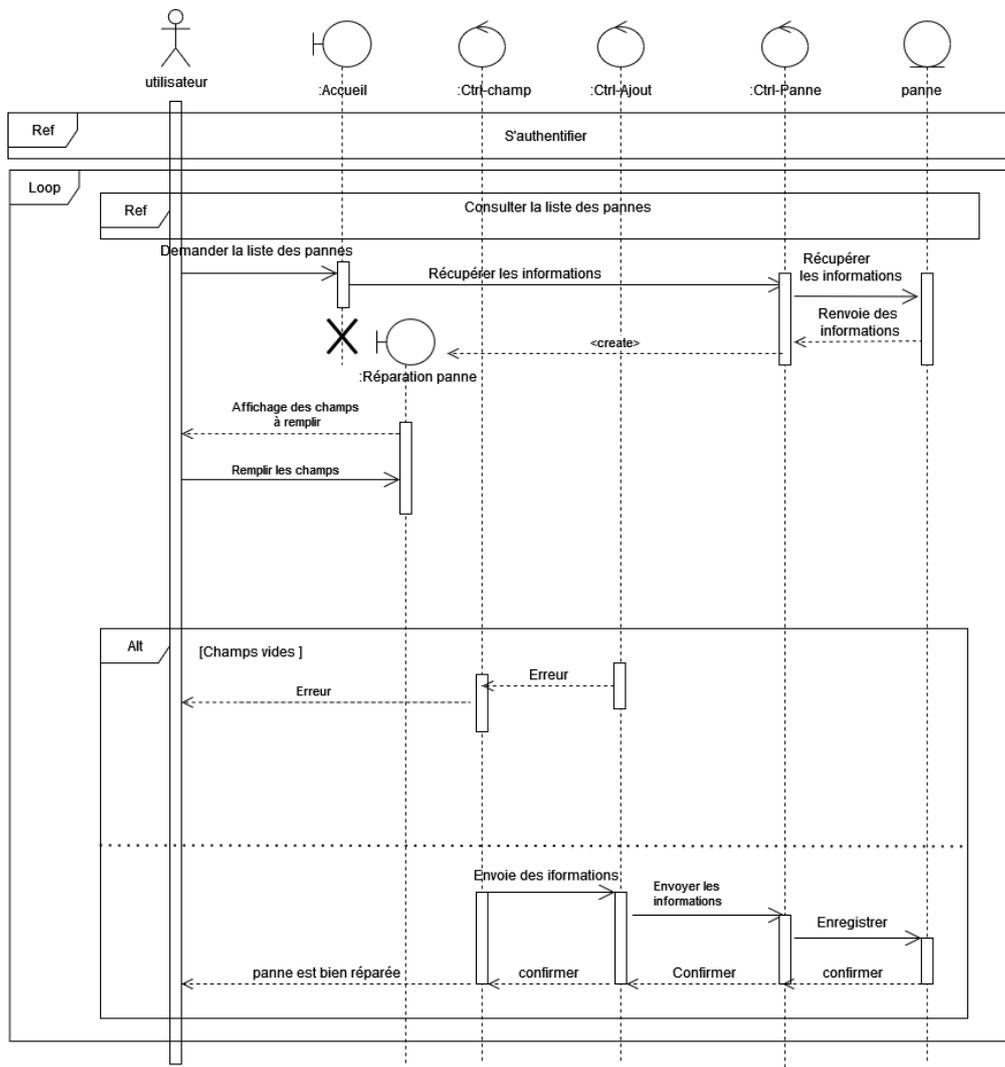


FIGURE C.4 – Diagramme d'interaction de cas d'utilisation «Réparer une panne»

C.1.5 Diagramme d'interaction de cas d'utilisation «Lister un équipement»

La figure suivante représente le diagramme d'interaction «Lister un équipement» :

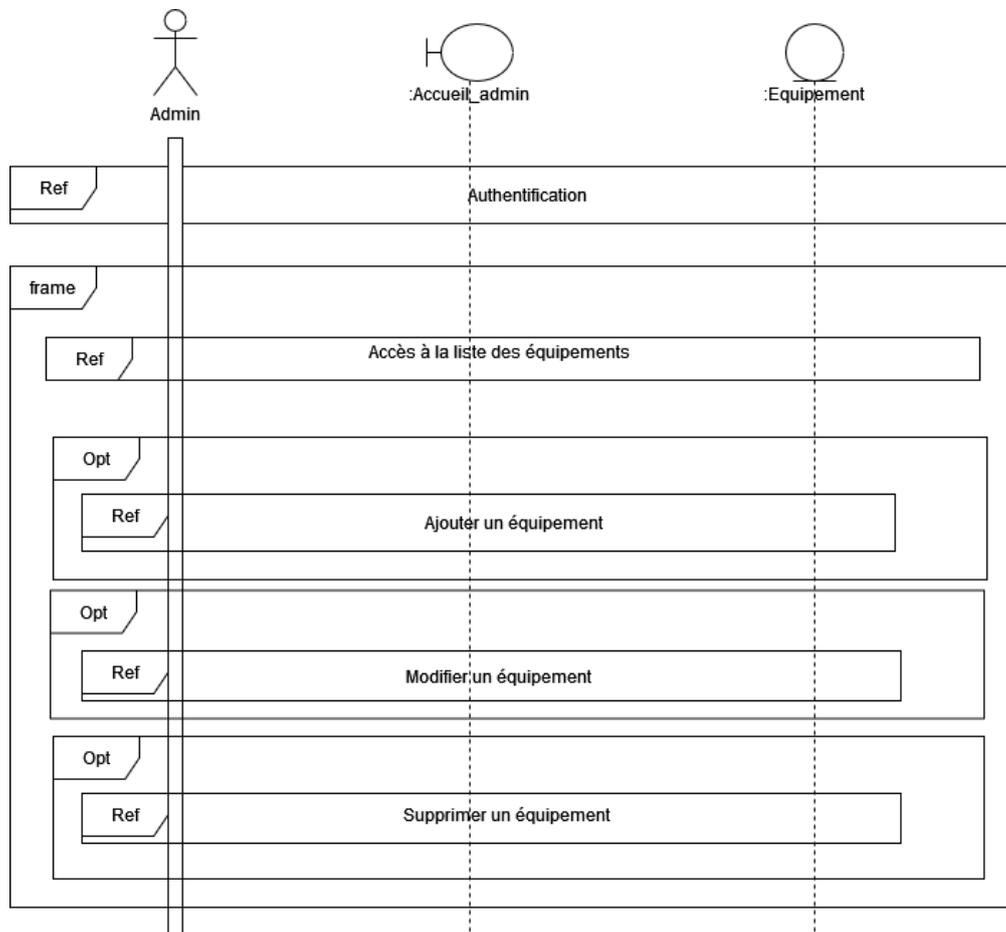


FIGURE C.5 – Diagramme d'interaction de cas d'utilisation «Lister un équipement»

RÉSUMÉ

Dans l'ère de la veille technologique et informatique, et l'application des nouvelles technologies d'information et de communication (N-TIC), des bases de données et réseaux dans les grandes entreprises, notre travail de projet de fin cycle (PFC), qui consiste en une application de gestion des pannes des équipements informatiques de SONATRACH et leurs maintenances, s'inscrit dans cette optique d'informatisation des procédés manuelles dans les organismes. Pour réaliser notre projet, nous avons suivi une démarche de développement logiciel qui se situe à mi-chemin entre UP et XP, en utilisant le langage de modélisation UML. Techniquement parlant, notre application est basée sur le principe SPA (Single Page Application) implémenté avec le framework VueJS-3 et la bibliothèque CSS Bootstrap. Ceci en ce qui concerne la partie frontend, et pour la partie backend, nous avons utilisé Node-JS pour implémenté une API RESTFull avec les méthodes HTTP : GET, POST, PUT et DELETE. Cette dernière partie accède à une base de données relationnelle MySQL. Cette séparation du frontend et backend permet de rendre facile l'extension de notre application d'autres solutions (applications de bureau et mobiles).

Mots clés : Gestion des pannes ,VueJS, NodeJS ,MySQL ,SPA , Frontend , Backend ,UML ,SONATRACH ,Equipements. .

ABSTRACT

In the era of the technological and data-processing watch, and the application of the new technologies of information and communication (N-TIC), data bases and networks in the large companies, our work of project of end cycle (PFC), which consists of an application of management of the breakdowns of the data-processing equipments of SONATRACH and their maintenances, falls under this optics of computerization of the manual processes in the organizations. To carry out our project, we followed a software development approach that is halfway between UP and XP, using the UML modeling language. Technically speaking, our application is based on the SPA (Single Page Application) principle implemented with the VueJS-3 framework and the CSS Bootstrap library. For the frontend part, we used Node-JS to implement a RESTFull API with HTTP methods : GET, POST, PUT and DELETE. This last part accesses a MySQL relational database. This separation of the frontend and backend makes it easy to extend our application to other solutions (desktop and mobile applications).

Key words : fault management,VueJS, NodeJS ,MySQL ,SPA, Frontend , Backend ,UML ,SONATRACH ,equipment.