

**RÉPUBLIQUE DU BÉNIN**

**\*\*\*\*\*\*\*\***

**MINISTERE DE L’ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE**

**\*\*\*\*\*\*\*\***

**ÉCOLE SUPERIEURE DE GESTION D’INFORMATIQUE ET DES SCIENCES**

**\*\*\*\*\*\*\*\***

**MÉMOIRE DE FIN D’ÉTUDE POUR L’OBTENTION DES CRÉDITS ASSOCIÉS AU DIPLÔME DE LICENCE PROFESSIONNELLE EN INFORMATIQUE, RÉSEAU ET TÉLÉCOMMUNICATION**

**Filière : Informatique, Réseaux et Télécommunication**

**Option : Architecture des Logiciels**

**Système de Gestion des Equipements de Protection Individuelle (EPI) au Port Autonome de Cotonou (PAC)**

**Réalisé Par :**

Besilia Sènan AHOMAGNON & Léa Tayé AKOSSETAN

**Maître de mémoire : Maître de Stage :**

**M. da Eleison MATHA SANT’ANNA M. Verbeck DEGBESSE**

**Année académique 2024-2025**

APPROBATION DES MAÎTRES MEMOIRES ET DU MAÎTRE DE STAGE

**SIGNATURES**

**Maître de mémoire**

**M. da Eleison MATHA SANT’ANNA**

**Maître de stage**

**M. Verbeck DEGBESSE**

DEDICACES

Besilia AHOMAGNON

Je dédie ce mémoire à mon père et ma mère,

Léa AKOSSETAN

Je dédie ce mémoire à mon père et ma mère,

REMERCIEMENTS

Nous rendons grâce à Dieu, le Tout-Puissant, de nous avoir accordé la vie, la lucidité, la sagesse et les facultés intellectuelles nécessaires à l’accomplissement de ce projet de fin d’études, ainsi que d’avoir permis le bon déroulement de ce processus.

Nous exprimons également notre profonde gratitude à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Nous exprimons notre gratitude envers notre encadreur M. Eleison da MATHA SANT’ANNA, pour sa disponibilité, ses conseils éclairés, son accompagnement constant et sa bienveillance tout au long de la rédaction de ce mémoire.

Nous remercions également notre maître de stage M. Verbeck DEGBESSE pour avoir partager avec nous ses connaissances et pour ses conseils qui ont été une aide précieuse.

Nos sincères remerciements aux membres du corps professoral de l’ESGIS, et en particulier le Dr. Béthel ATOHOUN, pour leurs précieux enseignements et leur soutien académique.

Nos remerciements vont également à nos professeurs pour leurs enseignements et leurs conseils.

Enfin, nous remercions profondément, nos différentes familles, pour leur amour, leur patience et leur soutien indéfectible tout au long de nos études.

GLOSSAIRES ET SIGLES

* API : Application Programming Interface
* API REST : Representational State Transfer API
* CI/CD : Continuous Integration / Continuous Deployment
* CSS : Cascading Style Sheets
* DI : Dependency Injection
* DQHSE : Direction/ Directeur de la Qualité, de l’Hygiène, de la Sécurité et de l’Environnement
* EcoOnline EHS : EcoOnline Environment, Health, and Safety
* EPI : Équipement de Protection Individuelle
* GEC : Gestion Électronique de Courrier
* HTML : HyperText Markup Language
* HTTP : HyperText Transfer Protocol
* IoC : Inversion of Control
* JPA : Java Persistence API
* MariaDB : Maria Database
* MVC : Model-View-Controller
* MySQL : My Structured Query Language
* PAC : Port Autonome de Cotonou
* POO : Programmation Orientée Objet
* UML : Unified Modeling Language
* UI/UX : User Interface / User Experience
* XAMPP : Cross-platform, Apache, MariaDB, PHP, Perl

RESUME

Dans un contexte professionnel où la sécurité des travailleurs est une exigence primordiale, le Port Autonome de Cotonou (PAC) fait face à d’importantes difficultés liées à la gestion manuelle des Équipements de Protection Individuelle (EPI). Ce mode de gestion, fondé sur des registres papier et des démarches physiques, engendre des retards dans les livraisons, un manque de traçabilité et un risque accru d’erreurs ou de ruptures de stock.

Ce mémoire propose une solution numérique innovante visant à automatiser et optimiser la gestion des EPI au sein du PAC. Après une analyse approfondie des insuffisances du système existant, une plateforme web a été conçue et développée à l’aide de technologies modernes telles que Spring Boot pour le backend, React et Tailwind CSS pour le frontend, et MySQL pour la base de données. Cette solution repose sur une architecture microservices, conforme au modèle MVC, et intègre des fonctionnalités clés telles que : gestion des demandes, traitement et validation par le DQHSE, gestion des livraisons, traçabilité des stocks, alertes automatiques, authentification sécurisée et génération de rapports.

Le projet a été conduit au sein du Département des Systèmes d’Information du PAC, dans le cadre d’un stage académique. Il a permis aux auteurs d’appliquer leurs compétences en programmation orientée objet (POO), modélisation UML, développement web, intégration continue (CI/CD), et collaboration en environnement professionnel.

En plus de répondre aux besoins opérationnels du PAC, cette application pose les bases d’une transformation digitale plus large, en renforçant la sécurité du personnel, la fiabilité des processus, et la qualité de service. Ce travail constitue une contribution concrète à l'amélioration des systèmes d’information dans un environnement portuaire stratégique.

ABSTRACT

In a professional context where worker safety is paramount, the Autonomous Port of Cotonou (PAC) faces significant challenges due to the manual management of Personal Protective Equipment (PPE). This traditional paper-based system leads to delivery delays, poor traceability, and an increased risk of errors and stock shortages.

This thesis presents an innovative digital solution designed to automate and optimize PPE management at PAC. Following a thorough analysis of the existing system’s shortcomings, a web platform was developed using modern technologies such as Spring Boot for the backend, React and Tailwind CSS for the frontend, and MySQL for data storage. The solution is based on a microservices architecture adhering to the MVC model, featuring key functionalities including request management, validation by the QHSE department, delivery tracking, stock traceability, automated alerts, secure authentication, and report generation.

Conducted within the PAC’s Information Systems Department during an academic internship, this project allowed the authors to apply object-oriented programming, UML modeling, web development, continuous integration (CI/CD), and professional collaboration skills.

Beyond addressing PAC’s operational needs, this application lays the groundwork for a broader digital transformation by enhancing personnel safety, process reliability, and service quality. It constitutes a concrete contribution to the improvement of information systems in a strategic port environment.

SOMMAIRE

[**INTRODUCTION 1**](#_Toc201154315)

[CHAPITRE I : PRESENTATION DU CADRE INSTITUTIONNEL DE STAGE 3](#_Toc201154316)

[I-1- PRESENTATION GENERALE DU PORT AUTONOME DE COTONOU (PAC) 4](#_Toc201154317)

[I-2- DEROULEMENT DU STAGE 11](#_Toc201154318)

[I-2-1-Travaux effectués 11](#_Toc201154319)

[I-2-2- Acquis du Stage 12](#_Toc201154320)

[I-2-3- Difficultés rencontrées 13](#_Toc201154321)

[CHAPITRE II : CADRE THEORIQUE ET METHODOLOGIQUE 14](#_Toc201154322)

[II-1- Étude de l’existant 15](#_Toc201154323)

[II-1-1 Présentation de l’existant 15](#_Toc201154324)

[II-1-2 Critique de l’existant 16](#_Toc201154325)

[II-2- Approche de solution 17](#_Toc201154326)

[II-2-1 État de l’art 17](#_Toc201154327)

[II-2-2 Proposition d’approche de solution 18](#_Toc201154328)

[II-3 Cahier des charges fonctionnel 18](#_Toc201154329)

[II-3-1 Cahier des charges fonctionnel 18](#_Toc201154330)

[II-3-2 Clarification conceptuelle 21](#_Toc201154331)

[II-3-3 Démarche méthodologique 22](#_Toc201154332)

[CHAPITRE III : MISE EN ŒUVRE DE LA SOLUTION ET TESTS 24](#_Toc201154333)

[III-1 Étude Comparative ou Analyse Informatique 25](#_Toc201154334)

[III-1-1 Choix de l’approche d’analyse informatique 25](#_Toc201154335)

[III-1-2 Diagramme UML 27](#_Toc201154336)

[III-1-2-1 Diagramme de cas d’utilisation 27](#_Toc201154337)

[III-1-2-2 Diagramme des classes 32](#_Toc201154338)

[III-1-2-3 Diagramme d’activité 34](#_Toc201154339)

[III-2 Mise en Œuvre de la Solution 36](#_Toc201154340)

[III-2-1 Choix Techniques 36](#_Toc201154341)

[III-2-2 Techniques Utilisées 37](#_Toc201154342)

[III-2-2-1 Frontend 38](#_Toc201154343)

[III-2-1-2 Backend 39](#_Toc201154344)

[III-3 : Résultats et Discussion 43](#_Toc201154345)

[III-3-1 : Captures d’écrans 43](#_Toc201154346)

[III-3-2 : Tests, discussion 43](#_Toc201154347)

[**CONCLUSION** 45](#_Toc201154348)

LISTE DES TABLEAUX

Tableau 1 : Diagrammes UML utilisés.........................................................................26

Tableau 2 : Acteurs du système.....................................................................................28

Tableau 3 : Étude comparatif des technologies disponibles..........................................38

LISTE DES FIGURES

Figure n°1 : Organigramme du Port Autonome de Cotonou (PAC) .....................................5

Figure n°2 : Diagramme de cas d’Utilisation................................................................29

Figure n°3 : Diagramme de classe.................................................................................33

Figure n°4 : Diagramme de l’activité “Créer un compte”.............................................34

Figure n°5 : Diagramme de l’activité “Traiter une demande”......................................35

Figure n°6 : Diagramme de l’activité “Traiter une demande”......................................35

INTRODUCTION

**INTRODUCTION**

La gestion des Équipements de Protection Individuelle (EPI) constitue un élément fondamental des politiques de sécurité dans toute organisation soucieuse de la santé et du bien-être de son personnel. Au-delà de leur simple distribution, ces équipements doivent faire l’objet d’un suivi rigoureux afin d’assurer leur disponibilité, leur conformité et leur efficacité.

Dans un contexte où la transformation digitale devient incontournable pour optimiser les processus internes, le Port Autonome de Cotonou, structure stratégique dans la chaîne logistique régionale, fait face à des défis importants dans la gestion quotidienne des EPIs : manque de traçabilité, lenteur dans le traitement des demandes, absence d’indicateurs fiables, etc.

Ce mémoire s’inscrit dans une démarche d’amélioration continue et propose le développement d’un système informatisé visant à moderniser l’ensemble du cycle de vie des EPIs au PAC. Nous y décrivons la problématique initiale, la méthodologie adoptée, les choix technologiques effectués, ainsi que les résultats obtenus dans le cadre de ce projet innovant.

CHAPITRE I : PRESENTATION DU CADRE INSTITUTIONNEL DE STAGE

I-1- PRESENTATION GENERALE DU PORT AUTONOME DE COTONOU (PAC)

I-1-1- Historique

Aux Xe siècle, Cotonou était un petit village côtier du Dahomey (ancien nom du Bénin). Avec l'arrivée des colons français, l'idée de développer un port à Cotonou a émergé pour faciliter le commerce avec l'arrière-pays et promouvoir l'influence économique de la France dans la région. Ainsi, en 1863, la construction du port a débuté, marquant le point de départ de son histoire.

Avec l'accession du Bénin à l'indépendance en 1960, le port de Cotonou est devenu un pilier essentiel de l'économie nationale. Il a joué un rôle central dans les échanges commerciaux du pays avec d'autres nations africaines et européennes. Le gouvernement béninois a progressivement investi dans le développement et la modernisation du port, afin d'améliorer ses infrastructures et d'accroître sa capacité.

Depuis lors, le Port Autonome de Cotonou a continué d'évoluer et de se développer. Des projets d'expansion et de modernisation ont été entrepris pour accroître sa capacité, améliorer l'efficacité des opérations et renforcer sa compétitivité régionale. Ces efforts ont permis de consolider la position du port en tant que plaque tournante commerciale majeure en Afrique de l'Ouest.

Situé dans la zone portuaire, le PAC est limité au Nord par le boulevard de la Marina, au sud par l'océan Atlantique, à l'Ouest par l'esplanade des AMAZONE, à l'Est par la traverse Est. C'est l'établissement public chargé de la gestion de toutes les infrastructures et supers structures que constitue le Port de Cotonou.

**1. Organigramme du PAC**

La *Figure n°1* présente l’organigramme du PAC structuré en Directions et Départements comprenant des services pour une exécution optimale de ses activités.



*Figure n°1 : Organigramme du Port Autonome de Cotonou (PAC)*

Une présentation générale de cette structure se présente comme suit :

**A. Le Conseil d'Administration (CA)**

Disposant de tous les pouvoirs pour agir en toute circonstance au nom de l'entreprise, le conseil d'administration élabore la politique générale du PAC en conformité avec le plan de développement économique et social tout en agissant sur les objectifs à atteindre. Il est composé de dix (10) membres que sont :

✔ Sept (07) Administrateurs :

* Le Représentant du Ministre du Cadre de Vie et des Transports
* Le Représentant du Ministre du Développement et de la Coordination de l'Action Gouvernementale
* Le Directeur des Affaires Portuaires, Maritimes et Fluvio-Lagunaires
* Le Représentant du Ministre de l'Économie et des Finances
* Le Représentant de la Chambre de Commerce et d'Industrie du Bénin
* Le Représentant de la Caisse des Dépôts et Consignations du Bénin
* Le Représentant de la Présidence de la République

✔ Trois (03) Observateurs :

* Le Représentant de la République du Niger
* Le Représentant de la République du Burkina Faso
* Le Représentant de la République du Mali

**B. La Direction Générale (DG)**

Elle a pour mission d'assurer la gestion complète de la société en vue de sa modernisation. Elle est composée de :

* Un Secrétariat Particulier
* Un Département du Système d'Information Portuaire
* Un Secrétariat Général

**C. Direction de Contrôle des Marchés Publics (DCMP)**

La Direction du Contrôle des Marchés Publics est chargée d'assurer la mission de contrôle des marchés publics conformément à la réglementation en vigueur notamment le code des marchés publics et ses décrets d'application.

**D. Direction des Marchés Publics (DMP)**

La Direction des Marchés Publics est chargée d'assurer l'ensemble des procédures de passation des marchés du PAC, dans le respect des dispositions du code des marchés publics et des textes d'application. Elle comprend deux (02) départements dont :

* Le Département de la Passation des Marchés Publics
* Le Département de l'Exécution des Marchés Publics

**E. La Direction des Infrastructures (DI)**

La Direction des Infrastructures a pour mission d'assurer la maîtrise d'ouvrage de tous les projets d'infrastructures routières, de rénovation et d'expansion du Port de Cotonou ainsi que les Aménagements Paysagers inclus dans la Convention Tripartite. Elle comprend :

* Le Département Technique
* Le Département Infrastructures Terrestres
* Le Département Infrastructures Portuaires
* Le Service Topographie, Hydrographie et Océanographie

**F. La Direction des Opérations Portuaires et de la Sécurité (DOPS)**

Chargée d'assurer le bon déroulement des opérations portuaires ainsi que leur sécurité, cette direction est composée de :

* Le Commandant du Port
* Le Département de la Qualité, de l'Hygiène, de la Sécurité et de l'Environnement
* Le Département Remorquage et Engins Flottants

**G. La Direction de l'Audit Interne et du Contrôle Financier (DAICF)**

Elle est chargée entre autres de concevoir et de coordonner la mise en œuvre des dispositifs de l'audit interne et du contrôle financier, de coordonner l'élaboration et la vulgarisation des manuels de procédures, d'élaborer, de faire approuver et de mettre en œuvre le planning annuel des missions d'audit interne et externe. Elle est composée de :

* Le Service Audit Interne
* Le Service PMO et Rapportage
* Le Service Contrôle Interne

**H. La Direction des Affaires Juridiques et du Contentieux (DAJC)**

Cette direction a pour mission de prévenir les litiges et d'assurer la veille juridique et le suivi des concessions portuaires. Elle comporte un département et quatre (04) services que sont :

* Le Département Juridique et Compliance
* Le Service Règlementation et Contentieux
* Le Service Veille Juridique et Compliance
* Le Service Gestion des Risques et Assurances
* Le Service Suivi des Concessions

**I. La Direction Commerciale et du Marketing (DCM)**

Quant à la Direction Commerciale et du Marketing, elle s'occupe d'évaluer les performances des activités du PAC afin d'intégrer dans l'action de marketing la qualité des services rendus et, de promouvoir la coopération internationale avec les Etats-clients et autres institutions maritimes. Au sein de cette direction, sont logés :

* Le Département Facturation
* Le Service BESC et Redevances PAC-SIRAT
* Le Service Facturation Commerciale
* Le Service Intelligence Économique
* Le Service Veille Commerciale
* Le Service Communication
* Le Service Gestion du Domaine Portuaire

**J. La Direction de l'Administration et des Finances (DAF)**

Elle est chargée de la gestion administrative, financière, comptable, budgétaire et de la fiscalité du PAC. Elle assure la prévision de la trésorerie, la mobilisation des ressources financières, le contrôle des coûts, la prévision des résultats, le management général des Ressources Humaines et la Sécurité des Systèmes d'Information du PAC. Elle est structurée en quatre (04) départements à savoir :

* Le Département Comptable
* Le Département Financier
* Le Département des Ressources Humaines
* Le Département des Systèmes d'Information

Voir l'organigramme schématique du PAC en annexe

Nous nous intéresserons au Département des Systèmes d'Information, structure dans laquelle s'est déroulé mon stage.

**2. Organisation du DSI**

Le Département des Systèmes d'Information du Port Autonome de Cotonou est responsable de la production, de la maintenance et du développement du système d'information. Le système d'information comprend des applications métiers, des infrastructures techniques (réseaux, système et base de données), ainsi que les outils du centre d'assistance. La responsabilité de la sécurité des systèmes d'information du PAC incombe à la DSI. Ce dernier compte en son sein trois services :

* **Le Service Infrastructures et Réseaux (SIR)**

Le SIR a pour objectif principal est de garantir la disponibilité, la stabilité et la sécurité des infrastructures informatiques du PAC.

* **Le Service Projets et Développements (SPD)**

Ce service est responsable de la planification, du suivi et de la mise en œuvre des projets, en veillant au respect des délais, des coûts et de la qualité.

* **Le Service Supports et Applications (SSA)**

Quant au SSA, il assure le support technique et fonctionnel des utilisateurs des systèmes d'information du Port Autonome de Cotonou. Ce service répond aux demandes d'assistance, résout les problèmes techniques, effectue des diagnostics et assure la formation des utilisateurs. Il est également responsable de la gestion des applications utilisées au sein du PAC, en assurant leur disponibilité, leur évolution et leur maintenance.

I-2- DEROULEMENT DU STAGE

I-2-1-Travaux effectués

Durant notre stage au Port Autonome de Cotonou, plusieurs activités ont été réalisées sous la supervision de l’équipe informatique. Ces travaux ont été à la fois techniques et organisationnels, contribuant à renforcer nos compétences professionnelles et notre compréhension du fonctionnement d’un système d’information dans un contexte réel. Les principales tâches effectuées sont les suivantes :

* **Rédaction du cahier des charges fonctionnel et technique** : Cette étape a consisté à recenser les besoins des utilisateurs finaux pour la gestion des Équipements de Protection Individuelle (EPI), à définir les objectifs de la solution, ainsi que les contraintes techniques et fonctionnelles. Ce document a servi de base pour la conception de l’application.
* **Assistance au déploiement continu d’une plateforme d’affectation des dockers** : Nous avons participé à la mise en place d’un processus d’intégration et de déploiement continu (CI/CD) pour automatiser les mises à jour d’une application utilisée pour l’affectation des dockers. Cette tâche nous a permis de découvrir des outils comme GitLab, Jenkins et Docker.
* **Présentation d’une plateforme de Gestion Électronique des Courriers (GEC)** : Nous avons assisté à la présentation et à la démonstration d’une solution numérique permettant la gestion, la traçabilité et le suivi des courriers entrants et sortants. Ce système est utilisé pour améliorer la circulation de l’information au sein du PAC.
* **Développement de la solution de gestion des EPI** : Dans le cadre de notre projet principal, nous avons travaillé à la conception, au développement et à l’intégration d’une application web basée sur Spring Boot pour la gestion des demandes, des validations, et des livraisons d’EPI.
* Développement d’une mini-application d’avis client, permettant aux utilisateurs de soumettre leurs retours sur les services rendus, et d’une application d’affectation des étudiants à des filières, utilisée pour faciliter la répartition automatique des apprenants selon des critères prédéfinis; ces deux projets ont été réalisés dans le cadre de l’apprentissage pratique du framework **Spring Boot.**

I-2-2- Acquis du Stage

Au cours de notre stage au Port Autonome de Cotonou, nous avons pu acquérir de solides compétences à la fois techniques et professionnelles. Cette immersion dans un environnement réel de travail nous a permis de mieux comprendre les exigences d’un système d’information à l’échelle d’une grande structure, ainsi que les bonnes pratiques à adopter en entreprise.

Sur le plan technique, nous avons consolidé nos connaissances en développement web, notamment en apprenant à utiliser le framework **Spring Boot**, basé sur le langage **Java**. Nous avons pu explorer plusieurs modules essentiels de cet écosystème, tels que **Spring Security** pour la gestion de l’authentification et de l’autorisation, **Spring Web** pour la création d’API REST, ainsi que **Spring Data JPA** pour les opérations de persistance avec la base de données. Par ailleurs, l’adoption de l’**architecture microservices** nous a permis de structurer notre application en services indépendants, facilitant la scalabilité, la maintenance et le déploiement continu. Ces technologies nous ont permis de concevoir une solution logicielle performante, modulaire et adaptée aux besoins des entreprises modernes.

Nous avons également été formés à l’utilisation d’outils modernes de **Déploiement Continu (CI/CD)**, une approche qui facilite les mises à jour fréquentes et sécurisées des applications. Une autre compétence importante acquise concerne la **Gestion Electronique des Courriers (GEC)**, à travers laquelle nous avons compris l’importance de la numérisation des processus administratifs pour une meilleure traçabilité.

Sur le plan humain, nous avons appris à travailler efficacement en équipe, à nous adapter à un rythme de travail professionnel, à participer activement aux réunions, et à documenter nos avancées régulièrement. Cette expérience nous a permis de développer un sens de la rigueur, de l’autonomie et de la collaboration.

I-2-3- Difficultés rencontrées

Comme toute expérience d’apprentissage, notre stage n’a pas été exempt de difficultés. Dès le début, nous avons été confrontés à un défi technique majeur : la prise en main du **framework Spring Boot**. Bien que disposant de bases en Java, la richesse de l’écosystème Spring et la complexité de sa configuration initiale ont nécessité un temps d’apprentissage important. La compréhension des mécanismes tels que l’injection de dépendances, la sécurisation des routes, et la structuration en microservices a demandé beaucoup d’efforts.

Sur le plan relationnel, les premiers jours ont été marqués par une certaine appréhension dans notre communication avec les membres de l’équipe. Il nous a fallu du temps pour nous adapter à la culture d’entreprise, aux outils de travail collaboratif, et aux méthodes de suivi utilisées par nos encadreurs. Cependant, grâce à leur disponibilité et à l’ambiance professionnelle, nous avons progressivement gagné en confiance.

Par ailleurs, la gestion simultanée de plusieurs activités telles que la rédaction du cahier des charges, le développement de l’application et la participation à d’autres projets internes a parfois constitué un défi en termes de gestion du temps et des priorités.

**CHAPITRE** II : CADRE THEORIQUE ET METHODOLOGIQUE

II-1- Étude de l’existant

II-1-1 Présentation de l’existant

Face à l'évolution rapide et inexorable des technologies numériques, l'automatisation des processus est devenue une impérative stratégique pour toute organisation, quel que soit son secteur d'activité. Cette transformation permet non seulement une optimisation des opérations, mais assure également une adaptabilité cruciale dans un environnement en constante mutation.

Actuellement, au sein du Port Autonome de Cotonou (PAC), la gestion des Équipements de Protection Individuelle (EPI) demeure ancrée dans des pratiques traditionnelles, révélant des marges d'amélioration significatives. Concrètement, le suivi des stocks est laborieusement effectué dans des registres papier, une méthode sujette aux erreurs et difficilement consultable en temps réel. De même, le processus de demande d'EPI requiert des déplacements physiques, engendrant une perte de temps considérable pour le personnel. En cas d'indisponibilité de l'équipement sollicité, un nouveau déplacement s'avère nécessaire, prolongeant inutilement le délai d'obtention.

Par ailleurs, le circuit de demande actuel impose une contrainte supplémentaire : tout employé du PAC, bien qu'ayant droit à des EPI, doit impérativement soumettre sa requête par l'intermédiaire de son chef de département. Cette centralisation du processus introduit un point de blocage potentiel. En cas d'indisponibilité ou d'oubli de la part du supérieur hiérarchique, l'employé se trouve dans l'obligation de patienter, ce qui peut impacter la fluidité des opérations et potentiellement compromettre la sécurité.

Un aspect crucial qui complexifie davantage la situation est l'absence de suivi structuré de la livraison des EPI. Une fois la demande approuvée (et si l'équipement est disponible), la distribution des EPI aux employés ne fait pas l'objet d'un enregistrement systématique et centralisé. Il n'existe pas de mécanisme fiable pour confirmer la réception effective par l'employé, ni pour assurer la traçabilité des équipements distribués.

II-1-2 Critique de l’existant

Ce mode de fonctionnement traditionnel dans la gestion des Équipements de Protection Individuelle au niveau du Port Autonome de Cotonou présente plusieurs faiblesses majeures qui méritent d'être soulignées :

* **Inefficacité et perte de temps :** Les déplacements physiques pour les demandes et la consultation manuelle des stocks sont chronophages et détournent le personnel de tâches à plus forte valeur ajoutée.
* **Manque de visibilité et de suivi en temps réel :** La tenue des registres papier ne permet pas d'avoir une vision claire et actualisée de l'état des stocks, rendant difficile l'anticipation des besoins, la prévention des ruptures ainsi que des alertes des dates de péremption.
* **Risque d'erreurs et de pertes :** La gestion manuelle des informations est plus susceptible aux erreurs de saisie, aux oublis et aux difficultés de traçabilité des EPI distribués.
* **Dépendance et points de blocage :** Le processus de demande centralisé sur les chefs de département introduit une dépendance qui peut entraîner des délais et des frustrations en cas d'indisponibilité ou de surcharge de travail de ces derniers.
* **Difficulté d'accès et de transparence pour les employés :** Les employés n'ont pas un accès direct et transparent à l'état des stocks ou à l'historique de leurs propres dotations en EPI.
* **Absence de suivi de la livraison :** Le manque de traçabilité dans la distribution des EPI rend difficile la confirmation de la réception par l'employé, augmente le risque de perte ou de non-distribution effective, et complique l'audit et la responsabilisation.
* **Impact potentiel sur la sécurité :** Les délais dans l'obtention des EPI nécessaires, combinés à l'incertitude quant à leur livraison effective, peuvent potentiellement exposer les employés à des risques professionnels accrus.
* **Frein à l'adaptabilité et à l'optimisation :** L'absence d'outils numériques entrave la collecte de données fiables et l'analyse des tendances de consommation d'EPI, limitant ainsi les possibilités d'optimisation des coûts et de planification stratégique.

II-2- Approche de solution

Suite aux constats récurrents liés aux insuffisances de la gestion actuelle des Équipements de Protection Individuelle (EPI), le Port Autonome de Cotonou (PAC) nous a officiellement sollicités pour réfléchir à une solution informatique permettant d'améliorer ce système. En réponse à cette demande, nous avons effectués des recherches approfondies, et avions fais le point des solutions existantes sur le marché.

II-2-1 État de l’art

Parmi les applications existantes, **OpenSafe** est une plateforme collaborative dédiée à la gestion QHSE, incluant un module spécifique pour la gestion des EPI. Elle propose une centralisation des données, un suivi en temps réel des dotations, ainsi qu’une gestion fine des alertes liées aux renouvellements et contrôles. **Registre-EPI** se concentre sur la digitalisation du registre des EPI avec une interface intuitive. Il facilite la traçabilité des équipements depuis leur attribution jusqu’à leur retour ou renouvellement. **BluKanGo** est une solution complète de gestion QHSE et RSE, offrant un suivi des EPI intégré à une gestion globale des risques. Elle se distingue par ses fonctionnalités de workflow automatisés, sa capacité à gérer plusieurs sites et utilisateurs, ainsi que par ses outils d’analyse et de reporting avancés. **Blubthings** est une plateforme innovante qui mise sur la mobilité et la connectivité des objets. Elle permet notamment un suivi des EPI via des dispositifs connectés (IoT), offrant une traçabilité en temps réel et une gestion automatisée des stocks et des alertes. Cette approche technologique est particulièrement pertinente pour les environnements industriels modernes.

Ces solutions présentent toutes des atouts indéniables pour la gestion précise des équipements, leur attribution, leur état et leur renouvellement. Elles proposent également des fonctionnalités complémentaires telles que la gestion des utilisateurs, la centralisation des données et l’interfaçage avec d’autres modules QHSE. Malgré leurs atouts, ces solutions présentent certaines limites dans le contexte spécifique d’une organisation publique ou portuaire comme le Port Autonome de Cotonou (PAC) :

**OpenSafe** offre une bonne modularité mais peut demander une adaptation importante pour les spécificités portuaires.

**Registre-EPI** est facile à prendre en main mais manque de fonctionnalités avancées de gestion proactive.

**BluKanGo** est très complet mais plus adapté aux grandes structures et nécessite un accompagnement pour la personnalisation.

**Blubthings** innove par l’IoT mais engendre un coût et une complexité technique non négligeables.

II-2-2 Proposition d’approche de solution

Face aux limites des solutions existants et dans le souci de disposer d’un outil pleinement adapté aux spécificités organisationnelles, nous proposons de développement d’une plateforme numérique intégrée dédiée à la gestion des Équipements de Protection Individuelle (EPI). Cette solution offrira un contrôle total sur les accès, l’infrastructure et la sécurité des données tout en étant alignée sur les processus métier du Port Autonome de Cotonou (PAC). Elle visera à centraliser et automatiser l’ensemble du cycle de gestion des EPI : de la demande à l’attribution, en passant par le suivi et la gestion des stocks. Par ailleurs, elle renforcera la traçabilité, la conformité réglementaire ainsi que la sécurité du personnel.  
Conçue pour être intuitive et contextualisée, cette plateforme contribuera à améliorer significativement l’efficacité opérationnelle des actions liées à la sécurité au sein du PAC.

II-3 Cahier des charges fonctionnel

II-3-1 Cahier des charges fonctionnel

Le présent cahier des charges fonctionnel décrit les objectifs spécifiques et les principales fonctionnalités attendues du système numérique de gestion des Équipements de Protection Individuelle (EPI) au Port Autonome de Cotonou. Il s’agit de mettre en place une application web intuitive, sécurisée et accessible, permettant d’automatiser, tracer et optimiser l’ensemble du processus de gestion des EPI.

* **Objectifs spécifiques**

Pour répondre aux limites de la gestion actuelle (papier, Excel, lenteurs, oublis, manque de traçabilité), la solution visée devra permettre de :

* Numériser le processus de demande et d’attribution des EPI pour gagner en rapidité et en clarté ;
* Améliorer le suivi des stocks et des dotations grâce à une base de données centralisée et mise à jour en temps réel ;
* Automatiser les notifications et rappels pour éviter les ruptures de stock, les péremptions et les oublis de renouvellement ;
* Garantir une traçabilité complète des opérations pour répondre aux exigences de conformité et de sécurité ;
* Renforcer la sécurité d’accès grâce à un système d’authentification adapté aux différents profils utilisateurs.
* **Fonctionnalités principales attendues**
* **Gestion des demandes d’EPI :**
* Chaque employé doit pouvoir soumettre une demande d’EPI en ligne à partir de son compte personnel.
* Les formulaires doivent être simples à remplir, avec choix du type d’EPI, justification, et possibilité de suivre l’état d’avancement de la demande.
* Un système de suivi en temps réel permet à l’agent de savoir si sa demande est en attente, validée ou rejetée, livraison en attente ou livrée.
* **Traitement des demandes par le DQHSE :**
* Le gestionnaire (DQHSE) accède à une interface dédiée pour consulter, analyser et traiter les demandes reçues.
* Il peut approuver ou rejeter une demande avec ajout de commentaires si nécessaire.
* En cas de validation, l’attribution de l’EPI est automatiquement enregistrée dans le système, avec la date, le type d’équipement et le nom du bénéficiaire.
* **Suivi des stocks et gestion logistique**
* L’application doit permettre une visualisation en temps réel des niveaux de stock par type d’EPI.
* Chaque mouvement de stock (entrée, sortie, retour) est enregistré automatiquement avec la date, le responsable et le motif.
* Des alertes automatiques sont générées lorsque le stock devient faible et les dates de péremption EPI sont proches.
* **Notifications intelligentes et rappels**

Le système doit générer des notifications pour prévenir :

* Les utilisateurs lorsqu’une EPI arrive à péremption ;
* Les gestionnaires lorsqu’un stock est presque épuisé ;
* Le personnel concerné lorsqu’un renouvellement est requis selon la durée d’utilisation définie.  
  Ces alertes peuvent être envoyées par e-mail ou directement via l’application.
* **Historique et traçabilité des dotations**
* Chaque dotation d’EPI est enregistrée avec les informations suivantes : nom de l’utilisateur, type d’équipement, date de remise, durée d’utilisation, état initial.
* Il doit être possible de consulter l’historique complet des EPI attribués à chaque agent.
* **Gestion des comptes et des rôles utilisateurs**
* Le système doit permettre la création de profils avec des rôles définis : employé, gestionnaire stock, DQHSE, administrateur.
* Chaque rôle a des droits d’accès spécifiques aux fonctionnalités de l’application.
* Une double authentification est prévue pour renforcer la sécurité des connexions.

II-3-2 Clarification conceptuelle

**Plateforme numérique intégrée** : Une plateforme numérique intégrée est une application centrale qui regroupe plusieurs fonctionnalités en un seul outil. Elle permet aux utilisateurs de gérer facilement leurs activités sans passer d’un logiciel à un autre. Dans notre projet, cette plateforme permet de suivre la demande, la distribution, le stock et le renouvellement des EPI. Grâce à cette intégration, les services concernés (logistique, sécurité, gestion du personnel, etc.) peuvent travailler ensemble de manière plus fluide, ce qui évite les oublis, les retards ou les erreurs, et améliore la gestion globale des équipements.

**Gestion centralisée des demandes et des stocks** : La gestion centralisée signifie que toutes les informations liées aux EPI (demandes, affectations, quantités disponibles, dates de péremption, etc.) sont réunies au même endroit (dans un seul système). Cela facilite le suivi en temps réel, évite les pertes de données (comme c’est souvent le cas avec les documents papier ou Excel), et permet une meilleure prise de décision. Le responsable peut par exemple voir en un coup d'œil les équipements manquants ou ceux à renouveler prochainement.

**Traçabilité des opérations :** La traçabilité est un aspect fondamental de la gestion des EPI. La plateforme permet de suivre chaque mouvement ou action : qui a demandé quoi, quand, quel équipement a été attribué, dans quel état, et avec quelle date de péremption. En cas d’incident ou de vérification, il est facile d’accéder à l’historique complet des opérations, ce qui garantit une transparence totale. Cette traçabilité renforce la responsabilité des utilisateurs, sécurise la gestion des équipements, et facilite les audits externes ou internes.

**Interface utilisateur (UI) et expérience utilisateur (UX)** : L'interface utilisateur (UI), c’est ce que l’on voit et utilise sur l'application (boutons, menus, tableaux, formulaires, etc.). L’expérience utilisateur (UX), c’est la manière dont l’utilisateur ressent l’utilisation de l’application : est-ce simple, rapide, logique ? Dans notre projet, l’objectif est de proposer une interface claire, facile à utiliser même pour les personnes peu à l’aise avec l’informatique, afin que les demandes et les suivis d’EPI puissent se faire rapidement, sans confusion.

**Systèmes de notification et de suivi** : Ces systèmes servent à informer les utilisateurs automatiquement à différents moments : par exemple lorsqu’un stock est presque vide, quand un équipement approche de sa date de péremption, ou quand une demande a été validée. Ces notifications permettent d’agir à temps et d’éviter les oublis, ce qui améliore la réactivité et la sécurité sur le terrain.

II-3-3 Démarche méthodologique

Dans le cadre de notre stage au sein du Département des Systèmes d’Information (DSI) du Port Autonome de Cotonou, nous avons adopté une démarche méthodologique structurée pour assurer la réussite de notre projet.

1. **Analyse et compréhension du besoin :** Nous avons débuté par une phase d’analyse approfondie, au cours de laquelle nous avons collecté les exigences auprès des utilisateurs et des responsables. Cette étape a impliqué des réunions, des entretiens et l’étude des documents existants afin de bien cerner les attentes et les contraintes du projet.
2. **Conception de la solution :** Suite à cette analyse, nous avons procédé à la conception de la solution. Nous avons modélisé le système à l’aide de diagrammes UML (cas d’utilisation, classes, séquence) pour structurer le projet. Par ailleurs, nous avons conçu la base de données en utilisant MySQL, en veillant à respecter les règles d’intégrité et d’optimisation.
3. **Développement et mise en œuvre :** Le développement a été réalisé en utilisant le langage Java avec le framework Spring Boot pour créer une API REST sécurisée et performante. La base de données MySQL a été mise en place pour assurer la gestion des données. Nous avons utilisé des environnements de développement modernes, notamment Visual Studio Code et IntelliJ IDEA, facilitant l’écriture et le débogage du code.
4. **Tests et validation :** Pour garantir la qualité et la fiabilité de notre application, nous avons effectué plusieurs tests fonctionnels et techniques. Nous avons notamment utilisé Postman pour tester les différentes requêtes HTTP de notre API (GET, POST, PUT, DELETE), ce qui nous a permis de valider la conformité des fonctionnalités aux besoins exprimés.
5. **Documentation et suivi :** Enfin, nous avons rédigé une documentation technique complète afin de faciliter la maintenance et l’évolution du système. Cette documentation inclut la description des fonctionnalités, l’architecture du système, les modèles de données, ainsi que les procédures d’installation et d’utilisation.

CHAPITRE III : MISE EN ŒUVRE DE LA SOLUTION ET TESTS

III-1 Étude Comparative ou Analyse Informatique

III-1-1 Choix de l’approche d’analyse informatique

Cette section est dédiée à la conception détaillée de notre application web, une étape primordiale pour garantir la qualité et la robustesse de la solution finale. Notre objectif est de présenter une analyse fonctionnelle, structurelle et dynamique claire de l'application, en justifiant nos choix conceptuels à travers l'utilisation du Langage de Modélisation Unifié (UML).

Conformément aux exigences du processus unifié, notre modélisation s'appuiera sur les diagrammes UML. L'UML, sigle de "Unified Modeling Language", est un langage de modélisation graphique standardisé, largement utilisé en développement logiciel et en conception orientée objet. Il offre une méthode normalisée pour visualiser la conception d'un système et permet d'élaborer des modèles indépendamment de tout langage de programmation.

L'adoption de la modélisation UML présente des avantages significatifs. Elle facilite la compréhension des aspects de conception et d'architecture du logiciel par toutes les parties prenantes, y compris les clients, grâce à sa représentation visuelle et standardisée. De plus, elle optimise la maintenance et la prise en main du logiciel par d'autres développeurs, en fournissant une documentation claire et structurée de la solution.

L'UML propose une variété de diagrammes, classés en deux familles principales : Les diagrammes de structure (Ils illustrent l'organisation des modules et composants de l'application, ainsi que leurs interactions internes). Les diagrammes de comportement (Ils décrivent les différents états et les scénarios d'utilisation de la solution).

Pour modéliser notre application, nous avons spécifiquement utilisé les diagrammes UML suivants :

|  |  |  |
| --- | --- | --- |
| Diagrammes | Objectifs | Types |
| Diagramme de classes | La modélisation du système vise à représenter de manière claire et structurée l’ensemble des éléments qui composent le système ainsi que leurs interactions. Cette vue d’ensemble permet de comprendre le rôle global du système, les besoins auxquels il doit répondre et sert de base à sa conception technique. | Statique |
| Diagramme de cas d’utilisation | Le diagramme de cas d’utilisation représente le comportement du système en illustrant comment celui-ci réagit et évolue en fonction des actions ou événements qui surviennent. Il regroupe les différentes fonctionnalités du système et met en évidence les interactions entre le système et ses utilisateurs ou d’autres systèmes, en précisant les actions réalisées pour déclencher ces fonctionnalités. | Fonctionnel et Statique |
| Diagramme d’activité | Le diagramme d’activité montre l’enchaînement des différentes actions ou tâches dans le système, en décrivant comment elles se succèdent pour assurer son bon fonctionnement. Il présente de manière claire les processus métier en illustrant les étapes du travail à accomplir, les décisions à prendre ainsi que les règles qui régissent ces enchaînements, offrant ainsi une vue précise du déroulement des opérations dans le système. | Fonctionnel et dynamique |

*Tableau 1 : Diagrammes UML utilisés*

III-1-2 Diagramme UML

III-1-2-1 Diagramme de cas d’utilisation

1. **Identification des acteurs du système**

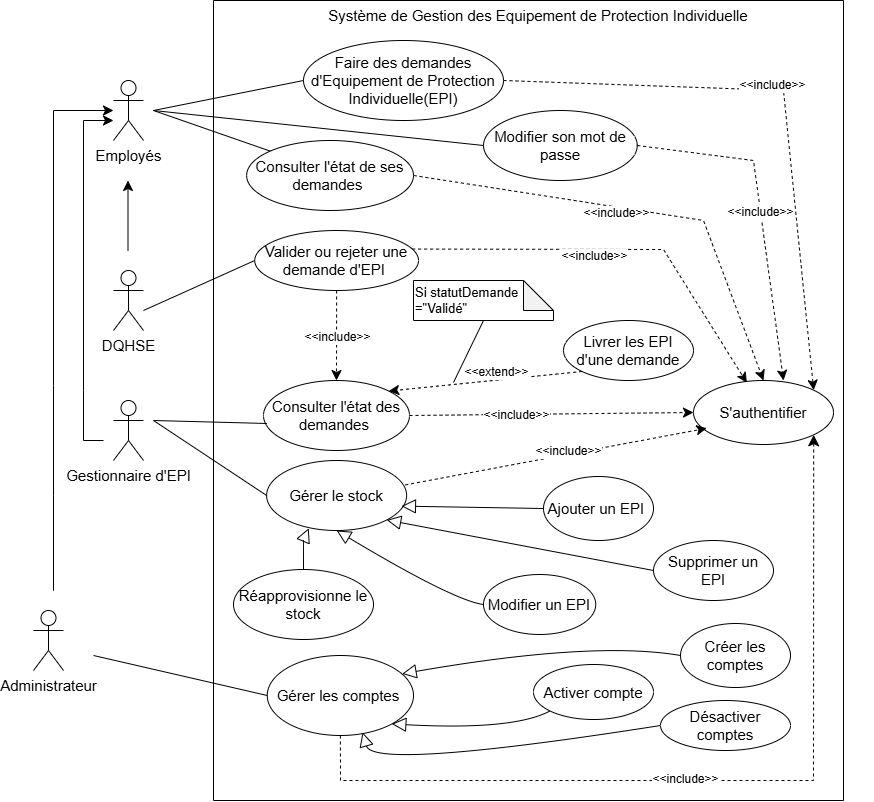
Un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé. Il joue un rôle déterminant en initiant des actions ou en sollicitant des services auprès du système. En réponse, ce dernier exécute les traitements nécessaires, qu’il s’agisse de fournir des informations, de modifier des données ou de déclencher un processus particulier.

Ainsi, les acteurs constituent les points d’entrée des fonctionnalités du système et déclenchent les interactions qui donnent vie à l’application, créant un lien dynamique entre l’utilisateur et la solution numérique. L’étude préliminaire nous a permis d’identifier les principaux acteurs intervenant dans le système. Ils sont présentés dans le tableau suivant :

|  |  |
| --- | --- |
| **Acteurs** | **Utilisation du système** |
| **Demandeur (Utilisateurs : Employés)** | - Se connecter à la plateforme  - Soumettre une demande d’EPI  - Consulter l’état de ses demandes - Accuser réception des EPI remis  - Recevoir des notifications de demande prête pour livraison |
| **Directeur QHSE** | - Se connecter à la plateforme  - Vérifier et valider (ou rejeter) les demandes d’EPI - Consulter les rapports d’attribution et de gestion du stock |
| **Gestionnaire d’EPI** | - Se connecter à la plateforme  - Enregistrer les livraisons d’EPI  - Enregistre les entrées/sorties - Gérer le stock d’EPI (  - Consulter les niveaux de stock  - Déclencher les commandes en cas de seuil critique  - Gérer les alertes de péremption ou de renouvellement) |
| **Administrateur** | - Se connecter à la plateforme  - Gérer les comptes utilisateurs - Définir les droits d’accès - Assurer la maintenance de la plateforme |

*Tableau 2 : Acteurs du système*

La *figure n°2* présente le diagramme des cas d’utilisation de notre système :



*Figure n°2 : Diagramme de cas d’Utilisation*

Ce diagramme de cas d’utilisation représente les principales interactions entre les différents acteurs du système de gestion des Équipements de Protection Individuelle (EPI) au Port Autonome de Cotonou. Il met en lumière les fonctionnalités clés de l’application, réparties selon les rôles : employés, DQHSE, gestionnaire d’EPI et administrateur. Le processus débute par l’authentification de l’utilisateur, suivie de la soumission d’une demande d’EPI. Celle-ci est ensuite traitée par le DQHSE, qui décide de sa validation. En cas d’approbation, la demande est prise en charge pour la livraison par le gestionnaire. Le système inclut également des fonctionnalités de gestion du stock, de réapprovisionnement, ainsi que de création et d’administration des comptes utilisateurs. Les relations d’inclusion et d’extension permettent de structurer les cas d’utilisation en soulignant les dépendances fonctionnelles, renforçant ainsi la cohérence et la sécurité du processus global.

1. **Identification des cas d’utilisation**

Un cas d'utilisation correspond à un ensemble d'actions réalisées par le système en réponse à une demande spécifique d'un acteur. Il représente les différentes interactions entre l'acteur et le système, aboutissant à un résultat observable. On l’identifie en recherchant les différentes interactions avec lesquelles un acteur utilise le système.

Dans notre cas d’étude, nous distinguons de plusieurs cas d’utilisation comme évoqués dans les descriptions fonctionnelles. En voici quelques exemples significatifs :

**Cas 1 : S’authentifier**

* **Acteurs :** Tous les utilisateurs
* **Description :** Permet à l'utilisateur de se connecter à la plateforme en créant une session de navigation sécurisée.
* **Précondition :** L’utilisateur dispose d’un compte valide (identifiants corrects, compte actif)
* **Scénario nominal :**

1. L’utilisateur accède à la plateforme web ;
2. Le système affiche le formulaire de connexion ;
3. L’utilisateur saisit son NOM d’UTILISATEUR et son MOT DE PASSE ;
4. Le système vérifie l'authenticité des identifiants ;
5. Si l’utilisateur est reconnu, le système le redirige vers son tableau de bord ;

* **Scénario alternatif :**
* L’utilisateur informe l’administrateur de la situation ;
* **Scénario d’Exception 1 : Identifiants incorrectes**

1. Le système affiche un message d’erreur invitant l’utilisateur à corriger et réessayer ;
2. Après 3 tentatives échouées, le système bloque temporairement le compte pour 1 heure ;
3. Après 3 blocages temporaires successifs, le système désactive définitivement le compte, et affiche un message indiquant à l’utilisateur de contacter l’administrateur ;

* **Scénario d’Exception 2 : service temporairement indisponible**

1. L’utilisateur reçoit un message d’erreur indiquant que le service indiquant que le service est temporairement indisponible.

* **Postcondition :** l'utilisateur est authentifié et a accès à son Dashboard.

**Cas 2 : Traiter une demande d’EPI**

* **Acteur** : Tous les utilisateurs du système
* **Description** : Effectuer une demande d’attribution d’un ou plusieurs EPI, la soumission, le suivi du statut (validation ou refus), et la notification des acteurs concernés.
* **Précondition** : L’utilisateur est connecté au système
* **Scénario** **nominal** :
  1. L’utilisateur accède au formulaire de demande, le remplit et le soumet ;
  2. Le système notifie le Directeur QHSE pour validation.
  3. Si la demande est à “Validée”, notifier le gestionnaire pour prise en charge ;
* **Scénario** **alternatif :**
* Si la demande est “refusée” par le Directeur QHSE, le système notifie l’utilisateur avec le motif du refus.
* **Postcondition** : La demande est soumise, validée et en attente de livraison

**Cas 3 : Réapprovisionner le stock**

* **Acteur** : Gestionnaire
* **Description** : Permet au gestionnaire de réapprovisionner le stock d’EPI en cas de stock insuffisant, ou alerte de rupture de stock.
* **Précondition** : Le gestionnaire est connecté et le système a détecté une alerte de rupture de stock.
* **Scénario nominal :**
  1. Le gestionnaire remplit et soumet le formulaire de réapprovisionnement d’EPI ;
  2. Le système transmet la demande au Directeur QHSE pour validation ;
  3. Si la demande est validée, le stock est réapprovisionné par la suite
* **Scénario** **alternatif :**

1. **Si la demande est rejetée, le système notifie le gestionnaire ;**
2. **Le gestionnaire soumet une justification plus appropriée**

* **Scénario d’exception 1 : Erreur de saisie du formulaire**
  1. Le gestionnaire soumet un formulaire incomplet ou avec des données invalides ;
  2. Le système affiche un message d’erreur et demande la correction avant validation.
* **Scénario d’exception 2 : Problème de communication avec le Directeur DQHSE**
  1. Le système ne parvient pas à notifier le Directeur QHSE (problème réseau ou serveur) ;
  2. Le système affiche une alerte au gestionnaire pour réessayer ultérieurement ou contacter l’administrateur.
* **Postcondition** : La stock est réapprovisionner ;

**Cas 4 : Livrer une demande**

* **Acteur** : Gestionnaire
* **Description** : Permet de gérer la livraison des EPI aux utilisateurs dont la demande a été validée.
* **Précondition** : Une demande d’EPI a été validée par le Directeur QHSE.
* **Scénario nominal :**

1. Le système vérifie la disponibilité des EPI ;
2. Si les EPI sont disponibles, le système notifie le gestionnaire ;
3. Le système envoie une notification au demandeur indiquant que la livraison est prête ;
4. Lorsque l’utilisateur se présente pour le retrait, le gestionnaire enregistre la livraison dans le système ;
5. Le système met automatiquement à jour le stock ;

* **Scénario alternatif :** 
  1. Si les EPI ne sont pas disponibles, le système informe le gestionnaire de l’indisponibilité ;
  2. Le gestionnaire initie une demande de réapprovisionnement.
* **Scénario d’exception :**

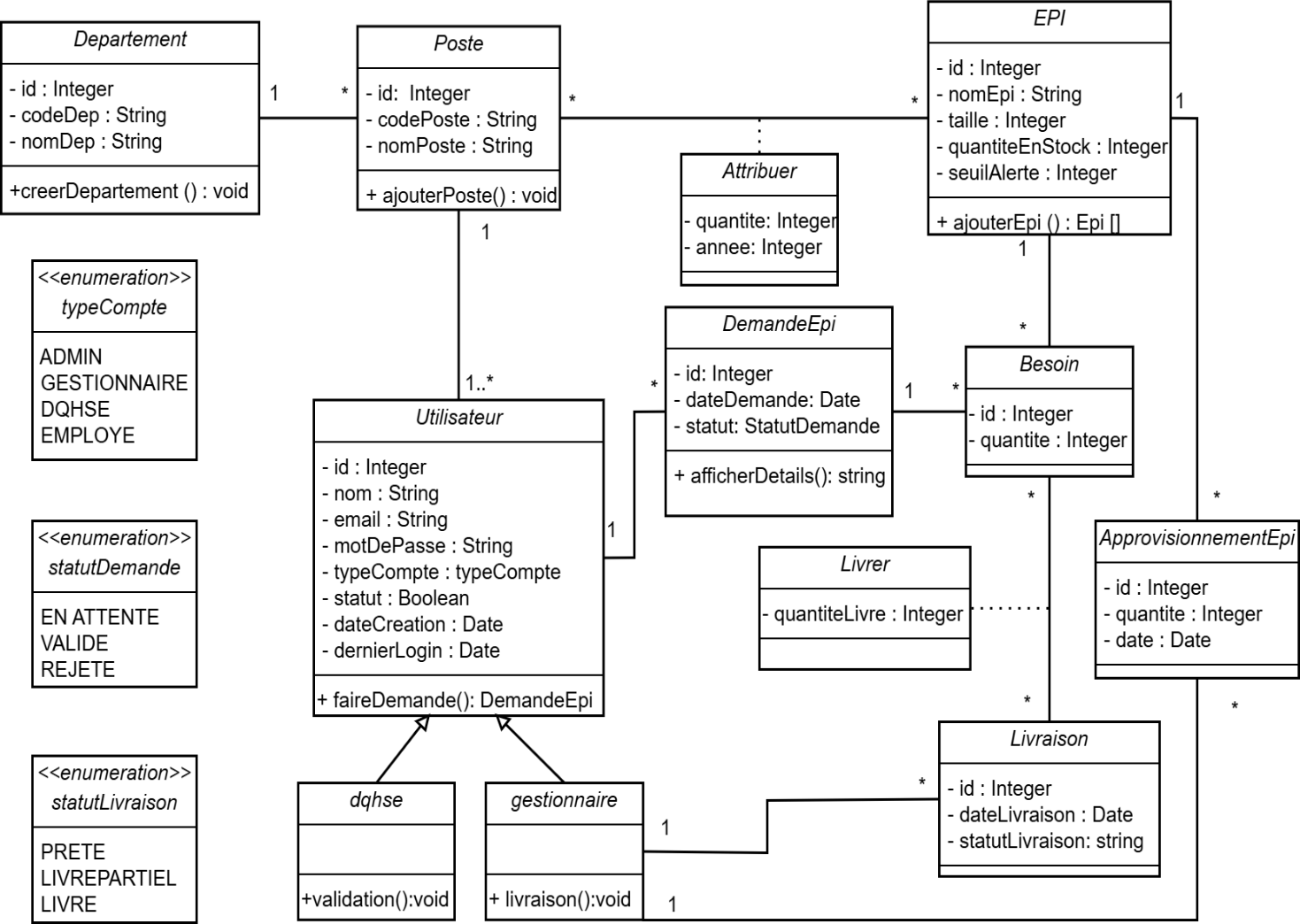
1. Après enregistrement de la livraison, si la mise à jour du stock échoue (erreur base de données) ;
2. Le système affiche une alerte critique au gestionnaire et journalise l’erreur pour intervention technique.

* **Postcondition** : L’utilisateur est livré et le stock est mise à jour.

III-1-2-2 Diagramme des classes

Le diagramme des classes est une composante essentielle de la modélisation orientée objet. Contrairement au diagramme des cas d'utilisation qui met l'accent sur les interactions entre le système et les acteurs, le diagramme des classes se concentre sur la structure interne du système. Il offre une vue abstraite des objets qui composent le système et montre comment ils interagissent pour réaliser les fonctionnalités décrites dans les cas d'utilisation. En résumé, le diagramme des classes permet de représenter de manière conceptuelle les entités et les relations clés du système, fournissant ainsi une base solide pour la compréhension et la conception de l'architecture logicielle.

La *figure* *n°3* montre une représentation du diagramme des classes de notre système :



*Figure n°3 : Diagramme de classe*

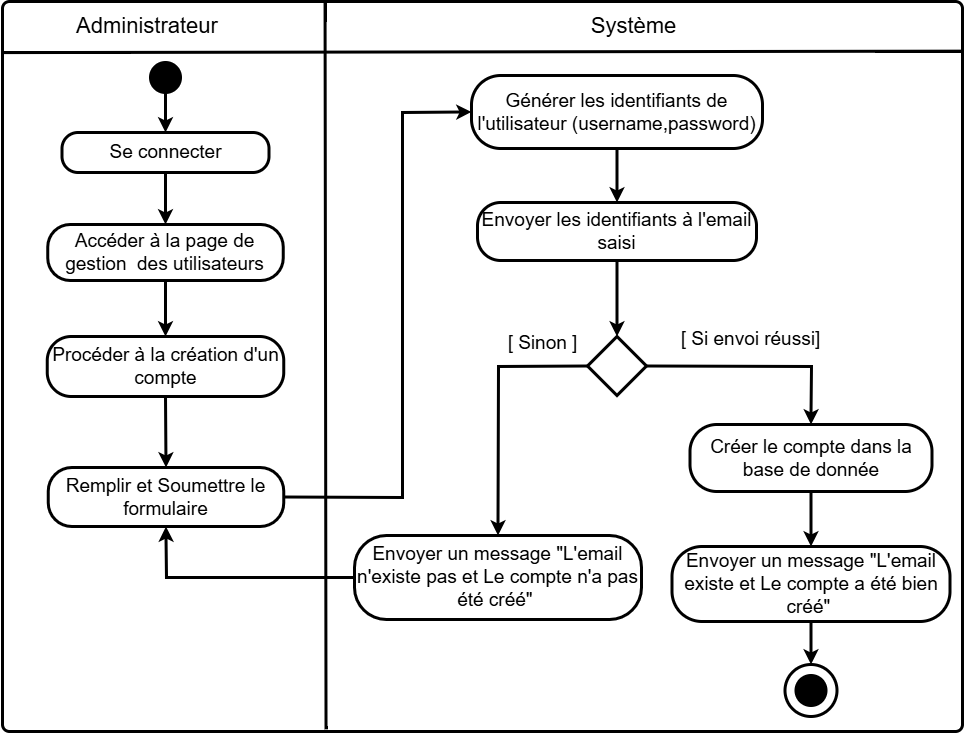
Ce diagramme de classes représente le modèle conceptuel de l’application de gestion des Équipements de Protection Individuelle (EPI) au Port Autonome de Cotonou. Il met en évidence les principales entités du système (utilisateur, demande, EPI, besoin, approvisionnement, livraison) et leurs interactions. Le processus commence par la soumission d'une demande d’EPI par un utilisateur, suivie de la validation par le DQHSE et de la livraison par le gestionnaire. Des énumérations permettent de suivre les différents statuts des demandes et des livraisons, facilitant ainsi la traçabilité et la gestion du stock d’EPI.

III-1-2-3 Diagramme d’activité

Un diagramme d'activité est une représentation graphique des processus ou des flux de travail dans un système. Il met en évidence l'ordre des activités, des décisions, et des interactions nécessaires pour accomplir une tâche ou un processus particulier. Ce type de diagramme illustre de manière claire et structurée comment les différentes étapes d'un processus sont enchaînées, y compris les éventuelles branches conditionnelles et les activités parallèles. Les diagrammes d'activité se concentrent sur la séquence des actions plutôt que sur les détails temporels ou les interactions entre objets spécifiques. Ils sont organisés de manière à suivre le flux des opérations de haut en bas, montrant clairement les transitions entre les activités et les points de décision. Le diagramme d'activité est un outil précieux pour modéliser, comprendre et améliorer les processus métier ou techniques, en fournissant une vue d'ensemble des étapes nécessaires et des possibles points de blocage ou d'amélioration dans un système.

1. **Diagramme de l’activité Créer compte utilisateur**

La *figure n°4* présente le diagramme de l’activité “Créer un compte” :

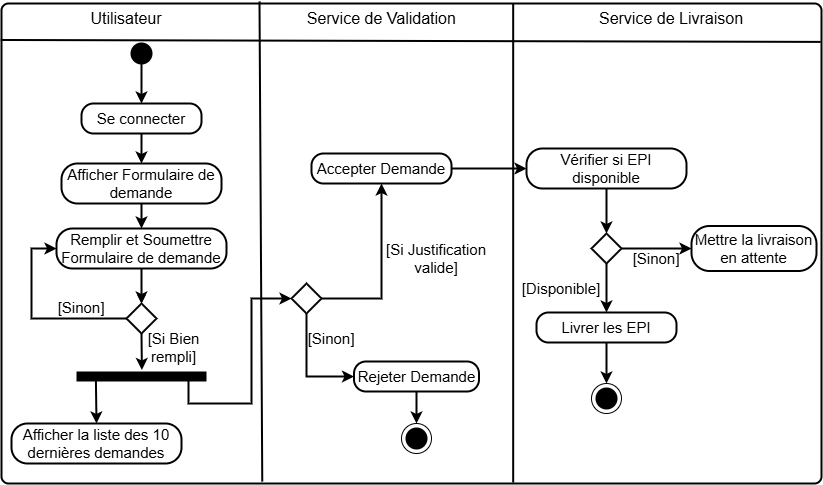


*Figure n°4 : Diagramme de l’activité “Créer un compte”*

Ce diagramme d’activité décrit le processus de création de compte utilisateur par un administrateur. Après authentification et accès à la page de gestion, l’administrateur remplit et soumet un formulaire. Le système génère ensuite les identifiants de l'utilisateur, le lui envoie par email, puis crée le compte en base de données si l’envoi est réussi. Des messages de confirmation ou d’échec sont affichés selon le résultat, assurant ainsi un retour clair à l’administrateur.

1. **Diagramme de l’activité Effectuer une demande d’EPI**

La *figure n°5* présente le diagramme de l’activité “Traiter une demande” :



*Figure n°5 : Diagramme de l’activité “Traiter une demande”*

Ce diagramme d’activité illustre le processus de demande d’EPI par un utilisateur. Après s’être authentifié, l’utilisateur remplit et soumet un formulaire. Si la demande est bien enregistrée, elle est transmise au service de validation et complétée avec un statut de validation en attente à la liste des demandes de l’utilisateur qu’il peut afficher. Une fois validée, la demande est envoyée au service de livraison. Ce dernier vérifie la disponibilité des EPI : s’ils sont en stock, la livraison est effectuée ; sinon, elle est mise en attente. La livraison peut également être rejetée si la justification ne convient pas.

III-2 Mise en Œuvre de la Solution

III-2-1 Choix Techniques

Pour notre projet, nous avons choisi d’utiliser une architecture **microservices**. Cette architecture consiste à diviser l’application en plusieurs petits services indépendants, où chaque service est chargé d’une fonctionnalité bien précise. Par exemple, un service peut s’occuper de la gestion des utilisateurs, un autre de la gestion des EPI, un autre des demandes, etc.

Chaque service fonctionne de manière autonome, comme une petite application à part entière. Cela facilite l’organisation du code, la compréhension du système, et permet de modifier ou de corriger un service sans toucher aux autres.

Cette approche rend également le projet plus souple et plus facile à faire évoluer. Elle permet de mieux répartir le travail pendant le développement, et de rendre l’application plus stable, car une erreur dans un service n’empêche pas forcément les autres de fonctionner.

Chaque microservice est développé avec Spring Boot selon l’approche MVC (Model-View-Controller), qui permet une structuration claire en séparant la logique métier, la gestion des données et l’interface utilisateur. Elle se compose de trois modules : modèle, vue, contrôleur.

• modèle : un noyau de l’application qui gère les données, permet de récupérer les informations dans la base de données, de les organiser pour qu’elles puissent ensuite être traitées par le contrôleur.

• vue : composant graphique de l’interface qui permet de présenter les données du modèle à l’utilisateur.

• contrôleur : composant responsable des prises de décision, gère la logique du code qui prend des décisions, il est l’intermédiaire entre le modèle et la vue.

Notre choix s’est porté sur cette architecture parce qu’elle permet une meilleure organisation du code, offre une possibilité de réutilisation et beaucoup plus de faciliter pour les tests unitaires.

|  |  |  |  |
| --- | --- | --- | --- |
| Technologie utilisée | Alternative(s) | Type | Pourquoi ce choix |
| React | Angular, Vue.js | Bibliothèque JS | Permet la création d'interfaces dynamiques avec des composants réutilisables. Flexible, largement adopté et soutenu par une communauté très active. |
| Tailwind CSS | Bootstrap, Material UI | Framework CSS | Offre un contrôle précis du design via des classes utilitaires. Léger, responsive, favorise une personnalisation rapide sans surcharge CSS. |
| Java | Node.js, Python | Langage backend | Langage robuste et sécurisé, adapté aux besoins des grandes entreprises. Haute scalabilité et maintenabilité. Recommandé dans le cadre du stage. |
| Spring Boot | Express (Node.js), Django | Framework backend | Facilite le développement d’applications Java. Prise en charge des microservices, configuration automatique, productivité élevée. |
| MySQL (via XAMPP) | PostgreSQL, MongoDB | SGBD relationnel | Fiable et facile à utiliser. Parfaitement compatible avec Spring Boot. Fournit un environnement local complet avec PhpMyAdmin via XAMPP. |
| Draw.io | Lucidchart, StarUML | Outil de modélisation | Gratuit, accessible en ligne, permet de créer rapidement des diagrammes clairs et professionnels pour la documentation technique. |

*Tableau 3 : Étude comparatif des technologies disponibles*

III-2-2 Techniques Utilisées

Après avoir analysé différentes technologies possibles (voir Tableau 3 ci-dessus), nous avons retenu les solutions les plus adaptées aux besoins du projet. Cette section présente en détail les technologies choisies ainsi que leur rôle dans l’implémentation du système.

III-2-2-1 Frontend

1. **React**

React est une bibliothèque JavaScript permettant de construire des interfaces utilisateur interactives et dynamiques. Basée sur une architecture de composants réutilisables, elle facilite le développement d’interfaces modulaires, maintenables et performantes. Grâce à son système de Virtual DOM, React optimise le rendu des éléments à l’écran et améliore la réactivité de l’application.

Dans notre projet, React a été utilisé pour développer l’interface utilisateur de la plateforme de gestion des EPI. Il a permis d’afficher les données en temps réel, de gérer les interactions comme la soumission de demandes, la navigation entre les pages, et la consultation des états de traitement. Sa flexibilité, combinée à un écosystème riche, nous a permis de créer une interface fluide, intuitive et adaptée aux besoins des utilisateurs du Port Autonome de Cotonou.

1. **Tailwindcss**

Tailwind CSS est un framework utilitaire qui permet d’appliquer rapidement des styles aux éléments HTML à l’aide de classes prédéfinies, sans écrire de CSS personnalisé. Dans notre projet, Tailwind CSS a été utilisé pour concevoir l’interface graphique de l’application, en facilitant la mise en page des formulaires de demande, des tableaux de données et des tableaux de bord. Grâce à sa grille flexible et ses classes responsives, il nous a permis de créer une interface épurée, cohérente et adaptée à tous types d’écrans, tout en accélérant le développement frontend et en assurant une bonne expérience utilisateur.

III-2-1-2 Backend

1. **Java**

Java est un langage de programmation orienté objet (POO), largement utilisé dans le développement d'applications professionnelles pour sa fiabilité, sa sécurité et sa portabilité, notamment grâce à la Machine Virtuelle Java (JVM) qui permet d'exécuter le code sur différentes plateformes. Ces caractéristiques en font une technologie particulièrement adaptée aux besoins des grandes entreprises, ce qui explique pourquoi il s’est imposé comme le choix idéal pour notre projet. Recommandé par notre tuteur de stage, il répondait aux standards technologiques du Port Autonome de Cotonou. Dans l’implémentation de notre système, Java a été utilisé pour développer la logique métier côté serveur, structurer les microservices et gérer les traitements liés aux demandes, à la validation et à la gestion des livraisons. Grâce à sa scalabilité, facilitée par le multithreading et les frameworks comme Spring, ainsi qu’à sa structure modulaire, il a permis une organisation claire et maintenable du code, adaptée à un environnement d’entreprise exigeant.

1. **Spring Boot**

Spring Boot est une extension du framework Spring conçue pour simplifier le développement d'applications Java robustes, évolutives et prêtes à être déployées. Il repose sur des principes comme l’inversion de contrôle (IoC) et l’injection de dépendances (DI), qui favorisent une architecture modulaire et facilement testable. Grâce à sa configuration automatique, son serveur embarqué et son approche basée sur les conventions plutôt que la configuration, Spring Boot permet un gain significatif en productivité. Dans notre projet, il a joué un rôle central dans l’implémentation de la couche backend : il a facilité la structuration des microservices, la gestion des dépendances, la création d’API REST, la connexion à la base de données MySQL ainsi que la sécurisation des accès. Ce choix nous a permis de développer une application performante, cohérente et facilement déployable dans un contexte professionnel exigeant.

III-2-1-2 Outils de développement et de gestion de données

1. **XAMPP**

XAMPP est un environnement de développement local open source qui combine Apache, MySQL (ou MariaDB), PHP et Perl, offrant une solution simple pour simuler un serveur web complet sur un ordinateur personnel. Bien que principalement utilisé pour le développement PHP, nous l’avons choisi pour héberger localement la base de données MySQL nécessaire à notre application Spring Boot. XAMPP a joué un rôle clé dans l’implémentation en nous permettant de simuler un environnement serveur, de gérer facilement la base via PhpMyAdmin, et de tester efficacement les connexions entre le backend Java et la base de données, facilitant ainsi le développement et les phases de test en local.

1. **MySQL**

MySQL est un système de gestion de base de données relationnelle open source, reconnu pour sa stabilité, ses bonnes performances et sa facilité d’utilisation. Largement adopté dans le secteur professionnel, il garantit une gestion fiable et sécurisée des données complexes. Nous avons choisi MySQL pour son intégration fluide avec Spring Boot, sa compatibilité avec des outils tels que XAMPP et PhpMyAdmin, ainsi que pour sa robustesse éprouvée, ce qui a permis de structurer efficacement la persistance des données de notre application et d’assurer une gestion optimale des informations liées aux EPI. De plus, MySQL est l’un des systèmes de gestion de bases de données utilisés au Port Autonome de Cotonou, ce qui facilite l’intégration de notre solution dans l’environnement technologique existant.

1. **Draw.io**

Draw.io, aujourd’hui intégré à diagrams.net, est un outil en ligne puissant et intuitif de création de diagrammes UML, de schémas de base de données et d’organigrammes. Il a joué un rôle essentiel dans l’implémentation de notre projet en nous permettant de réaliser facilement les diagrammes UML (cas d’utilisation, classes, séquence, etc.), de visualiser clairement l’architecture du système et de documenter de manière collaborative la conception logicielle, facilitant ainsi la communication entre les membres de l’équipe et la compréhension globale du projet.

III-3 : Résultats et Discussion

III-3-1 : Fonctionnement général

L’application de gestion des Équipements de Protection Individuelle (EPI) permet aux employés de formuler des demandes d’EPI en ligne, de consulter l’historique de leurs dotations et de suivre le statut de leurs requêtes. Une fois connectés à leur espace, ils peuvent également relancer une demande en cas de besoin.

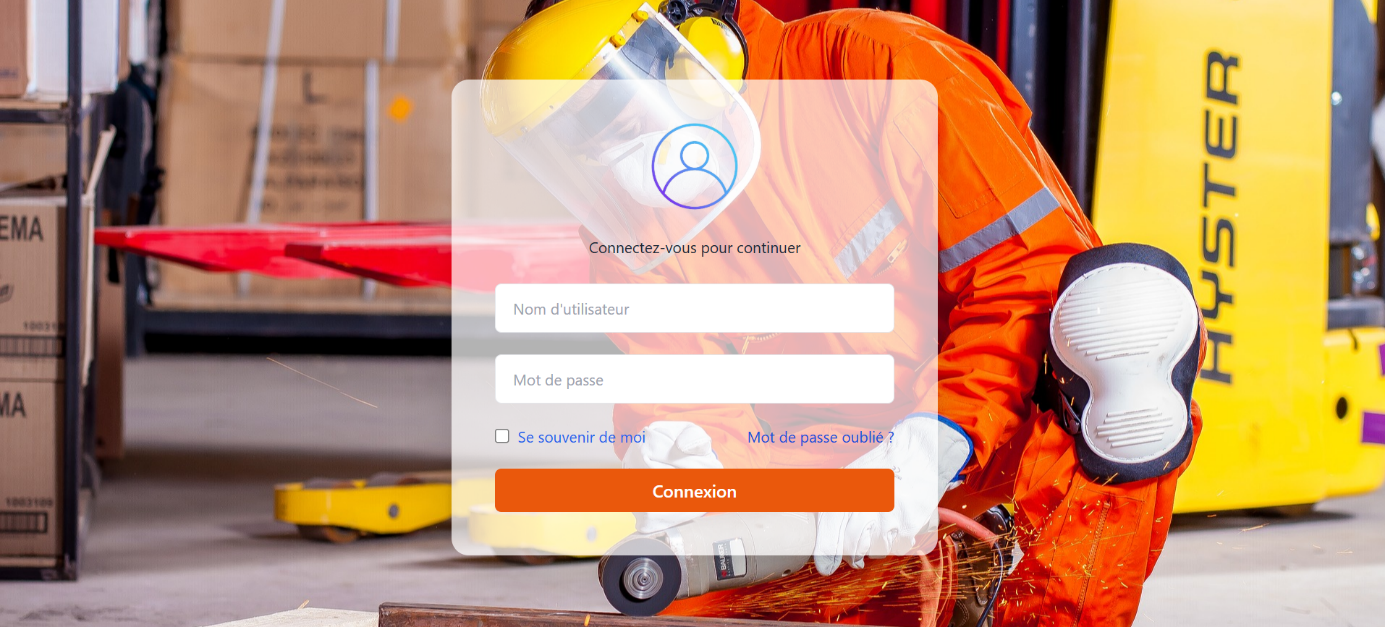
Le gestionnaire EPI accède à un tableau de bord centralisé lui permettant de recevoir et traiter les demandes, de gérer les stocks, d’ajouter de nouveaux équipements, et d’enregistrer les livraisons effectuées. Il bénéficie aussi de statistiques sur la distribution des EPI afin d’anticiper les besoins et les réapprovisionnements.

L’administrateur supervise l’ensemble du système via une interface dédiée. Il peut créer, activer ou désactiver des comptes utilisateurs (employés ou gestionnaires), et consulter l’activité récente de la plateforme. Il assure ainsi le bon fonctionnement et la sécurité du système.

Cette plateforme intégrée vise à automatiser la gestion des EPI, optimiser les flux de distribution, garantir la traçabilité des dotations, et offrir une expérience simple et fluide à chaque profil d’utilisateur.

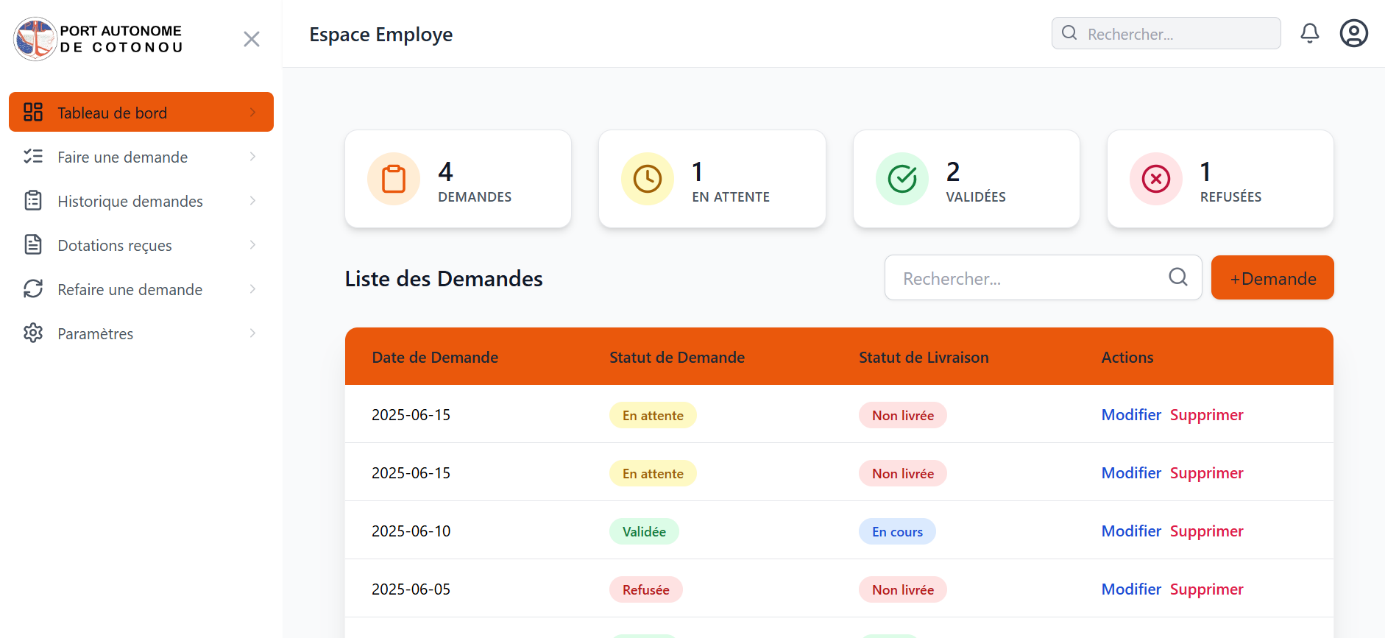
III-3-2 : Quelques captures d’écrans

La figure n°6 présente la page de connexion de l’application



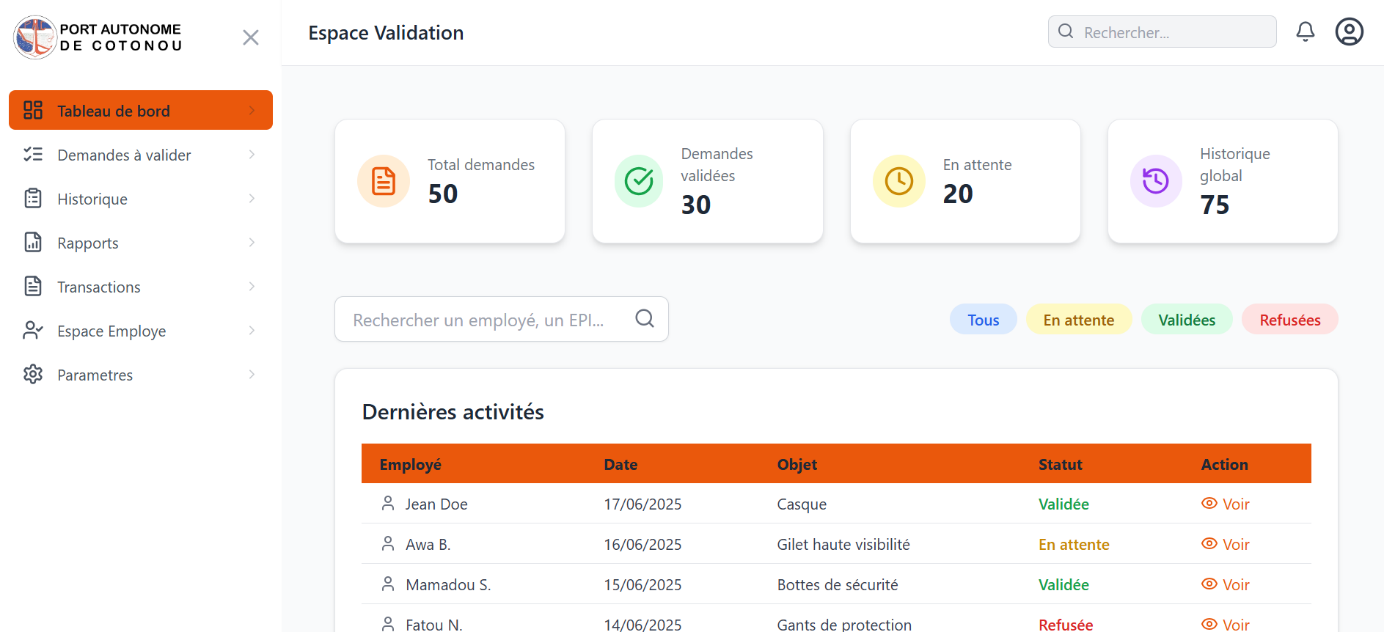
*Figure n°6 : Page d’authentification*

La figure n°7illustre le tableau de bord de l’employé pour la gestion des demandes



*Figure n°7 : Présentation du tableau de bord Employé*

La figure n°7présente le tableau de bord de validation des demandes



*Figure n°7 : Présentation du tableau de bord de Validation*

La figure n°8 présente le tableau de bord du gestionnaire



*Figure n°8 : Présentation du tableau de bord Gestion des EPIs*

La figure n°9présente la page de création de compte utilisateur



*Figure n°9 : Page de Création de Compte*

La figure n°10 présente le tableau de bord de l’administrateur



*Figure n°10 : Le Tableau de bord de l’administrateur*

III-3-2 : Difficultés rencontrées et accès sécuritaire

Les tests réalisés ont porté essentiellement sur les fonctionnalités principales de la plateforme : la soumission de demande d’EPI, le traitement par le DQHSE, la livraison par le gestionnaire, la gestion du stock et la génération des notifications automatiques. Ces tests ont été effectués à l’aide de Postman pour le backend et par des simulations d’utilisation du frontend développé avec React. Chaque scénario d’utilisation a été validé afin de s’assurer que le système répond correctement aux entrées des utilisateurs et traite les erreurs de manière adéquate.

Les résultats ont montré que la plateforme fonctionne de manière fluide, les interactions entre les différents modules sont cohérentes, et les actions réalisées sont bien enregistrées dans la base de données. Le système parvient à notifier les utilisateurs au bon moment, détecte automatiquement les stocks faibles, et déclenche les alertes configurées.

Toutefois, quelques difficultés ont été observées notamment :

* le traitement simultané de plusieurs requêtes pouvait occasionner un ralentissement du serveur en local, ce qui nécessite une optimisation pour une montée en charge réelle ;
* des erreurs de validation non interceptées au niveau du frontend ont parfois permis la soumission de formulaires incomplets, montrant la nécessité de renforcer les contrôles de saisie côté client.

Ces observations ouvrent la voie à des améliorations futures, notamment en matière d’optimisation des performances, d’ajout de tests unitaires automatisés, et de renforcement de la gestion des erreurs côté client.

**CONCLUSION**

# **CONCLUSION**

Le développement d’un système numérique de gestion des Équipements de Protection Individuelle au Port Autonome de Cotonou a permis de répondre à des lacunes majeures constatées dans la gestion actuelle. Grâce à une approche centrée sur les utilisateurs et une architecture technique robuste, cette solution contribue à améliorer la réactivité, la sécurité et la performance opérationnelle du PAC.

Cette expérience a été particulièrement enrichissante sur le plan académique, professionnel et humain. Elle nous a permis d’approfondir notre maîtrise de technologies modernes telles que Java, Spring Boot et React, tout en prenant conscience des enjeux organisationnels liés à la mise en œuvre de solutions informatiques dans un environnement réel.

Ce projet ouvre également des perspectives d’évolution, notamment vers l’intégration d’une gestion multi-sites, de tableaux de bord analytiques ou d’interconnexions avec d’autres systèmes d’information. Il constitue un exemple concret d'application de la technologie au service de la sécurité et de l'efficacité dans un contexte professionnel.

REFERENCES WEBOGRAPHIES

[1] <https://spring.io/projects/spring-authorization-server$>

[2] <https://start.spring.io/index.html>

[1] <https://www.youtube.com/watch?v=k6Nmt-l1Bzc>

[3] <https://gayerie.dev/docs/spring/index.html>

[4] <https://www.jetbrains.com/idea/download/?section=windows>

[5] <https://www.youtube.com/watch?v=UY-9noR8dQ4>

[6] <https://dartagnan.cg.helmo.be/~p150107/tutoriels/uml-uc/>

[7] https://www.bing.com/images/search?q=logo+du+parc+b%C3%A9nin&id=96803

764F59F2028594E8BE7DDD87D0685C0AE79&FORM=IACFIR

REFERENCES BIBLIOGRAPHIQUES

**TABLE DES MATIERES**

[APPROBATION DES MAÎTRES MEMOIRES ET DU MAÎTRE DE STAGE i](#_Toc201154306)

[DEDICACES ii](#_Toc201154307)

[REMERCIEMENTS iii](#_Toc201154308)

[GLOSSAIRES ET SIGLES iv](#_Toc201154309)

[RESUME v](#_Toc201154310)

[ABSTRACT vi](#_Toc201154311)

[SOMMAIRE vii](#_Toc201154312)

[LISTE DES TABLEAUX viii](#_Toc201154313)

[LISTE DES FIGURES ix](#_Toc201154314)

[I**NTRODUCTION** 1](#_Toc201154315)

[CHAPITRE I : PRESENTATION DU CADRE INSTITUTIONNEL DE STAGE 3](#_Toc201154316)

[I-1- PRESENTATION GENERALE DU PORT AUTONOME DE COTONOU (PAC) 4](#_Toc201154317)

[I-2- DEROULEMENT DU STAGE 11](#_Toc201154318)

[I-2-1-Travaux effectués 11](#_Toc201154319)

[I-2-2- Acquis du Stage 12](#_Toc201154320)

[I-2-3- Difficultés rencontrées 13](#_Toc201154321)

[CHAPITRE II : CADRE THEORIQUE ET METHODOLOGIQUE 14](#_Toc201154322)

[II-1- Étude de l’existant 15](#_Toc201154323)

[II-1-1 Présentation de l’existant 15](#_Toc201154324)

[II-1-2 Critique de l’existant 16](#_Toc201154325)

[II-2- Approche de solution 17](#_Toc201154326)

[II-2-1 État de l’art 17](#_Toc201154327)

[II-2-2 Proposition d’approche de solution 18](#_Toc201154328)

[II-3 Cahier des charges fonctionnel 18](#_Toc201154329)

[II-3-1 Cahier des charges fonctionnel 18](#_Toc201154330)

[II-3-2 Clarification conceptuelle 21](#_Toc201154331)

[II-3-3 Démarche méthodologique 22](#_Toc201154332)

[CHAPITRE III : MISE EN ŒUVRE DE LA SOLUTION ET TESTS 24](#_Toc201154333)

[III-1 Étude Comparative ou Analyse Informatique 25](#_Toc201154334)

[III-1-1 Choix de l’approche d’analyse informatique 25](#_Toc201154335)

[III-1-2 Diagramme UML 27](#_Toc201154336)

[III-1-2-1 Diagramme de cas d’utilisation 27](#_Toc201154337)

[III-1-2-2 Diagramme des classes 32](#_Toc201154338)

[III-1-2-3 Diagramme d’activité 34](#_Toc201154339)

[III-2 Mise en Œuvre de la Solution 36](#_Toc201154340)

[III-2-1 Choix Techniques 36](#_Toc201154341)

[III-2-2 Techniques Utilisées 37](#_Toc201154342)

[III-2-2-1 Frontend 38](#_Toc201154343)

[III-2-1-2 Backend 39](#_Toc201154344)

[III-3 : Résultats et Discussion 43](#_Toc201154345)

[III-3-1 : Captures d’écrans 43](#_Toc201154346)

[III-3-2 : Tests, discussion 43](#_Toc201154347)

[**CONCLUSION** 45](#_Toc201154348)

[ANNEXES 46](#_Toc201154349)

[REFERENCES WEBOGRAPHISUES 47](#_Toc201154350)

[REFERENCES BIBLIOGRAPHIQUES 48](#_Toc201154351)