

**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 PROFESSIONNEL EN INFORMATIQUE, RÉSEAU ET TÉLÉCOMMUNICATION**

**Option : Architecture des Logiciels**

**Filière : Informatique, Réseaux et Télécommunication**

**Système de Gestion des Equipements de Protection Individuelle (EPI)**

**Réalisé Par :**

Besilia Sènan AHOMAGNON & Léa Tayé AKOSSETAN

**Encadreur de mémoire : Tuteur de Stage :**

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

**Année académique 2024-2025**

**APPROBATION DES MAÎTRES MEMOIRES ET DU MAÎTRE DE STAGE**

**SIGNATURE**

**Maître de mémoire**

**M. da Eleison MATHA SANT’ANNA**

**Maître de stage**

**M. Verbeck DEGBESSE**

**DEDICACES**

Nous dédions ce mémoire à nos parents, pour avoir toujours cru en nous et pour nous avoir inculqué les valeurs de persévérance et de travail acharné.

A nos frères, sœurs, amis et camarades pour leurs accompagnements et leurs soutiens indéfectibles.

**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. da Eleison MATHA SANT’ANNA, pour sa disponibilité, ses conseils éclairés, son accompagnement constant et sa bienveillance tout au long de la réalisation de ce travail.

Nous remercions également notre maître de stage M. Verbeck DEGBESSE pour avoir partager avec nous 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. ATOHOUN Béthel, pour leurs enseignements précieux et leur soutien académique.

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

Sans oublier nos camarades de promotion et amis, pour leur esprit de camaraderie, leur soutien moral et les moments de partage qui ont constitué une source de motivation et d’encouragement tout au long de ce parcours.

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

**GLOSSAIRE ET SIGLES**

* PAC : Port Autonome de Cotonou
* EPI : Équipement de Protection Individuelle
* DQHSE : Direction/ Directeur de la Qualité, de l’Hygiène, de la Sécurité et de l’Environnement
* GEC :
* HTTP :
* XAMPP :
* MySQL :
* MariaDB :
* HTML :
* CSS :
* POO :
* MVC :
* API :
* API REST :
* UML :
* Spring :
* CI/CD :
* JPA :
* DI :
* IoC :
* UI/UX :
* EcoOnline EHS :

**SOMMAIRE**

[**INTRODUCTION** 7](#_Toc200917196)

[**I-1- PRESENTATION GENERALE DU PORT AUTONOME DE COTONOU (PAC)** 9](#_Toc200917197)

[**I-2- DEROULEMENT DU STAGE** 16](#_Toc200917198)

[**I-2-1-Travaux effectués** 16](#_Toc200917199)

[**I-2-2- Acquis du Stage** 17](#_Toc200917200)

[**I-2-3- Difficultés rencontrés** 18](#_Toc200917201)

[**II-1- Étude de l’existant** 19](#_Toc200917202)

[**II-1-1 Présentation de l’existant** 20](#_Toc200917203)

[**II-1-2 Critique de l’existant** 21](#_Toc200917204)

[**II-2- Approche de solution** 22](#_Toc200917205)

[**II-2-1 État de l’art** 22](#_Toc200917206)

[**II-2-2 Proposition d’approche de solution** 23](#_Toc200917207)

[**II-3 Cahier des charges fonctionnel** 23](#_Toc200917208)

[**II-3-1 Cahier des charges fonctionnel** 23](#_Toc200917209)

[**II-3-2 Clarification conceptuelle** 26](#_Toc200917210)

[**II-3-3 Démarche méthodologique** 27](#_Toc200917211)

[**III-1 Étude Comparative ou Analyse Informatique** 29](#_Toc200917212)

[**III-1-1 Choix de l’approche d’analyse informatique** 30](#_Toc200917213)

[**III-1-2 Diagramme UML** 31](#_Toc200917214)

[**III-2 Mise en Œuvre de la Solution** 40](#_Toc200917215)

[**III-2-1 Choix Techniques** 40](#_Toc200917216)

[**III-2-2 Techniques Utilisées** 41](#_Toc200917217)

[**III-3 : Résultats et Discussion** 45](#_Toc200917218)

[**CONCLUSION** 47](#_Toc200917219)

**LISTE DES TABLEAUX**

**LISTE DES FIGURES**

**INTRODUCTION**

# **INTRODUCTION**

**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.
* **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 Git, 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 électronique 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 rencontrés**

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**.

# **II-1- Étude de l’existant**

**CHAPITRE II : CADRE THEORIQUE ET METHODOLOGIQUE**

## **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, après nos recherches approfondies, nous avons identifié plusieurs solutions existantes sur le marché.

## **II-2-1 État de l’art**

Parmi les applications existantes, **EPI Manager** propose une plateforme de gestion des dotations EPI avec traçabilité, planification et alertes de renouvellement. **EcoOnline EHS**, solution plus globale de gestion QHSE, intègre un module pour le suivi des EPI, le tout dans une approche de conformité réglementaire. **Synergi Life**, de DNV, permet un suivi détaillé des EPI, notamment pour les industries à haut risque, avec un historique des dotations par employé et des indicateurs de performance. **Workhub**, anciennement SafetySync, permet la gestion des formations obligatoires et du suivi des EPI avec une interface simple et des notifications automatiques. Enfin, **IntelliPERMIT** offre un système d’autorisation de travail intégré à la gestion des EPI, garantissant que seuls les employés correctement équipés accèdent aux zones à risque.

Ces outils permettent un suivi précis des équipements, de leur attribution à leur état d’usure, avec des fonctionnalités complémentaires comme la gestion des utilisateurs, la centralisation des données, ou encore 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) :

* **EPI Manager** est reconnu pour sa spécialisation, mais reste peu flexible pour les environnements multisites ou très réglementés.
* **EcoOnline EHS** est très complet, adapté aux grandes structures, mais peut être complexe à déployer et nécessite une formation préalable des utilisateurs.
* **Synergi Life** offre une couverture fonctionnelle étendue, mais son coût peut être prohibitif pour les structures moyennes.
* **Workhub** est simple d’utilisation et abordable, mais présente des limites pour une gestion avancée des stocks ou une intégration poussée aux systèmes d’information internes.
* **IntelliPERMIT** est efficace dans les contextes industriels à haut risque, mais son approche très centrée sur les permis de travail peut être excessive pour des besoins de gestion standard des EPI.

## **II-2-2 Proposition d’approche de solution**

Face aux limites des méthodes manuelles actuellement en place, nous proposons de développer une plateforme numérique intégrée dédiée à la gestion des EPI. Cette solution aura pour objectif de centraliser et d’automatiser les processus de demande, d’attribution et de suivi des équipements, tout en optimisant la gestion des stocks. Elle renforcera également la traçabilité, la conformité réglementaire et la sécurité du personnel. Intuitive et adaptée au contexte spécifique du PAC, cette plateforme contribuera significativement à l’amélioration de l’efficacité des opérations liées à la sécurité des employés.

# **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.

**Rapports et exportation des données**

* L’application devra générer automatiquement des rapports périodiques (hebdomadaires, mensuels ou personnalisés) sur les dotations, les demandes, les niveaux de stock, etc.
* Ces rapports peuvent être exportés en formats courants (PDF, Excel) et utilisés pour les réunions, audits ou décisions de gestion.

## **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.

1. **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.

1. **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.

1. **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.

1. **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.

# **III-1 Étude Comparative ou Analyse Informatique**

**CHAPITRE III : MISE EN ŒUVRE DE LA SOLUTION ET TESTS**

## **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**

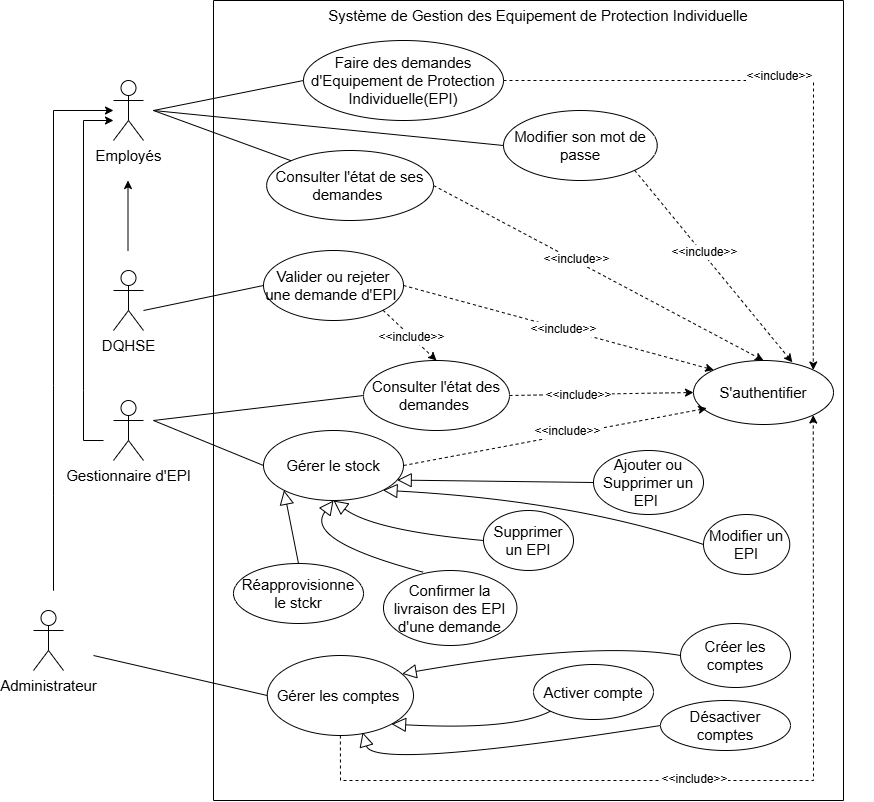
Un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé. Il joue un rôle déterminant en initiant des actions ou en sollicitant des services auprès du système. En réponse, ce dernier exécute les traitements nécessaires, qu’il s’agisse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

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 :** créer une session de navigation.
* **Précondition :** L’utilisateur doit avoir un compte valide
* **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 entrés ;
5. Si l’utilisateur est reconnu, le système le redirige vers son tableau de bord ;

* **Scénario alternatif :**

1. L’utilisateur informe l’administrateur de la situation ;

* **Scénario d’Exception :**

1. Le système affiche un message d’erreur en demandant à l’utilisateur de corriger et de réessayer.
2. 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 : Effectuer une demande d’EPI**

* **Acteur** : Tous les utilisateurs du système
* **Description** : Soumettre une demande d’attribution d’un ou plusieurs EPI
* **Précondition** : L’utilisateur est connecté
* **Scénario** **nominal** :
  1. L’utilisateur remplit et soumet le formulaire de demande ;
  2. Le système notifie le Directeur QHSE pour validation.
  3. Si la demande est à “Validé”, notifier le gestionnaire ;
* **Postcondition** : La demande est soumise et en attente de validation

**Cas 3 : Réapprovisionner le stock**

* **Acteur** : Gestionnaire
* **Description** : Permet de gérer les stocks d'EPI
* **Précondition** : Le gestionnaire est connecté et Le système envoie un alerte de rupture de stock
* **Scénario nominal :**
  1. Le gestionnaire remplit et soumet le formulaire de réapprovisionnement d’EPI ;
  2. Le système envois la demande au DQHSE et le notifie ;
* **Postcondition** : La demande de réapprovisionnement est soumise ;

**Cas 5 : Livrer une demande**

* **Acteur** : Gestionnaire
* **Description** : Permet de gérer les livraisons d'EPI
* **Précondition** : Une demande est validée
* **Scénario nominal :**

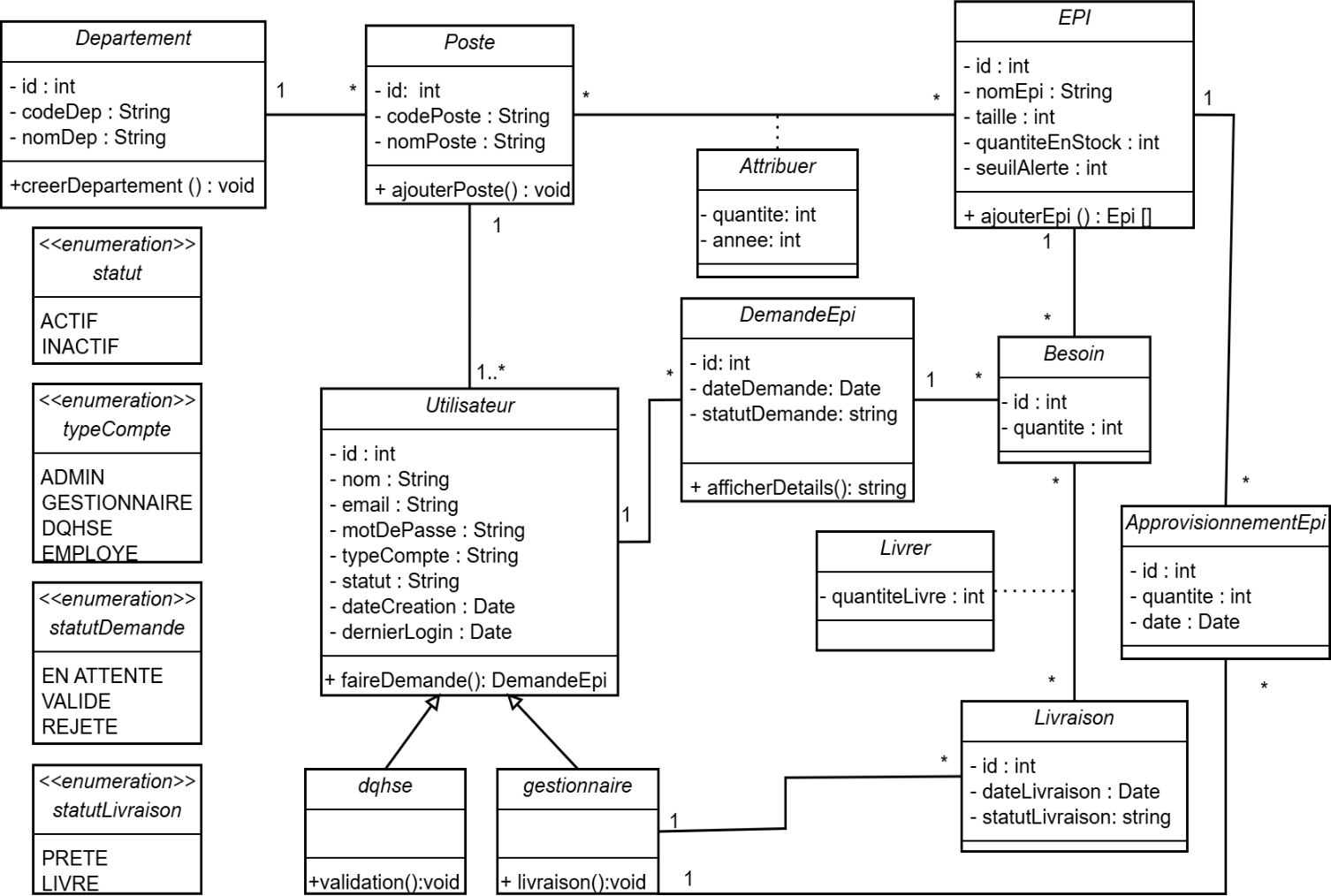
1. Le système vérifie la disponibilité des EPI
2. Si EPI disponible, le système notifie le gestionnaire
3. Et envois une notification de livraison prête au demandeur ;
4. Si l’utilisateur viens récupérer les EPI, le gestionnaire enregistre la livraison ;
5. Le système met à jour le stock ;

* **Scénario alternatif : EPI indisponible**
  + Le gestionnaire remplit et soumet le formulaire de réapprovisionnement d’EPI ;
* **Postcondition** : L’utilisateur est livré

**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*

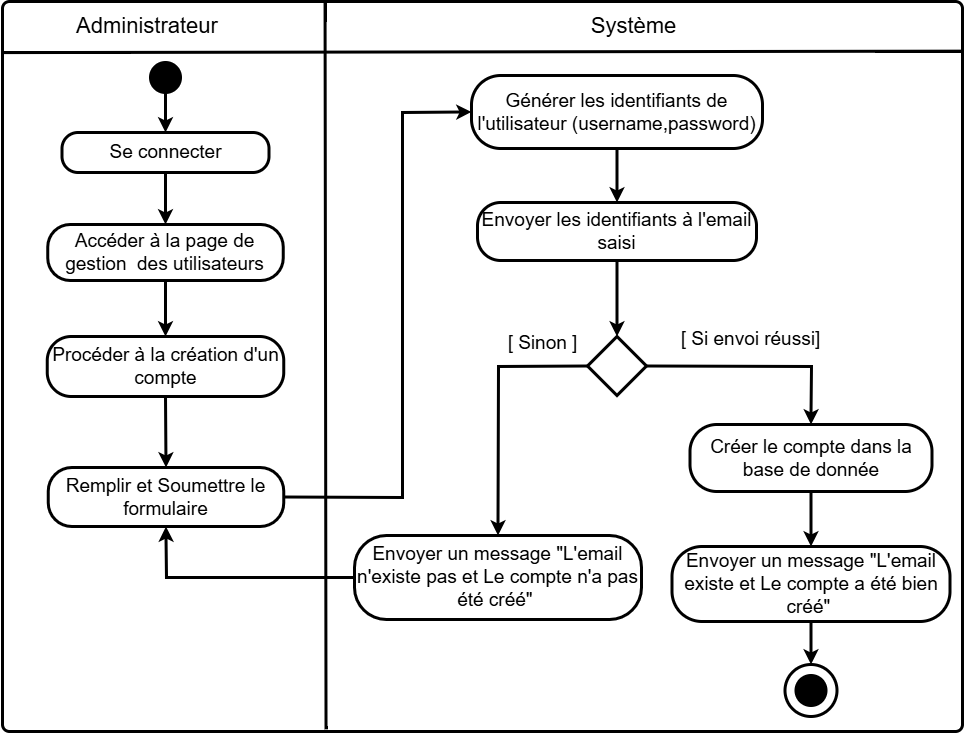
Ce diagramme de classes représente le modèle conceptuel de l’application de gestion des Équipements de Protection Individuelle (EPI) au Port Autonome de Cotonou. Il met en évidence les principales entités du système (utilisateur, demande, EPI, besoin, approvisionnement, livraison) et leurs interactions. 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” :

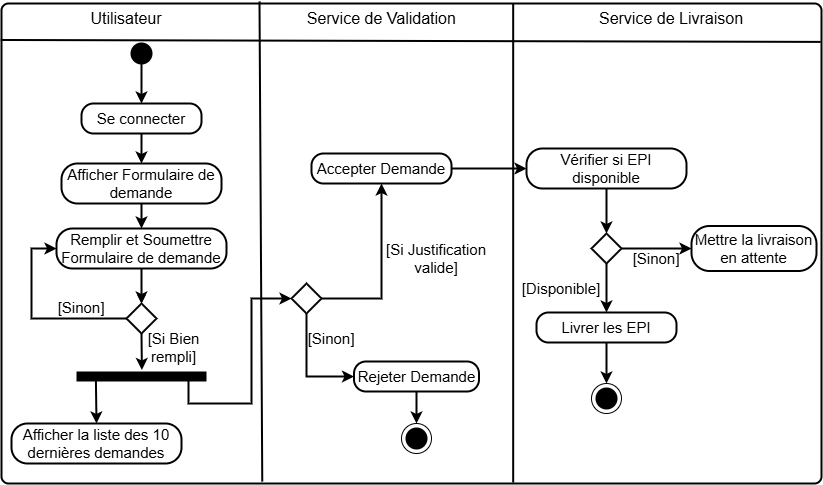


*Figure n°4 : Diagramme de l’activité “Créer un compte”*

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 “Effectuer une demande” :



*Figure n°5 : Diagramme de l’activité “Effectuer 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.

## **III-2-2 Techniques Utilisées**

**III-2-2-1 Frontend**

Plusieurs technologies peuvent être utilisées pour le développement de l’interface utilisateur, telles qu’Angular : Framework JavaScript complet basé sur TypeScript, adapté aux applications complexes ; Vue.js : Framework progressif, léger et facile à prendre en main, React, ou encore des frameworks CSS comme Tailwind CSS.

Dans notre projet, nous avons choisi : React et Tailwind CSS

1. **React**

**React** est une bibliothèque JavaScript utilisée pour construire des interfaces utilisateur interactives et des applications web modernes. Elle permet aux développeurs de créer des **composants réutilisables** et modulaires, ce qui simplifie la gestion et l'organisation du code. React se distingue par sa flexibilité et sa capacité à s'intégrer facilement avec d'autres bibliothèques ou projets existants. Nous avons choisi React pour son **architecture basée sur les composants**, son écosystème robuste et sa capacité à créer des interfaces utilisateur dynamiques avec une réactivité optimale. De plus, la **grande communauté active de React** offre un large éventail d'outils, de bibliothèques et de ressources pour améliorer notre projet. En utilisant React, nous avons pu construire une application front-end réactive et performante qui répond aux besoins des utilisateurs.

1. **Tailwindcss**

Tailwind CSS est un framework CSS utilitaire qui permet de concevoir rapidement des interfaces web. Au lieu de fournir des composants pré-définis, il propose une collection de classes CSS qui peuvent être directement appliquées aux éléments HTML. Ces classes permettent de définir rapidement des styles et des mises en page personnalisées sans avoir à écrire de CSS traditionnel. Tailwind CSS facilite également la création de designs réactifs et adaptatifs grâce à sa grille flexible et ses classes responsives. Ce framework nous a aidé à rendre notre interface plus attrayante et cohérente.

**III-2-1-2 Backend**

Pour développer le backend d'une application web, il existe un large éventail de technologies. Parmi elles, **Node.js** est souvent choisi pour ses performances en temps réel et sa nature asynchrone, idéale pour les applications légères et rapides. **Python**, avec des frameworks comme Django ou Flask, est apprécié pour sa simplicité, sa rapidité de développement et son expressivité. **Ruby** on Rails permet de concevoir rapidement des applications web grâce à son approche basée sur les conventions.

Dans le cadre de notre projet, nous avons retenu les technologies suivantes :

1. **Java**

Pour le développement de notre solution au profit d'une grande entreprise, nous avons sélectionné le langage Java. Ce choix repose sur ses qualités intrinsèques et sa pertinence avérée pour les applications d'entreprise exigeantes. Java est un langage de programmation orienté objet (POO) polyvalent, reconnu pour sa portabilité ("écrire une fois, exécuter partout" grâce à la Machine Virtuelle Java), sa sécurité et sa robustesse. Il est omniprésent dans les applications web, mobiles et embarquées.

Nous avons spécifiquement choisi Java pour ses atouts majeurs dans un contexte d'entreprise :

**Scalabilité** : Les grandes entreprises traitent d'énormes volumes de données et de transactions. Java, grâce à sa gestion efficace du multithreading et un écosystème riche en frameworks tels que Spring et Spring Boot, permet de concevoir des architectures hautement scalables et performantes, idéales pour les microservices et les systèmes distribués.

**Maintenabilité** : La nature orientée objet de Java encourage un code propre, modulaire et structuré. Cette qualité est essentielle pour les applications d'entreprise qui évoluent sur le long terme et sont gérées par différentes équipes, garantissant une facilité de maintenance et d'évolution.

**Fiabilité et Écosystème** : Java bénéficie d'une fiabilité éprouvée, d'un écosystème mature et vaste (bibliothèques, outils, communauté). Cette richesse assure un développement efficace, une résolution rapide des problèmes et une pérennité du projet.

En somme, l'adoption de Java pour notre solution est une garantie de fiabilité, scalabilité, sécurité et maintenabilité, des qualités indispensables pour le développement d'un système complexe et critique au sein d'une grande entreprise. De plus c’est le langage qui nous a été recommandé à notre lieu de stage pour la conception de cette application.

1. **Spring Boot**

Spring Boot est un framework open-source qui accélère et simplifie le développement d'applications web et de microservices en Java. Il s'appuie sur le Spring Framework, qui est la base de nombreuses applications Java d'entreprise.

Le Spring Framework est la fondation qui gère l'infrastructure technique des applications Java, utilisant l'Inversion de Contrôle (IoC) et l'Injection de Dépendances (DI). Ces principes permettent de réduire le couplage entre les composants du code, le rendant plus modulaire et facile à tester, tout en fournissant des modules pour l'accès aux données (Spring Data Jpa), le web (Spring Web), la sécurité (Spring Security), etc.

Spring Boot agit comme une surcouche du Spring Framework en le rendant beaucoup plus facile à utiliser. Il se distingue par trois fonctionnalités principales qui minimisent la configuration et l'installation nécessaires :

**Configuration automatique** : Spring Boot devine et configure la majeure partie de votre application, vous évitant ainsi des heures de paramétrage manuel.

**Approche directive de la configuration** : Il propose des conventions par défaut intelligentes, vous guidant vers les meilleures pratiques sans vous imposer de lourdes décisions.

**Capacité à créer des applications autonomes** : Grâce à des serveurs web embarqués (comme Tomcat), les applications Spring Boot peuvent être démarrées directement, simplifiant grandement le déploiement.

Nous avons choisi Spring Boot pour sa productivité exceptionnelle, sa facilité de déploiement, et sa capacité à créer rapidement des applications robustes et scalables. Ces atouts sont essentiels pour répondre aux exigences d'une grande entreprise et construire des solutions performantes.

1. **XAMPP**

XAMPP est un environnement de développement web local gratuit et open-source qui permet de simuler un serveur web complet sur un ordinateur personnel. Pour exécuter et tester notre application localement, nous avons choisi d’utiliser XAMPP, principalement pour sa composante MySQL (ou MariaDB) qui nous permet de disposer d’un serveur de base de données local. Bien que XAMPP soit souvent associé au développement en PHP, il reste tout à fait compatible avec les applications développées en Java, notamment avec Spring Boot, dès lors que celles-ci interagissent avec une base de données MySQL.

Nous avons opté pour XAMPP car il fournit une solution simple et rapide pour héberger une base de données en local, via MySQL et PhpMyAdmin, tout en nous permettant de connecter facilement notre application Spring Boot à cette base. De plus, son interface de gestion centralisée facilite le démarrage et l’arrêt des services, ce qui est particulièrement pratique pendant le développement et les tests.

1. **MySQL**

MySQL est un système de gestion de base de données relationnelle open source réputée pour sa fiabilité, sa facilité d'utilisation et sa performance. Il est largement utilisé dans l'industrie pour gérer efficacement de grandes quantités de données. Nous avons opté pour MySQL en raison de sa réputation de stabilité et de l'intégrité de ses données. Il offre également des fonctionnalités robustes et innovantes, ce qui en fait un choix solide pour notre application. Grâce à MySQL, nous pouvons être confiants dans la gestion sécurisée de nos données complexes et dans la fourniture de solutions performantes.

1. **Draw.io**

Draw.io est un logiciel de diagramme en ligne simple et puissant qui permet de créer facilement des diagrammes, des organigrammes, des schémas, et bien plus encore. Il est largement utilisé dans l'industrie pour sa flexibilité et sa facilité d'intégration. Nous avons opté pour Draw.io en raison de son interface intuitive et conviviale, qui offre une large gamme d'outils de dessin et de formes prédéfinies pour vous aider à donner vie à vos idées visuelles. Grâce à Draw.io, nous pouvons être confiants dans notre capacité à communiquer visuellement des informations complexes de manière claire et efficace, et à fournir des solutions performantes.

# **III-3 : Résultats et Discussion**

**III-3-x : (Environnement et outils de déploiement, captures d’écrans, tests, discussion)**

**CONCLUSION**

# **CONCLUSION**