



Université Mohammed V - Rabat
École Nationale d'Informatique
et d'Analyse des Systèmes

Rapport de Stage

FILIÈRE

Génie Logiciel

SUJET :

Conception et développement d'une solution pour le volet "facility maintenance" de l'aéroport Mohammed V

Réalisé par :

Abdessamad HAMMA

Encadré par :

M. El Madani KHALILI

Jury :

M. Abdellatif EL FAKER

M. Ahmed ETTALBI

Remerciements

À l'achèvement de ce projet, je tiens à exprimer ma sincère gratitude envers tous ceux qui ont joué un rôle, qu'il soit direct ou indirect, dans sa réalisation. Leur engagement et leur soutien indéfectible ont véritablement enrichi cette expérience, en lui insufflant une énergie positive et une collaboration exceptionnelle.

Je souhaite ainsi adresser des remerciements spéciaux à mon encadrant M. El Madani KHALIL pour sa guidance éclairée, ses conseils avisés et sa disponibilité tout au long de ce projet. Son expertise et son accompagnement ont été essentiels pour mener à bien ce travail.

En espérant que ce travail sera à la hauteur de vos attentes.

Résumé

Mon stage avait pour objectif la conception et le développement d'une solution intelligente pour le volet "facility maintenance" de l'aéroport Mohammed V. Ce projet visait à optimiser la gestion des opérations de maintenance des installations aéroportuaires en utilisant des technologies avancées, dans le but d'améliorer l'efficacité opérationnelle et la qualité des services.

Mon travail s'est déroulé en plusieurs phases, en commençant par une analyse approfondie des besoins et des processus existants. Cette étape a été cruciale pour comprendre les défis auxquels fait face l'aéroport en matière de maintenance des installations et pour identifier les opportunités d'amélioration. L'analyse a révélé la nécessité d'une solution centralisée permettant une meilleure traçabilité des interventions et une coordination optimisée des équipes de maintenance.

Sur la base de cette analyse, j'ai entamé une phase de conception de la solution, en définissant l'architecture système, les fonctionnalités clés et les interfaces utilisateur. L'objectif était de créer une plateforme conviviale et efficace pour faciliter la gestion des opérations de maintenance, tout en assurant une intégration harmonieuse avec les systèmes existants de l'aéroport.

Dans la phase de développement, j'ai mis en œuvre une architecture moderne et évolutive, en utilisant des technologies de pointe telles que Node.js et Express.js pour le backend, ainsi que HTML, CSS et Angular pour le frontend. L'intégration d'une base de données PostgreSQL a permis de gérer efficacement les informations relatives aux équipements, aux tâches de maintenance et aux interventions. Cette solution a abouti à une plateforme robuste qui optimise le flux de travail des équipes de maintenance et améliore significativement la gestion des installations aéroportuaires.

Abstract

My internship focused on designing and developing an intelligent solution for the facility maintenance division of Mohammed V Airport. This project aimed to optimize the management of airport facility maintenance operations using advanced technologies, with the goal of improving operational efficiency and service quality.

The work was conducted in several phases, beginning with a comprehensive analysis of existing needs and processes. This crucial step was essential for understanding the challenges faced by the airport in terms of facility maintenance and identifying opportunities for improvement. The analysis revealed the need for a centralized solution enabling better intervention tracking and optimized coordination of maintenance teams.

Based on this analysis, I initiated the solution design phase, defining the system architecture, key functionalities, and user interfaces. The objective was to create a user-friendly and efficient platform to facilitate maintenance operations management, while ensuring seamless integration with the airport's existing systems.

In the development phase, I implemented a modern and scalable architecture using cutting-edge technologies such as Node.js and Express.js for the backend, along with HTML, CSS, and Angular for the frontend. The integration of a PostgreSQL database enabled efficient management of equipment information, maintenance tasks, and interventions. This solution resulted in a robust platform that optimizes maintenance team workflows and significantly improves airport facility management.

The implemented system features a comprehensive ticket management workflow, real-time monitoring capabilities, and advanced reporting tools, providing maintenance supervisors and administrators with enhanced control over operations. This digital transformation project has demonstrated significant potential for improving maintenance efficiency and reducing operational costs while maintaining high safety and service standards at Mohammed V Airport.

Table des figures

1.1	logo ONDA	7
2.1	use-case diagramme	15
2.2	Diagramme des cas d'utilisation lié à l'administrateur	16
2.3	Diagramme des cas d'utilisation lié au distributeur	17
2.4	Diagramme de classes	17
2.5	workflow de gestion ticket	19
2.6	Diagramme de séquence relatif à la création du ticket	20
2.7	Diagramme de séquence relatif à l'attribution du Ticket	21
2.8	Diagramme de séquence relatif à l'intervention technique	21
2.9	Diagramme de séquence relatif au processus de validation	22
3.1	Logo de HTML	25
3.2	Logo de CSS	25
3.3	Logo de Bootstrap	25
3.4	Logo de JavaScript	26
3.5	Node.js logo	26
3.6	Logo de express	26
3.7	Logo de angular	27
3.8	Logo de postgres	27
3.9	Tableau de bord de l'administrateur	28
3.10	Ajout utilisateur	29
3.11	Tableau de bord de l'agent	29
3.12	Création d'un nouveau ticket	30
3.13	Tableau de bord du distributeur	31
3.14	Attribution ticket pour distributeur	31
3.15	Tableau de bord du technicien	32
3.16	Détails ticket	33
3.17	Tableau de bord du superviseur	33
3.18	Liste des tickets assignés	35
3.19	Détails et traitement d'un ticket	35
3.20	Création d'un nouveau ticket	36
3.21	Suivi des tickets créés	36
3.22	Interface de validation du superviseur	37

Table des matières

Remerciements	2
Résumé	3
Abstract	4
Remerciements	2
Résumé	3
Abstract	4
Introduction	5
Chapitre 1	6
1 Présentation du cadre général du stage	6
1.1 Description de l'entreprise	7
1.1.1 Statut et missions fondamentales	7
1.1.2 Engagements et objectifs stratégiques	8
1.2 Présentation de la mission du stage	8
1.2.1 Contexte général	8
1.2.2 Sujet du stage	9
1.2.3 Problématique	10
1.2.4 Solution proposée	10
1.2.5 Besoins fonctionnels	11
1.2.6 Besoins non fonctionnels	12
Chapitre 2	13
2 Conception Globale du Projet	13
2.1 Analyse théorique.	14
2.2 Conception globale	14
2.3 Conception détaillée	14
2.3.1 Diagramme de cas d'utilisation	15
2.3.2 Diagramme de cas d'utilisation détaillé	16
2.3.3 Diagramme de classes	17
2.3.4 Workflow de gestion des tickets	18
2.3.5 Diagrammes de séquences	20

Chapitre 3	24
3 Réalisation	24
3.1 Outils de travail	25
3.1.1 Le langage HTML	25
3.1.2 Le langage CSS	25
3.1.3 L'outil Bootstrap	25
3.1.4 Le langage JavaScript	26
3.1.5 L'outil Node.js	26
3.1.6 L'outil Express.js	26
3.1.7 L'outil Angular	27
3.1.8 L'outil PostgreSQL	27
3.2 Réalisation de l'application	28
3.2.1 Tableau de bord de l'administrateur	28
3.2.2 Tableau de bord de l'agent	29
3.2.3 Création d'un nouveau ticket	30
3.2.4 Tableau de bord du distributeur	30
3.2.5 Tableau de bord du technicien	31
3.2.6 Tableau de bord du superviseur	33
3.3 Interfaces Mobiles	34
3.3.1 Interface Mobile du Technicien	35
3.3.2 Interface Mobile de l'Agent	36
3.3.3 Interface Mobile du Superviseur	37
Conclusion	38
Bibliographie	39



Université Mohammed V - Rabat
École Nationale d'Informatique
et d'Analyse des Systèmes

Rapport de Stage

FILIÈRE

Génie Logiciel

SUJET :

Conception et développement d'une solution pour le volet "facility maintenance" de l'aéroport Mohammed V

Réalisé par :

Abdessamad HAMMA

Encadré par :

M. El Madani KHALILI

Jury :

M. Abdellatif EL FAKER

M. Ahmed ETTALBI

Remerciements

À l'achèvement de ce projet, je tiens à exprimer ma sincère gratitude envers tous ceux qui ont joué un rôle, qu'il soit direct ou indirect, dans sa réalisation. Leur engagement et leur soutien indéfectible ont véritablement enrichi cette expérience, en lui insufflant une énergie positive et une collaboration exceptionnelle.

Je souhaite ainsi adresser des remerciements spéciaux à mon encadrant M. El Madani KHALIL pour sa guidance éclairée, ses conseils avisés et sa disponibilité tout au long de ce projet. Son expertise et son accompagnement ont été essentiels pour mener à bien ce travail.

En espérant que ce travail sera à la hauteur de vos attentes.

Résumé

Mon stage avait pour objectif la conception et le développement d'une solution intelligente pour le volet "facility maintenance" de l'aéroport Mohammed V. Ce projet visait à optimiser la gestion des opérations de maintenance des installations aéroportuaires en utilisant des technologies avancées, dans le but d'améliorer l'efficacité opérationnelle et la qualité des services.

Mon travail s'est déroulé en plusieurs phases, en commençant par une analyse approfondie des besoins et des processus existants. Cette étape a été cruciale pour comprendre les défis auxquels fait face l'aéroport en matière de maintenance des installations et pour identifier les opportunités d'amélioration. L'analyse a révélé la nécessité d'une solution centralisée permettant une meilleure traçabilité des interventions et une coordination optimisée des équipes de maintenance.

Sur la base de cette analyse, j'ai entamé une phase de conception de la solution, en définissant l'architecture système, les fonctionnalités clés et les interfaces utilisateur. L'objectif était de créer une plateforme conviviale et efficace pour faciliter la gestion des opérations de maintenance, tout en assurant une intégration harmonieuse avec les systèmes existants de l'aéroport.

Dans la phase de développement, j'ai mis en œuvre une architecture moderne et évolutive, en utilisant des technologies de pointe telles que Node.js et Express.js pour le backend, ainsi que HTML, CSS et Angular pour le frontend. L'intégration d'une base de données PostgreSQL a permis de gérer efficacement les informations relatives aux équipements, aux tâches de maintenance et aux interventions. Cette solution a abouti à une plateforme robuste qui optimise le flux de travail des équipes de maintenance et améliore significativement la gestion des installations aéroportuaires.

Abstract

My internship focused on designing and developing an intelligent solution for the facility maintenance division of Mohammed V Airport. This project aimed to optimize the management of airport facility maintenance operations using advanced technologies, with the goal of improving operational efficiency and service quality.

The work was conducted in several phases, beginning with a comprehensive analysis of existing needs and processes. This crucial step was essential for understanding the challenges faced by the airport in terms of facility maintenance and identifying opportunities for improvement. The analysis revealed the need for a centralized solution enabling better intervention tracking and optimized coordination of maintenance teams.

Based on this analysis, I initiated the solution design phase, defining the system architecture, key functionalities, and user interfaces. The objective was to create a user-friendly and efficient platform to facilitate maintenance operations management, while ensuring seamless integration with the airport's existing systems.

In the development phase, I implemented a modern and scalable architecture using cutting-edge technologies such as Node.js and Express.js for the backend, along with HTML, CSS, and Angular for the frontend. The integration of a PostgreSQL database enabled efficient management of equipment information, maintenance tasks, and interventions. This solution resulted in a robust platform that optimizes maintenance team workflows and significantly improves airport facility management.

The implemented system features a comprehensive ticket management workflow, real-time monitoring capabilities, and advanced reporting tools, providing maintenance supervisors and administrators with enhanced control over operations. This digital transformation project has demonstrated significant potential for improving maintenance efficiency and reducing operational costs while maintaining high safety and service standards at Mohammed V Airport.

Introduction

Ce stage représente l'accomplissement d'une étape cruciale de ma formation à l'ENSIAS. J'ai eu le privilège de l'effectuer au sein de l'Office National Des Aéroports (ONDA), une opportunité qui s'inscrit parfaitement dans mon parcours académique et mes aspirations professionnelles.

Mon objectif principal à travers ce rapport est de présenter de manière structurée les connaissances et les technologies que j'ai employées pour mener à bien les missions qui m'ont été confiées, ainsi que de partager l'expérience enrichissante que j'ai acquise au sein de l'ONDA. Durant ce stage, mon travail consistait à concevoir et développer une solution intelligente pour le volet "facility maintenance" de l'aéroport Mohammed V. Ce projet avait pour but d'optimiser la gestion des opérations de maintenance des installations aéroportuaires en utilisant des technologies avancées. Ce rapport détaille la méthodologie que j'ai adoptée pour réaliser ce projet, depuis l'analyse des besoins jusqu'à la mise en production de la solution. Il met en lumière les différentes étapes franchies, les choix technologiques effectués et les défis rencontrés.

Chapitre 1

Présentation du cadre général du stage

Ce chapitre présentera une description générale du sujet « application de gestion des stages » puis présentera notre objectif et notre démarche.

1.1 Description de l'entreprise

L'histoire de la gestion aéroportuaire au Maroc est marquée par une évolution progressive vers une autonomie de gestion. Jusqu'en 1980, les aéroports et les services de navigation aérienne étaient directement gérés par le Ministère du Transport. La construction et la mise en service du terminal de l'aéroport Mohammed V ont marqué un tournant décisif dans cette évolution, conduisant à la création du premier établissement public de gestion aéroportuaire : l'Office des Aéroports de Casablanca (OAC).

Le succès de cette première expérience a conduit à son extension progressive. En 1990, l'OAC est transformé en Office National Des Aéroports (ONDA) en vertu du décret n° 2-89-480 du 30 décembre 1989, élargissant ainsi ses compétences à l'ensemble des aéroports nationaux. Cette évolution a été renforcée par une Décision Royale en 1991, rattachant l'ensemble des services de la Navigation Aérienne à l'ONDA.



FIG. 1.1 : logo ONDA

1.1.1 Statut et missions fondamentales

L'ONDA est aujourd'hui un établissement public à caractère industriel et commercial, doté d'une autonomie financière et administrative. Ses missions s'articulent autour de quatre axes principaux :

1. Sécurité aérienne :

- Garantie de la sécurité de la navigation aérienne au niveau des aéroports
- Gestion de l'espace aérien sous juridiction nationale

2. Gestion et développement infrastructurel :

- Aménagement et exploitation des aéroports civils
- Maintenance et développement des infrastructures
- Gestion des services aux voyageurs et marchandises

3. Coopération internationale :

- Liaison avec les organismes internationaux
- Coordination avec les aéroports internationaux

4. Formation et développement des compétences :

- Formation d'ingénieurs de l'aéronautique civile
- Formation des contrôleurs aériens
- Formation des électroniciens de la sécurité aérienne

1.1.2 Engagements et objectifs stratégiques

L'ONDA poursuit des objectifs stratégiques qui se déclinent en plusieurs exigences fondamentales :

- **Excellence opérationnelle** : Garantie d'une qualité de service conforme aux normes internationales dans toutes les prestations rendues aux compagnies et aux passagers
- **Innovation technologique** : Développement continu des ressources pour répondre aux évolutions technologiques du secteur aéronautique
- **Développement durable** : Adaptation et croissance pour répondre aux besoins croissants du transport aérien tout en respectant les normes environnementales

Ces engagements s'inscrivent dans une vision à long terme visant à positionner le Maroc comme hub aéroportuaire régional majeur, tout en contribuant au développement économique et touristique du Royaume.

1.2 Présentation de la mission du stage

1.2.1 Contexte général

Dans le cadre de sa stratégie de modernisation, l'aéroport Mohammed V, infrastructure clé du royaume du Maroc, entreprend une transformation digitale majeure de ses processus opérationnels. Cette initiative s'inscrit dans une vision globale visant à optimiser l'ensemble des services aéroportuaires et à renforcer la position de l'aéroport comme hub régional de premier plan.

La maintenance des infrastructures aéroportuaires, composante critique de cette transformation, a jusqu'à présent été gérée de manière traditionnelle avec des processus majoritairement manuels. Cette approche conventionnelle, bien qu'ayant servi l'aéroport pendant de nombreuses années, nécessite aujourd'hui une refonte complète pour répondre aux exigences croissantes du secteur aéroportuaire. Cette digitalisation devient particulièrement cruciale face à l'augmentation continue du trafic aérien, la complexification des installations aéroportuaires modernes, les nouvelles normes internationales de sécurité et de qualité, ainsi que la nécessité d'une meilleure optimisation des ressources.

Le processus actuel de gestion de la maintenance présente des limitations significatives dans plusieurs domaines. La communication et le suivi reposent essentiellement sur des méthodes traditionnelles, avec une documentation majoritairement manuelle, entraînant des délais de traitement prolongés et une traçabilité limitée des opérations de maintenance. L'absence d'un système d'information centralisé complexifie considérablement la gestion de l'information, rendant difficile le suivi en temps réel des opérations et limitant la visibilité sur l'historique des interventions.

La coordination entre les différentes équipes d'intervention souffre également de ces limitations, impactant directement l'efficacité opérationnelle. La synchronisation complexe entre les équipes, combinée à une optimisation limitée des ressources techniques disponibles, résulte en des temps de réponse parfois prolongés et une difficulté accrue dans la planification préventive des maintenances. Cette situation affecte non seulement la réactivité des équipes de maintenance, mais aussi la qualité du service fourni aux usagers de l'aéroport. Face à ces défis multiples, la digitalisation devient non seulement une opportunité d'amélioration mais une nécessité stratégique pour maintenir la compétitivité et l'excellence opérationnelle de l'aéroport Mohammed V. Dans un contexte d'évolution rapide du secteur aéroportuaire international, cette transformation digitale représente un levier essentiel pour optimiser les processus de maintenance et répondre aux exigences croissantes en termes de performance et de fiabilité des infrastructures aéroportuaires.

1.2.2 Sujet du stage

Le stage s'inscrit dans cette démarche de transformation digitale entreprise par l'aéroport Mohammed V et porte spécifiquement sur la conception et le développement d'une solution intelligente pour la gestion de la maintenance des installations aéroportuaires. Cette initiative représente une étape cruciale dans la modernisation des processus opérationnels de l'aéroport, visant à améliorer significativement l'efficacité et la traçabilité des opérations de maintenance.

Ce projet a pour objectif principal de digitaliser l'intégralité du processus de facility maintenance, en procédant à une transformation profonde des méthodes de travail actuelles. Cette digitalisation implique le remplacement systématique des procédures manuelles, jusqu'alors utilisées pour la gestion des interventions, par un système informatisé moderne et efficient. Cette transition représente un changement fondamental dans la manière dont la maintenance est planifiée, exécutée et suivie au sein de l'infrastructure aéroportuaire.

La mission centrale de ce stage consiste à développer une plateforme web intégrée, conçue pour gérer l'ensemble du cycle de vie des interventions de maintenance. Cette plateforme se positionne comme un outil complet et centralisé, capable de prendre en charge toutes les étapes du processus de maintenance, depuis la détection initiale d'un besoin d'intervention jusqu'à la validation finale des travaux réalisés. Cette approche globale permet d'assurer une continuité et une cohérence dans le suivi des opérations de maintenance.

L'objectif final est de créer un outil qui s'intègre parfaitement dans l'écosystème technique existant de l'aéroport, tout en apportant une valeur ajoutée significative en termes d'efficacité opérationnelle. Cette intégration harmonieuse est essentielle pour garantir l'adoption réussie de la solution par les équipes et maximiser son impact positif sur les opérations quotidiennes de maintenance.

1.2.3 Problématique

La problématique centrale de ce projet de digitalisation s'articule autour de la modernisation complète du processus de gestion de la maintenance. Il s'agit de transformer un système manuel en une solution digitale performante, tout en prenant en compte les contraintes spécifiques du secteur aéroportuaire et les besoins des différents acteurs impliqués.

Cette transformation soulève plusieurs questions fondamentales. Comment concevoir une solution digitale qui améliore significativement l'efficacité des processus tout en garantissant une adoption optimale par les utilisateurs ? Comment assurer une transition fluide vers le nouveau système tout en maintenant la continuité des opérations ? Comment intégrer les meilleures pratiques de l'industrie tout en s'adaptant aux spécificités de l'aéroport Mohammed V ?

1.2.4 Solution proposée

1.2.4.1 Concept du ticket de maintenance

Un ticket de maintenance, également appelé "fiche d'intervention" ou "fiche de panne", représente l'unité fondamentale de suivi d'une opération de maintenance. Il s'agit d'un document numérique qui centralise toutes les informations relatives à une intervention : la nature du problème, sa localisation, son niveau d'urgence, l'équipement concerné, ainsi que l'historique complet des actions entreprises. Ce concept, largement utilisé dans l'industrie, permet une traçabilité complète des opérations de maintenance et facilite la coordination entre les différents intervenants.

1.2.4.2 Architecture et fonctionnalités de la solution

La solution développée consiste en une application web moderne qui digitalise intégralement le processus de gestion des tickets de maintenance de l'aéroport Mohammed V. Cette plateforme implémente un workflow structuré qui transforme les processus manuels existants en un système informatisé efficace et transparent. L'application permet une gestion complète du cycle de vie des tickets de maintenance, depuis leur création jusqu'à leur clôture, en passant par les phases d'affectation, de traitement et de validation.

Le système s'appuie sur une architecture moderne utilisant des technologies web robustes, garantissant ainsi performance et évolutivité. L'interface utilisateur intuitive facilite l'adoption par les différents acteurs tout en optimisant leur productivité. Cette solution représente une avancée significative dans la modernisation des processus de maintenance de l'aéroport.

L'application de gestion des tickets de maintenance doit répondre à un ensemble de besoins fonctionnels précis, organisés selon les différents aspects du processus de maintenance. Ces besoins ont été identifiés à travers une analyse approfondie des processus existants et des attentes des utilisateurs finaux.

1.2.5 Besoins fonctionnels

Administration :

- Gestion complète des utilisateurs et de leurs droits d'accès
- Configuration des rôles et permissions spécifiques par profil utilisateur
- Paramétrage des workflows de validation selon les types d'intervention
- Définition et maintenance des référentiels (équipements, localisations, types de pannes)

Gestion des tickets :

- Création de tickets avec spécification des informations essentielles (localisation, équipement, priorité)
- Suivi en temps réel de l'état d'avancement des tickets
- Modification des informations selon les droits d'accès
- Catégorisation des tickets par type d'intervention et niveau d'urgence
- Historisation complète des modifications apportées aux tickets

Gestion des interventions :

- Affectation des tickets aux techniciens appropriés selon leurs compétences
- Prise en charge et mise à jour du statut par les techniciens
- Documentation des interventions avec supports multimédia (photos, documents)
- Planification et ordonnancement des interventions
- Suivi des temps d'intervention et des ressources utilisées

Validation et contrôle :

- Examen et validation des interventions par les superviseurs
- Réouverture possible des tickets non satisfaisants avec justification
- Processus de clôture formalisé après validation finale
- Génération de rapports de contrôle et de suivi des interventions
- Évaluation de la qualité des interventions réalisées

1.2.6 Besoins non fonctionnels

Les exigences non fonctionnelles nécessaires au bon fonctionnement de l'application sont :

Sécurité :

- Authentification robuste des utilisateurs
- Gestion des droits d'accès par rôles
- Protection des données sensibles

Ergonomie :

- Interface intuitive et responsive
- Compatibilité multi-supports
- Messages d'erreur explicites

Disponibilité et Fiabilité ::

- Haute disponibilité du système
- Gestion efficace des erreurs
- Mécanismes de sauvegarde automatique des données

Chapitre 2

Conception Globale du Projet

Ce chapitre abordera une analyse du sujet pour présenter une approche originale et une modélisation de l'application.

2.1 Analyse théorique.

L'analyse théorique constitue la première étape de notre projet de digitalisation de la maintenance de l'aéroport Mohammed V, où nous approfondissons notre compréhension du sujet. Cette phase est essentielle car elle nous permet de définir les objectifs, les besoins et les contraintes de notre application de gestion des tickets de maintenance aéroportuaire.

Au cours de cette analyse, nous avons examiné les différents aspects liés à la gestion de la maintenance aéroportuaire. Nous avons étudié les processus et les workflows actuellement en place, identifié les principales inefficacités du système manuel existant, et analysé les attentes des différents acteurs impliqués, notamment les agents créateurs de tickets, les distributeurs, les techniciens de maintenance, les superviseurs et les administrateurs du système.

De plus, nous avons réalisé une étude approfondie des fonctionnalités nécessaires pour répondre aux besoins spécifiques de notre application. Cette étude a mis en évidence l'importance d'un système de suivi en temps réel, d'une gestion efficace des affectations des tickets, et d'une documentation structurée des interventions. Nous avons également pris en compte les aspects techniques essentiels tels que la performance du système face à une charge importante de tickets, la sécurité des données sensibles de l'aéroport, et la capacité d'évolution de l'application pour s'adapter aux besoins futurs.

Cette analyse théorique nous a permis d'établir une base solide pour la conception et le développement de notre application. Nous avons pu définir un workflow précis pour le traitement des tickets, depuis leur création jusqu'à leur clôture, en passant par les phases d'affectation, de traitement et de validation. En comprenant profondément les enjeux de la maintenance aéroportuaire, nous sommes en mesure de concevoir une application qui transformera efficacement les processus manuels existants en un système informatisé performant et transparent, répondant aux exigences croissantes du secteur aéroportuaire.

2.2 Conception globale

Notre projet de maintenance aéroportuaire vise à développer une application web moderne offrant un ensemble de services de gestion des tickets de maintenance accessibles à travers le réseau de l'aéroport Mohammed V. Cette application adopte une architecture à trois niveaux, comprenant : — Un client léger, accessible via un navigateur web, offrant une interface intuitive adaptée aux différents profils d'utilisateurs (agents, distributeurs, techniciens et superviseurs). — Un serveur d'application qui orchestre le workflow de maintenance, gère les affectations des tickets et assure le traitement des interventions. — Une base de données robuste qui centralise les informations des tickets, le suivi des interventions et les données des utilisateurs. Cette architecture spécifique guide notre conception pour garantir une gestion efficace et transparente des opérations de maintenance au sein de l'aéroport Mohammed V.

2.3 Conception détaillée

Dans cette phase, nous abordons la conception approfondie de notre projet de stage. Nous allons concrétiser notre vision en présentant des diagrammes de cas d'utilisation, des diagrammes

de classes et des diagrammes d'activité. Ces éléments illustreront de manière précise la structure et le fonctionnement de notre application.

2.3.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation ci-dessous illustre les interactions entre les différents acteurs du système et les principales fonctionnalités de la solution de gestion de la maintenance des installations aéroportuaires. propose deux cas de figures :

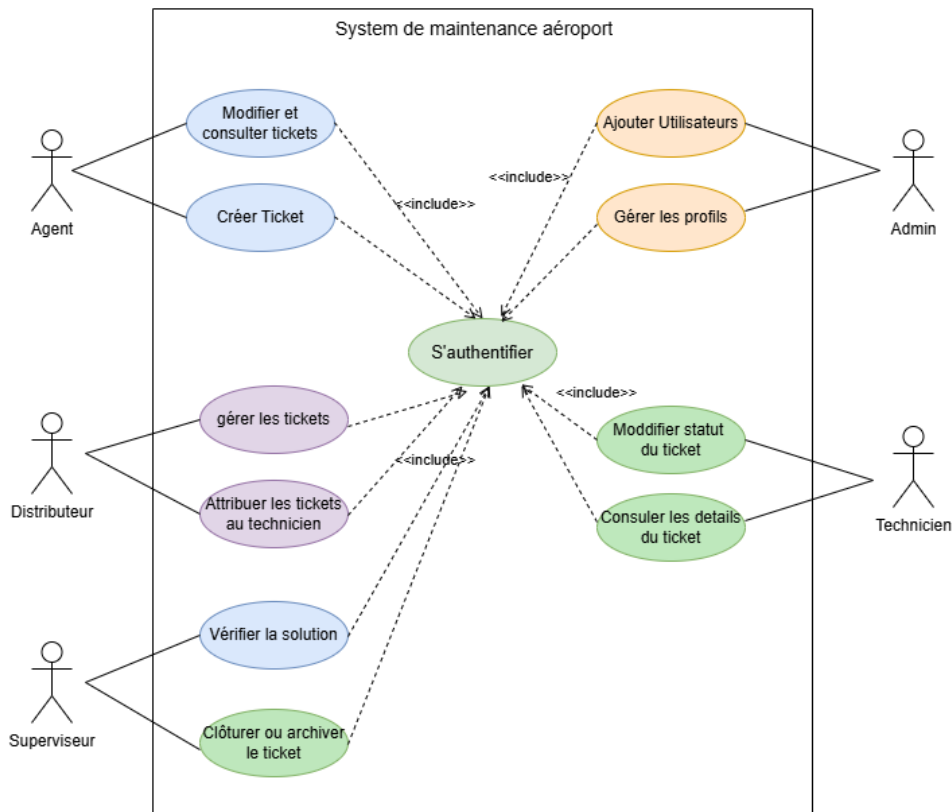


FIG. 2.1 : use-case diagramme

Le diagramme de cas d'utilisation illustre les différentes interactions entre les acteurs et notre système de maintenance aéroportuaire. Au centre du système se trouve le cas d'utilisation "S'authentifier" qui est commun à tous les acteurs, assurant ainsi la sécurité et la traçabilité des actions. L'Agent dispose de deux fonctionnalités principales : la création des tickets ("Créer Ticket") et la possibilité de modifier et consulter les tickets existants ("Modifier et consulter tickets"). Ces cas d'utilisation permettent d'initier et de suivre le processus de maintenance.

Le Distributeur intervient avec deux cas d'utilisation essentiels : le tri et la modification des tickets ("Trier et modifier les tickets") ainsi que l'attribution des tickets aux techniciens ("Attribuer les tickets au technicien"). Ces fonctionnalités sont cruciales pour la bonne coordination des interventions.

Le Technicien peut interagir avec le système à travers deux cas d'utilisation : la modification du statut du ticket ("Modifier statut du ticket") et la consultation des détails du ticket

(”Consulter les détails du ticket”), lui permettant ainsi de gérer efficacement ses interventions.

Le Superviseur dispose de deux cas d’utilisation spécifiques : la vérification de la solution proposée (”Vérifier la solution”) et la clôture ou l’archivage du ticket (”Clôturer ou archiver le ticket”), assurant ainsi le contrôle qualité des interventions.

Enfin, l’Administrateur peut gérer les utilisateurs à travers deux cas d’utilisation : l’ajout d’utilisateurs (”Ajouter Utilisateurs”) et la gestion des profils (”Gérer les profils”), garantissant une administration efficace du système.

2.3.2 Diagramme de cas d’utilisation détaillé

Dans cette partie, nous présentons les diagrammes des cas d’utilisation détaillé ainsi que leur descriptions textuelles.

2.3.2.1 Cas d’utilisation ”Gestion des profils”

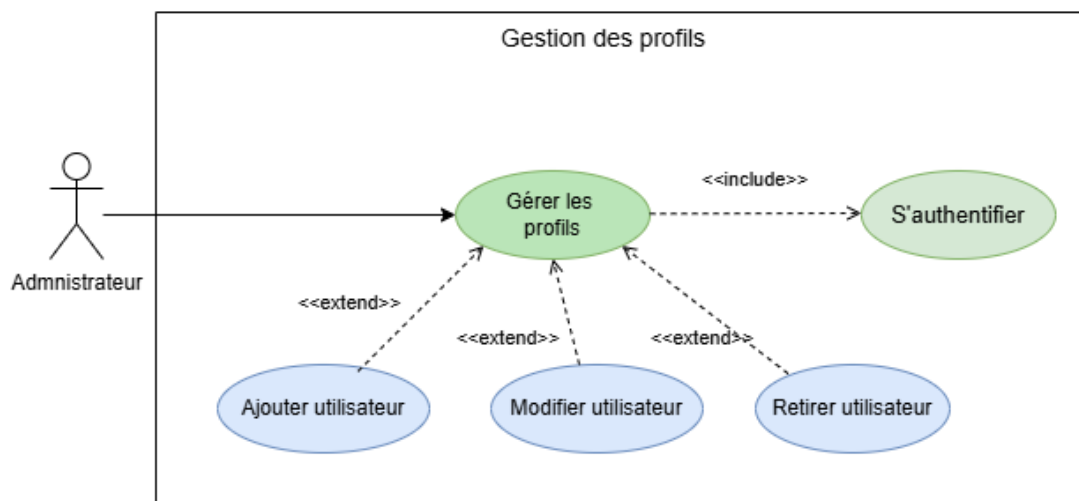


FIG. 2.2 : Diagramme des cas d’utilisation lié à l’administrateur

Ce diagramme de cas d’utilisation illustre les fonctionnalités de gestion des profils accessibles à l’administrateur du système. Le cas d’utilisation principal ”Gérer les profils” nécessite une authentification préalable et se décline en trois opérations principales : l’ajout, la modification et le retrait d’utilisateurs.

2.3.2.2 Cas d'utilisation "Gestion des tickets par distributeur"

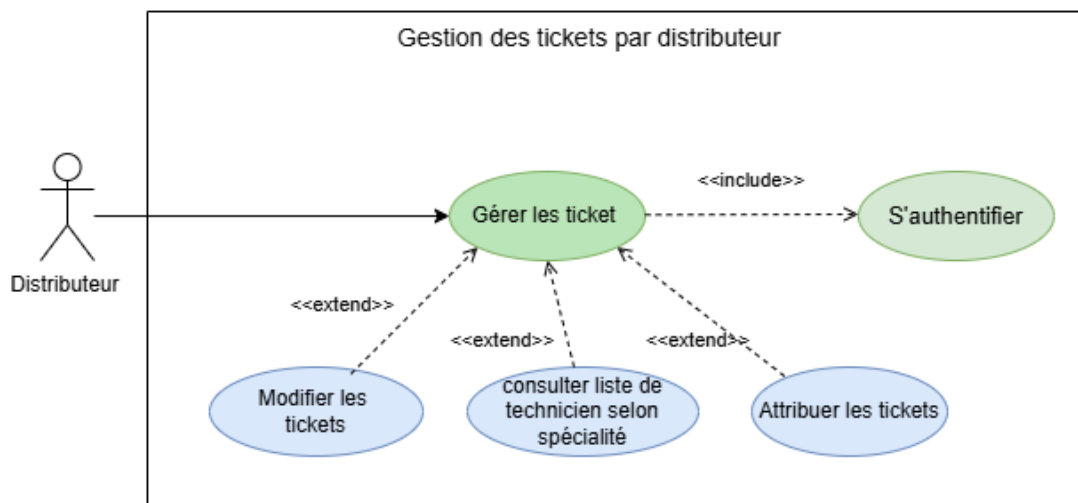


FIG. 2.3 : Diagramme des cas d'utilisation lié au distributeur

2.3.3 Diagramme de classes

Le diagramme de classes donne une vue d'ensemble d'un système en montrant ses classes, interfaces, collaborations et leurs relations. La Figure illustre le diagramme de classes pour notre projet.

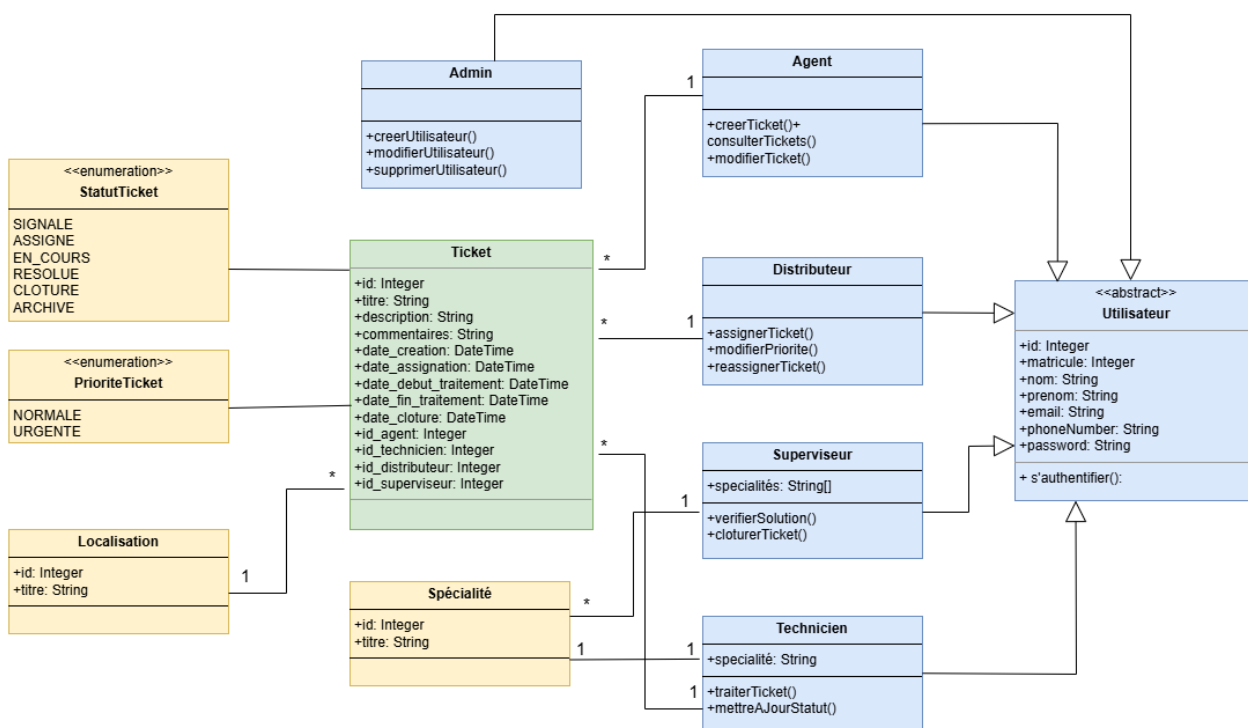


FIG. 2.4 : Diagramme de classes

Le diagramme de classes ci-dessus présente la structure statique de notre système de maintenance aéroportuaire. Il met en évidence les différentes classes qui composent l'application ainsi que leurs relations.

Au cœur du système se trouve la classe **Ticket**, qui représente les demandes de maintenance. Chaque ticket est caractérisé par des attributs essentiels tels que son identifiant, son titre, sa description, et diverses dates marquant son cycle de vie (création, assignation, début et fin de traitement, clôture). Les tickets sont liés à plusieurs acteurs du système via leurs identifiants (`id_agent`, `id_technicien`, `id_distributeur`, `id_superviseur`).

La classe abstraite **Utilisateur** définit la structure commune à tous les types d'utilisateurs du système, avec les attributs de base comme l'identifiant, le matricule, le nom, le prénom, l'email et le mot de passe. Cette classe est étendue par cinq classes spécialisées :

- **Agent** : responsable de la création et du suivi des tickets
- **Admin** : gère les utilisateurs du système
- **Distributeur** : assure l'assignation et la gestion des priorités des tickets
- **Technicien** : traite les tickets et met à jour leur statut
- **Superviseur** : vérifie et valide les solutions apportées

Le système utilise également deux énumérations importantes :

- **StatutTicket** : définit les différents états possibles d'un ticket (Signalé, Assigné ...)
- **PrioriteTicket** : indique le niveau d'urgence (Normale, Urgente)

Les classes **Localisation** et **Spécialité** servent de référentiels, où **Spécialité** est spécifiquement liée aux techniciens et superviseurs, permettant une attribution appropriée des tickets selon les compétences requises.

2.3.4 Workflow de gestion des tickets

Le workflow ci-dessous illustre le processus complet de gestion des tickets de maintenance aéroportuaire, de leur création jusqu'à leur clôture.

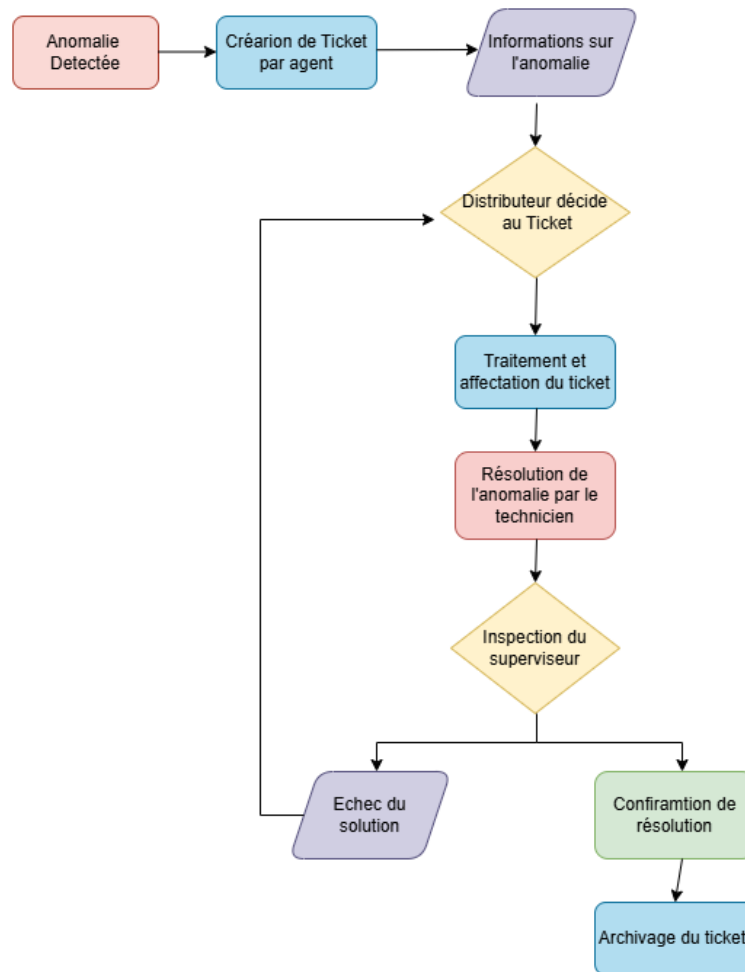


FIG. 2.5 : workflow de gestion ticket

Le processus commence par la détection d'une anomalie dans les installations aéroportuaires. Un agent crée alors un ticket dans le système en renseignant toutes les informations essentielles : description détaillée de l'anomalie, localisation précise, niveau de priorité et éventuellement des photos. Cette étape initiale est cruciale car la qualité des informations influence directement l'efficacité du traitement.

Un distributeur examine ensuite le ticket pour évaluer sa complexité et son urgence. Il procède à l'affectation en sélectionnant un technicien approprié selon plusieurs critères : sa spécialité technique, sa disponibilité, sa charge de travail actuelle et la localisation de l'intervention.

Le technicien désigné prend alors en charge l'intervention en suivant plusieurs étapes : mise à jour du statut à "En cours", évaluation sur site, réalisation de l'intervention technique, puis actualisation du statut à "Résolu".

Un superviseur procède ensuite à une inspection approfondie pour vérifier que l'anomalie a été correctement traitée et que la solution est conforme aux standards de qualité. Si validé, le ticket est clos et les informations sont archivées. En cas de rejet, un nouveau ticket est créé automatiquement avec l'historique complet et réassigné à un autre technicien.

Ce processus structuré assure un traitement méthodique et rigoureux des anomalies, tout

en garantissant une traçabilité complète et une haute qualité des interventions pour la sécurité des installations aéroportuaires.

2.3.5 Diagrammes de séquences

La description textuelle des cas d'utilisation facilite la communication avec les utilisateurs, mais elle a des limites, notamment pour représenter la séquence d'actions. Pour remédier à cela, il est conseillé d'ajouter des diagrammes de séquence. Dans cette section, nous présentons quelques exemples de ces diagrammes pour illustrer la dynamique de notre système.

2.3.5.1 Diagramme de séquence relatif à la création du ticket

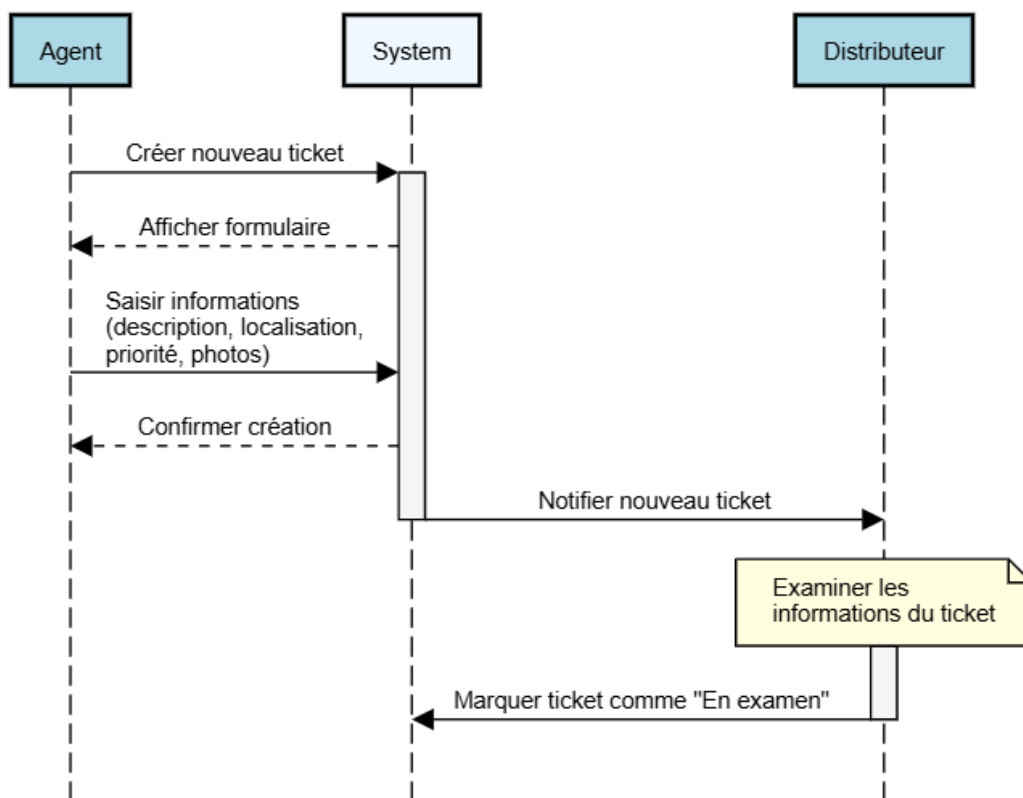


FIG. 2.6 : Diagramme de séquence relatif à la création du ticket

Ce diagramme de séquence illustre l'initialisation d'un ticket de maintenance. Le processus débute avec la création du ticket par l'Agent, qui saisit les informations nécessaires (description, localisation, priorité, photos) via le formulaire fourni par le Système. Une fois la création confirmée, le Distributeur est notifié et procède à l'examen initial du ticket, le marquant comme "En examen" pour traitement ultérieur.

2.3.5.2 Diagramme de séquence relatif à l'attribution du Ticket

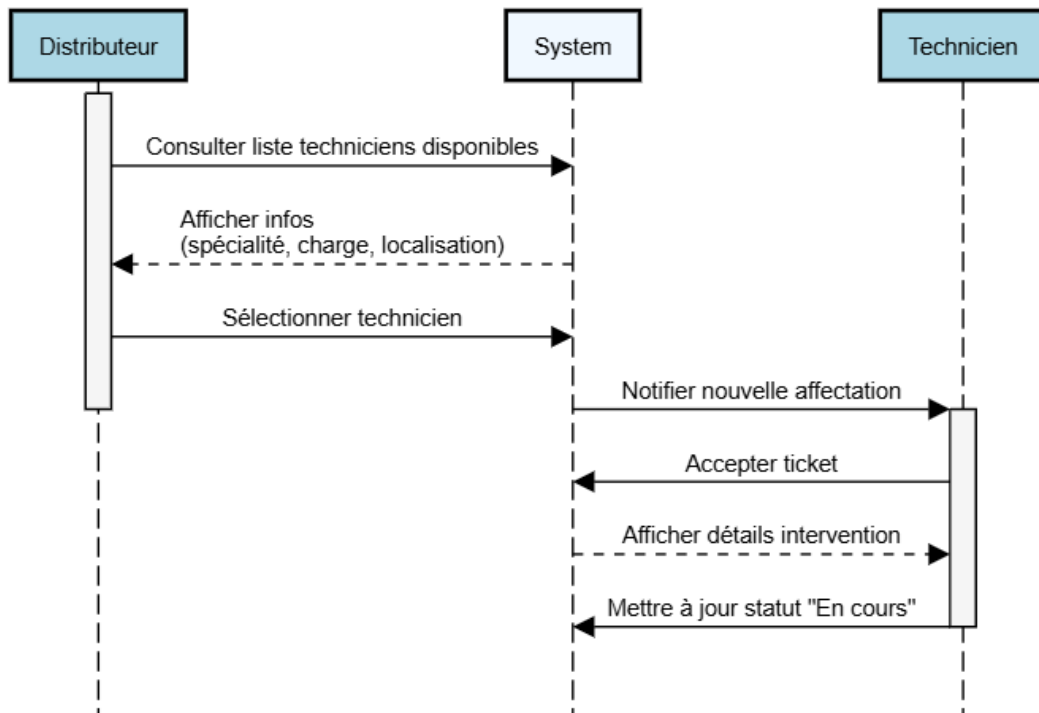


FIG. 2.7 : Diagramme de séquence relatif à l'attribution du Ticket

Ce diagramme de séquence illustre le processus d'attribution et de prise en charge d'un ticket de maintenance. Le Distributeur consulte d'abord la liste des techniciens disponibles avec leurs caractéristiques avant de sélectionner le technicien approprié.

2.3.5.3 Diagramme de séquence relatif à l'intervention technique

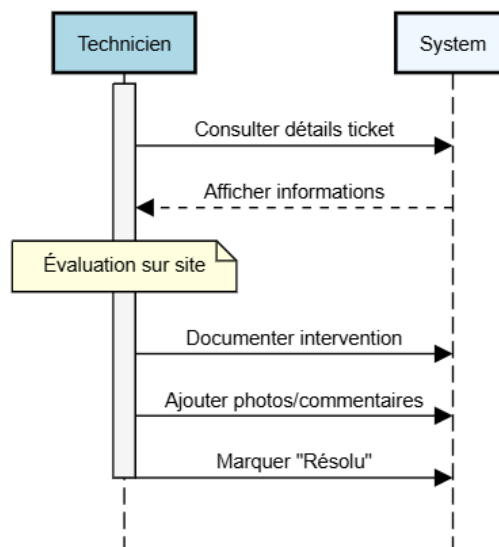


FIG. 2.8 : Diagramme de séquence relatif à l'intervention technique

Ce diagramme de séquence détaille le processus d'intervention technique sur site. Le processus débute lorsque le Technicien consulte les détails complets du ticket dans le système, lui permettant d'avoir une vision claire de la problématique à traiter. Il procède ensuite à une évaluation approfondie sur site pour identifier précisément la nature de l'intervention requise. Une fois l'intervention effectuée, le Technicien documente minutieusement son travail dans le système en ajoutant des photos illustratives et des commentaires détaillés qui serviront de référence pour le suivi et la traçabilité de l'intervention.

2.3.5.4 Diagramme de séquence relatif au processus de validation

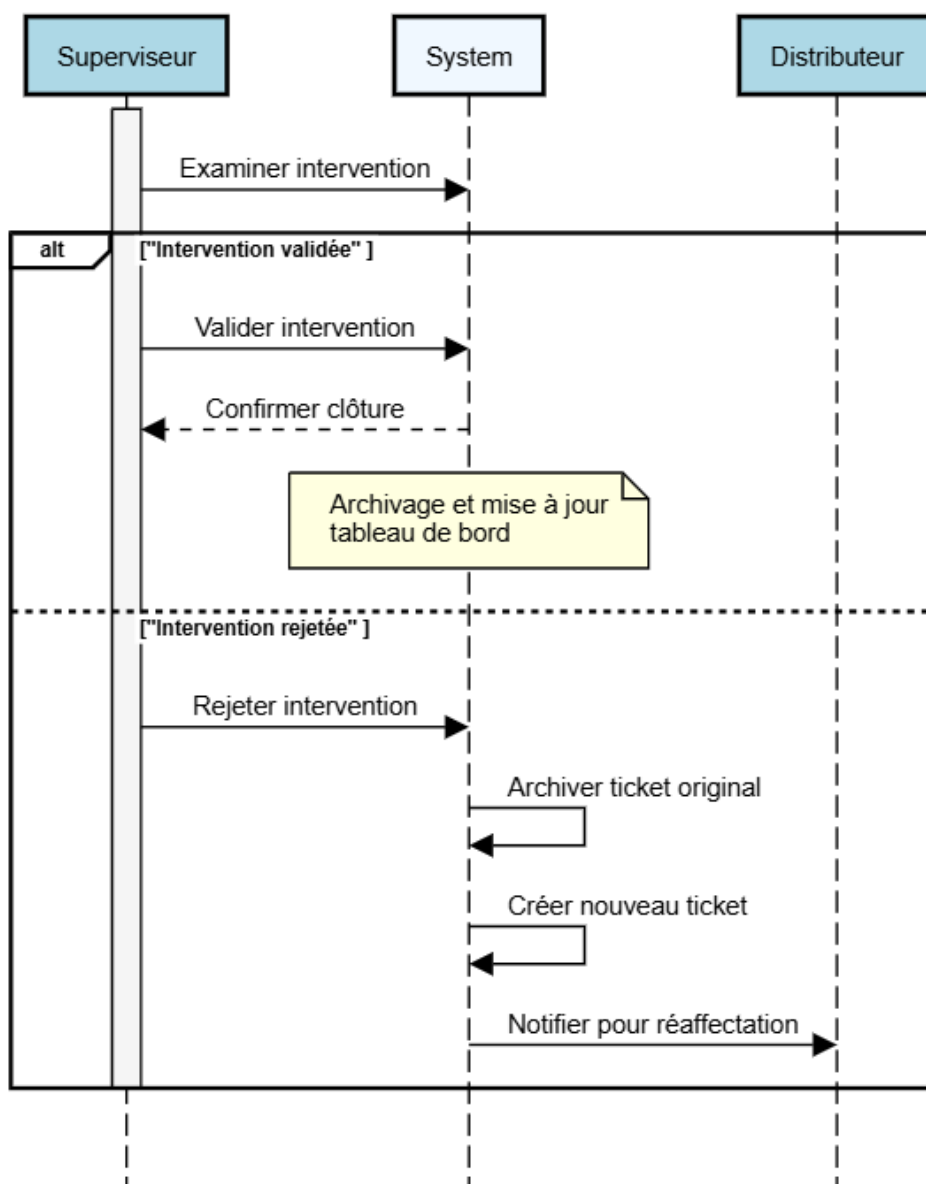


FIG. 2.9 : Diagramme de séquence relatif au processus de validation

Ce diagramme de séquence illustre le processus critique de validation d'une intervention technique. Le Superviseur commence par examiner en détail l'intervention réalisée, en analysant la documentation fournie, les photos et les commentaires du technicien pour s'assurer de la

conformité et de la qualité du travail effectué. Suite à cette analyse approfondie, deux chemins sont possibles : dans le cas favorable, le Superviseur valide l'intervention, ce qui déclenche automatiquement une série d'actions système comprenant la clôture officielle du ticket, l'archivage des données dans la base de données, et la mise à jour du tableau de bord pour les statistiques de maintenance. Dans le cas contraire, si l'intervention ne répond pas aux standards de qualité requis, le Superviseur peut la rejeter en fournissant une justification. Ce rejet entraîne un processus automatisé où le système archive le ticket original avec son historique complet, crée immédiatement un nouveau ticket en y attachant toutes les informations pertinentes de l'intervention précédente, et notifie le Distributeur pour une nouvelle affectation, permettant ainsi une nouvelle tentative de résolution du problème initial.

Chapitre 3

Réalisation

Ce chapitre abordera les outils que nous avons choisis d'utiliser, et présentera le résultat final de l'application.

3.1 Outils de travail

3.1.1 Le langage HTML

L'**Hypertext Markup Language**, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images et des formulaires de saisie.



FIG. 3.1 : Logo de HTML

3.1.2 Le langage CSS

Le **Cascading Style Sheets : feuilles de style en cascade** généralement abrégé **CSS**, est un langage informatique qui sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.



FIG. 3.2 : Logo de CSS

3.1.3 L'outil Bootstrap

Bootstrap est un Framework destiné aux applications web. Développé par Twitter et distribué sous licence Apache 2, c'est un outil à considérer lors du développement rapide d'applications web. L'utilisation combinée du HTML, du CSS, et du JavaScript.



FIG. 3.3 : Logo de Bootstrap

3.1.4 Le langage JavaScript

Le **JavaScript Language**, généralement abrégé **JS**, est un langage de programmation de scripts principalement utilisé dans les pages web interactives. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés.



FIG. 3.4 : Logo de JavaScript

3.1.5 L'outil Node.js

Node.js est un environnement d'exécution open source côté serveur, qui permet aux développeurs de créer et de faire fonctionner des applications côté serveur en utilisant JavaScript. Il est basé sur le moteur JavaScript V8 développé par Google et offre une architecture basée sur des événements et des opérations non bloquantes, ce qui le rend particulièrement adapté à la création d'applications web évolutives et en temps réel.



FIG. 3.5 : Node.js logo

Justification du choix : Node.js a été sélectionné pour sa performance exceptionnelle dans la gestion des opérations asynchrones et sa capacité à gérer efficacement de multiples connexions simultanées, caractéristiques essentielles pour notre application de gestion de tickets qui nécessite des interactions en temps réel et un traitement rapide des requêtes.

3.1.6 L'outil Express.js

Express.js est un framework web minimaliste et flexible pour Node.js, utilisé pour créer des applications web côté serveur. Il simplifie le processus de création d'applications web en fournissant des fonctionnalités et des outils prêts à l'emploi, notamment pour le routage, la gestion des requêtes HTTP, la création de modèles de vue, et bien plus encore.



FIG. 3.6 : Logo de express

Justification du choix : Express.js s'impose comme le choix idéal pour notre API REST grâce à sa légèreté, sa flexibilité et sa grande communauté active. Sa facilité d'intégration avec Node.js et sa riche bibliothèque de middleware permettent un développement rapide et une maintenance simplifiée de notre backend.

3.1.7 L'outil Angular

Angular est un framework open-source développé par Google pour la création d'applications web dynamiques et modernes. Utilisant TypeScript, Angular permet de construire des applications web monopage (SPA) en offrant une architecture MVC (Modèle-Vue-Contrôleur) ou MVVM (Modèle-Vue-ViewModèle). Il est conçu pour simplifier le développement d'applications complexes en fournissant des outils pour la gestion des formulaires, des routes, des requêtes HTTP, et des composants réutilisables.



FIG. 3.7 : Logo de angular

Justification du choix : Angular a été retenu pour sa robustesse et son architecture complète, particulièrement adaptée aux applications d'entreprise complexes. Son système de composants, sa gestion efficace de l'état et ses outils de développement avancés facilitent la création d'une interface utilisateur réactive et maintenable pour notre système de gestion de tickets.

3.1.8 L'outil PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet open-source. Il est reconnu pour sa robustesse, sa conformité aux standards SQL, et sa capacité à gérer de grandes quantités de données de manière efficace. PostgreSQL supporte les transactions ACID, l'intégrité référentielle, les vues, les déclencheurs, et les procédures stockées. Il est extensible, permettant d'ajouter de nouvelles fonctionnalités comme des types de données, des opérateurs, et des langages de procédure.



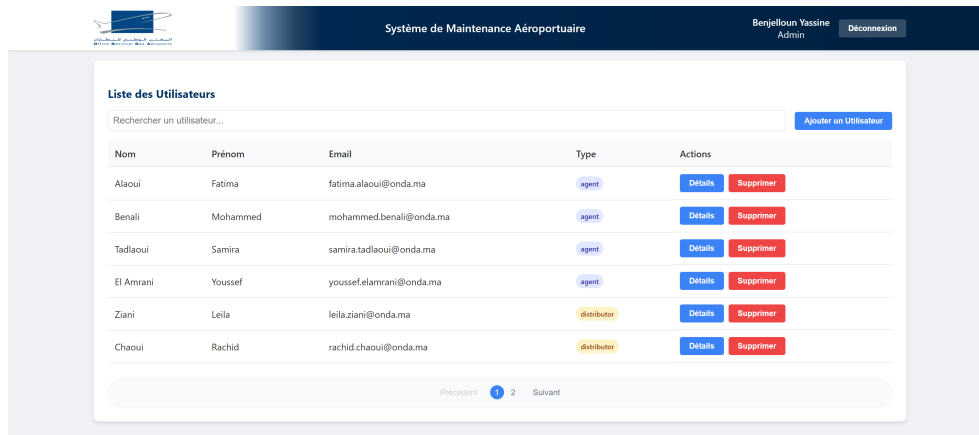
FIG. 3.8 : Logo de postgres

Justification du choix : PostgreSQL répond parfaitement aux besoins de notre application grâce à sa fiabilité, sa gestion avancée des transactions ACID et ses fonctionnalités de sécurité robustes. Sa capacité à gérer efficacement les relations complexes entre les données et sa scalabilité en font le choix optimal pour une application de maintenance aéroportuaire critique.

3.2 Réalisation de l'application

3.2.1 Tableau de bord de l'administrateur

La figure montre le tableau de bord de l'administrateur, qui permet de gérer les utilisateurs du système.



Nom	Prénom	Email	Type	Actions
Alaoui	Fatima	fatima.alaoui@onda.ma	agent	Détails Supprimer
Benali	Mohammed	mohammed.benali@onda.ma	agent	Détails Supprimer
Tadlaoui	Samira	samira.tadlaoui@onda.ma	agent	Détails Supprimer
El Amrani	Youssef	youssef.elamrani@onda.ma	agent	Détails Supprimer
Ziani	Leila	leila.ziani@onda.ma	distributeur	Détails Supprimer
Chaoui	Rachid	rachid.chaoui@onda.ma	distributeur	Détails Supprimer

FIG. 3.9 : Tableau de bord de l'administrateur

Le tableau de bord de l'administrateur affiche la liste des utilisateurs avec les informations suivantes :

- Nom : Le nom de l'utilisateur.
- Prénom : Le prénom de l'utilisateur.
- Email : L'adresse e-mail de l'utilisateur.
- Type : Le type d'utilisateur (agent, distributeur, technicien, etc.).
- Actions : Les actions disponibles pour l'administrateur, telles que la modification et la suppression de l'utilisateur.

L'administrateur peut également ajouter de nouveaux utilisateurs en spécifiant leur nom, prénom, e-mail, numéro de téléphone, matricule et type.

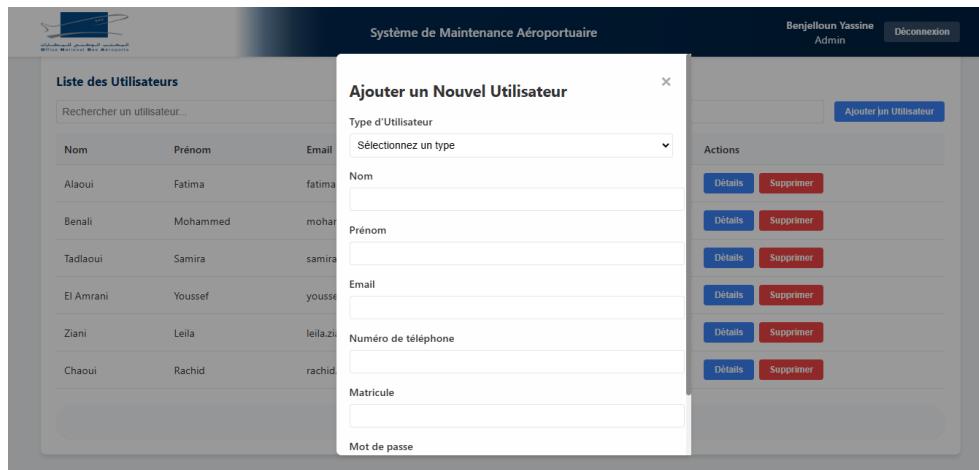


FIG. 3.10 : Ajout utilisateur

3.2.2 Tableau de bord de l'agent

La figure illustre le tableau de bord de l'agent, qui permet de créer et de suivre les tickets de maintenance.

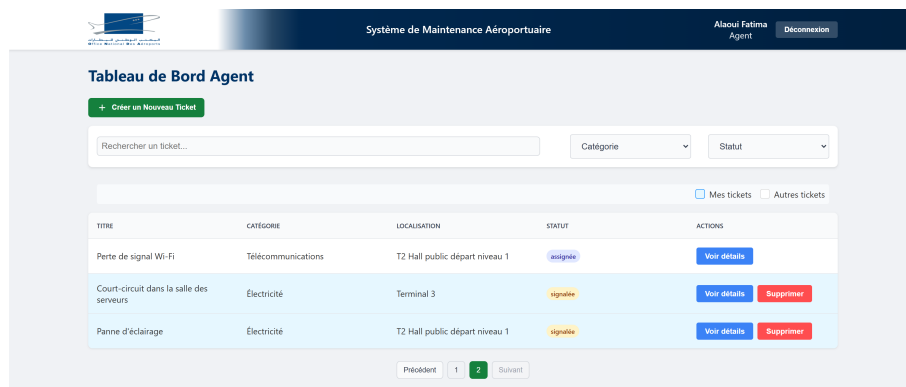


FIG. 3.11 : Tableau de bord de l'agent

Le tableau de bord Agent constitue l'interface principale permettant aux agents de gérer et suivre les tickets de maintenance au sein du système. Nous avons implémenté un système de différenciation visuelle simple mais efficace : les tickets créés par l'agent connecté apparaissent sur un fond bleu clair, tandis que les tickets créés par d'autres agents sont affichés sur un fond blanc. le tableau affiche également les informations :

- Titre : Le titre descriptif du ticket.
- Catégorie : La catégorie du problème (électricité, plomberie, etc.).
- Localisation : L'emplacement où le problème a été signalé.
- Statut : L'état actuel du ticket (signalé, assigné, en cours, résolu, etc.).
- Actions : Les actions disponibles pour l'agent, telles que la visualisation des détails du ticket.

3.2.3 Création d'un nouveau ticket

La figure montre le formulaire de création d'un nouveau ticket par l'agent.

Pour créer un nouveau ticket, l'agent doit fournir les informations suivantes :

- Titre : Un titre descriptif du problème.
- Catégorie : La catégorie du problème (électricité, plomberie, etc.).
- Localisation : L'emplacement où le problème a été signalé.
- Description du problème : Une description détaillée du problème rencontré.
- Priorité : Le niveau de priorité du ticket (normale ou urgente).
- Image : Une image illustrant le problème (facultatif).

Une fois le ticket créé, il apparaît dans le tableau de bord de l'agent avec le statut "signalé".

The screenshot shows the 'Créer un Nouveau Ticket' form. The fields are filled as follows:

- Titre:** Panne d'éclairage
- Catégorie:** Électricité
- Localisation:** T2 Hall public départ niveau 1
- Description du Problème:** Plusieurs luminaires ne fonctionnent pas dans la zone d'enregistrement du Terminal 2. Cela crée des zones sombres et gêne les passagers
- Priorité:** Urgente
- Image:** Choose File: panne.jpeg

FIG. 3.12 : Création d'un nouveau ticket

3.2.4 Tableau de bord du distributeur

La figure présente le tableau de bord du distributeur, qui permet de visualiser et de gérer les tickets de maintenance.

Le tableau de bord affiche les informations suivantes pour chaque ticket :

- ID : L'identifiant unique du ticket.
- Titre : Le titre descriptif du ticket.
- Emplacement : L'emplacement où le problème a été signalé.
- Priorité : Le niveau de priorité du ticket (normale ou urgente).
- Statut : L'état actuel du ticket (signalé, assigné, en cours, résolu, etc.).

Système de Maintenance Aéroportuaire

Ziani Leila Distributor Déconnexion

Rechercher un ticket... Tous les statuts ▼

ID	Titre	Emplacement	Priorité	Statut	Action
32	test	Taxiway T4	normale	en cours	Modifier et réattribuer
30	Dysfonctionnement des prises électriques	T1 Livraison bagages	urgente	assignée	Modifier et réattribuer
29	Interférences radio sur fréquence tour de contrôle	T2 Hall public arrivée niveau 0	normale	résolue	Modifier et réattribuer
28	Perte de signal Wi-Fi	T2 Hall public départ niveau 1	normale	assignée	Modifier et réattribuer
26	Court-circuit dans la salle des serveurs	Terminal 3	normale	signalée	Modifier et réattribuer

Précédent 1 2 Suivant

Statistiques

2 Nouveaux tickets

2 Tickets assignés

FIG. 3.13 : Tableau de bord du distributeur

- Action : Les actions disponibles pour le distributeur, telles que la modification et l'attribution du ticket.

Le tableau de bord fournit également des statistiques sur les tickets, telles que le nombre de nouveaux tickets et le temps moyen d'attribution.

Modifier et attribuer le ticket

Titre : Interférences radio sur fréquence tour de contrôle

Catégorie : Télécommunications Priorité : Normale

Emplacement : T2 Hall public arrivée niveau 0

Description : Des interférences ont été signalées sur la fréquence principale de communication avec la tour de contrôle. Cela pourrait affecter la sécurité des communications

Attribuer à un technicien :

Annuler Attribuer le ticket

FIG. 3.14 : Attribution ticket pour distributeur

3.2.5 Tableau de bord du technicien

La figure montre le tableau de bord du technicien, qui permet de visualiser les tickets qui lui sont assignés et de les traiter.

Système de Maintenance Aéroportuaire

Berrada Hassan
Technician

Déconnexion

Tableau de Bord Technicien

Rechercher un ticket... Tous les statuts ▼

■ Tickets assignés ■ Tickets de la catégorie

ID	Titre	Localisation	Priorité	Statut	Action
29	Interférences radio sur fréquence tour de contrôle	T2 Hall public arrivée niveau 0	normale	assignée	Voir détails
27	Dysfonctionnement des prises électriques	T1 Livraison bagages	urgente	assignée	Voir détails
26	Court-circuit dans la salle des serveurs	Terminal 3	normale	signalée	Voir détails
25	Panne d'éclairage	T2 Hall public départ niveau 1	urgente	signalée	Voir détails

Précédent 1 Suivant

FIG. 3.15 : Tableau de bord du technicien

Le tableau de bord du technicien affiche deux listes de tickets :

- Tickets assignés : Les tickets qui ont été assignés au technicien par le distributeur et qui nécessitent une intervention.
- Tickets de la catégorie : Les tickets qui correspondent à la catégorie d'expertise du technicien.

Pour chaque ticket, les informations suivantes sont affichées :

- ID : L'identifiant unique du ticket.
- Titre : Le titre descriptif du ticket.
- Localisation : L'emplacement où le problème a été signalé.
- Priorité : Le niveau de priorité du ticket (normale ou urgente).
- Statut : L'état actuel du ticket (assigné dans ce cas).
- Action : L'action disponible pour le technicien, qui est de consulter les détails du ticket.

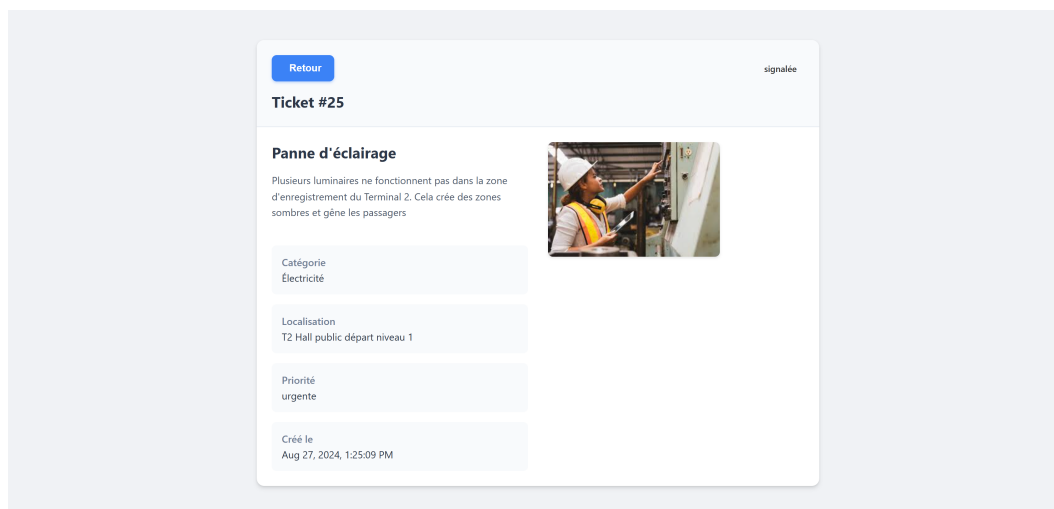


FIG. 3.16 : Details ticket

Lorsque le technicien consulte les détails d'un ticket qui lui est assigné, le statut du ticket passe automatiquement à "en cours". Le technicien peut alors effectuer l'intervention, documenter les actions réalisées et les pièces jointes, puis marquer le ticket comme résolu une fois le problème traité.

Les tickets résolus par le technicien sont ensuite soumis à la vérification du superviseur pour validation finale avant la clôture.

3.2.6 Tableau de bord du superviseur

La figure présente le tableau de bord du superviseur, qui permet de vérifier et de valider les tickets résolus par les techniciens.

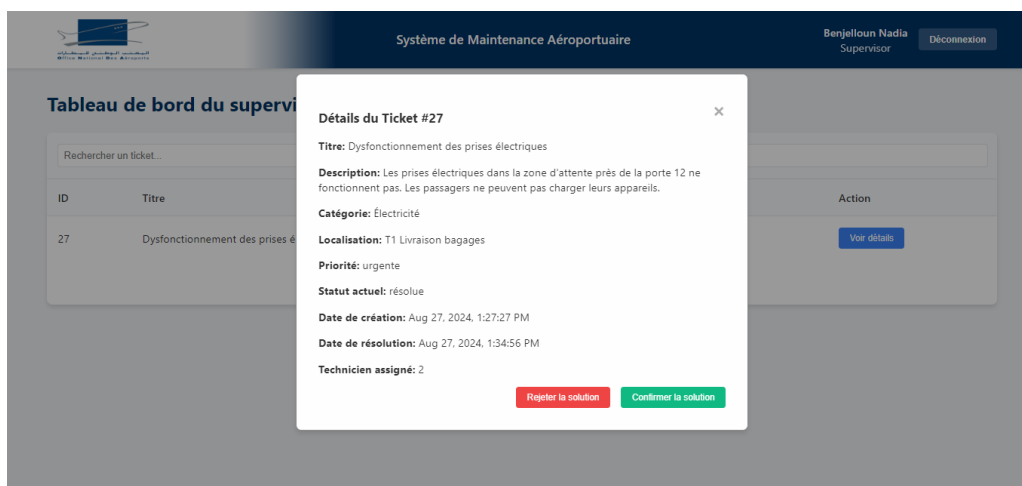


FIG. 3.17 : Tableau de bord du superviseur

Le tableau de bord du superviseur affiche les tickets résolus avec les informations suivantes :

- ID : L'identifiant unique du ticket.
- Titre : Le titre descriptif du ticket.

- Description : Une description détaillée du problème et de la solution apportée.
- Catégorie : La catégorie du problème.
- Localisation : L'emplacement où le problème a été signalé.
- Priorité : Le niveau de priorité du ticket (normale ou urgente).
- Statut actuel : L'état actuel du ticket (résolue dans ce cas).
- Date de création : La date à laquelle le ticket a été créé.
- Date de résolution : La date à laquelle le ticket a été résolu par le technicien.
- Technicien assigné : Le technicien qui a résolu le ticket.
- Action : Les actions disponibles pour le superviseur, telles que la vérification de la solution.

Le superviseur peut consulter les détails du ticket et l'image associée pour évaluer la qualité de la résolution. Il a ensuite deux options :

- Rejeter la solution : Si la solution n'est pas satisfaisante, le superviseur peut rejeter le ticket, qui retourne alors à l'état "assigné" pour que le technicien puisse apporter des corrections.
- Confirmer la solution : Si la solution est satisfaisante, le superviseur confirme la résolution du ticket, qui passe alors à l'état "clos".

3.3 Interfaces Mobiles

La mobilité est un aspect crucial de notre système de gestion de maintenance, particulièrement dans le contexte d'un aéroport où les intervenants doivent pouvoir accéder et interagir avec le système depuis différentes zones de l'infrastructure. L'implémentation d'interfaces mobiles réactives répond à plusieurs besoins essentiels :

- Permettre aux techniciens d'accéder aux détails des tickets et de documenter leurs interventions directement sur site
- Faciliter la création rapide de tickets par les agents lors de la détection d'anomalies sur le terrain
- Offrir aux superviseurs la possibilité de valider les interventions et gérer les urgences en mobilité
- Réduire les délais de traitement en permettant une interaction immédiate avec le système

3.3.1 Interface Mobile du Technicien

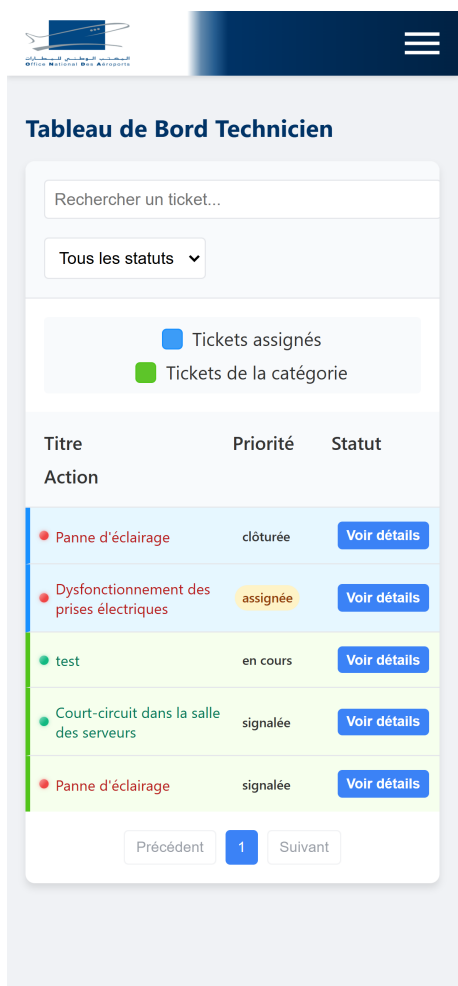


FIG. 3.18 : Liste des tickets assignés



FIG. 3.19 : Détails et traitement d'un ticket

L'interface mobile du technicien est conçue pour optimiser la gestion des interventions sur le terrain. La première vue présente une liste claire et organisée des tickets assignés, avec des indicateurs visuels de priorité et de statut. La seconde vue permet au technicien de consulter les détails complets d'un ticket et de documenter son intervention avec photos et commentaires, offrant ainsi tous les outils nécessaires pour un traitement efficace des demandes de maintenance.

3.3.2 Interface Mobile de l'Agent

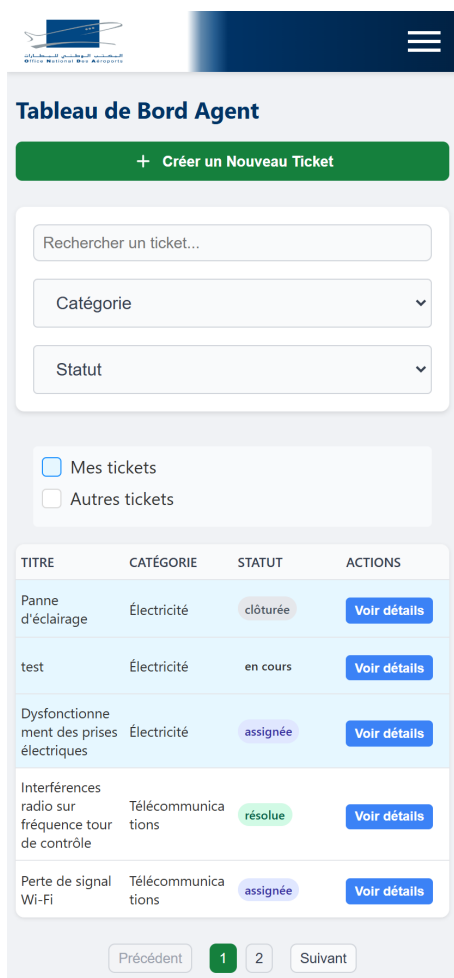


FIG. 3.20 : Création d'un nouveau ticket



FIG. 3.21 : Suivi des tickets créés

L'interface agent se concentre sur deux fonctionnalités principales : la création rapide de tickets et le suivi des demandes. L'interface de création est optimisée pour une saisie efficace des informations essentielles, avec la possibilité d'ajouter des photos directement depuis l'appareil mobile. La vue de suivi permet aux agents de monitorer l'état d'avancement de leurs tickets et d'accéder à l'historique de leurs demandes.

3.3.3 Interface Mobile du Superviseur

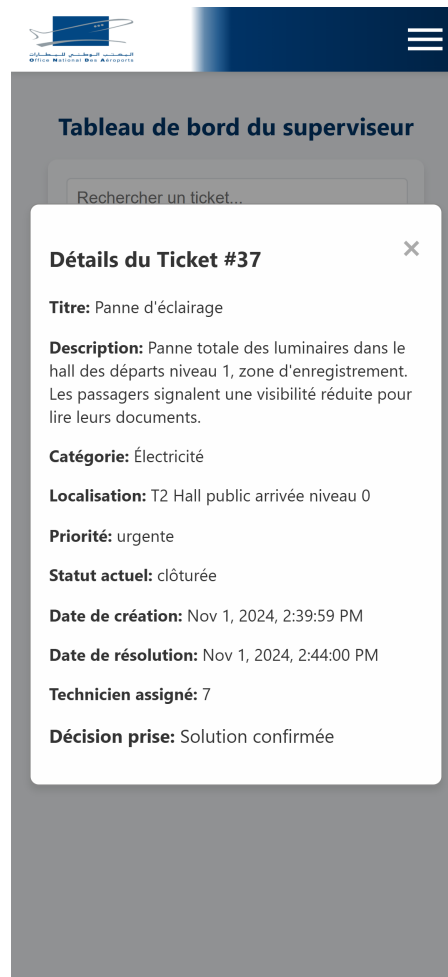


FIG. 3.22 : Interface de validation du superviseur

L'interface du superviseur est conçue pour faciliter la validation rapide des interventions terminées. Elle offre une vue synthétique des tickets en attente de validation, avec accès aux détails des interventions et aux photos associées. Les fonctionnalités de filtrage et de tri permettent une gestion efficace des priorités, tandis que les outils de validation/rejet sont optimisés pour une utilisation tactile.

Conclusion

Au cours de mon stage au sein de l'Office National Des Aéroports (ONDA), j'ai travaillé sur la conception et le développement d'une solution intelligente pour optimiser la gestion des opérations de maintenance des installations de l'aéroport Mohammed V. Ce rapport a présenté en détail ma démarche, depuis l'analyse des besoins jusqu'à la réalisation de la plateforme. La solution développée offre une interface conviviale et efficace pour la création, l'assignation et le suivi des tickets de maintenance. Elle permet une meilleure coordination entre les différents acteurs (agents, distributeurs, techniciens et superviseurs) et fournit des outils de reporting et d'analyse pour une gestion optimisée de la maintenance. Cependant, il existe encore des pistes d'amélioration pour enrichir la plateforme. L'une des perspectives envisagées est l'intégration d'un système de recommandations intelligent, capable de suggérer des actions de maintenance préventive en fonction de l'historique des interventions et des données collectées. Cela permettrait d'anticiper les problèmes potentiels et d'optimiser les ressources allouées à la maintenance.

Bibliographie

- [1] Documentation Angular, *Guide complet du framework Angular*, Google LLC, 2023. <https://angular.io/docs>
- [2] Documentation Node.js, *Guide de développement Node.js v18.x*, OpenJS Foundation, 2023. <https://nodejs.org/fr/docs/>
- [3] Documentation Express.js, *Référence API Express 4.x*, OpenJS Foundation, 2023. <https://expressjs.com/fr/>
- [4] Documentation TypeScript, *TypeScript : Guide du développeur*, Microsoft Corporation, 2023. <https://www.typescriptlang.org/fr/>
- [5] Auth0 Blog, *Authentification JWT avec Node.js et Express.js*, 2023. <https://auth0.com/blog/node-js-and-express-tutorial-building-and-securing-rest-apis/>
- [6] OpenClassrooms, *Sécurisez vos API avec JSON Web Tokens*, 2023. <https://openclassrooms.com/fr/courses/6390246-securisez-vos-applications-web-avec-les-json-web-tokens>