

Université Assane Seck de Ziguinchor

UFR Sciences et Technologies

Département Informatique



*L'excellence, ma référence*

Mémoire de fin d'études

**Pour l'obtention du diplôme de master**

Mention Informatique

Spécialité Génie Logiciel

**Systeme de contrôle d'accès et de planification de rendez-vous  
pour les sociétés externes à la plateforme logistique de  
distribution du Port Autonome de Dakar.**

Présenté et soutenu par

Mouhamadou DIAMANKA

le 17/04/2024

Sous la direction de

Dr Gorgoumack SAMBE

M. Moussa DIOP

Sous la supervision de

Pr Khadim DRAME

Membres du jury

Pr Ibrahima DIOP	Professeur assimilé	Président — Examineur	UASZ
Pr Ousmane DIALLO	Professeur assimilé	Rapporteur	UASZ
Pr Khadim DRAME	Professeur assimilé	Directeur de mémoire	UASZ
Dr Gorgoumack SAMBE	Maitre de conférences titulaire	Co-directeur de mémoire	UASZ
M. Moussa DIOP	Directeur Atlantic DS	Maitre de stage	ADS

## Résumé

L'optimisation des opérations logistiques au sein de la plateforme de distribution du Port Autonome de Dakar représente un défi majeur pour assurer la fluidité des activités de distribution. Gérée par un concessionnaire, elle comprend diverses installations et procédures. Actuellement, le contrôle d'accès au niveau de la plateforme de distribution est géré manuellement. Cependant, ce mode de gestion présente des vulnérabilités, notamment des risques d'accès non autorisés et des complexités sur la gestion des mouvements d'entrée et de sortie des individus et des véhicules. Ces défis suscitent des préoccupations quant à la sécurité des marchandises et des installations portuaires.

Afin de résoudre ces problématiques, nous avons réalisé un système sécurisé et centralisé répondant aux objectifs majeurs suivants : la gestion des accès des sociétés externes, la gestion des ressources (personnel et véhicules), la planification des rendez-vous et le suivi des entrées/sorties.

De manière spécifique, notre solution comprend une API indépendante permettant aux sociétés d'intégrer notre solution à leurs applications existantes ou futures, ainsi que des interfaces web et mobile pour plus de flexibilité.

Dans cette démarche, l'option s'est portée sur la méthodologie Scrum dans la gestion du projet, avec une architecture en microservices. De plus, le système de contrôle d'accès repose sur du code QR. Pour illustrer son fonctionnement, un système de contrôle d'accès utilisant des composants Arduino a été développé, accompagné d'une simulation virtuelle.

L'initiative présente une première version du système de contrôle d'accès, visant à résoudre des problèmes de sécurité et d'efficacité. Ces avancées jettent les bases d'une solution complète, parfaitement alignée sur les exigences strictes du commerce international.

*Mots clés: plateforme de distribution, code QR, port autonome de Dakar, API, contrôle d'accès*

## Abstract

Optimizing logistics operations within the distribution platform of the Autonomous Port of Dakar represents a major challenge to ensure the fluidity of distribution activities. Operated by a dealer, it includes various facilities and procedures.

Currently, access control at the distribution platform level is managed manually. However, this management method presents vulnerabilities, including risks of unauthorized access and complexities in the management of entry and exit movements of individuals and vehicles. These challenges raise concerns about the security of cargo and port facilities.

In order to resolve these issues, we have created a secure and centralized system meeting the following major objectives: management of access by external companies, management of resources (personnel and vehicles), management of parking spaces and monitoring of entries/exits.

Specifically, our solution includes an independent API allowing companies to integrate our solution into their existing or future applications, as well as web and mobile interfaces for greater flexibility.

In this approach, the option was based on the Scrum methodology in project management, with a microservices architecture. In addition, the access control system is based on QR code. To illustrate its operation, an access control system using Arduino components was developed, accompanied by a virtual simulation.

The initiative presents a first version of the access control system, aimed at solving security and efficiency problems. These advances lay the foundations for a comprehensive solution, perfectly aligned with the stringent requirements of international trade.

*Keywords: distribution platform, QR code, autonomous port of Dakar, API, access control*

## **Dédicaces**

*Je rends grâce à DIEU et je dédie ce modeste travail :*

*À ma chère mère, même si tu n'es plus parmi nous physiquement, ton amour, ta sagesse et ta bienveillance continuent à me guider chaque jour. Tu restes gravée dans mon cœur. Que le paradis soit ton éternel demeure et que cet humble hommage témoigne de l'amour éternel que je te porte.*

*À mon cher père, qui n'a jamais cessé de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.*

*À mon père, Alioune Badara CISSE, aucune dédicace ne saurait exprimer l'estime et le respect que j'ai pour toi. Rien au monde ne valent les efforts fournis pour mon éducation et mon bien-être. Ce travail est le fruit des sacrifices que tu as consentis pour mon éducation et ma formation le long de ces années.*

*À ma mère, Aramatoulaye SAGNA, quoi que je fasse ou que je dise, je ne pourrais jamais te remercier assez. Merci pour tout ce que tu as fait, pour toujours croire en moi, même quand je doutais.*

*À mon oncle Youssoupha CISSE, qui m'a offert mon tout premier ordinateur, déclenchant ainsi ma passion pour l'informatique.*

*À Fatou Kiné BADIANE et Tonton Abdoulaye BA, qui ont facilité tout mon cursus à l'université Alioune DIOP de Bambey.*

*À mes tantes Aissatou CISSE, Diariatou CISSE, Ndeye Mariama CISSE, Safietou CISSE, Mouskeba MANKAL, Aminata DIOP, que ce modeste travail soit l'expression des vœux que vous n'avez cessé de formuler dans vos prières.*

*À mes amis (es) et promotionnaires, qui m'ont toujours encouragé et à qui je souhaite plus de succès.*

*À mes chers collègues Balla GNINGUE, Rokhaya Sarr DIOP et Papa Matar DIENG, que Dieu vous guide vers les sommets de l'excellence et du succès.*

*Que Dieu le Tout-Puissant vous garde et vous procure santé, bonheur et longue vie.*

*Enfin, je dédie ce travail à tous mes proches, amis(es) et promotionnaires en témoignage de l'affection et la sincère amitié qui nous lient.*

## Remerciements

*Je rends grâce à Dieu le Tout-Puissant et Miséricordieux de m'avoir accordé la volonté d'entamer et de terminer ce mémoire.*

*Je tiens à exprimer ma sincère et profonde gratitude envers mes directeurs de mémoire, le Pr Khadim DRAME et le Dr. Gorgoumack SAMBE, Enseignants-Chercheurs à l'Université Assane Seck de Ziguinchor, pour leur expertise et leur soutien tout au long de l'élaboration de ce travail. Leurs conseils et leur disponibilité ont grandement contribué à sa réussite.*

*Je souhaite également exprimer ma reconnaissance envers Monsieur Moussa DIOP, mon maître de stage à Atlantic Digital Solution. Sa collaboration active, ses conseils pertinents et son accompagnement ont enrichi mon expérience professionnelle et ont grandement contribué à l'enrichissement de mon mémoire.*

*Je tiens à remercier Messieurs les membres du jury qui m'honorent en évaluant mon travail :*

- *Pr Ibrahima DIOP d'avoir accepté d'être président du jury et examinateur ;*
- *Pr Ousmane DIALLO d'être rapporteur ;*
- *Pr Khadim DRAME et Dr Gorgoumack SAMBE pour leur encadrement.*

*Un chaleureux merci à Atlantic Digital Solution pour avoir fourni un cadre propice à l'application pratique des connaissances acquises au cours de mon cursus universitaire. Mes remerciements s'étendent à l'ensemble du personnel de Business Logistique et IZICARS.*

*Mes remerciements vont également à tous les professeurs du département informatique de l'Université Assane Seck de Ziguinchor, pour leur générosité et la patience dont ils ont fait preuve malgré leurs charges académiques.*

*Mes sincères remerciements vont également à l'endroit de mes amis(es) et camarades de promotion du master Informatique.*

*Je remercie également mes amis(es), tantes, oncles et youmpagnes.*

*Enfin, je remercie toutes celles et tous ceux qui de près ou de loin ont participé à l'accomplissement de ce travail.*

# Tables des matières

Résumé.....	ii
Abstract .....	iii
Dédicaces .....	iv
Remerciements.....	v
Listes des figures.....	ix
Listes des tableaux .....	xi
Listes des sigles et abréviations .....	xii
Introduction générale .....	1
Chapitre 1 - Description du sujet .....	3
Introduction.....	3
1.1    Présentation de Atlantic Digital Solutions .....	3
1.1.1    Mission et organigramme .....	3
1.1.2    Services.....	4
1.2    Présentation de la plateforme logistique .....	5
1.3    Etude de l'existant.....	6
1.4    Problématiques et objectifs .....	8
Conclusion .....	9
Chapitre 2 - Généralités sur les systèmes de contrôles d'accès .....	11
Introduction.....	11
2.1    Types de contrôle d'accès .....	11
2.2    Etapas de gestion du contrôle d'accès.....	12
2.3    Types de systèmes de contrôle d'accès .....	12
2.4    Technologies de contrôle d'accès.....	13
2.4.1    Code QR .....	13
2.4.2    RFID .....	14
2.4.3    Technologies basées sur l'IA .....	15
2.4.4    Etude comparative .....	16
2.5    Modèles de blocage d'accès.....	18
2.5.1    Barrière .....	19

2.5.2	Tourniquet tripode.....	19
	Conclusion .....	19
Chapitre 3 - Méthodologies de développement .....		21
	Introduction.....	21
3.1	Méthodologies.....	21
3.1.1	Modèle en cascade .....	21
3.1.2	Processus Unifié.....	22
3.1.3	Scrum .....	23
3.1.4	Etude comparative des méthodes.....	25
3.2	UML.....	26
3.3	Architecture.....	27
3.3.1	Architecture monolithique .....	27
3.3.2	SOA .....	28
3.3.3	Architecture microservices .....	29
3.3.4	Etude comparative .....	29
	Conclusion .....	31
Chapitre 4 - Spécification et analyse des besoins .....		32
	Introduction.....	32
4.1	Spécification des besoins .....	32
4.1.1	Expressions des besoins.....	32
4.1.2	Identification des acteurs .....	33
4.1.3	Identification des cas d'utilisation .....	33
4.1.4	Diagrammes de cas d'utilisation.....	34
4.2	Analyse des besoins .....	36
4.2.1	Description textuelle des cas d'utilisation .....	36
4.2.2	Diagramme de séquences.....	37
4.2.3	Diagramme d'activité .....	39
	Conclusion .....	40

Chapitre 5 - Conception du système .....	41
Introduction.....	41
5.1 Conception générale.....	41
5.1.1 Architecture système .....	41
5.1.2 Diagramme de composants .....	42
5.1.3 Diagramme de paquetage.....	43
5.1.4 Diagramme de déploiement .....	43
5.2 Conception détaillée.....	44
5.2.1 Dictionnaire de données.....	44
5.2.2 Diagramme de classes.....	46
Conclusion .....	48
Chapitre 6 - Implémentation et présentation de « AccessLog » .....	49
Introduction.....	49
6.1 Implementation .....	49
6.1.1 Technologies .....	49
6.1.2 Outils.....	50
6.2 Presentation du système.....	54
6.2.1 API.....	54
6.2.2 Interfaces de l'application web.....	55
6.2.3 Interfaces de l'application mobile .....	56
6.2.4 Simulateur de contrôle d'accès.....	57
Conclusion .....	60
Conclusion generale.....	61
Annexe .....	62
Bibliographie.....	67

# Listes des figures

Figure 1-1 Organigramme Atlantic Digital Solutions .....	4
Figure 1-2 Plateforme de distribution (google earth).....	6
Figure 1-3 Etude de l'existant.....	8
Figure 2-1 Etapes de gestion de contrôle d'accès .....	12
Figure 2-2 Exemple de code QR .....	14
Figure 2-3 Tag et Lecteur RFID .....	15
Figure 2-4 Empreinte digitale .....	16
Figure 2-5 Barrière routière.....	19
Figure 2-6 Tourniquet tripode .....	19
Figure 3-1 Cycle de vie modèle en cascade .....	21
Figure 3-2 Cycle de vie du processus unifié .....	23
Figure 3-3 Cycle de vie Scrum.....	24
Figure 3-4 Hiérarchie des diagrammes d'uml 2.0 .....	27
Figure 3-5 Architecture monolithique .....	28
Figure 3-6 Architecture SOA .....	29
Figure 3-7 Architecture microservices .....	29
Figure 4-1 Diagramme de cas d'utilisation de l'administrateur .....	35
Figure 4-2 Diagramme de cas d'utilisation du Responsable société.....	35
Figure 4-3 Diagramme de séquences « Contrôle d'accès des camions».....	38
Figure 4-4 Diagramme de séquences « Planifier un rendez-vous » .....	38
Figure 4-5 Diagramme d'activités « Faire une demande de rendez-vous » .....	39
Figure 4-6 Diagramme d'activités « Planifier un rendez-vous » .....	40
Figure 5-1 Architecture du système .....	42
Figure 5-2 Diagramme de composants.....	42
Figure 5-3 Diagramme de package du microservice « Panification de rendez-vous » .....	43
Figure 5-4 Diagramme de déploiement.....	44
Figure 5-5 Diagramme de classes MS: Gestion sociétés et ressources .....	47
Figure 5-6 Diagramme de classes MS: Planification de rendez-vous.....	47
Figure 5-7 Diagrammes de classes MS: Gestion de contrôle d'accès .....	48
Figure 6-1 Exemple de Token généré par l'API .....	53
Figure 7-1 Diagramme de séquences « Faire une demande de rendez-vous ».....	62
Figure 7-2 Diagramme de package microservice « Gestion du contrôle d'accès » .....	63

Figure 7-3 Diagramme de package microservice « Gestion des ressources » .....	63
Figure 7-4 Gate pass.....	64
Figure 7-5 Prometheus .....	64
Figure 7-6 Zipkin 1/3 .....	65
Figure 7-7 Zipkin 2/3 .....	65
Figure 7-8 Zipkin 3/3 .....	65
Figure 7-9 Tableau de bord grafana.....	66
Figure 7-10 Spring boot admin .....	66

## Listes des tableaux

Tableau 2-1 Tableau comparative des technologies de contrôle d'accès.....	18
Tableau 3-1 Les phases du processus unifié.....	23
Tableau 3-2 Evènements et artefacts Scrum.....	24
Tableau 3-3 Etude comparative des modèles .....	25
Tableau 3-4 : Organisation de l'équipe Scrum.....	26
Tableau 3-5 Etude comparative entre architecture .....	31
Tableau 4-1 Liste de modules et cas d'utilisation .....	34
Tableau 4-2 Description textuelle du cas d'utilisation « Contrôle d'accès des camions » .....	36
Tableau 4-3 Description textuelle du cas d'utilisation « Planifier un rendez-vous ».....	37
Tableau 5-1 Dictionnaire de données .....	46
Tableau 6-1 : Avantages et limites de Angular .....	49
Tableau 6-2 : Outils intégrés à Spring Boot .....	50
Tableau 7-0-1 Description textuelle du cas d'utilisation « Faire une demande de rendez-vous » .....	62

# Listes des sigles et abréviations

<b>ACL</b>	: Access Control List
<b>ADN</b>	: Acide Désoxyribonucléique
<b>API</b>	: Application Programming Interface
<b>CCTV</b>	: Closed-Circuit Television
<b>CMA CGM</b>	: Compagnie Maritime d'Affrètement – Compagnie Générale Maritime
<b>ECDSA</b>	: Elliptic Curve Digital Signature Algorithm
<b>ERP</b>	: Enterprise Resource Planning
<b>GIE</b>	: Groupement d'Intérêt Économique
<b>HMAC</b>	: Hash-based Message Authentication Code
<b>IA</b>	: Intelligence Artificielle
<b>IFF</b>	: Identification Friend or Foe
<b>IDE</b>	: Integrated Development Environment
<b>IPFS</b>	: InterPlanetary File System
<b>JWT</b>	: JSON Web Token
<b>JSON</b>	: JavaScript Object Notation
<b>LPR</b>	: License Plate Recognition
<b>MIT</b>	: Massachusetts Institute of Technology
<b>OAS</b>	: OpenAPI Specification
<b>OS</b>	: Operating System
<b>PAD</b>	: Port Autonome de Dakar
<b>PHP</b>	: PHP Hypertext Preprocessor
<b>QR</b>	: Quick Response
<b>REST</b>	: REpresentational State Transfer
<b>RFC</b>	: Request for Comments
<b>RFID</b>	: Radio-Frequency Identification
<b>RSA</b>	: Rivest–Shamir–Adleman
<b>RUP</b>	: Rational Unified Process
<b>SOA</b>	: Service-Oriented Architecture
<b>SPA</b>	: Single Page Application
<b>SQL</b>	: Structured Query Language
<b>TOM</b>	: Terminal des Operations Maritimes

**2TUP** : Two-Track Unified Process  
**UML** : Unified Modeling Language  
**UP** : Unified Process  
**VCS** : Version Control System

# Introduction générale

Dans le cadre de l'optimisation des opérations logistiques au sein de la plateforme de distribution du Port Autonome de Dakar (PAD), la planification de rendez-vous est un élément essentiel pour garantir la fluidité des activités de distribution. Les sociétés externes jouent un rôle important dans cet écosystème, nécessitant un accès facilité à la plateforme logistique. Ainsi, ce mémoire se consacre à l'implémentation d'une plateforme de planification des rendez-vous dédiée aux sociétés externes de la plateforme de distribution, intégrant un système de contrôle d'accès.

L'objectif fondamental de cette initiative est d'améliorer la coordination et l'efficacité des opérations logistiques, en offrant aux sociétés externes un moyen systématique de planifier et de gérer leurs rendez-vous au sein de la plateforme logistique. L'intégration d'un système de contrôle d'accès vise à renforcer la sécurité et la traçabilité des activités, assurant ainsi un accès autorisé tout en facilitant le suivi des activités. Ce mémoire montre les défis spécifiques rencontrés par les sociétés lors de la planification des rendez-vous au sein de la plateforme de distribution et propose une approche méthodique pour la mise en place d'un système efficace.

Ce mémoire vise à contribuer à l'optimisation des processus logistiques en fournissant une solution intégrée qui répond aux besoins spécifiques des sociétés externes, tout en assurant une gestion sécurisée et transparente des rendez-vous au sein de la plateforme logistique du PAD.

Ce rapport de mémoire est organisé en six chapitres, abordant différents aspects du projet :

1. Description du projet : Ce premier chapitre s'articule autour de la présentation de la structure d'accueil, à savoir Atlantic Digital Solutions. Il fournit une description de la plateforme logistique et de son fonctionnement. De plus, il englobe une analyse de l'existant, dégage les problématiques, et définit les objectifs visés.
2. Généralités sur les systèmes de contrôle d'accès : Le deuxième chapitre explore les notions fondamentales des systèmes de contrôle d'accès, abordant divers aspects tels que les types de contrôles d'accès, les étapes de contrôle d'accès, les technologies utilisées, et les méthodes de blocage d'accès.
3. Méthodologies de développement : Ce troisième chapitre se concentre sur la méthodologie de développement adoptée, les concepts clés d'UML ainsi que l'architecture adoptée pour le projet.

4. Spécification et analyse des besoins : Le quatrième chapitre est dédié à une analyse des besoins des utilisateurs du système, identifiant les exigences fonctionnelles et non fonctionnelles. Il vise à établir une base solide pour la conception et l'implémentation ultérieures.
5. Conception du système : Le cinquième chapitre se penche sur la conception du système, décrivant l'architecture globale, les choix de conception, et les interactions entre les différents modules pour répondre aux besoins spécifiés.
6. Implémentation et présentation du système : Enfin, le sixième chapitre se concentre sur l'implémentation concrète du système, détaillant les outils et technologies utilisés et présentant le système finalisé.

# Chapitre 1 - Description du sujet

## Introduction

Le commerce maritime joue un rôle essentiel dans l'économie mondiale, facilitant l'importation et l'exportation de marchandises à travers les ports du monde entier. Le Sénégal, situé sur la côte ouest de l'Afrique, bénéficie d'une position géographique stratégique en tant que porte d'entrée pour les échanges commerciaux avec l'Atlantique et les pays voisins de la sous-région. Au cœur de cette activité, le PAD est un acteur clé de la distribution des marchandises importées et exportées vers le Sénégal et les pays avoisinants.

Pour assurer une gestion efficace des flux de marchandises et des opérations logistiques au sein du port, une plateforme de distribution est mise en place. Cette plateforme constitue un ensemble complexe d'infrastructures, de services et de processus visant à optimiser les opérations et à faciliter les échanges internationaux. Cependant, malgré son importance, la plateforme logistique du PAD fait face à des défis en matière de sécurité, de traçabilité et de gestion des autorisations.

La mise en place du système « AccessLog » est supervisée par Atlantic Digital Solutions entité qui sera présentée ultérieurement dans le document, au sein de laquelle nous avons effectué notre stage.

Ce chapitre débutera par une présentation de Atlantic Digital Solutions, suivie d'une introduction à la plateforme de distribution, et sera ensuite suivi d'une analyse de la situation actuelle. Enfin, la problématique sera identifiée afin de définir les objectifs.

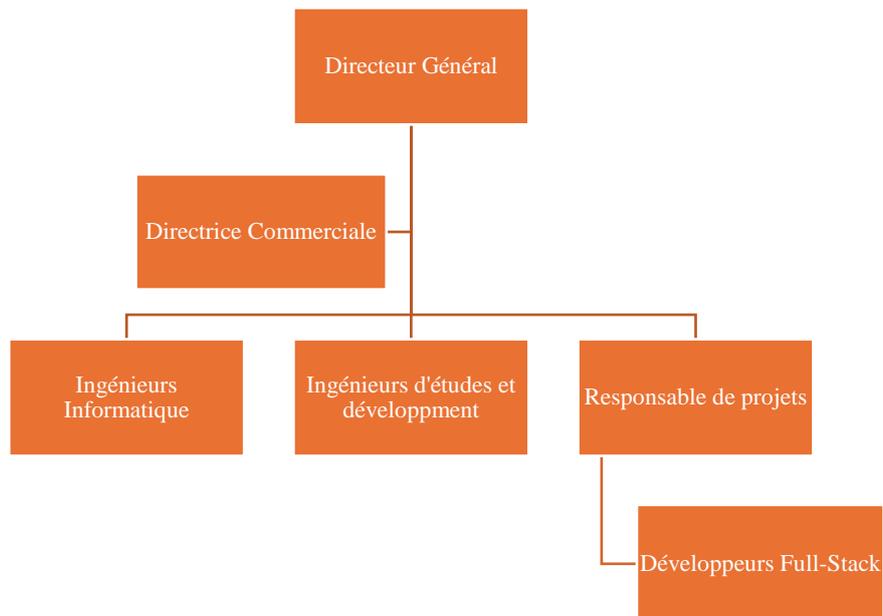
## 1.1 Présentation de Atlantic Digital Solutions

Atlantic Digital Solutions est une structure spécialisée dans la fourniture de services de transformation digitale.

### 1.1.1 Mission et organigramme

La mission d'Atlantic Digital Solutions se matérialise à travers leur dévouement à devenir le partenaire de choix pour les entreprises engagées dans leur transformation digitale. Ils s'engagent à offrir à leurs clients une expertise et un savoir-faire pour la création de produits uniques et adaptés sur mesure. Ces solutions sont conçues en intégrant les toutes dernières avancées technologiques, assurant ainsi une approche innovante parfaitement adaptée au contexte particulier de chaque entreprise.

La figure 1-1 montre l'organigramme de Atlantic Digital Solutions.



*Figure 1-1 Organigramme Atlantic Digital Solutions*

### 1.1.2 Services

Atlantic Digital Solutions propose des services qui englobe l'ensemble du processus de digitalisation. Depuis la phase de consultation en transformation digitale jusqu'à la concrétisation des solutions. Les services incluent des consultations personnalisées, le développement d'applications, l'intégration de systèmes, ainsi qu'une gestion efficace des données. Chaque service est élaboré pour répondre aux besoins spécifiques de chaque client, garantissant ainsi une transition numérique réussie et adaptée à l'écosystème particulier de chaque entreprise.

Ces services sont :

- le développement d'application web et mobile ;
- l'hébergement et nom de domaine ;
- l'implémentation de modules ERP ;
- la fourniture de licence office et Kaspersky ;
- la gestion de parc informatique ;
- la formation ;
- etc...

En somme, cette présentation souligne le rôle que joue Atlantic Digital Solutions dans le domaine de la transformation digitale. Elle souligne son engagement en faveur de l'innovation et du succès à long terme de ses clients dans cet environnement numérique en constante évolution.

## 1.2 Présentation de la plateforme logistique

La plateforme de distribution représente un système intégré d'infrastructures, de services et de processus conçu pour faciliter de manière efficace le mouvement des marchandises et des cargaisons à travers le port. L'objectif principal est d'optimiser les opérations logistiques, d'accroître la productivité et de favoriser le commerce international.

La plateforme dispose d'une capacité de stockage significative et comprend diverses installations, notamment :

- 1 entrée ;
- 1 sortie ;
- 8 hangars ;
- 2 postes de gendarmeries ;
- 1 poste de douane ;
- 1 zone de dépotage ;
- 2 opérateurs (TOM, CMA CGM) ;
- 1 parking véhicule ;
- 2 ponts bascules ;
- 1 pompe à essence.

La gestion globale de la plateforme est confiée à concessionnaire, qui assure la location des espaces aux différentes sociétés. Chaque société établie sur la plateforme est responsable de sa propre zone, supervisant le flux des marchandises et des véhicules entrants et sortants.

La figure 1-2 montre la plateforme de distribution.



Figure 1-2 Plateforme de distribution (google earth)

### 1.3 Etude de l'existant

La plateforme de distribution du PAD, gérée par un concessionnaire, occupe une position centrale dans la supervision des activités logistiques et de distribution. Elle comporte deux points d'entrée distincts, chacun équipé d'un poste de contrôle accueillant des agents de la gendarmerie.

L'analyse de la situation actuelle révèle plusieurs aspects de la plateforme, incluant la gestion des accès, l'émission de badges, la présence des sociétés établies, les interactions avec les Groupements d'Intérêt Économique (GIE), les procédures liées aux opérations de distribution et les processus de prise de rendez-vous.

- gestion des accès et émission de badges :
  - contrôle d'accès manuel : actuellement, la plateforme de distribution gère manuellement les accès, aussi bien à l'entrée qu'à la sortie, en impliquant des vigiles pour superviser ces processus. Bien que des tourniquets soient installés, ils demeurent inutilisés par les piétons. Pour les véhicules, l'accès implique le passage par une barrière levante manuelle, une procédure dirigée par les vigiles.
  - badges temporaires : le concessionnaire fournit des badges temporaires aux sociétés établies, lesquelles les attribuent ensuite à leurs visiteurs pour un accès limité à la plateforme.
- sociétés établies

À l'intérieur de la plateforme, plusieurs sociétés ont installé leurs opérations, chacune ayant son propre personnel, parc de véhicules et exigences logistiques. Ces sociétés jouent un rôle central dans les processus de distribution et de réception de marchandises, fonctionnant de manière indépendante et autonome pour superviser leurs opérations internes.

➤ GIE

Les sociétés établies entretiennent une collaboration avec des Groupements d'Intérêt Économique (GIE) afin de garantir l'approvisionnement en ressources essentielles à leurs activités logistiques. Les dirigeants de chaque GIE soumettent des demandes de badges d'accès pour leurs manœuvres au concessionnaire de la plateforme. Ces badges ont une validité déterminée, permettant ainsi l'accès à la plateforme.

➤ opérations de distribution et de réception

Les activités centrales de la plateforme se concentrent sur la distribution et la réception des marchandises, avec plusieurs dispositifs clés en place pour assurer leur bon déroulement :

- contrôle d'accès des camions externes : Les camions des sociétés externes doivent présenter un bon d'entrée pour l'accès à la plateforme, soumis à une vérification par les vigiles pour autoriser leur accès.
- contrôle de sortie : Lorsque les camions quittent la plateforme, un bon de sortie est requis. Ce document est inspecté et validé par les autorités douanières, aussi bien pour les camions des sociétés externes que pour ceux des sociétés établies. Cette procédure de contrôle de sortie revêt une importance cruciale pour garantir la conformité aux réglementations douanières.

➤ Rendez-vous

Chaque société établie dispose d'un nombre limité de badges qu'elle peut attribuer à ses visiteurs une fois à la porte principale. Ces badges sont récupérés après la visite.

Cependant, il est important de noter qu'en raison de la pénurie d'espaces de stationnement disponibles à l'extérieur de la plateforme, certains véhicules ont été autorisés à entrer, même en l'absence de justifications valables.

Cette situation compromet à la fois la sécurité et l'efficacité du contrôle d'accès. Bien que la présence des vigiles dans la configuration actuelle vise à renforcer la sécurité en surveillant les entrées et les sorties, en vérifiant les autorisations et en intervenant au besoin, elle ne parvient pas à résoudre entièrement le problème des accès non autorisés.

L'étude de l'existant est résumée par la figure 1-3.

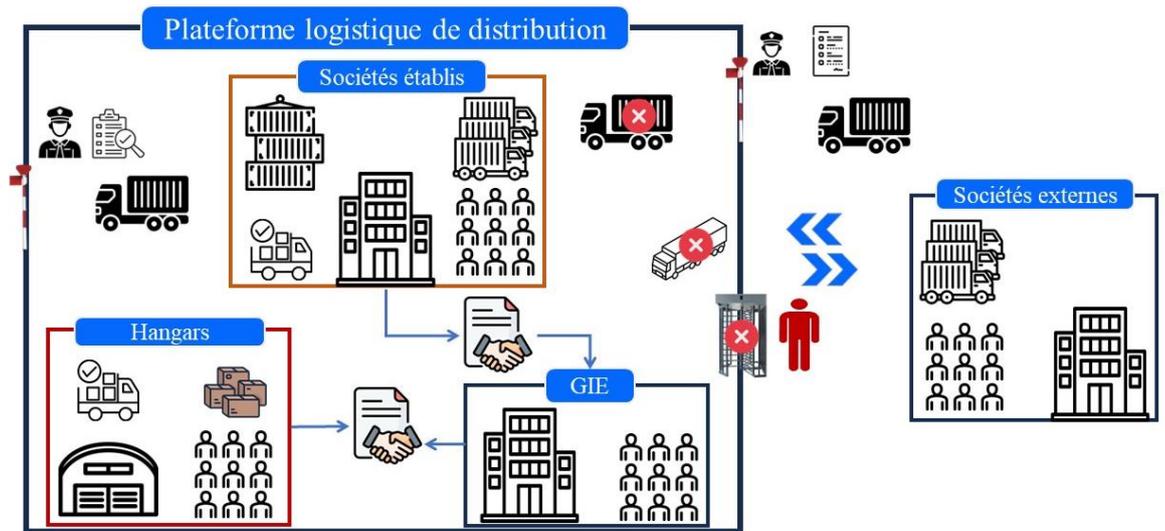


Figure 1-3 Etude de l'existant

## 1.4 Problématiques et objectifs

Le PAD est un centre névralgique dans la région. Cependant, les sociétés qui interagissent avec la plateforme de distribution du port font face à des défis majeurs en matière de planification des rendez-vous et de contrôle d'accès. Actuellement, il n'existe pas de système centralisé efficace pour gérer les rendez-vous et le contrôle d'accès des sociétés, ce qui entraîne des retards, des congestions et des inefficacités opérationnelles.

L'absence d'une plateforme dédiée à la planification des rendez-vous pour les sociétés externes entraîne une gestion désorganisée des flux logistiques au sein de la plateforme du PAD. De plus, le manque d'un système de contrôle d'accès engendre :

- des risques de sécurité ;
- des erreurs de traitement des données ;
- une faible traçabilité des mouvements des sociétés externes sur le site ;
- une difficulté d'obtenir des statistiques fiables ;
- etc.

Pour remédier à ces enjeux, le projet a été subdivisé en deux sous-projets complémentaires, chacun abordant des besoins spécifiques :

- sous-projet 1 (Monsieur Balla GNINGUE) : il est axé sur la mise en place d'un système de contrôle d'accès dédié aux sociétés établies dans la plateforme de distribution. Il vise à améliorer la fluidité des opérations tout en renforçant la sécurité, grâce à une identification automatisée des personnes et véhicules autorisés.

- sous-projet 2 (Monsieur Mouhamadou DIAMANKA) : il se concentre sur la mise en place d'un système de contrôle d'accès et de planification de rendez-vous pour les sociétés externes à la plateforme de distribution.

L'objectif principal est la mise en place d'une plateforme web et mobile dédiée à la planification des rendez-vous pour les sociétés externes à la plateforme de distribution du PAD. Ce système vise à résoudre les problèmes de coordination en offrant une solution centralisée, automatisée, et accessible. La planification des rendez-vous sera facilitée par l'utilisation de codes QR, visant à optimiser le processus de contrôle d'accès et de validation des rendez-vous.

De plus, une API est mise en place pour permettre aux sociétés ne souhaitant pas utiliser notre solution, mais qui veulent intégrer le système dans leurs applications existantes ou développer leurs propres systèmes, de se connecter de manière transparente à notre solution. Cette API offrira une flexibilité aux sociétés, leur permettant d'exploiter les fonctionnalités de planification des rendez-vous de manière efficace et personnalisée. Ainsi, notre approche inclut une solution complète, allant de la plateforme web et mobile à une API, afin de répondre aux besoins variés des sociétés et de favoriser une intégration avec leurs systèmes existants.

Les objectifs spécifiques de ce projet sont :

- résoudre les problèmes de coordination en introduisant une solution centralisée et automatisée pour la gestion des rendez-vous, visant à optimiser l'utilisation des ressources ;
- intégrer un système de contrôle d'accès basé sur les codes QR pour renforcer la sécurité des accès à la plateforme ;
- automatiser les processus de planification afin d'optimiser l'utilisation des ressources et d'éliminer les conflits d'horaire ;
- mettre à disposition des indicateurs et des statistiques sur les accès et les rendez-vous des sociétés.

L'ensemble de ces objectifs vise à instaurer une solution complète, efficace et moderne pour la planification des rendez-vous, améliorant la coordination des opérations et renforçant la sécurité et la facilité d'utilisation pour les sociétés externes.

## Conclusion

En somme, ce chapitre a fourni une vue d'ensemble de l'importance du commerce maritime dans l'économie mondiale, mettant en lumière le rôle clé du PAD en tant que point

d'entrée stratégique pour les échanges commerciaux dans la région. La mise en place d'une plateforme logistique vise à optimiser les opérations, mais des défis subsistent, notamment en matière de sécurité, de traçabilité et de gestion des autorisations d'accès.

La situation actuelle met en évidence la nécessité d'un système de planification des rendez-vous associé à un système de contrôle d'accès sécurisé, facile à intégrer aux applications existantes. Les prochains chapitres se concentreront sur l'exploration de potentielles solutions afin d'améliorer l'efficacité des opérations et la gestion des autorisations d'accès.

# Chapitre 2 - Généralités sur les systèmes de contrôles d'accès

## Introduction

Un système de contrôle d'accès a pour objectif principal de limiter l'accès à certaines ressources (zones, matériels ou informations) à un ensemble de personnes bien identifiées durant des périodes bien définies et de garder la trace des demandes d'accès autorisées ou refusées. Il donne à une entité l'autorisation d'accéder aux ressources demandées. Celles-ci peuvent être physiques, régulant l'entrée dans l'environnement bâti à l'aide de portes, de serrures, de portails et d'autres barrières, ou logiques, régulant l'accès aux réseaux, aux fichiers ou aux données dans un système informatique [1].

Tout d'abord, ce chapitre explorera les différents types de contrôle d'accès. Ensuite, seront définies les étapes nécessaires à sa mise en œuvre, suivies par l'analyse des technologies associées au contrôle d'accès et au blocage.

## 2.1 Types de contrôle d'accès

Principalement, il existe deux types de contrôle d'accès : accès physique et accès logique.

### ➤ contrôle d'accès physique

Le contrôle d'accès physique englobe les méthodes et les dispositifs déployés pour limiter l'entrée à des zones déterminées, telles que des dépôts de conteneurs, des installations, etc. Les méthodes et dispositifs déployés pour le contrôle d'accès physique incluent des portes de sécurité, des badges d'accès, des codes d'accès, etc. Son objectif principal est de permettre l'accès uniquement aux individus autorisés, renforçant ainsi la sécurité des lieux concernés.

### ➤ contrôle d'accès logique

Le contrôle d'accès logique se réfère aux mécanismes et dispositifs mis en place pour réguler l'accès à des systèmes informatiques, des réseaux, des données ou des applications. Contrairement au contrôle d'accès physique qui concerne les zones physiques, le contrôle d'accès logique vise à autoriser l'entrée uniquement aux utilisateurs autorisés dans le domaine numérique. Les méthodes et dispositifs déployés pour le contrôle d'accès logique incluent des mots de passe, des listes de contrôle d'accès (ACL), des groupes d'utilisateurs, etc.

En résumé, les contrôles d'accès physique et logique sont deux aspects importants de la sécurité informatique qui visent à protéger les données et les systèmes contre les accès non autorisés.

Dans ce document, nous nous concentrerons sur le contrôle d'accès physique.

## 2.2 Etapes de gestion du contrôle d'accès

Au niveau de chaque point de contrôle, l'accès n'est autorisé que si la personne est porteuse d'une autorisation valable (badge, QR code ... ). La figure 2-1 montre le déroulement du contrôle d'accès :

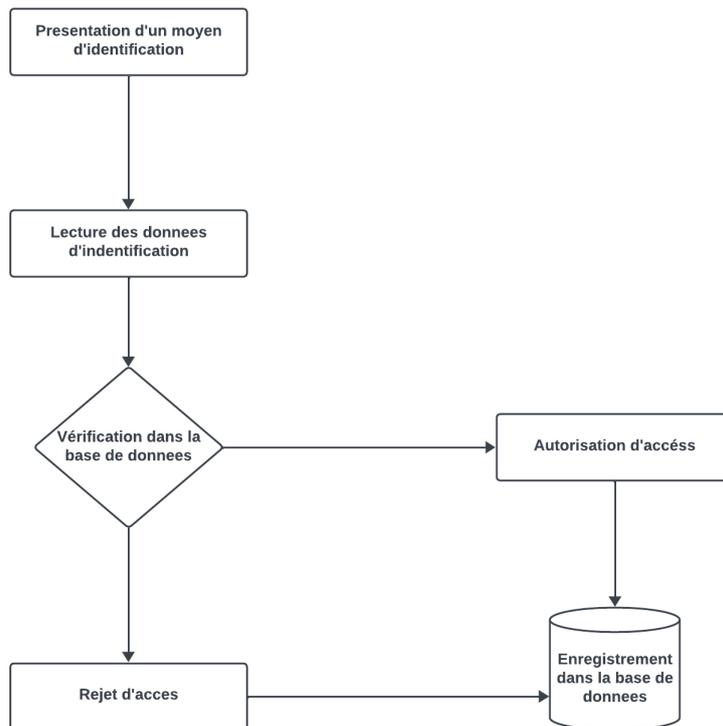


Figure 2-1 Etapes de gestion de contrôle d'accès

## 2.3 Types de systèmes de contrôle d'accès

Au sein des systèmes de contrôle d'accès, nous pouvons différencier trois types fondamentaux :

➤ systèmes autonomes

Les systèmes ne requièrent aucun type d'infrastructure ni de connectivité, le terminal dispose d'une mémoire propre et d'une capacité pour la gestion des utilisateurs. Habituellement ces types de systèmes fonctionnent avec un nombre d'utilisateurs limité et ne permettent pas la gestion des temps de travail. Ils permettent uniquement de travailler avec un ou 2 point d'accès et s'installent normalement dans des bureaux, boutiques ou entrepôts pour gérer un nombre maximum de 30 utilisateurs et un niveau de sécurité moyen [1].

➤ systèmes managés

Les systèmes managés disposent d'un logiciel qui permet d'administrer l'enregistrement et l'annulation des utilisateurs, la gestion des horaires et des autorisations. De plus, ils permettent de faire une visualisation du système et d'obtenir un registre complet des accès. Ces fonctions alliées à la vidéo surveillance permettent d'offrir un niveau de sécurité élevé afin de gérer jusqu'à 2000 employés, à un coût économique [1].

➤ systèmes corporatifs

Les systèmes corporatifs disposent d'une gamme complète d'applications web, multi-plateformes, multisites qui permettent de gérer tout type d'accès. Ces solutions ont l'avantage de permettre l'intégration avec d'autres systèmes tels que : CCTV, anti-intrusion, accès de véhicules, interphonie, alertes, messagerie, gestion d'actifs, cyber sécurité (accès physiques ou logiques), systèmes de contrôles d'assistance ou présence [1].

## 2.4 Technologies de contrôle d'accès

Il existe diverses technologies pour l'identification et le contrôle d'accès des individus et des véhicules. Parmi celles-ci, nous avons les codes QR, la technologie RFID, les technologies basées sur l'intelligence artificielle.

### 2.4.1 Code QR

Un code QR est un type de code barres matriciel ou de code bidirectionnel pouvant stocker des informations sur des données et conçu pour être lu par les smartphones. QR signifie « Quick response », ce qui signifie que le contenu du code doit être décodé très rapidement et à grande vitesse. Le code se compose de modules noirs disposés en carrée sur fond blanc. Les informations codées peuvent être du texte, une url ou d'autres types de données [2], [3].

Le code QR peut être :

- dynamique : les informations peuvent être mises à jour à tout moment ;
- statique : il stocke des informations qui ne peuvent pas être modifiées une fois le code créé.

Ils sont utilisés dans différentes domaines comme :

- le paiement par code QR : c'est une alternative à la caisse enregistreuse et à l'acquisition ;
- le contrôle des marchandises en production et par les autorités d'inspection ;
- le contrôle des membres du personnel : accès à certains locaux ou fichiers, identification ;

- mouvement des personnes : mouvement pendant la pandémie à l'intérieur d'un pays et lors du franchissement des frontières ;
- etc. ...

La figure 2-2 montre un exemple de code QR.



*Figure 2-2 Exemple de code QR*

### 2.4.2 RFID

L'identification par radiofréquence (RFID) est l'utilisation des ondes radio pour identifier des objets de façon automatique. Il est apparu durant la seconde guerre mondiale en 1939, sous l'appellation IFF (Identify Friend or Foe), où il fut utilisé par les alliés pour distinguer les avions amis de ceux ennemis. En substance, le système facilite l'identification automatique grâce à une combinaison d'étiquettes et de lecteurs [4] [5].

L'étiquette radiofréquence (transpondeur, étiquette RFID), est composée d'une puce reliée à une antenne, encapsulée dans un support (RFID Tag ou RFID Label). Elle est lue par un lecteur qui capte et transmet l'information vers un serveur. On distingue 3 catégories d'étiquettes RFID [6] :

- les étiquettes en lecture seule, non modifiables ;
- les étiquettes « écriture une fois, lecture multiple » ;
- les étiquettes en « lecture-réécriture ».

Par ailleurs, il existe deux types d'étiquettes RFID :

- les étiquettes actives : reliées à une source d'énergie embarquée (pile, batterie, etc.). Les étiquettes actives possèdent une meilleure portée, mais à un coût plus élevé et avec une durée de vie restreinte.
- les étiquettes passives : utilisant l'énergie propagée à courte distance par le signal radio de l'émetteur. Ces étiquettes à moindre coût sont généralement plus petites et possèdent une durée de vie quasi illimitée. En contrepartie, elles nécessitent une quantité d'énergie non négligeable de la part du lecteur pour pouvoir fonctionner.

Un système RFID est composé de deux entités qui communiquent entre elles : un tag ou étiquette (aussi appelé transpondeur), associé à l'élément à identifier. Il est capable de répondre à une demande venant d'un lecteur. Une station de base ou lecteur RFID qui a pour mission d'identifier le tag. Le lecteur envoie une onde électromagnétique en direction de l'élément à identifier. En retour, il reçoit l'information renvoyée par le tag.

La figure 2-3 illustre un tag et un lecteur RFID.



*Figure 2-3 Tag et Lecteur RFID*

### 2.4.3 Technologies basées sur l'IA

Il existe diverses technologies de contrôle d'accès basés sur l'intelligence artificielle comme : la reconnaissance faciale, l'empreinte digitale, la reconnaissance de la rétine, la reconnaissance de la démarche, la reconnaissance de l'iris, l'ADN, la reconnaissance de plaque d'immatriculation (LPR) etc. Parmi toutes ces technologies nous avons choisi de présenter : la reconnaissance faciale, l'empreinte digitale et la reconnaissance de l'iris.

#### ➤ la reconnaissance faciale

La technologie de reconnaissance faciale est un système qui identifie la présence d'une personne en analysant une image numérique ou une vidéo de son visage et en la comparant à des données préexistantes[7]. C'est une technologie qui permet à partir des traits du visage de :

- authentifier une personne : vérifier qu'une personne est bien celle qu'elle prétend être (dans le cadre de contrôle d'accès).
- identifier une personne : retrouver une personne au sein d'un groupe d'individus dans un lieu, une image ou une base de données.

#### ➤ l'empreinte digitale

L'empreinte digitale constitue une signature distinctive et singulière résultant des ridules et des crêtes présentes sur la pulpe des doigts. Chaque individu possède des empreintes

digitales uniques, une caractéristique qui en fait un moyen d'identification biométrique fiable et étendu. Même entre jumeaux identiques, les empreintes digitales demeurent distinctes, renforçant ainsi leur unicité en tant que moyen d'authentification.

La figure 2-4 est un exemple d'empreinte digitale.



*Figure 2-4 Empreinte digitale*

➤ Reconnaissance de l'iris

La reconnaissance de l'iris est une technologie fiable et semble être plus précise que certains autres moyens biométriques. Cela s'explique par la diversité des caractéristiques présentes dans notre iris, telles que la disposition unique des vaisseaux sanguins. Chaque iris est unique, et la probabilité de trouver deux iris identiques est très faible.

Une fois que le système biométrique a obtenu l'image de la configuration des vaisseaux sanguins, il fonctionne de manière similaire à celui analysant les empreintes digitales. Les détails tels que la taille des vaisseaux, leur positionnement et leurs bifurcations sont étudiés pour créer un algorithme spécifique. Cet algorithme est ensuite comparé avec une base de données de référence pour l'identification ou la vérification. Cependant, ce système a pour faiblesse sa difficulté à lire l'image de l'œil chez les personnes aveugles ou atteintes de cataracte [8].

#### 2.4.4 Etude comparative

Dans le tableau 2-1, nous effectuons une analyse comparative des différentes technologies de contrôle d'accès. Chaque technologie de contrôle d'accès a ses propres avantages et limitations. Cette comparaison nous permettra de mieux comprendre comment elles répondent à nos besoins. L'objectif est de choisir la technologie qui offre le meilleur équilibre entre sécurité, convivialité et coût, alignée sur les exigences du projet.

Technologies	Avantages	Limites
Code QR	<ul style="list-style-type: none"> <li>➤ peut contenir plus d'informations par rapport aux codes à barres ;</li> <li>➤ la lecture peut être plus rapide et</li> </ul>	<ul style="list-style-type: none"> <li>➤ certains codes QR peuvent être exploités à des fins malveillantes ;</li> <li>➤ la lecture nécessite un lecteur QR,</li> </ul>

	<p>plus précise ;</p> <ul style="list-style-type: none"> <li>➤ peut contenir des liens vers des sites Web, des informations de contact, des événements, etc.</li> </ul>	<p>qui n'est pas toujours présent sur tous les appareils.</p>
RFID	<ul style="list-style-type: none"> <li>➤ peut être lu sans nécessiter un contact direct ;</li> <li>➤ peut contenir plus d'informations que les codes à barres ;</li> <li>➤ peut être lu à distance, ce qui facilite la gestion des stocks et des processus logistiques.</li> </ul>	<ul style="list-style-type: none"> <li>➤ l'infrastructure RFID peut être coûteuse à mettre en place ;</li> <li>➤ risques potentiels liés à la sécurité et à la vie privée ;</li> <li>➤ les signaux radio fréquents peuvent être perturbés dans des environnements avec des interférences électromagnétiques.</li> </ul>
Reconnaissance faciale	<ul style="list-style-type: none"> <li>➤ identification rapide des individus ;</li> <li>➤ ne nécessite pas de contact physique ;</li> <li>➤ ne nécessite pas d'interaction physique, ce qui peut être plus convivial.</li> </ul>	<ul style="list-style-type: none"> <li>➤ préoccupations liées à la vie privée et à la surveillance ;</li> <li>➤ peut être moins précis avec certaines populations ou dans certaines conditions d'éclairage ;</li> <li>➤ la précision peut être affectée par des changements dans l'apparence physique, comme une coupe de cheveux ou le port de lunettes.</li> </ul>
Empreinte digitale	<ul style="list-style-type: none"> <li>➤ considérée comme une méthode de sécurité biométrique fiable ;</li> <li>➤ une méthode précise d'identification ;</li> <li>➤ le processus d'identification est souvent rapide et efficace.</li> </ul>	<ul style="list-style-type: none"> <li>➤ des préoccupations relatives à la protection de la vie privée peuvent surgir ;</li> <li>➤ la mise en place de lecteurs d'empreintes digitales peut être coûteuse ;</li> <li>➤ la sécurité des bases de données stockant les empreintes digitales est cruciale pour éviter tout accès non autorisé.</li> </ul>
Reconnaissance de l'iris	<ul style="list-style-type: none"> <li>➤ pas modifiable même par intervention chirurgicale ;</li> </ul>	<ul style="list-style-type: none"> <li>➤ aisément visible et peut être photographié ;</li> </ul>

	<ul style="list-style-type: none"> <li>➤ uniques et différent même pour les vrai jumeaux ;</li> <li>➤ contient une grande quantité d'information ;</li> <li>➤ dessin de l'iris indépendant du code génétique ;</li> <li>➤ ne varie presque pas au cours d'une vie [8].</li> </ul>	<ul style="list-style-type: none"> <li>➤ aspect psychologiquement invasif de la méthode ;</li> <li>➤ difficile de lire l'image de l'œil chez les personnes aveugles ou atteintes de cataracte [8].</li> </ul>
--	---	---

*Tableau 2-1 Tableau comparative des technologies de contrôle d'accès*

Nous avons opté pour les codes QR en raison de leur faible coût et de leur capacité à stocker des informations variées, ce qui nous permet d'inclure les détails du rendez-vous et des informations d'identification spécifiques.

Ils offrent une solution interactive pour la planification des rendez-vous, permettant aux sociétés de scanner rapidement le code avec leur téléphone ou un lecteur pour accéder aux informations de rendez-vous, aux autorisations d'accès et à la disponibilité de l'espace de la société de destination. Ils permettent d'éviter l'utilisation des données personnelles en utilisant des identifiants uniques et sécurisés générés pour chaque rendez-vous. Cette solution garantit la confidentialité des utilisateurs tout en facilitant la gestion efficace des rendez-vous et des accès.

Les codes QR sont plus évolutifs dans le sens où ils peuvent s'adapter à différentes tailles tout en conservant leur capacité de stockage d'informations.

## 2.5 Modèles de blocage d'accès

Les équipements d'accès sont utilisés pour gérer le contrôle de passage de personnes et de véhicules aux points de contrôle.

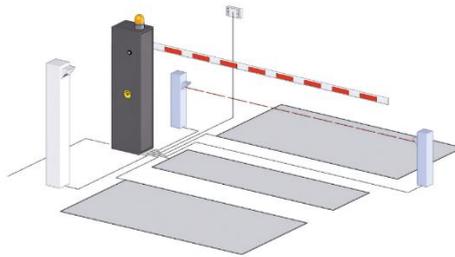
- lorsqu'une demande est reçue, le système de contrôle vérifie les données de la demande dans la base de données ;
- si la demande est autorisée, le mécanisme est débloqué pour permettre le passage. Le verrou interne est électriquement débloqué pendant un certain nombre de secondes (paramétrable).
- l'équipement revient à son état de blocage après le passage de la personne ou du véhicule une fois le délai de n secondes écoulées (Time out).

Il existe plusieurs types d'équipements de blocage d'accès, mais pour notre cas, nous nous concentrerons sur deux : les barrières et les tourniquets.

### 2.5.1 Barrière

Une barrière est un dispositif mécanique utilisé pour contrôler l'accès à un espace en restreignant ou en permettant le passage de véhicules. Elles sont souvent utilisées dans les parkings, les péages d'autoroute, les zones industriels et d'autres endroits où la gestion du flux de véhicules est nécessaire. Elles peuvent être actionnées manuellement ou automatiquement à l'aide d'un dispositif de contrôle d'accès. Les barrières contribuent à la sécurité et à la gestion efficace des espaces.

La figure 2-5 montre un exemple de barrière routière.



*Figure 2-5 Barrière routière*

### 2.5.2 Tourniquet tripode

Les tourniquets sont équipés d'un mécanisme à bras articulés. Le bras qui se trouve à l'horizontale bloque le passage. Après rotation d'un tiers ( $1/3$ ) de tour du mécanisme le bras tombe en position verticale libérant le passage d'un seul utilisateur.

La figure 2-6 montre un exemple de tourniquet tripode.



*Figure 2-6 Tourniquet tripode*

## Conclusion

La mise en place d'un système de contrôle d'accès revêt d'une importance cruciale dans la sécurisation des ressources, qu'elles soient physiques ou logiques. À travers ce chapitre, nous avons exploré les diverses facettes du contrôle d'accès, ses objectifs principaux, et les différents types de systèmes qui y sont associés. Que ce soit pour réguler l'accès à des

zones physiques à l'aide de barrières, tourniquets, et autres dispositifs, le contrôle d'accès joue un rôle central dans la gestion de la sécurité.

Nous avons également examiné les étapes impliquées dans la mise en œuvre d'un tel système, mettant en lumière l'importance de définir clairement les autorisations d'accès et de suivre les demandes autorisées ou refusées. Par la suite, nous avons exploré les technologies de contrôle d'accès, en mettant l'accent sur les mécanismes de blocage.

Enfin, nous avons comparé différentes technologies d'accès et avons conclu que l'utilisation de codes QR est préférable. Le choix des codes QR se justifie par la simplicité d'utilisation, la facilité d'installation, et la traçabilité efficace des accès autorisés.

# Chapitre 3 - Méthodologies de développement

## Introduction

Dans ce chapitre, nous explorerons d'abord différentes méthodes de développement logiciel. Ensuite, nous parlerons du langage de modélisation UML, qui est à la base de plusieurs approches méthodologiques. Enfin, nous aborderons l'architecture adaptée à la plateforme « AccessLog ».

## 3.1 Méthodologies

La gestion de projet est définie comme l'application de connaissances, de compétences, d'outils et de techniques aux activités d'un projet dans le but de répondre à ses exigences. Afin de faciliter cette gestion, diverses méthodologies ont été élaborées. Dans notre cas, nous explorerons les méthodologies suivantes : modèle en cascade, processus unifié (UP) et Scrum.

### 3.1.1 Modèle en cascade

Le modèle en cascade présente le développement logiciel comme une suite de phases qui s'enchaînent dans un déroulement linéaire, depuis l'analyse des besoins jusqu'à la livraison du produit au client. Chaque phase correspond à une activité. Le développement en cascade est rythmé par la génération de documents. Ceux-ci servent de support tangible lors des revues de validation, lors du passage d'une phase à une autre [9]."

Le cycle de vie du modèle en cascade est montré par la figure 3-1.

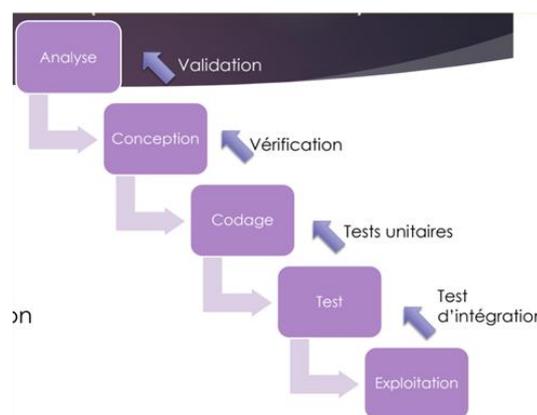


Figure 3-1 Cycle de vie modèle en cascade

Le cycle en cascade se caractérise par des phases séquentielles, qui se succèdent après la validation des livrables produits lors de la phase précédente [10]:

- tous les besoins sont exprimés et recueillis lors de la première phase, puisque l'analyse détaillée de ces besoins, puis la conception du système qui répondra à ces besoins, en dépendront.
- la conception du système, bien que textuelle ou représentée sous forme de diagrammes, doit être validée avant le démarrage des développements.
- les développements doivent être achevés pour permettre à l'équipe de testeurs de lancer ses campagnes de tests fonctionnels et techniques.
- enfin, une fois, et seulement une fois, que les anomalies ont été corrigées, on peut procéder à l'intégration globale finale et à la mise en production du système.

### 3.1.2 Processus Unifié

Le Processus Unifié (UP) est un processus de développement logiciel « itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques » [11]:

1. itératif et incrémental : Le projet est découpé en itérations de courte durée (environ 1 mois) qui aide à mieux suivre l'avancement global. A la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
2. centré sur l'architecture : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle etc...) doit être modélisée en UML et pas seulement documentée en texte.
3. piloté par les risques : Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
4. conduit par les cas d'utilisations : Le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisations du futur système sont identifiés, décrits avec précision et priorisés.

En somme, UP est un processus de développement logiciel construit sur UML, il est itératif et incrémental, centré sur l'architecture, piloté par les cas d'utilisation afin de diminuer au maximum les risques [12] [13].

UP gère le processus de développement par deux axes [13] comme illustré par la figure 3-2:

- axe vertical représente les phases ;
- axe horizontal représente les activités.

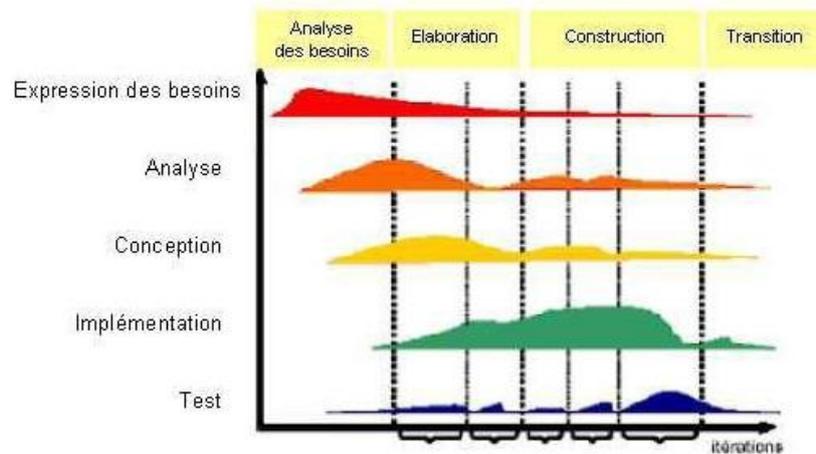


Figure 3-2 Cycle de vie du processus unifié

Le processus unifié répète un certain nombre de fois une série de cycle qui s’articule autour de 04 phases : analyse des besoins, élaboration, construction, transition [14]. Le tableau 3-1 résume les différentes phases du processus UP [10] :

Phase	Description
Analyse des besoins (initialisation)	A pour objectif d’obtenir un consensus de l’ensemble des parties prenantes sur la vision ou la portée du projet, le périmètre et l’estimation globale
Elaboration	A pour objectifs d’établir et de valider l’architecture de référence, de lever les principaux risques ainsi que de spécifier en détail les exigences ; c’est une phase exploratoire.
Construction	A pour objectif de livrer une première version opérationnelle du système, accompagnée de sa documentation.
Transition	A pour objectif de déployer et mettre en production la version finale de l’application, testée par les utilisateurs.

Tableau 3-1 Les phases du processus unifié

Il est important de noter que la méthode UP présente plusieurs variantes et implémentations dérivées telles que le Rational Unified Process (RUP), les Two-Track Unified Process (2TUP), entre autres.

### 3.1.3 Scrum

Scrum est un ensemble de processus destiné à une équipe unique, et utilisé pour développer des produits. Ce cadre de travail se compose de rôles, d’événements, d’artefacts et de règles. Il utilise une approche itérative pour livrer un produit opérationnel. Scrum est divisé

en blocs de temps d'une durée fixe d'un mois au maximum, appelés sprints, lors desquels un incrément de produit potentiellement livrable est créé [15].

L'équipe Scrum se compose d'un Product Owner, d'une équipe de développement et d'un Scrum Master [15] [16]:

- le Product Owner est responsable de maximiser la valeur du produit.
- l'équipe de développement est une équipe autoorganisée pluridisciplinaire dont les membres ont toutes les compétences requises pour livrer un produit opérationnel sans avoir à dépendre d'autres personnes extérieures à l'équipe.
- la mission du Scrum Master consiste à s'assurer que le processus Scrum est respecté et que l'équipe Scrum adhère aux pratiques et règles. Le Scrum Master guide afin de lever les obstacles éventuels.

Le tableau 3-2 énumère les événements et les artefacts Scrum utilisés lors de la réalisation d'un projet.

Evènements	Artefacts
Sprint	Backlog de produit
Planification du sprint	Backlog du sprint
Daily scrum	Increments
Revue du sprint	
Rétrospective du sprint	

Tableau 3-2 Evènements et artefacts Scrum

Le cycle de vie Scrum représenté par la figure 3-3, est rythmé par des itérations de quatre semaines (sprints); la liste des exigences initiales, dressée et hiérarchisée avec le client constitue le backlog de produit (product backlog) [10].

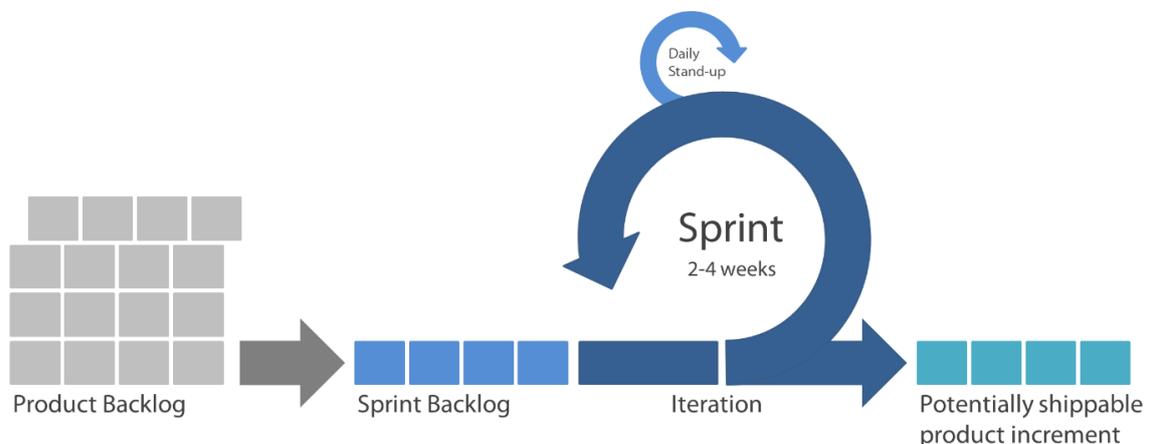


Figure 3-3 Cycle de vie Scrum

### 3.1.4 Etude comparative des méthodes

Afin de choisir la bonne méthode pour le processus de développement du système, une étude comparative des différentes méthodes précédemment étudiées est réalisée dans le tableau 3-3.

Méthodes	Résumé	Points forts	Points faibles
Cascade	Simple et directe	<ul style="list-style-type: none"> <li>➤ une planification structurée ;</li> <li>➤ une documentation détaillée ;</li> <li>un suivi facile.</li> </ul>	<ul style="list-style-type: none"> <li>➤ impossible de retourner sur une phase précédente ;</li> <li>➤ non transparente avec le client ;</li> <li>contrôle qualité tardif.</li> </ul>
UP	Gestion des risques	<ul style="list-style-type: none"> <li>➤ très bien documentée ;</li> <li>➤ est devenue un standard dans de nombreuses organisations ;</li> <li>➤ un bon intermédiaire pour le passage d'une méthode classique à une méthode agile [10].</li> </ul>	<ul style="list-style-type: none"> <li>➤ méthode la moins agile de toutes, prescriptive, qui peut être lourde à mettre en œuvre ;</li> <li>➤ beaucoup de livrables à produire [10].</li> </ul>
Scrum	Valeur ajoutée pour le client	<ul style="list-style-type: none"> <li>➤ les priorités sont gérées en fonction de la valeur ajoutée ;</li> <li>➤ bien adaptée au développement d'un logiciel au sein d'équipes produit, grâce à la présence d'un product owner ;</li> <li>➤ favorise la communication et la collaboration ;</li> <li>➤ propose un programme de certification [10].</li> </ul>	<ul style="list-style-type: none"> <li>➤ limite les changements avec des itérations de trente jours et un contenu figé durant le sprint ;</li> <li>➤ limitée à la discipline gestion de projet ;</li> <li>➤ ne propose aucune pratique technique [10].</li> </ul>

*Tableau 3-3 Etude comparative des modèles*

Toutes ces méthodes présentent des points forts ainsi que des points faibles. Aucune méthode n'est meilleure qu'une autre; le choix de la méthode dépend du contexte du projet, des ressources disponibles, de sa réalisation et de la taille du projet. Ainsi, pour notre cas, nous opterons pour la méthode Scrum dans le cadre de ce projet en raison de sa:

- flexibilité face aux changements ;
- transparence dans la communication ;
- capacité à assurer une livraison régulière de fonctionnalités opérationnelles.

En plus les rôles clairement définis et la nature collaborative de scrum favorisent une gestion efficace du projet, tandis que son approche itérative permet une adaptation continue aux évolutions du contexte. En somme, Scrum s’aligne parfaitement avec les besoins de notre projet en offrant une méthodologie agile, transparente et collaborative.

Le tableau 3-4 montre l’organisation de l’équipe Scrum chargée de la réalisation du projet.

Prénoms et noms	Rôles
Monsieur Moussa DIOP	Product owner
Madame Rokhaya Diop SARR	Scrum Master
Monsieur Balla GNINGUE Monsieur Mouhamadou DIAMANKA	Equipe de développement

*Tableau 3-4 : Organisation de l'équipe Scrum*

## 3.2 UML

UML est un langage de modélisation graphique et textuel qui permet de modéliser les activités des différents domaines et ce n’est pas seulement pour les applications informatiques. Plus clairement, UML permet d’offrir des outils d’analyse, de conception, de documentation et d’implémentation des systèmes logiciels [17] [18].

C’est un langage de modélisation universel qui unifie les notations et les concepts orientés objets et offre le moyen de modéliser les décisions prises depuis la spécification des besoins jusqu’au déploiement.

Il est constitué d’un ensemble de diagrammes, qui permettent de représenter le logiciel à développer : son fonctionnement, sa mise en route et les actions susceptibles d’êtres effectués par le logiciel. Un diagramme est la représentation graphique d’un ensemble d’éléments qui constitue un système. Il donne à l’utilisateur un moyen de visualiser et de manipuler les éléments de modélisation. UML modélise le système sur deux aspects :

- l’un concerne la structure du système (diagrammes structurels) : permettant de visualiser, spécifier construire et documenter les aspects statiques ou structurel du système.
- l’autre concerne sa dynamique de fonctionnement (diagrammes comportementaux) : modélisant les aspects dynamiques du système qui incluent les interactions entre le système et ses différents acteurs.

La figure 3-4 montre la hiérarchie des diagrammes d'UML 2.0 [19].

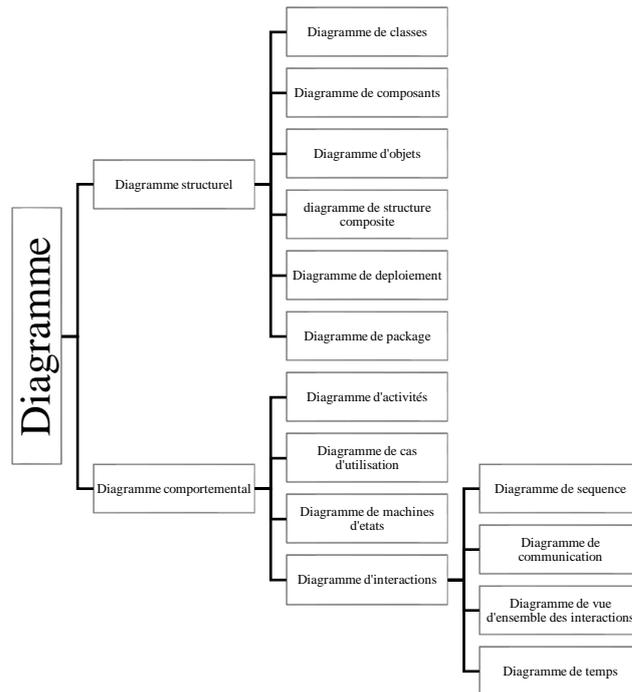


Figure 3-4 Hiérarchie des diagrammes d'uml 2.0

Durant notre stage, nous avons ainsi utilisé UML pour modéliser notre système à l'aide de plusieurs types de diagrammes tels que le diagramme de cas d'utilisation, de séquences, d'activités, de package, de déploiement, de composants et de classes.

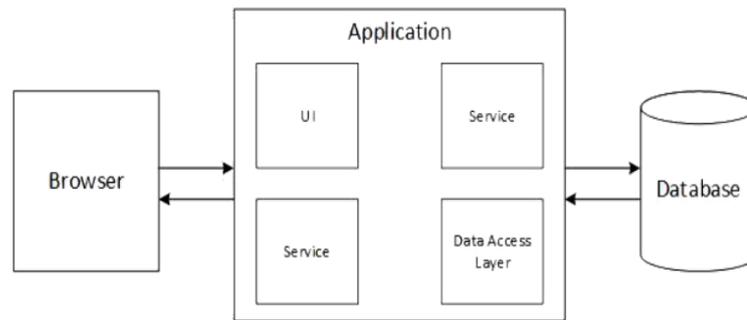
### 3.3 Architecture

Il existe diverses architectures logicielles, notamment l'architecture monolithique, l'architecture microservices, l'architecture orientée services (SOA), l'architecture n-tiers, l'architecture peer-2-peer, entre autres. Chacune de ces architectures présente des caractéristiques distinctes qui influent sur la manière dont les applications sont conçues, déployées et maintenues. Dans cette section, nous présenterons les différentes architectures logicielles, en mettant l'accent sur la comparaison entre l'architecture monolithique, l'architecture microservices et l'architecture orientée services (SOA).

#### 3.3.1 Architecture monolithique

Une architecture monolithique est une architecture dans laquelle une application logicielle est conçue pour fonctionner comme une unité unique et autonome. Les applications dotées de ce type d'architecture sont courantes. Les composants d'une architecture monolithique sont interconnectés et interdépendants, ce qui donne lieu à un code étroitement couplé [20].

La figure 3-5 illustre un système doté d'une architecture monolithique:



*Figure 3-5 Architecture monolithique*

Les différentes préoccupations d'une application, telles que l'interface utilisateur, la logique métier, l'autorisation, la journalisation et l'accès à la base de données, ne sont pas séparées. Ces différentes fonctionnalités sont entrelacées dans une application monolithique [20].

### 3.3.2 SOA

L'architecture orientée services (SOA) est un modèle architectural permettant de développer des systèmes logiciels en créant des services interopérables et faiblement couplés qui fonctionnent ensemble pour automatiser les processus métier. Un service est une partie d'une application logicielle qui exécute une tâche spécifique, fournissant des fonctionnalités à d'autres parties de la même application logicielle ou à d'autres applications logicielles. Quelques exemples de consommateurs de services incluent les applications Web, les applications mobiles, les applications de bureau et d'autres services.

Chaque service d'une SOA encapsule un certain élément de logique. Cette logique peut être responsable d'une tâche très spécifique, d'un processus métier ou d'un sous-processus. Les services peuvent varier en taille et un service peut être composé de plusieurs autres services pour accomplir sa tâche [20], [21].

La figure 3-6 montre l'architecture SOA :



Figure 3-6 Architecture SOA

### 3.3.3 Architecture microservices

L'architecture microservices crée des applications logicielles à l'aide de petits services autonomes, dotés de versions indépendantes et autonomes. Ces services utilisent des interfaces bien définies et communiquent entre eux via des protocoles standard et légers [20].

La figure 3-7 illustre un système doté d'une architecture de microservices :

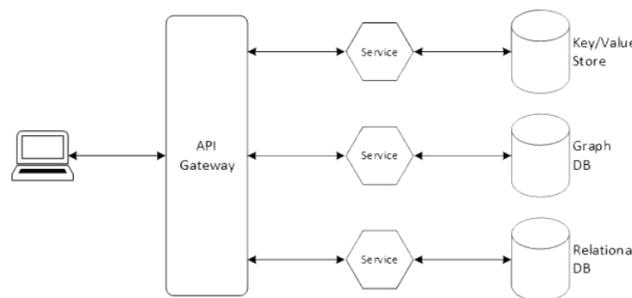


Figure 3-7 Architecture microservices

### 3.3.4 Etude comparative

Dans le tableau 3-5, nous faisons une étude comparative entre les différentes architectures suscitées, à savoir l'architecture monolithique, l'architecture microservices et l'architecture SOA.

Chaque architecture présente des avantages et des limites par rapport à l'autre. Le but est de faire un choix dépendant du projet et de ses spécifications. Ce qui permettra de trouver l'équilibre optimal entre les fonctionnalités recherchées et les contraintes particulières.

Architecture	Avantages	Limites
Monolithique	➤ plus faciles à déployer en raison de la simplicité de l'architecture de haut	➤ modifications deviennent plus complexes, car une modification dans une partie de l'application est

	<p>niveau ;</p> <ul style="list-style-type: none"> <li>➤ plus faciles à tester et à déboguer car elles sont plus simples et comportent moins de composants distincts à prendre en compte ;</li> <li>➤ une meilleure performance.</li> </ul>	<p>plus susceptible d'affecter d'autres parties de l'application ;</p> <ul style="list-style-type: none"> <li>➤ point de défaillance unique ;</li> <li>➤ même si une modification est apportée à un seul composant d'une application monolithique, l'ensemble du système logiciel devra être déployé.</li> </ul>
SOA	<ul style="list-style-type: none"> <li>➤ améliore l'alignement entre l'entreprise et la technologie ;</li> <li>➤ favorise la fédération au sein d'une organisation ;</li> <li>➤ permet la diversité des fournisseurs ;</li> <li>➤ augmente l'interopérabilité intrinsèque ;</li> <li>➤ fonctionne bien avec les méthodologies de développement agiles [20].</li> </ul>	<ul style="list-style-type: none"> <li>➤ l'introduction de nouvelles couches dans les architectures logicielles augmente la complexité du système ;</li> <li>➤ la gestion du déploiement de nouveaux services et des mises à jour des services existants devient plus complexe avec un nombre croissant de services.</li> </ul>
Microservice	<ul style="list-style-type: none"> <li>➤ services autonomes et déployables indépendamment, ce qui facilite les mises à jour et les déploiements continus ;</li> <li>➤ stockage de données indépendante, permettant aux microservices d'utiliser différents types de bases de données adaptés à leurs besoins spécifiques ;</li> <li>➤ meilleure isolation des pannes, car un échec dans un</li> </ul>	<ul style="list-style-type: none"> <li>➤ complexité opérationnelle en raison du nombre de services à gérer, déployer et surveiller ;</li> <li>➤ défi de garantir la cohérence des données à travers les microservices, en raison de la décentralisation des bases de données.</li> </ul>

	microservice n'affecte pas nécessairement l'ensemble du système.	
--	--	--

*Tableau 3-5 Etude comparative entre architecture*

Après une analyse comparative entre les différentes architectures suscitées, notre choix s'est porté sur l'architecture microservices. Les microservices offrent une flexibilité significative grâce à leur capacité à permettre des mises à jour indépendantes et des déploiements continus, simplifiant ainsi la gestion des évolutions fonctionnelles et des corrections d'erreurs sans perturber l'ensemble du système.

De plus, la possibilité d'utiliser des bases de données indépendantes pour chaque microservice offre une adaptabilité aux besoins de chaque composant, favorisant l'efficacité et la performance. La modularité inhérente à cette architecture facilite la gestion, le déploiement et la surveillance individuelle des microservices malgré la complexité opérationnelle associée. La résilience améliorée des microservices, assurant une meilleure isolation des pannes, a été un facteur déterminant dans notre choix.

## Conclusion

Ce chapitre a exploré diverses méthodologies de développement, notamment le modèle en cascade, UP et Scrum, privilégiant finalement Scrum suite à une étude comparative. Le langage de modélisation UML a également été examiné, soulignant son utilité pour représenter visuellement les concepts logiciels. En ce qui concerne les architectures, le choix s'est porté sur les microservices, jugés plus adaptatifs aux exigences évolutives du projets.

En combinant ces éléments, notre approche méthodologique intègre l'agilité de Scrum, la clarté visuelle d'UML et la flexibilité des microservices, visant ainsi à optimiser le processus de développement logiciel pour une meilleure réussite du projet.

# Chapitre 4 - Spécification et analyse des besoins

## Introduction

Ce chapitre identifie les fonctionnalités du système pour chaque type d'utilisateur en recensant les besoins fonctionnels. Il établit également la liste des exigences traduites par les besoins non fonctionnels. Ceci se fera par l'identification des acteurs et la définition de tous les besoins qui seront ensuite traduits en diagrammes de cas d'utilisation pour chaque acteur. Nous pourrons ensuite étudier toutes les interactions entre les acteurs et le système, et cela se fera à l'aide de diagramme de séquences et diagramme d'activités.

## 4.1 Spécification des besoins

Cette partie implique la nécessité de saisir le contexte dans lequel le système opère. De ce fait, pour mieux illustrer la spécification des besoins, nous allons définir les besoins fonctionnels et non fonctionnels, identifier les acteurs ainsi que leurs rôles.

### 4.1.1 Expressions des besoins

Les plateformes web et mobile auront comme objectifs d'assurer et de faciliter la planification de rendez-vous afin de gérer de manière efficace les flux de véhicules. Elle devra permettre de gérer facilement le contrôle d'accès ainsi que d'assurer un meilleur suivi de la situation sur la plateforme de distribution.

L'expression des besoins permet de définir les différents besoins en :

- recensant les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation ;
- appréhendant les besoins non fonctionnels (technique) et livrer une liste des exigences [13].

#### ❖ Besoins fonctionnels

Les besoins fonctionnels ci-dessous nous permettrons de préciser et d'identifier toutes les fonctionnalités de notre système qui représente les différents modules :

- gestion des sociétés et ressources ;
- planification de rendez-vous ;
- gestion de contrôles d'accès ;
- notifications sur les activités.

#### ❖ Besoins non fonctionnels

Pour les besoins non fonctionnels, il s'agit des besoins exprimés par le système. Ce sont des besoins en matière de performance, de type de matériel ou de conception.

Ces contraintes sont les suivantes :

- la rapidité du traitement

Vu le nombre important des transactions quotidiennes possibles, il est nécessaire que la durée d'exécution des traitements soit courte.

- la performance

Un système doit être avant tout performant, c'est-à-dire à travers ses fonctionnalités, il doit répondre d'une manière optimale à toutes les exigences des usagers.

- la convivialité

Les interfaces doivent être conviviales et faciles à utiliser. Autrement dit, elles doivent être simples, ergonomiques et adaptées à l'utilisateur.

- la confidentialité

Vu que les données manipulées par nos applications sont critiques, sensibles; nous devons garantir une sécurité optimale. Ainsi, les droits d'accès au système doivent être bien attribués afin d'assurer la sécurité des données.

### 4.1.2 Identification des acteurs

Un acteur représente une personne, un périphérique ou un autre système qui joue un rôle (interagit) avec le système [22].

En UML on distingue deux catégories d'acteurs [23]:

- acteurs primaires : l'objectif du cas d'utilisation est essentiel ;
- acteurs secondaires : interagissent avec le cas d'utilisation mais dont l'objectif n'est pas essentiel.

Les principaux acteurs de notre système sont le responsable société et l'administrateur.

### 4.1.3 Identification des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier [19] [11].

Chaque cas d'utilisation spécifie le comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera [19] [23].

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes intentions « métier » selon lesquelles il utilise le système.

Voici les principaux cas d'utilisation de notre système listés dans le tableau 4-1:

Modules	Cas d'utilisation
Planification de rendez-vous	<ul style="list-style-type: none"> <li>➤ faire une demande de rendez-vous</li> <li>➤ modifier le rendez-vous</li> <li>➤ valider le rendez-vous</li> <li>➤ annuler le rendez-vous</li> <li>➤ visualiser l'historique des rendez-vous</li> </ul>
Gestion des ressources	<ul style="list-style-type: none"> <li>➤ ajouter, modifier, activer, désactiver un employé</li> <li>➤ ajouter, modifier, activer, désactiver un camion</li> <li>➤ vérification de la disponibilité de l'espace</li> </ul>
Gestion de contrôles d'accès	<ul style="list-style-type: none"> <li>➤ octroyer une autorisation</li> <li>➤ visualiser les entrées/sorties</li> </ul>
Notifications	<ul style="list-style-type: none"> <li>➤ visualiser les activités</li> </ul>

*Tableau 4-1 Liste de modules et cas d'utilisation*

#### 4.1.4 Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation modélise le comportement d'un système et permet de capturer les exigences du système. Il décrit les fonctions générales et la portée d'un système. Il identifie également les interactions entre le système et ses acteurs [25].

Le diagramme de cas d'utilisation a pour rôle de servir à résumer les informations des acteurs et leurs interactions avec le système.

Afin de montrer les interactions entre les différents acteurs et le système, nous allons représenter le diagramme de cas d'utilisation correspondant à chaque acteur.

Les figures 4-1 et 4-2 montrent quelques cas d'utilisation du système.

- diagramme de cas d'utilisation de l'Administrateur

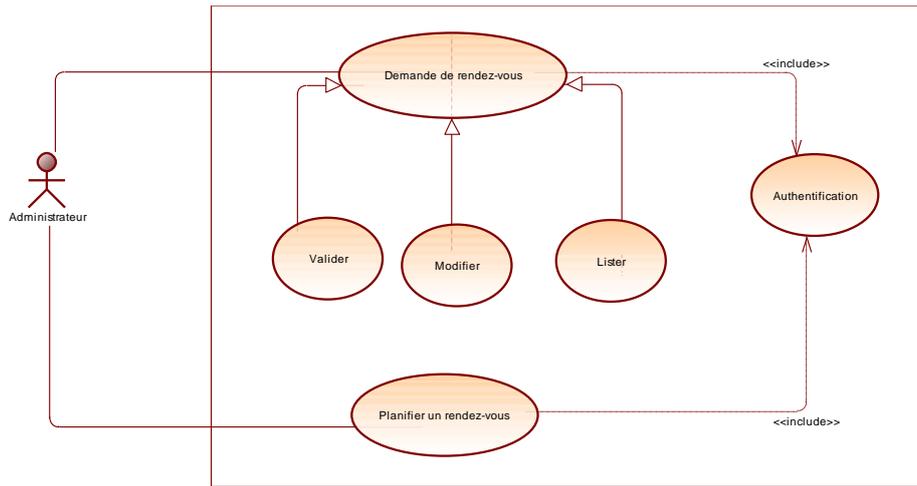


Figure 4-1 Diagramme de cas d'utilisation de l'administrateur

➤ diagramme de cas d'utilisation du Responsable société

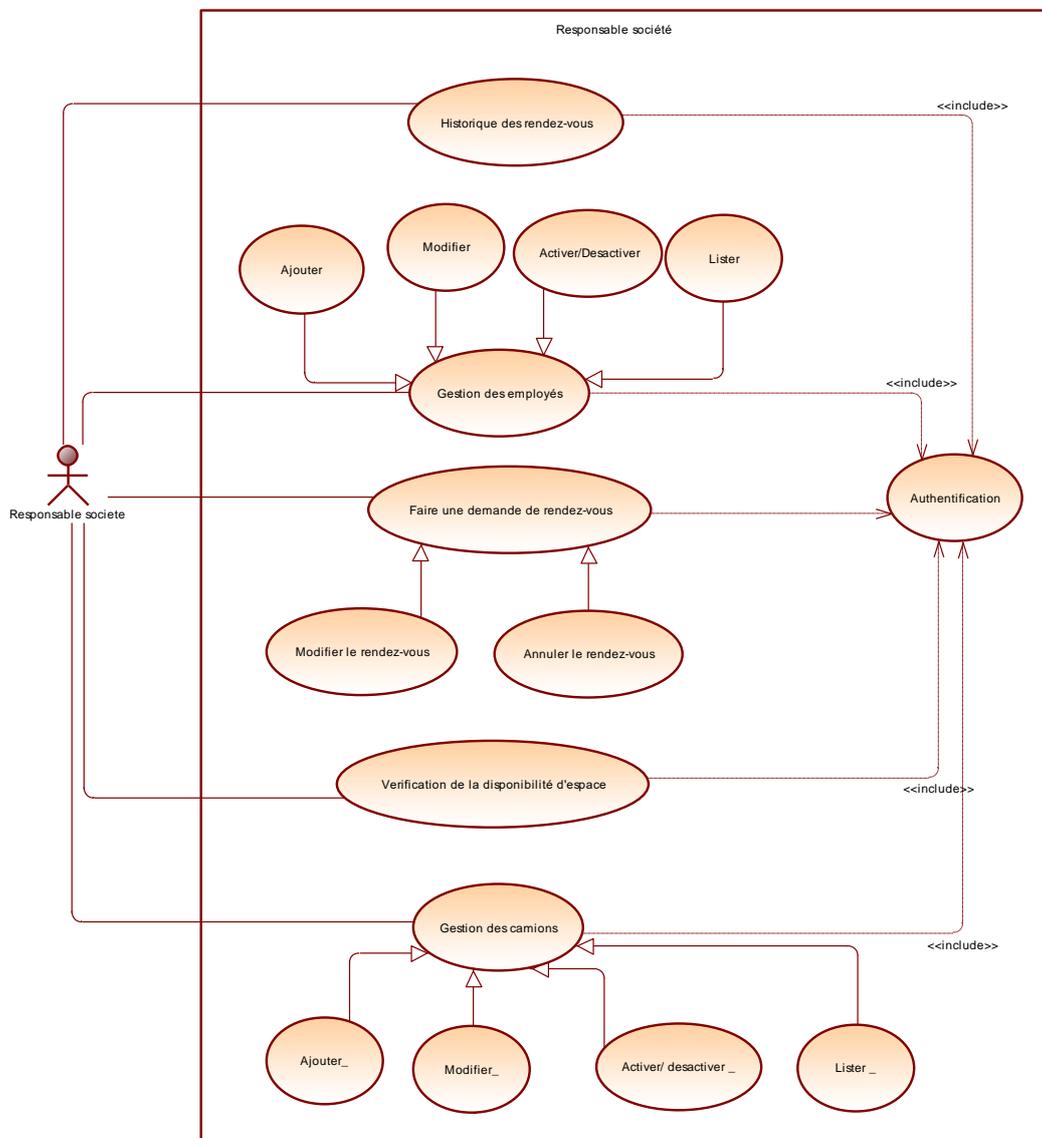


Figure 4-2 Diagramme de cas d'utilisation du Responsable société

## 4.2 Analyse des besoins

L'analyse des besoins constitue une phase essentielle dans le développement d'un système. Elle permet de comprendre en profondeur ce que les utilisateurs attendent et nécessitent. Dans cette partie, nous allons faire une description textuelle de quelques cas d'utilisation, représenter les interactions dans un cas d'utilisation à travers des diagrammes de séquences et d'activités.

### 4.2.1 Description textuelle des cas d'utilisation

Dans cette section, nous examinerons en détail le fonctionnement de certains cas d'utilisation précédemment identifiés. Cela implique la description textuelle de toutes les interactions entre les acteurs et le système.

Les tableaux 4-2 et 4-3 montrent la description textuelle de quelques cas d'utilisation.

- description textuelle du cas d'utilisation « Contrôle d'accès des camions »

Nom	Contrôle d'accès des camions
Acteurs	Système de contrôle d'accès.
Résumé	Gestion des entrées/sorties des camions des sociétés externes à la plateforme de distribution.
Précondition	Le camion dispose d'une autorisation valide
Scenario nominal	<ul style="list-style-type: none"><li>➤ le camion se présente à la portail(entrée ou sortie)</li><li>➤ il présente ses autorisations (Code QR, code barre)</li><li>➤ le système vérifie l'autorisation</li><li>➤ le camion est autorisé à entrer ou sortir.</li></ul>
Postcondition	Le camion est autorisé à entrer ou sortir de la plateforme.
Exception	Si le camion se présente avec une autorisation invalide, l'accès est refusé. Si le nombre de place alloué à la société de destination est atteint, l'accès est refusé.

Tableau 4-2 Description textuelle du cas d'utilisation « Contrôle d'accès des camions »

- description textuelle du cas d'utilisation « Planifier un rendez-vous »

Nom	Planifier un rendez-vous
Acteurs	Administrateur
Résumé	Planifier un rendez-vous pour une société externes à la plateforme de distribution.

Précondition	L'administrateur doit être authentifié
Scenario nominal	<ul style="list-style-type: none"> <li>➤ l'administrateur choisi une société externes</li> <li>➤ il rempli les informations de rendez-vous</li> <li>➤ il valide les informations de rendez-vous</li> <li>➤ le système enregistre et génère un code QR</li> <li>➤ le système envoie ce code QR à la société concerné</li> </ul>
Postcondition	Le rendez-vous est planifié.
Exception	Si les informations fournies sont incomplète ou incorrects, le système génère une erreur et guide l'utilisateur pour corriger les informations.

*Tableau 4-3 Description textuelle du cas d'utilisation « Planifier un rendez-vous »*

Pour faire la documentation de cas d'utilisation, la description textuelle est essentielle et efficace pour le recueil des exigences sur la base des scenarios d'utilisation du système car il se focalise sur les interactions des acteurs. Il permet également de préparer les tests de recettes basés sur l'utilisation du système.

Alors pour compléter la description textuelle nous allons faire plusieurs diagrammes de séquences afin de montrer comment les enchainements se succèdent.

#### 4.2.2 Diagramme de séquences

Le diagramme de séquences est considéré comme un des diagrammes d'UML les plus importants car il permet de comprendre le fonctionnement du logiciel. Il montre, selon un ordre chronologique, d'une part, les interactions entre les objets et l'extérieur du système au cours d'un scénario bien déterminé et d'autre part, les interactions entre les objets à l'intérieur du système.

En effet, le diagramme de séquence permet de représenter des collaborations entre objets selon un point de vue temporel, en mettant l'accent sur la chronologie des envois de messages. Dans les premières phases du cycle de développement, un diagramme de séquence est exploité pour illustrer des cas d'utilisation et offre en plus un excellent moyen pour communiquer les aspects dynamiques d'un système [26] [27].

Pour capturer toute la dynamique d'un système, il est nécessaire de faire plusieurs diagrammes de séquence. Dans notre cas, nous ferons appel à un ensemble de diagrammes de séquences (figure 4-3 et figure 4-4) chacun correspondant à un cas d'utilisation.

- diagramme de séquences « Contrôle d'accès des camions»

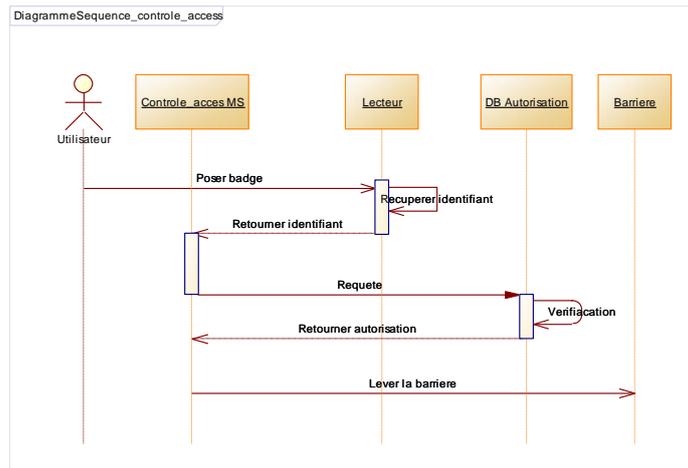


Figure 4-3 Diagramme de séquences « Contrôle d'accès des camions »

➤ diagramme de séquences « Planifier un rendez-vous »

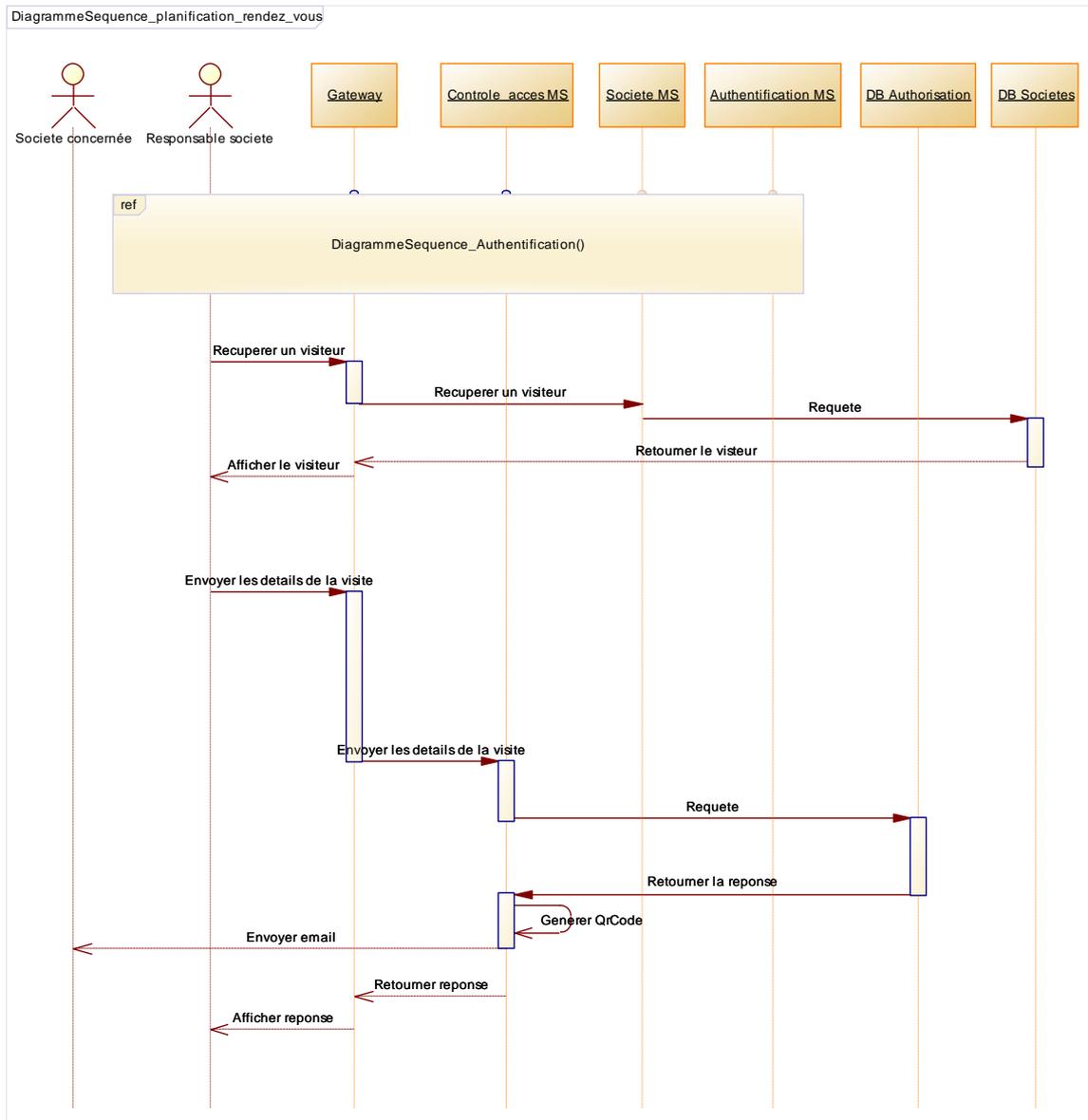


Figure 4-4 Diagramme de séquences « Planifier un rendez-vous »

### 4.2.3 Diagramme d'activité

Le diagramme d'activité représente graphiquement le comportement (séquences d'actions et conditions d'exécution) d'une méthode ou le déroulement d'un cas d'utilisation. Il est utile pour la représentation des processus métiers et les cas d'utilisation [28].

Le diagramme comprend :

- des activités : une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles.
- des transitions : qui sont automatiques entre activités, ils sont déclenchés par la fin d'une activité et provoquent le début immédiat d'une autre.

Ainsi, pour représenter le processus métier, les flux d'activités et les décisions dans le contexte des cas d'utilisation, nous allons tracer les diagramme d'activités correspondante à ces cas d'utilisation (figure 4-5 et figure 4-6) :

- diagramme d'activités « Faire une demande de rendez-vous »

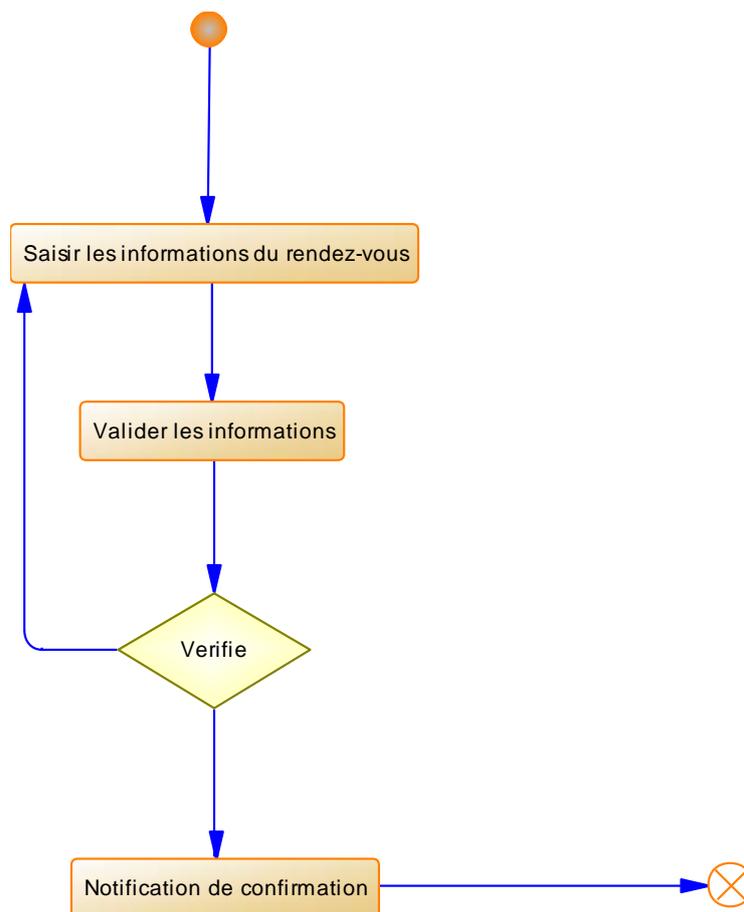


Figure 4-5 Diagramme d'activités « Faire une demande de rendez-vous »

➤ diagramme d'activités « Planifier un rendez-vous »

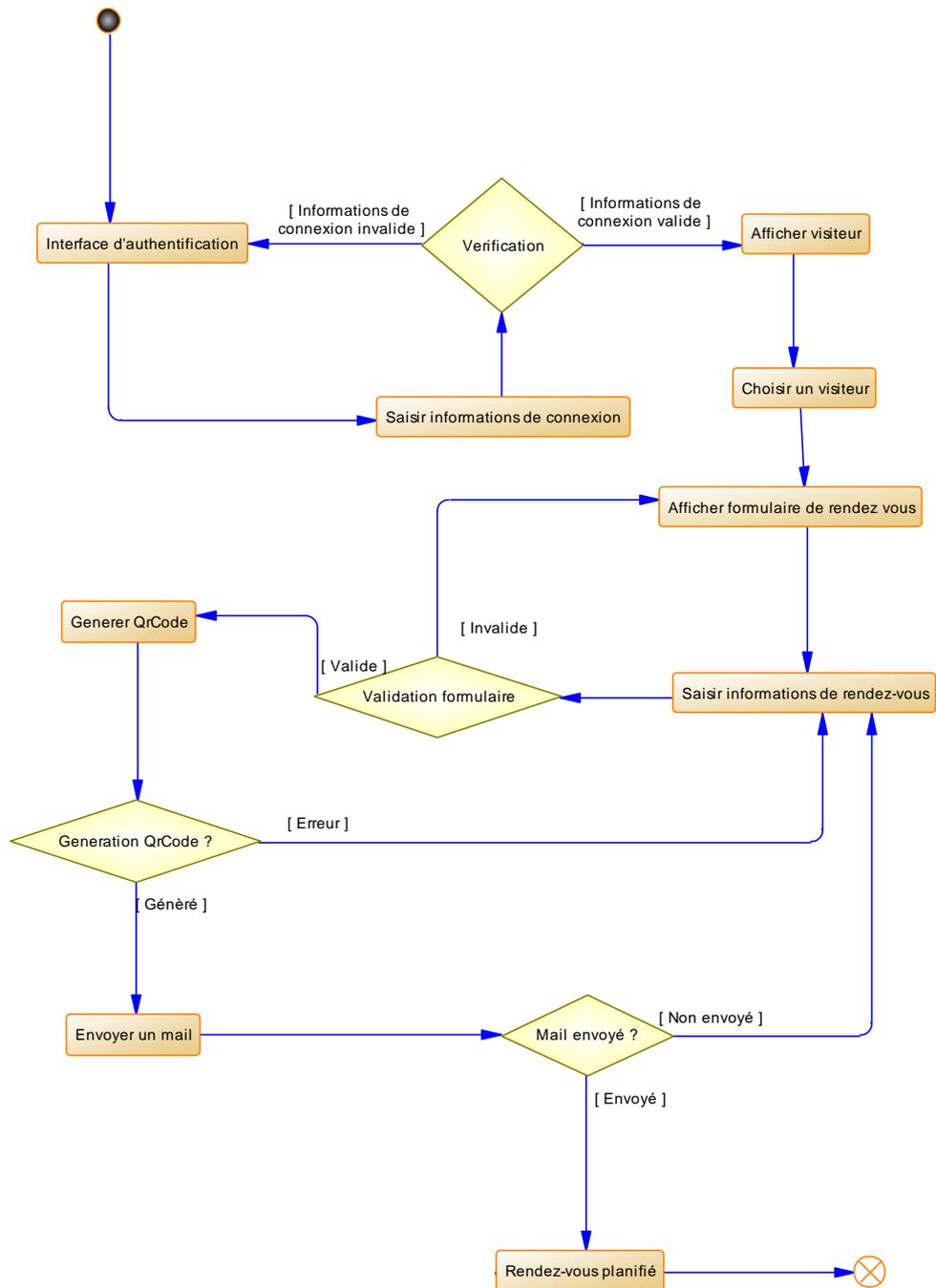


Figure 4-6 Diagramme d'activités « Planifier un rendez-vous »

## Conclusion

En résumé, cette première étape est importante pour la conception détaillée du système. Elle met en lumière les attentes spécifiques de chaque utilisateur et établit les paramètres essentiels à la réalisation du système. Ce chapitre marque une étape fondamentale vers la concrétisation de notre système, adapté aux besoins variés des utilisateurs.

# Chapitre 5 - Conception du système

## Introduction

La conception consiste à déterminer de manière détaillée et précise, ce que le système est en mesure de faire.

Dans le chapitre précédent, nous avons d'abord recensé les besoins fonctionnels et non fonctionnels pour chaque utilisateur, puis représenté ces besoins avec des diagrammes de cas d'utilisation. Ensuite, nous avons illustré les interactions entre les acteurs et le système à travers des descriptions textuelles, des diagrammes de séquence et d'activités.

Dans ce chapitre, nous élaborerons la conception générale du système en définissant son architecture logicielle, suivi du diagramme de composants, du diagramme de paquetage, et enfin du diagramme de déploiement. En poursuivant avec la conception détaillée, nous explorerons le dictionnaire de données et le diagramme de classes.

## 5.1 Conception générale

### 5.1.1 Architecture système

La figure 5-1 illustre l'architecture système sur laquelle repose la plateforme « AccessLog ». Cette architecture est composée d'une application web et d'une application mobile partageant les mêmes données. Pour ce faire, il faut que ces applications se connectent à une interface (API) pour accéder à celles-ci. A cet égard, l'utilisation d'une architecture microservices s'avère prédominante pour notre cas. Elle fournit sous forme de service les fonctionnalités qui ont trait à la logique métier et au système d'information. Ainsi, les applications clientes n'auront qu'à demander des services.

L'interface de programmation d'application (API) mise en place pour la communication entre différentes applications est ouverte aux sociétés qui souhaitent développer leurs propres applications de planification de rendez-vous. De plus, elle offre la possibilité d'intégrer l'API directement dans leurs applications existantes. Cette approche favorise la flexibilité et la personnalisation, permettant aux sociétés internes de tirer pleinement parti des fonctionnalités de planification offertes par l'API tout en adaptant ces capacités à leurs besoins spécifiques.

Le figure 5-1 montre l'architecture de notre système.

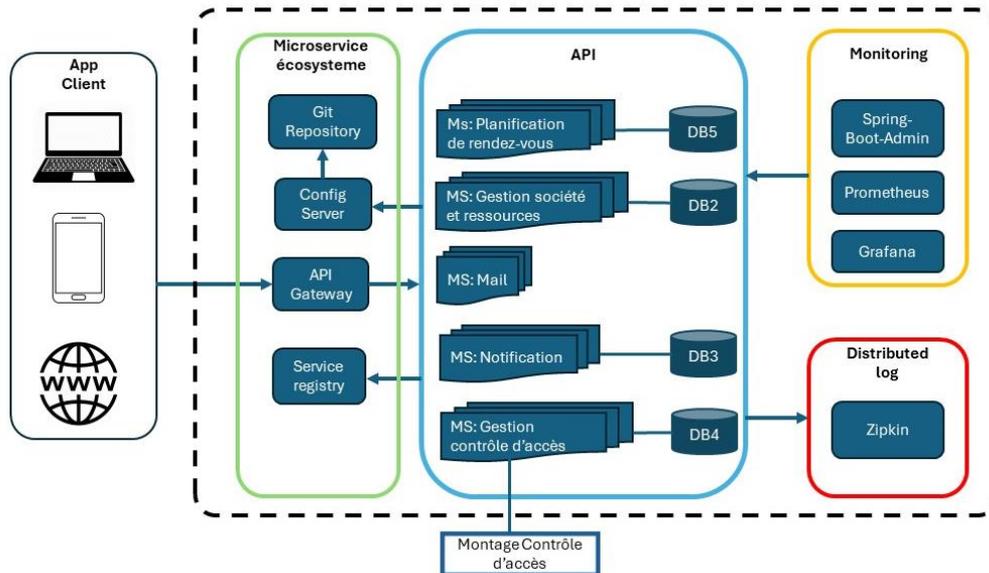


Figure 5-1 Architecture du système

### 5.1.2 Diagramme de composants

Le diagramme de composants décrit les composants et leurs dépendances dans l'environnement de réalisation. Il sont des vues statiques de l'implémentation des systèmes qui montrent les choix de réalisation. Un composant est un élément physique qui représente une partie implémentée d'un système. Il peut être du code, un script, un fichier de commande, des fichiers de données, une table etc... [9].

La figure 5-2 montre le diagramme de composants du système.

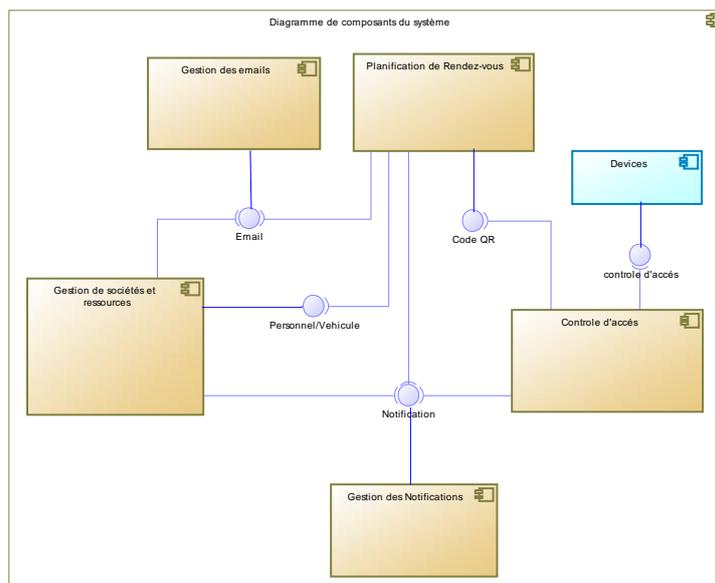


Figure 5-2 Diagramme de composants

### 5.1.3 Diagramme de paquetage

Le diagramme de paquetage est un type de diagramme UML qui permet de représenter l'organisation logique d'un système en termes de paquets (packages) et des dépendances entre eux. Les paquets sont des regroupements logiques d'éléments tels que des classes, des composants, des sous-systèmes, des packages, etc... [17]. Il est utile pour montrer la structure modulaire d'un système et comment ses différents éléments sont organisés en groupes cohérents.

La figure 5-3 montre le diagramme de package du microservices « Planification de rendez-vous ».

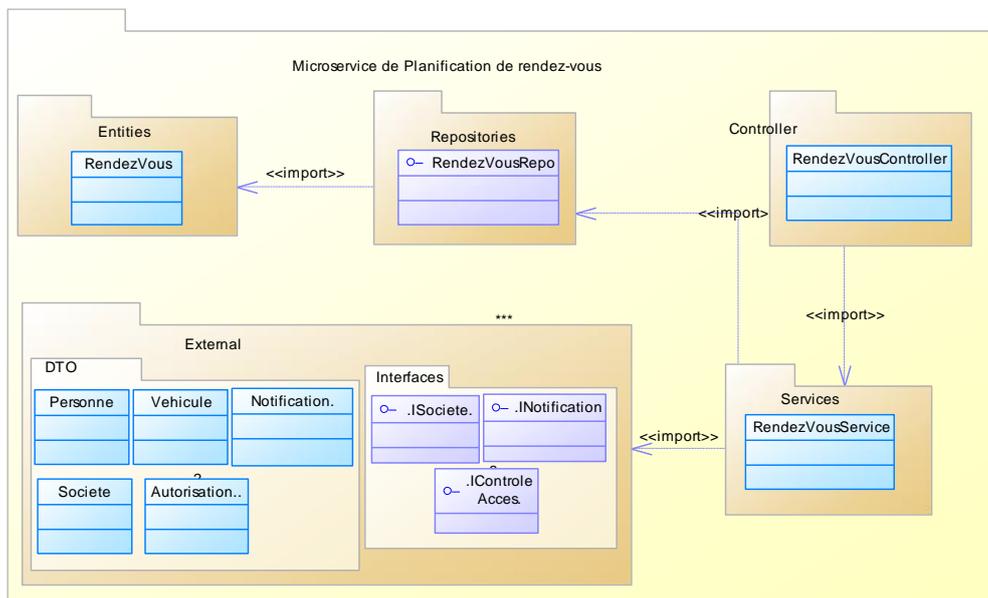


Figure 5-3 Diagramme de package du microservice « Panification de rendez-vous »

### 5.1.4 Diagramme de déploiement

Le diagramme de déploiement montre la disposition physique des éléments matériels (matériel informatique, serveurs, nœuds) dans un système, ainsi que les connexions entre eux. Il représente l'agencement physique d'un système montrant sur quels composants matériels les différents composants logiciels s'exécutent [29].

La figure 5-4 montre le diagramme de déploiement de notre système.

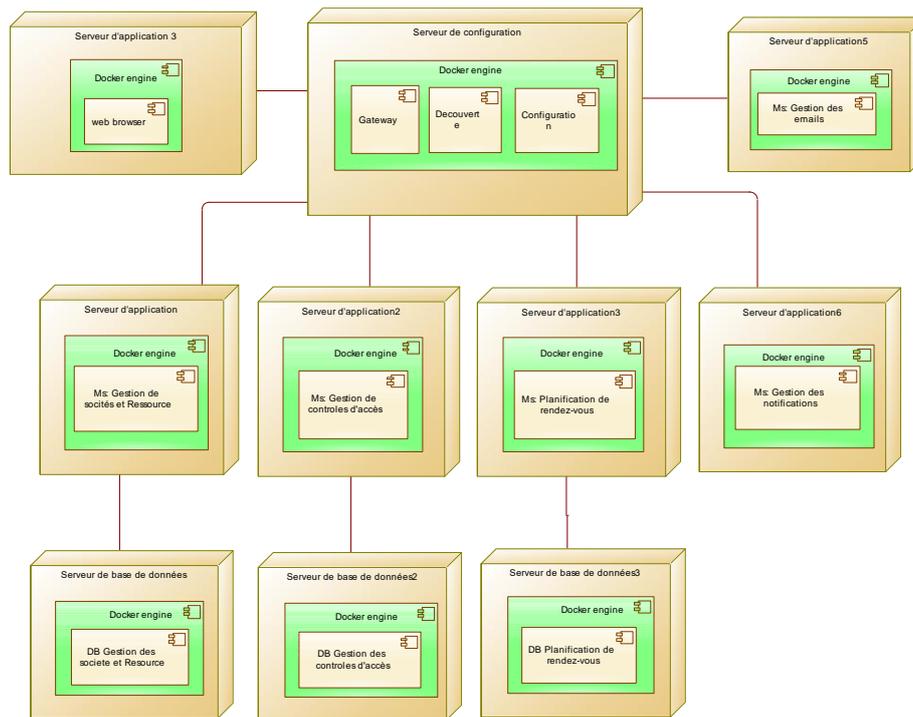


Figure 5-4 Diagramme de déploiement

## 5.2 Conception détaillée

Dans cette section, nous aborderons à travers des outils tels que le dictionnaire de données (jouant un rôle important en définissant les éléments de données), et le diagramme de classes (permettant de visualiser la structures des classe et leurs relations) la conception détaillée du système.

### 5.2.1 Dictionnaire de données

Un dictionnaire de données est une documentation qui définit et décrit les données utilisées dans un système d'information. Il s'agit d'un élément important dans le domaine de la gestion des données et de la conception de bases de données. Le dictionnaire de données contient des informations détaillées sur chaque élément de données (voir tableau 5-1).

Le tableau 5-1 représente le dictionnaire de données de notre système.

Attributs	Description	Type
<b>Société</b>		
id	Identifiant de la société	Long
nom	Nom de la société	String
rc	Registre de commerce de la société	String

ninea	Ninea de la société	String
email	Adresse email de la société	String
téléphone	Numéro de téléphone de la société	String
adresse	Adresse de la société	String
type	Type de la société (interne ou externe)	String
activo	Etat de la société(Actif ou Inactif)	Boolean
<b>Personne</b>		
id	Identifiant de la personne	Long
nom	Nom de la personne	String
prénom	Prénom de la personne	String
email	L'adresse email de la personne	String
telephone	Numéro de téléphone de la personne	String
activo	Etat de la personne (Actif ou Inactif)	Boolean
<b>Utilisateur</b>		
id	Identifiant de l'utilisateur	Long
email	Adresse email de l'utilisateur	String
password	Mot de passe de l'utilisateur	String
activo	Etat de l'utilisateur(Actif ou Inactif)	Boolean
<b>Accès</b>		
id	Identifiant de l'accès	Long
nom	Nom de l'accès	String
ipAdresse	L'adresse IP de l'accès	String
activo	Etat de l'accès(Actif ou Inactif)	Boolean
<b>Camion</b>		
id	Identifiant du véhicule	Long
immatriculation	Numéro d'immatriculation du camion	String
catégorie	Catégorie du camion	String
position	Position actuelle du camion(in ou out)	Boolean

activo	Etat du camion(Actif ou Inactif)	Boolean
<b>Autorisation</b>		
id	Identifiant de l'autorisation	Long
type	Type d'autorisation (Entrée ou sortie)	String
<b>Historique</b>		
id	Identifiant de l'historique	Long
date	Date de l'activité (Entrée ou Sortie)	Date Time
type	Type de l'activité(Entrée ou Sortie)	String
<b>Espace</b>		
id	Identifiant de l'espace	Long
capacité	Capacité d'accueil de camions alloué à une société locataire	Integer
occupé	Espace occupé de camions d'une société locataire	Integer
<b>Rendez-vous</b>		
id	UUID du rendez-vous	String
motif	Motif du rendez-vous	Text
début_rdv	Date et heure de début du rendez-vous	Date Time
duration	Marge de retard ou d'avance du rendez-vous	Integer
statut	Statut du rendez-vous	String
activo	Etat du rendez-vous (Actif ou Inactif)	Boolean

*Tableau 5-1 Dictionnaire de données*

### 5.2.2 Diagramme de classes

Le diagramme de classes est au centre du processus de modélisation objet. Il décrit les types des objets qui composent le système et les différents types de relations statiques qui existent entre eux. Il représente également les propriétés et les opérations des classes, ainsi que les contraintes qui s'appliquent sur la façon dont les objets sont connectés. En somme, il s'agit du diagramme principal permettant de capturer toutes les règles qui régissent la définition et l'utilisation des objets [19] [29].

Les figures 5-5, 5-6 et 5-7 présentent les diagrammes de classes des divers microservices constituant notre système.

➤ Microservices Gestion sociétés et ressources

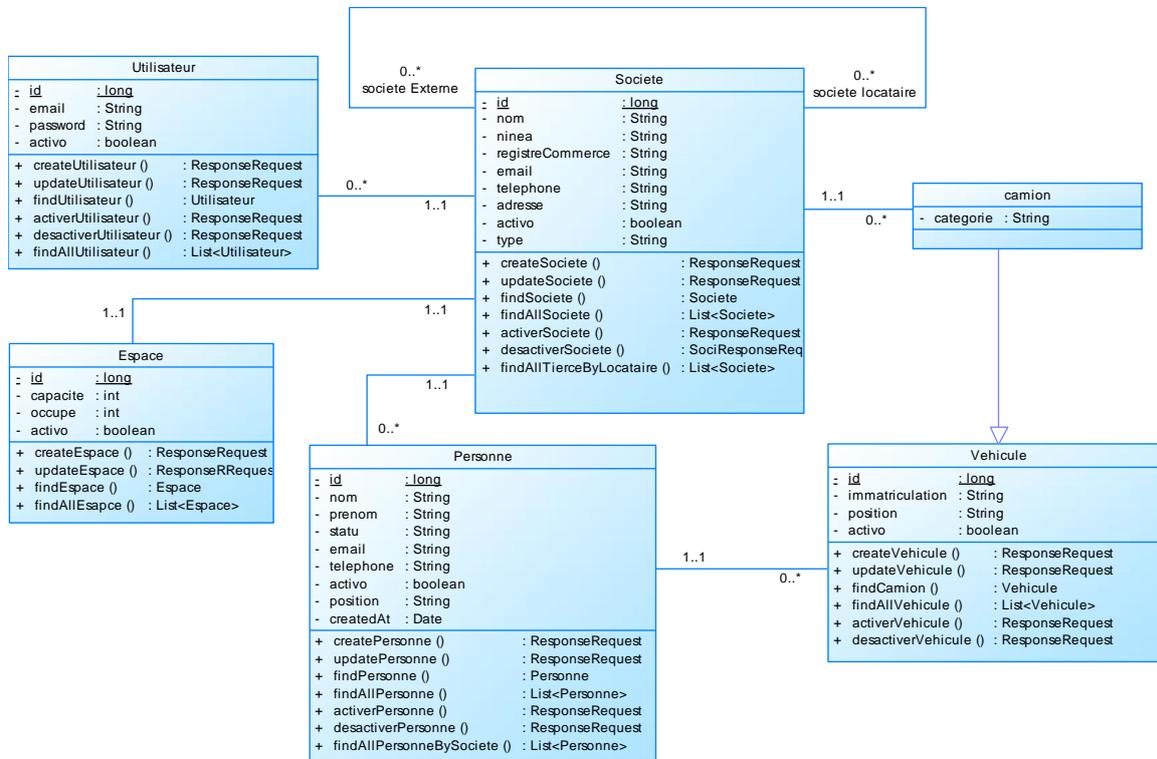


Figure 5-5 Diagramme de classes MS: Gestion sociétés et ressources

➤ Microservices Planification de rendez-vous

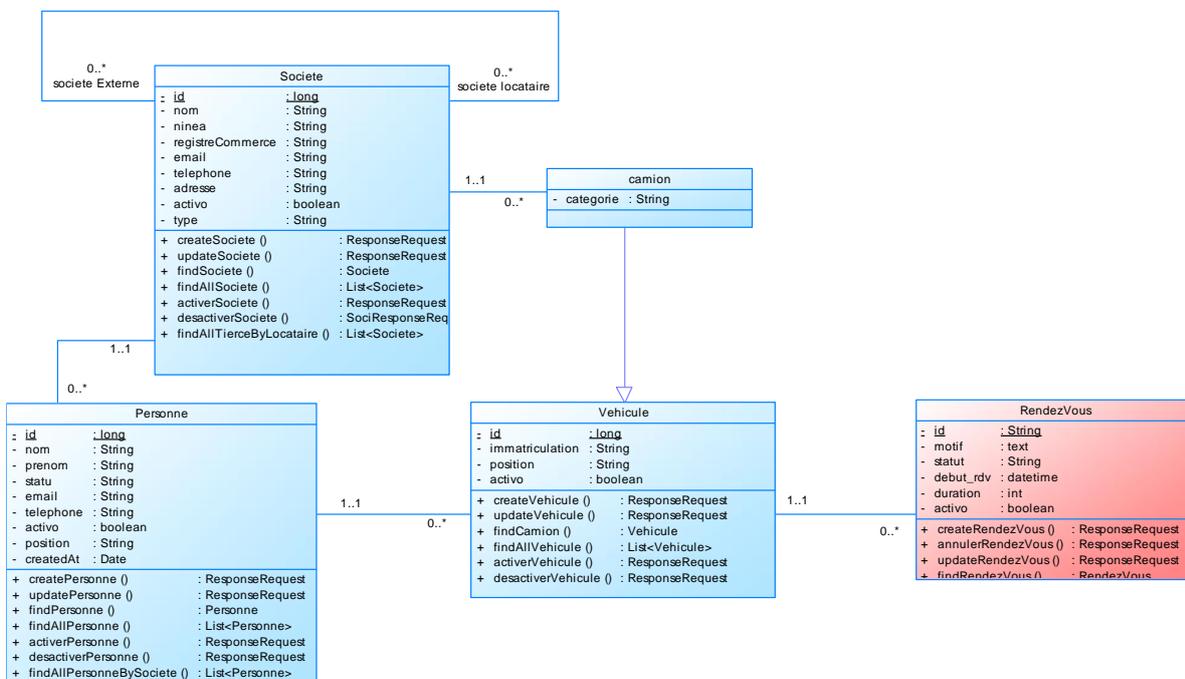


Figure 5-6 Diagramme de classes MS: Planification de rendez-vous

➤ Microservices Gestion de contrôle d'accès

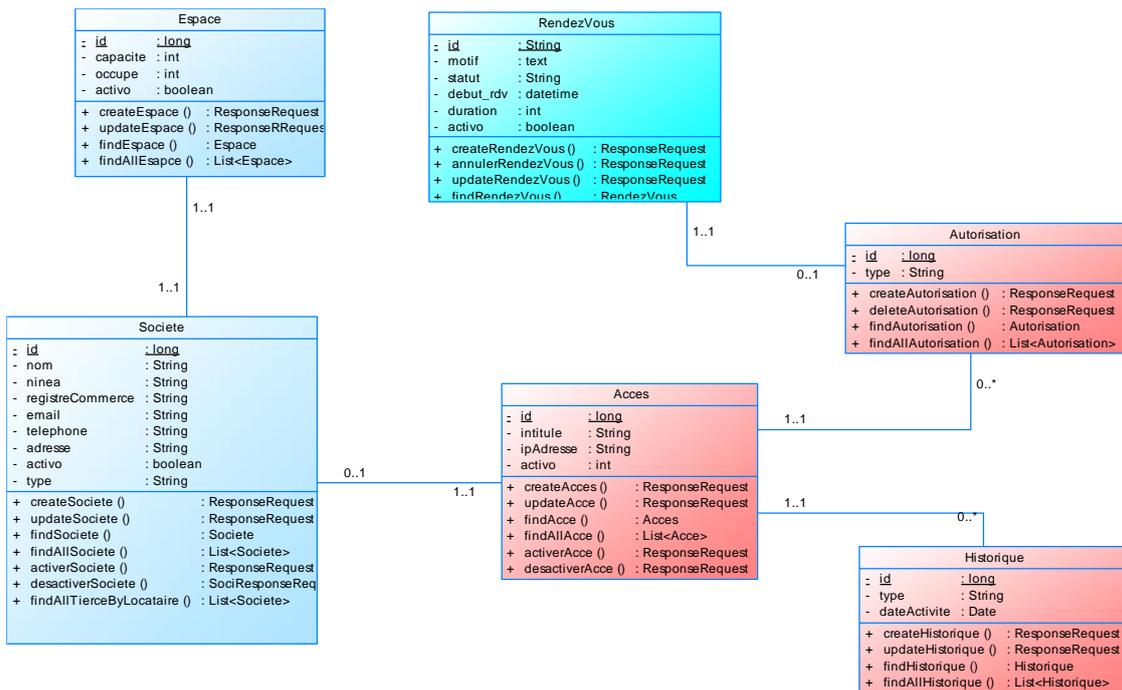


Figure 5-7 Diagrammes de classes MS: Gestion de contrôle d'accès

## Conclusion

Ce chapitre s'est principalement articulé sur la conception du système dans le processus de développement en détaillant ses capacités. Nous avons élaborer la conception générale du système, débutant par la définition de son architecture, puis en présentant les diagrammes de composants, de paquetage, et enfin de déploiement. Cette approche a offert une vue globale de la structure du système.

En approfondissant davantage avec la conception détaillée, nous avons exploré le dictionnaire de données et le diagramme de classes, apportant une clarté supplémentaire sur la manière dont les données seront gérées et la structure de classe du système. Ce processus pose des bases solides pour l'implémentation, assurant une compréhension approfondie et détaillée des fonctionnalités et des interactions du système.

# Chapitre 6 - Implémentation et présentation de « AccessLog »

## Introduction

Ce chapitre est consacré à la réalisation du système. Il comporte l'implémentation, et la présentation de quelques interfaces de « AccessLog ».

## 6.1 Implementation

Dans cette partie, nous détaillons les outils et technologies.

### 6.1.1 Technologies

Pour la réalisation de la plateforme AccessLog, nous avons choisi les technologies suivantes :

#### ➤ Angular

Angular est un framework open-source développé par Google, largement utilisé pour créer des applications web dynamiques et interactives. Basé sur TypeScript, Angular facilite la création de SPA (Single Page Applications) en fournissant une structure solide et des outils puissants pour le développement front-end.

Le tableau 6-1 résume les avantages et les limites d'Angular :

Avantages	Limites
➤ architecture robuste ;	➤ surcharge de performance ;
➤ liaison de données bidirectionnelle ;	➤ taille de l'application ;
➤ modularité et réutilisabilité ;	➤ courbe d'apprentissage ;
➤ large écosystème.	➤ mise à jour fréquente.

*Tableau 6-1 : Avantages et limites de Angular*

#### ➤ Spring boot

Spring Boot est un framework Java open-source utilisé pour créer des applications Spring autonomes et de qualité de production avec une configuration minimale. Il fait partie de l'écosystème plus large de Spring Framework, offrant une approche simplifiée pour développer et déployer des applications en fournissant des solutions prêtes à l'emploi pour des tâches courantes telles que la gestion des dépendances, la configuration et le démarrage de l'application.

Dans le tableau 6-2, nous avons quelques outils intégrés que nous avons utilisés avec Spring Boot :

Outils	Description
Spring Security	Intégré pour la gestion des aspects liés à la sécurité de l'application, y compris l'authentification et l'autorisation.
Spring Data – JPA	Employé pour la gestion simplifiée des interactions avec la base de données PostgreSQL.
Spring cloud – Gateway	Utilisé comme passerelle pour la gestion centralisée du trafic entre les clients et les microservices.
Spring cloud – Eureka	Intégré pour la découverte dynamique des services au sein de l'architecture distribuée.
Spring Cloud Discovery Client/Server	Adopté pour la découverte et l'enregistrement des microservices dans l'écosystème.

*Tableau 6-2 : Outils intégrés à Spring Boot*

➤ React native

React Native est un framework open-source développé par Facebook qui permet de créer des applications mobiles multiplateformes en utilisant la syntaxe et les concepts de développement de React, une bibliothèque JavaScript populaire pour la création d'interfaces utilisateur. Avec React Native, les développeurs peuvent utiliser des composants React pour créer des interfaces utilisateur riches et dynamiques, tout en bénéficiant de la réutilisation du code entre les plateformes iOS et Android.

➤ Expo

Expo est un ensemble d'outils et de services open-source construits autour de React Native pour faciliter le développement d'applications mobiles. Il vise à simplifier le processus de développement en offrant des fonctionnalités telles que le démarrage rapide de projets, l'accès à des API natifs, le déploiement facile d'applications et bien plus encore.

### 6.1.2 Outils

➤ PostgreSQL

PostgreSQL est un système de base de données objet-relationnel open source puissant avec plus de 35 ans de développement actif qui lui a valu une solide réputation en matière de fiabilité, de robustesse des fonctionnalités et de performances [30].

➤ Postman

Postman, en tant que plateforme API, simplifie de manière significative le cycle de vie des API en offrant des outils complets pour la création, le test, la documentation et la gestion des API. L'approche collaborative de Postman, illustrée par des fonctionnalités telles que le

référentiel API et les espaces de travail, permet de stocker, cataloguer et collaborer autour de tous les artefacts liés aux API de manière centralisée. Avec une intégration étroite dans le pipeline de développement logiciel et une gouvernance orientée vers le cycle de vie complet, Postman favorise des pratiques de développement qui conduisent à la création d'API de haute qualité.

➤ **GitHub**

GitHub est un service basé sur le cloud qui héberge un système de contrôle de version (VCS) appelé Git. Il permet aux développeurs de collaborer et d'apporter des modifications aux projets partagés tout en gardant un suivi détaillé de leur progression [31].

➤ **Expo Go**

Expo Go est une application mobile disponible sur iOS et Android qui permet de tester les applications Expo directement sur les appareils physiques, sans avoir besoin de les publier sur les stores.

➤ **Vs code**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS<sup>2</sup>. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. Le code source de Visual Studio Code provient du projet logiciel libre et open source VS Code de Microsoft publié sous la licence MIT permissive, mais les binaires compilés sont des logiciels gratuits pour toute utilisation.

➤ **IntelliJ**

IntelliJ est un environnement de développement intégré (IDE) utilisé principalement pour le développement de logiciels en Java, bien qu'il supporte également d'autres langages de programmation tels que Kotlin, Groovy, Scala et plus encore. C'est un produit développé par JetBrains. IntelliJ offre des fonctionnalités avancées telles que la saisie automatique de code, la navigation intelligente, le débogage, le profilage, la gestion de projet et une intégration avec des systèmes de contrôle de version tels que Git. C'est un outil populaire parmi les développeurs pour sa productivité accrue et son ensemble d'outils complet.

➤ **Docker**

Docker est une plateforme logicielle open source qui automatise le déploiement, la gestion et l'exécution d'applications dans des conteneurs. Les conteneurs permettent d'encapsuler une

application et ses dépendances, assurant ainsi une exécution cohérente et isolée sur n'importe quel environnement compatible avec Docker. Cette approche simplifie le déploiement des applications, améliore la portabilité et facilite la gestion des ressources.

En somme, Docker offre une solution de virtualisation légère, basée sur des conteneurs, pour faciliter le développement, le déploiement et la gestion d'applications.

➤ Swagger

Swagger est un ensemble d'outils open source permettant de concevoir, créer, documenter et consommer des services web RESTful. Il est principalement utilisé pour décrire l'interface d'API REST de manière standardisée. Swagger utilise une spécification appelée Spécification d'OpenAPI(OAS) pour définir la structure et la documentation d'une API.

En résumé, Swagger facilite le développement et la gestion des API en fournissant des outils cohérents pour la conception et la documentation, ce qui favorise la collaboration entre les développeurs, les testeurs et les équipes chargées de la gestion des API.

➤ JWT

JSON Web Token (JWT) est une norme ouverte (RFC 7519) qui définit un moyen compact et autonome pour transmettre en toute sécurité des informations entre les parties en tant qu'objet JSON. Ces informations sont fiables et peuvent être vérifiées car elles sont signées numériquement à l'aide d'une clé secrète (avec l'algorithme HMAC) ou d'une paire de clés publique/privée à l'aide de RSA ou ECDSA.

Bien que les JWT puissent être chiffrés pour assurer également la confidentialité entre les parties, nous nous concentrerons sur les jetons signés. Les jetons signés peuvent vérifier l'intégrité des revendications qu'ils contiennent, tandis que les jetons chiffrés cachent ces revendications à d'autres parties.

Notre utilisation de ce service va à l'endroit des autorisations accordées aux utilisateurs de l'application. Une fois que l'utilisateur est connecté, chaque demande suivante inclura le Token, permettant à l'utilisateur d'accéder aux routes, services et ressources qui sont autorisés avec ce jeton. L'authentification unique est une fonctionnalité que utilise largement JWT de nos jours, en raison de sa faible surcharge et de sa capacité à être facilement utilisée dans différents domaines.

Voici un exemple de représentation du Token généré suite à une authentification réussie (figure 6-1) :



C'est un projet communautaire qui fournit une application web pour gérer et surveiller les applications spring boot. Il offre des fonctionnalités telles que la visualisation des détails de l'application, les vérifications d'état, les métriques, les journaux, et plus encore. Il est utile pour surveiller et gérer plusieurs applications spring boot de manière centralisée.

## 6.2 Présentation du système

Notre solution se compose d'une application web, d'une application mobile, d'une API ainsi que d'un système de contrôle d'accès accompagné d'une simulation pour la gestion des accès aux zones et à l'entrée principale de la plateforme de distribution.

Dans cette partie nous présenterons quelques interfaces et captures des différents composant de notre solution.

### 6.2.1 API

Cette documentation offre aux sociétés locataires la flexibilité d'implémenter notre solution de planification de rendez-vous selon leurs exigences. En suivant les indications détaillées fournies, ces sociétés peuvent intégrer notre système dans leur infrastructure existante, adaptant ainsi la solution à leurs processus opérationnels uniques.

L'approche modulaire de notre solution permet une personnalisation, permettant aux sociétés de tirer parti des fonctionnalités tout en respectant leurs propres préférences et contraintes.

Les figures suivantes (figure 6-2 et figure 6-3) présentent la documentation de l'API.

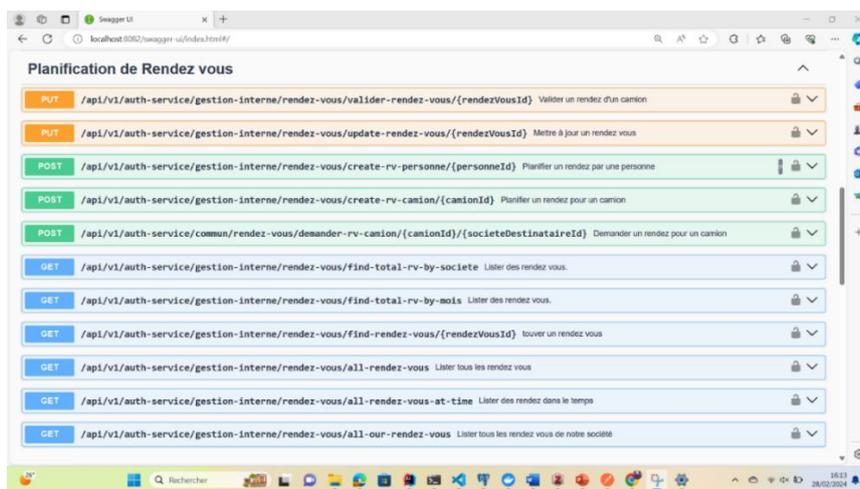


Figure 6-2 Documentation de l'API 1/2

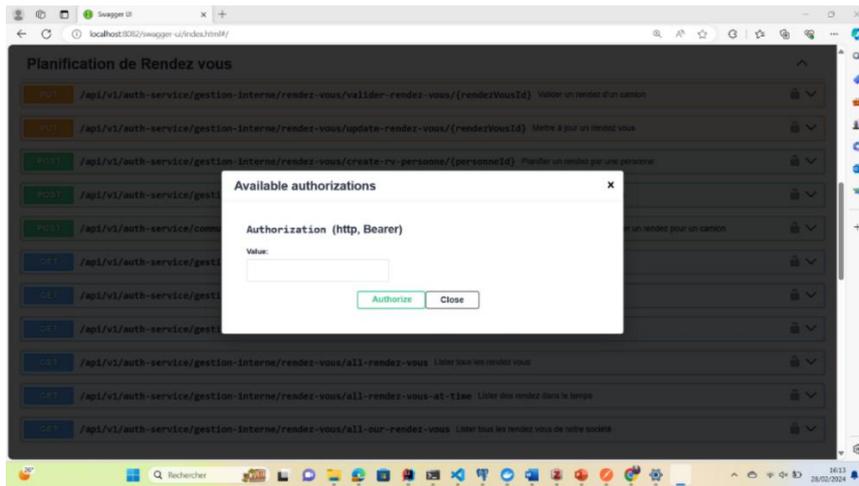


Figure 6-3 Documentation de l'API 2/2

## 6.2.2 Interfaces de l'application web

Les interfaces suivantes montrent les fonctionnalités implémentés dans l'application web permettant aux sociétés de planifier des rendez-vous.

Le responsable d'une société peut consulter toutes les statistiques de sa société, y compris le nombre d'employés, ses partenariats, l'espace qu'elle occupe, ses rendez-vous, ainsi que le nombre de camions qu'elle possède par catégorie (figure 6-4).

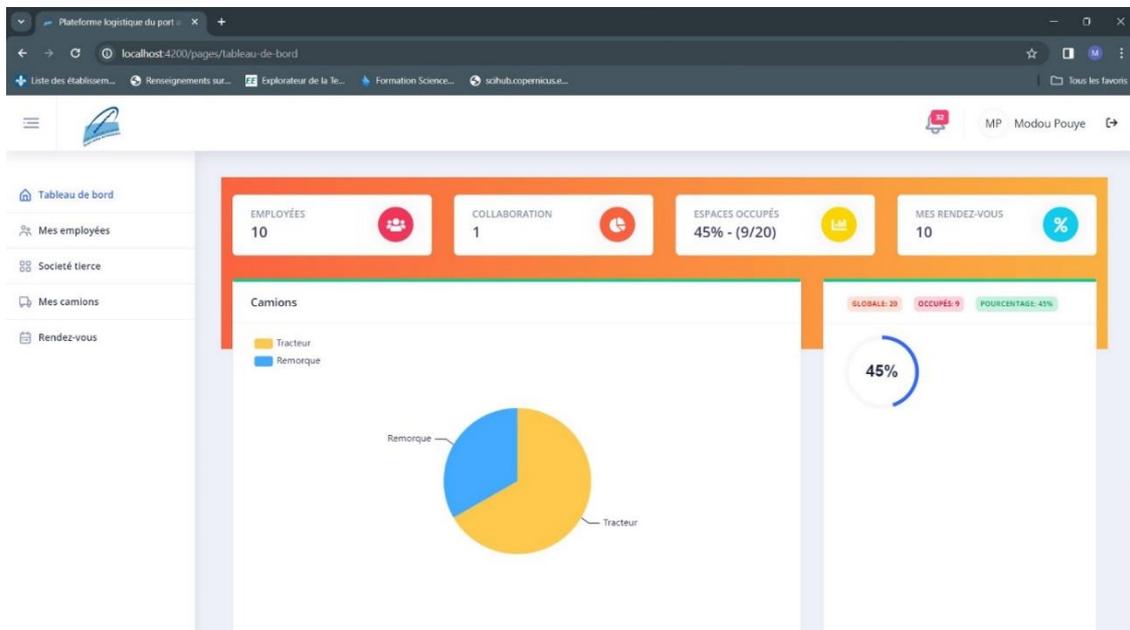


Figure 6-4 Tableau de bord du responsable de société

Le responsable d'une société a la possibilité de planifier ou de valider un rendez-vous pour une société (voir figure 6-5). Pour cela, il renseigne une marge de retard et peut modifier s'il s'agit d'une validation la date du rendez-vous en fonction des activités en cours. Une fois

la demande validée, un email de confirmation contenant les autorisations d'accès sera envoyé à la société à l'origine de la demande (figure 6-6).

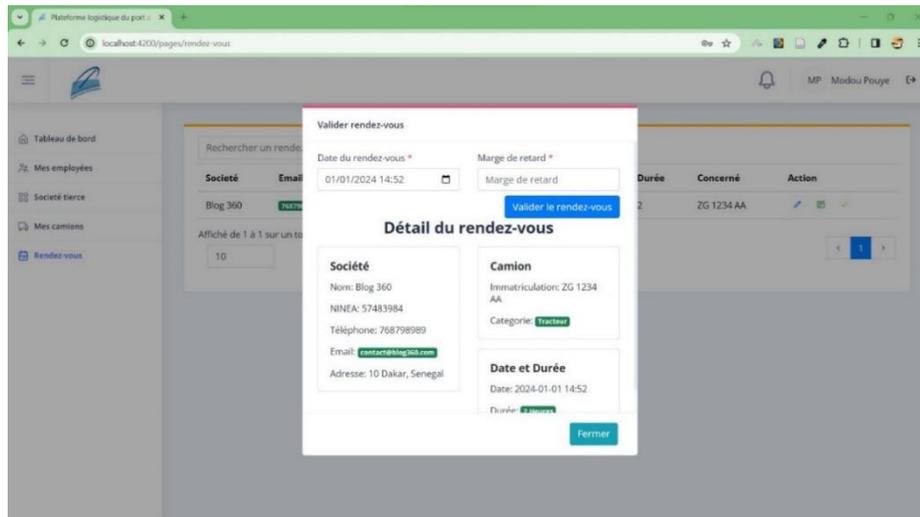


Figure 6-5 Validation du rendez-vous

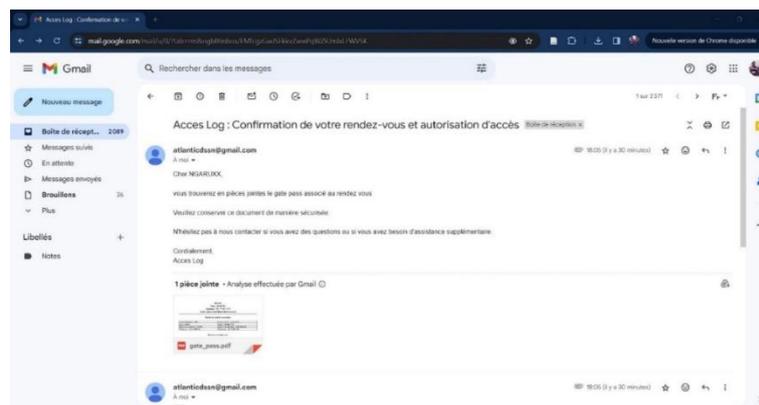


Figure 6-6 Email de confirmation de la demande

### 6.2.3 Interfaces de l'application mobile

Les quelques interfaces (figure 6-7) présentent les fonctionnalités disponibles après une authentification réussie dans notre application. Un responsable de société peut organiser des rendez-vous pour des sociétés externes à la plateforme logistique ou solliciter un rendez-vous pour accéder à l'espace d'une société locataire. Il peut valider ou modifier ou relancer (en cas d'expiration) une demande rendez-vous. Il a la possibilité d'ajouter ses propres camions ou ceux de ses partenaires, de faire des demandes de badges pour ses employés et ses véhicules.

De plus, il peut surveiller sa capacité disponible avant de planifier des opérations, tout en visualisant le nombre de rendez-vous effectués par les différentes sociétés à travers des

graphiques. Il peut également visualiser ces informations personnelles ainsi que celle de la société qu'il administre.

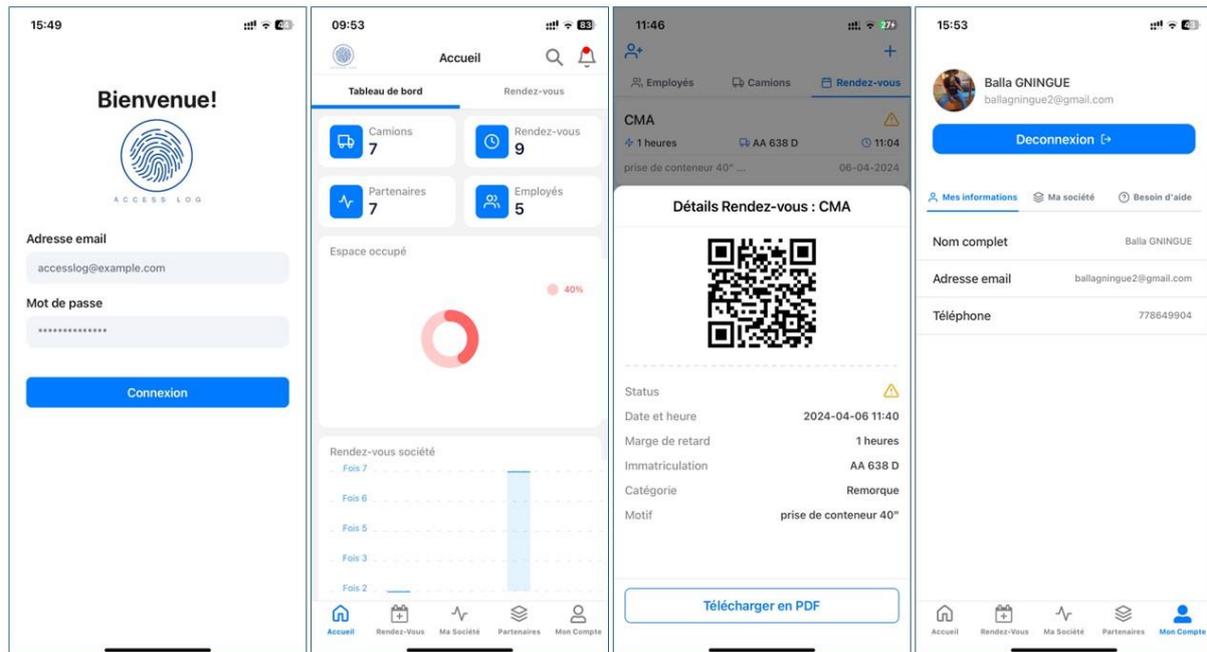
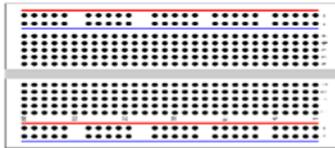


Figure 6-7 Quelques interfaces de l'application mobile

#### 6.2.4 Simulateur de contrôle d'accès

Un dispositif de contrôle d'accès a été mis en place à l'aide de composants Arduino afin de simuler le contrôle d'accès à la porte principale (figure-9). Parallèlement, un simulateur a été développé en vue de reproduire le contrôle d'accès dans les différentes zones situées au niveau de la plateforme de distribution (figure-10). Les outils et technologies utilisés à cette fin sont présentés dans le tableau 6-3 :

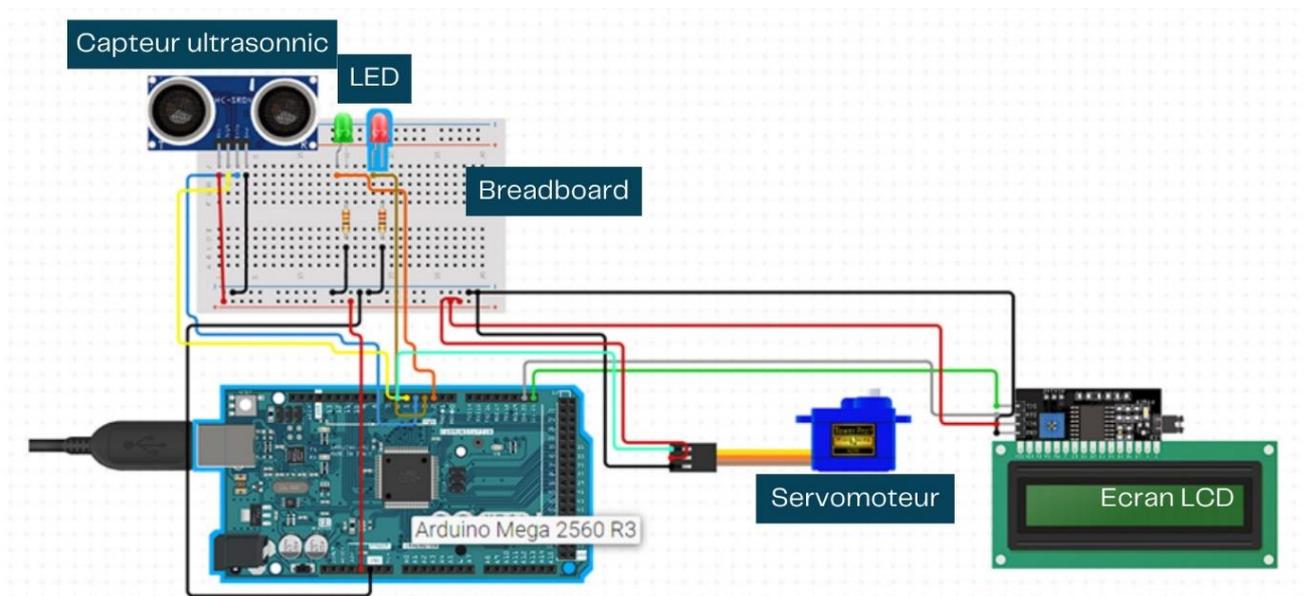
Nom	Description	Image
Arduino IDE	C'est un environnement de développement intégré (IDE) spécialement conçu pour la programmation des microcontrôleurs Arduino. Il facilite la rédaction, la compilation et le téléversement du code.	

Circuit.io	C'est une plateforme en ligne pour la conception et la simulation de circuits électroniques. Permettant de modéliser visuellement les connexions entre les composants avant leur implémentation physique. Elle est utile pour planifier et tester le montage.	
Langage C	C'est un langage de programmation utilisé dans le développement de logiciels pour Arduino. Le code est interprété et exécuté par le microcontrôleur Arduino pour contrôler le fonctionnement des montages électroniques.	
Arduino Mega 2560	Il gère le processus global du montage.	
Mini Breadboard	Il facilite les connexions entre les composants.	
Led verte  Led rouge	Indicateur visuel l'autorisation d'accès.  Indicateur visuel du refus d'accès.	
Ecran LCD	Il donne des indications visuelles, comme l'état de la barrière.	

Servomoteur	Il soulève et abaisse la barrière en fonction de l'autorisation d'accès.	
Capteur ultrasonnic	Il détecte la présence d'objets.	

*Tableau 6-3 Outils et technologies utilisés pour le système de contrôle d'accès*

La figure 6-8 montre notre montage du contrôle d'accès.



*Figure 6-8 Montage du contrôle d'accès*

La figure 6-9 montre le système de contrôle d'accès réalisé avec des composants Arduino.

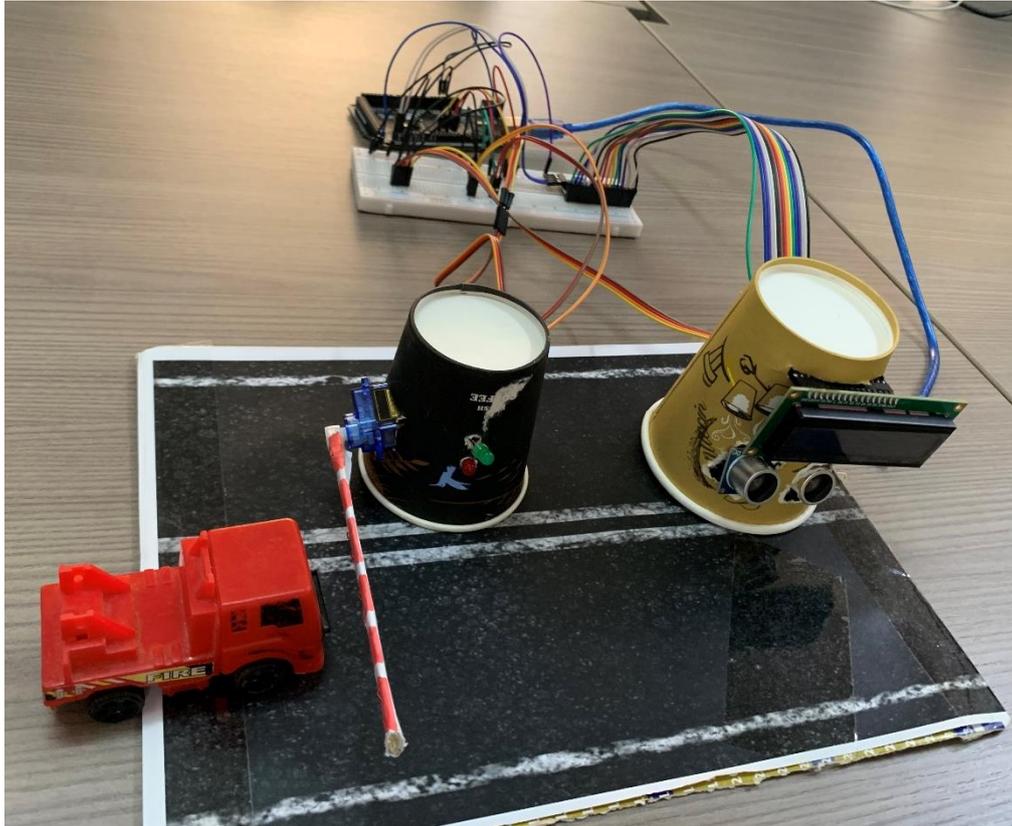


Figure 6-9 Système de contrôle d'accès

Chaque société dispose d'un espace qui lui est attribué dès sa création, et ce quota ne peut être dépassé. Lorsqu'un rendez-vous est planifié, un camion se voit accorder l'autorisation d'accéder à l'espace réservé à la société associée, sans toutefois lui permettre d'accéder à l'espace d'une autre société (voir figure 6-10).

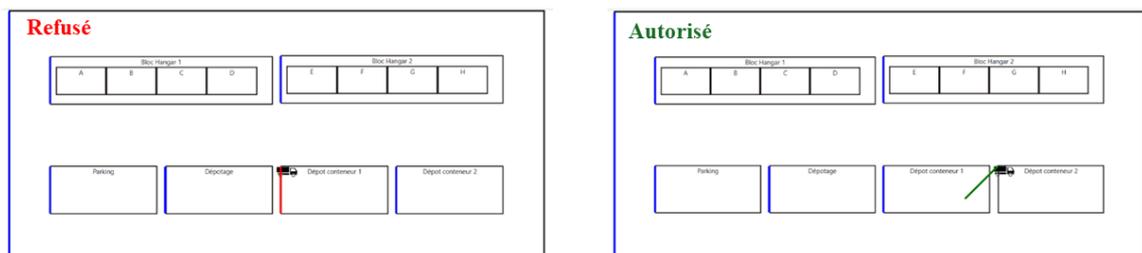


Figure 6-10 Simulateur de contrôle d'accès

## Conclusion

Dans tout projet informatique, il est important de définir une architecture solide avec un choix réfléchi et justifié des technologies utilisées dans son système. Dans ce chapitre, nous avons présenté l'architecture système de « AccessLog » ainsi que les outils et technologies utilisés pour la mise en œuvre avant d'illustrer quelques interfaces des applications et la documentation de l'API.

## Conclusion generale

Ce mémoire s'inscrit dans une démarche visant à améliorer les opérations logistiques au sein de la plateforme de distribution du PAD. Mettant l'accent sur la planification des rendez-vous et la sécurité des opérations, nous avons identifié ces aspects comme étant essentiels pour assurer la fluidité et l'efficacité des activités de distribution. Une attention est accordée aux sociétés externes qui jouent un rôle important dans cet écosystème dynamique.

La contribution principale réside dans la mise en place d'une application web et mobile pour la planification des rendez-vous, accompagnée d'un système de contrôle d'accès basé sur les codes QR. L'objectif est de coordonner les opérations logistiques, tout en garantissant un accès sécurisé et autorisé pour toutes les sociétés interagissant avec la plateforme de distribution.

L'introduction d'une API constitue une autre composante de notre approche, facilitant l'intégration fluide de nos fonctionnalités dans les infrastructures existantes des sociétés ne souhaitant pas utiliser nos applications. Cette ouverture à la collaboration et à l'interopérabilité permet répondre aux divers besoins des sociétés.

En anticipant l'avenir, l'intégration de technologies telles que le protocole InterPlanetary File System (IPFS) et la blockchain est envisagé. Aussi en utilisant de vrais barrières et tourniquets pour le contrôle d'accès. Cette perspective offre des avantages en termes de sécurité, de transparence et de résilience des données, ouvrant ainsi de nouvelles opportunités pour l'évolution de notre solution.

# Annexe

➤ Annexe 1: Analyse et spécification des besoins

- description textuelle du cas d'utilisation « Faire une demande de rendez-vous »

Nom	Faire une demande de rendez-vous
Acteurs	Responsable société
Résumé	Faire une demande de rendez-vous
Précondition	Les responsable de la société est authentifié
Scenario nominal	<ul style="list-style-type: none"> <li>➤ il remplit les informations nécessaires pour la demande ;</li> <li>➤ il valide la demande de rendez-vous ;</li> <li>➤ le système enregistre la demande de rendez-vous ;</li> <li>➤ une notification de confirmation est envoyée au responsable.</li> </ul>
Postcondition	La demande de rendez-vous est enregistrée et a été transmise avec succès à la société concernée.
Exception	Si les informations fournies sont incomplète ou incorrects, le système génère une erreur et guide l'utilisateur pour corriger les informations.

Tableau 7-0-1 Description textuelle du cas d'utilisation « Faire une demande de rendez-vous

»

- diagramme de séquences « Faire une demande de rendez-vous »

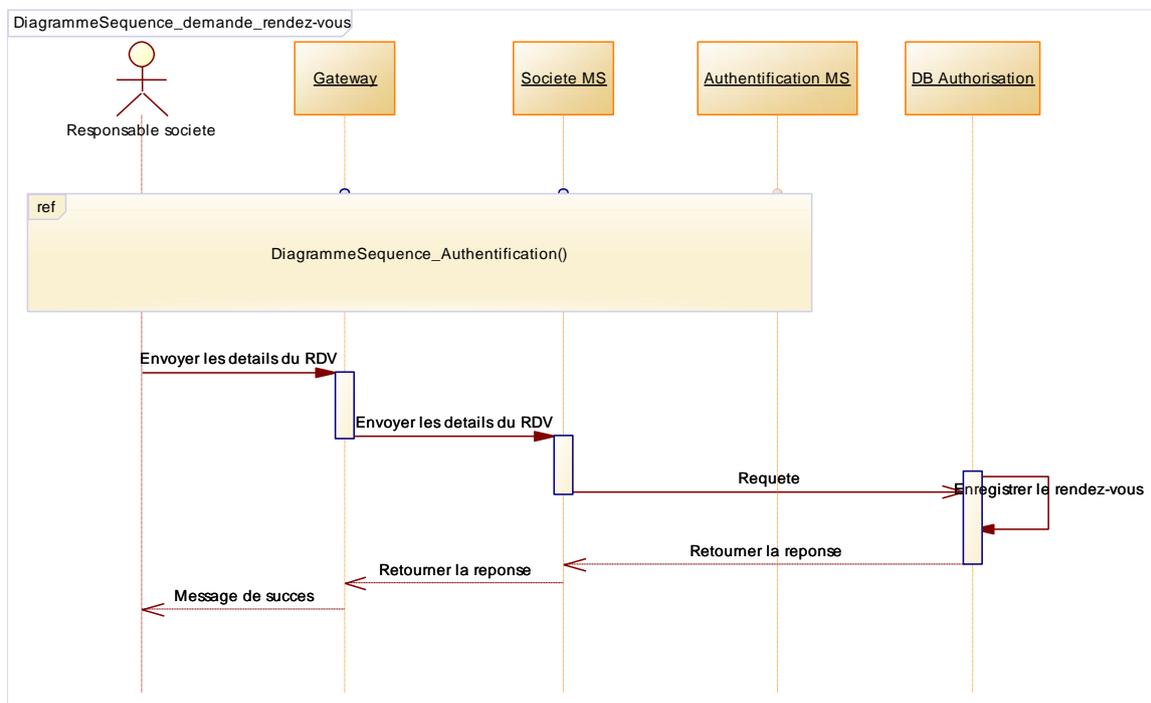


Figure 7-1 Diagramme de séquences « Faire une demande de rendez-vous »

➤ Annexe 2 : Conception du système

- diagramme de package du microservice « Gestion du contrôle d'accès »

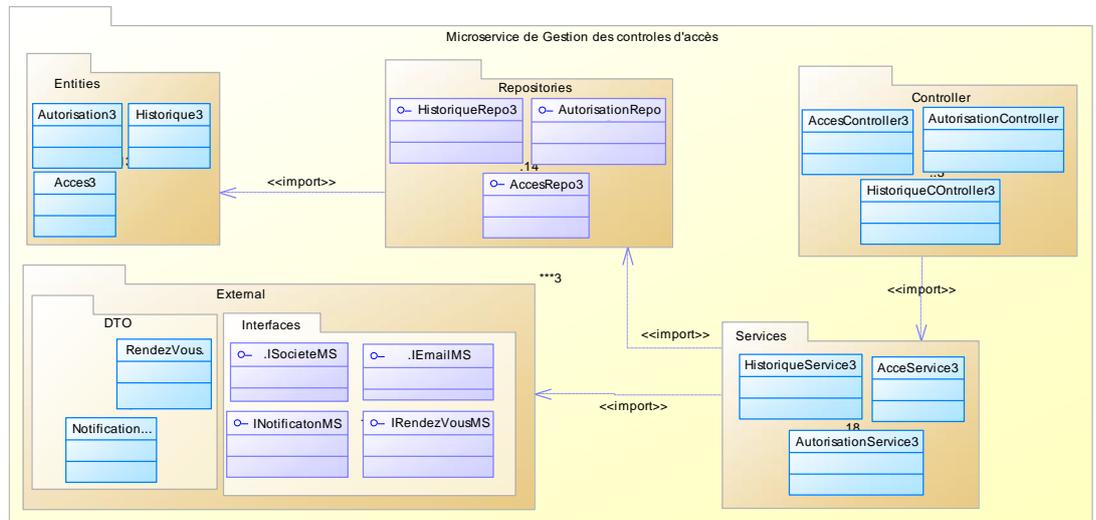


Figure 7-2 Diagramme de package microservice « Gestion du contrôle d'accès

»

- diagramme de package du microservice « Gestion des ressources »

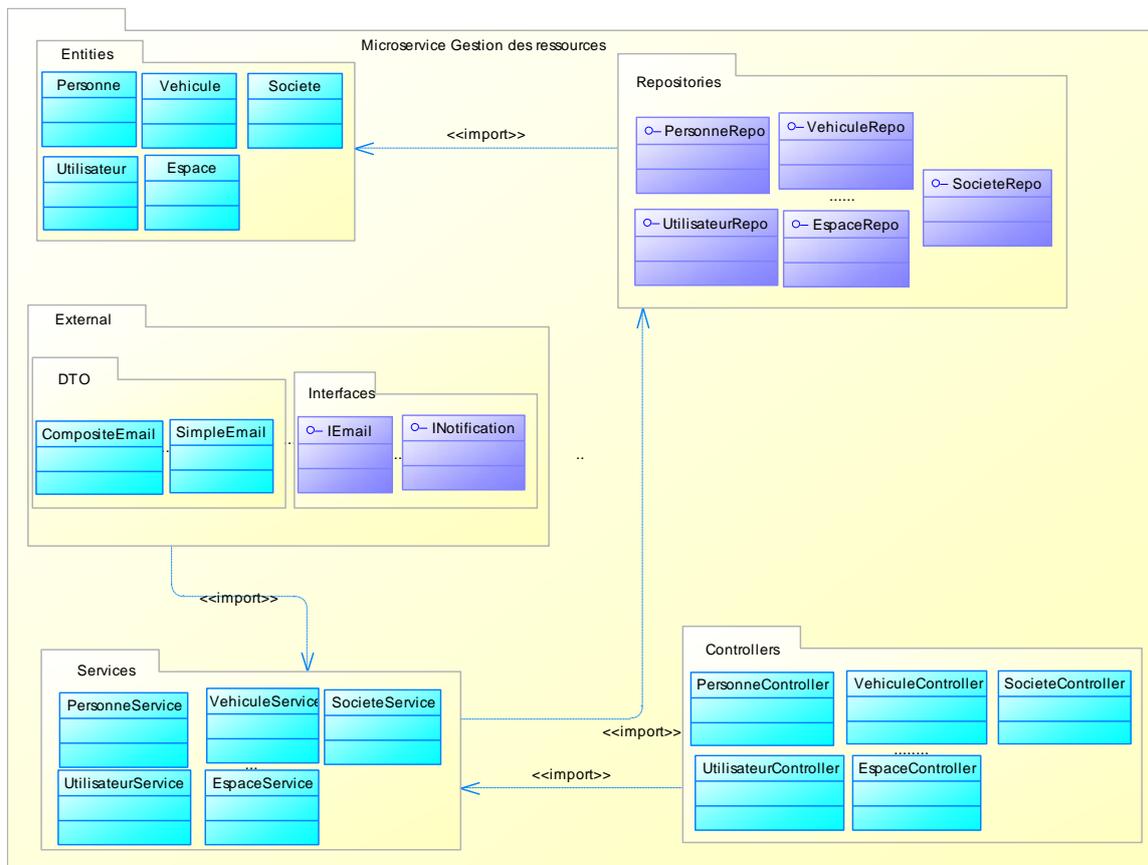


Figure 7-3 Diagramme de package microservice « Gestion des ressources »

➤ Annexe 3 : Présentation du système « AccessLog »

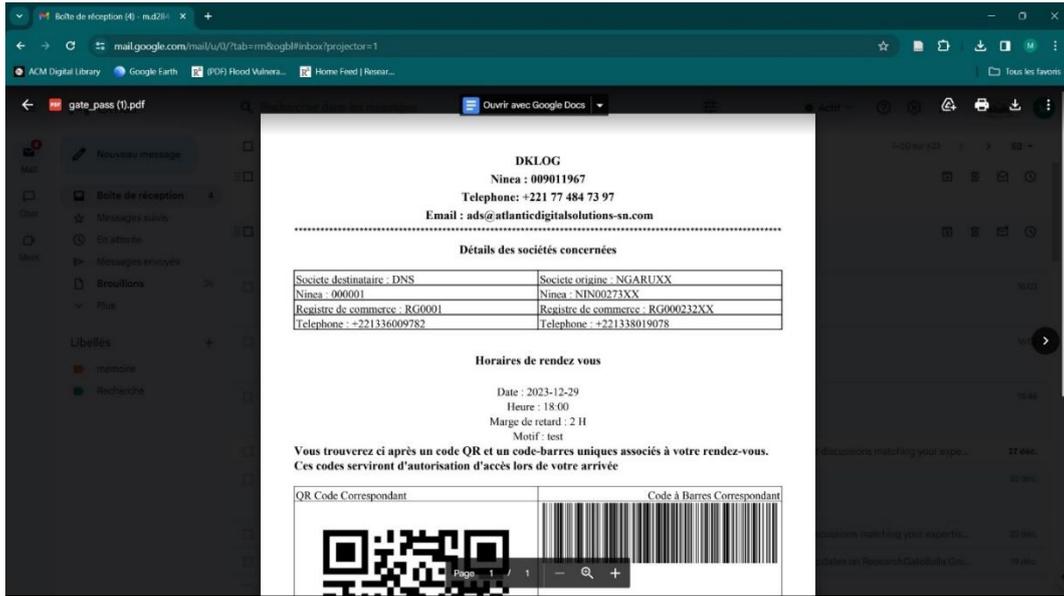


Figure 7-4 Gate pass

➤ Annexe 4: Outils de surveillances et de visualisations

- Prometheus

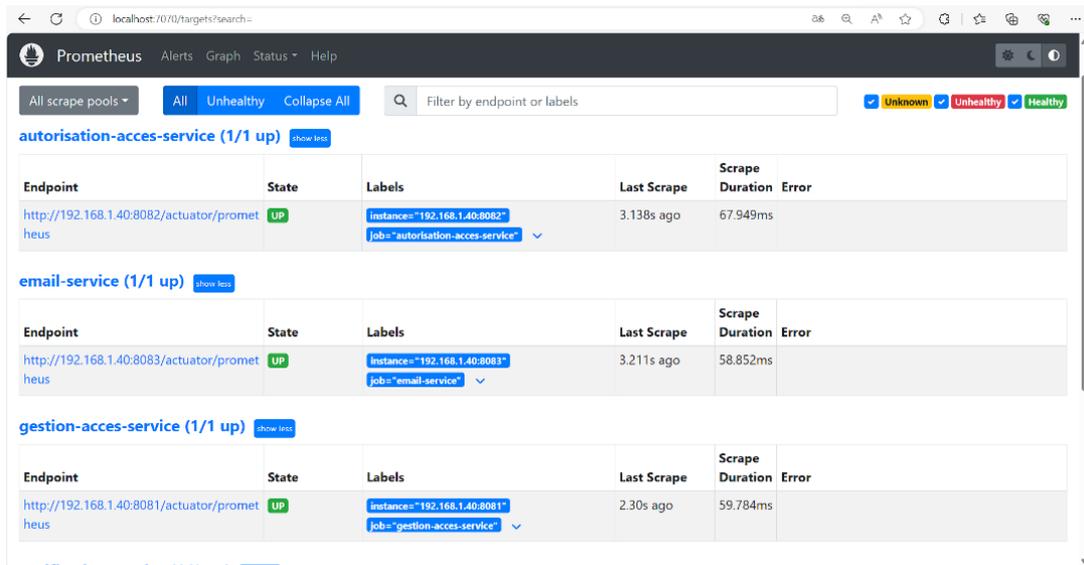


Figure 7-5 Prometheus

- Zipkin

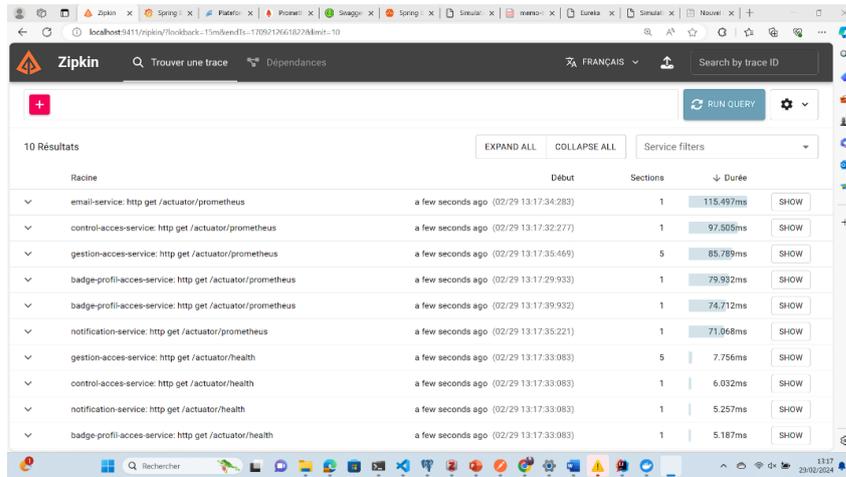


Figure 7-6 Zipkin 1/3

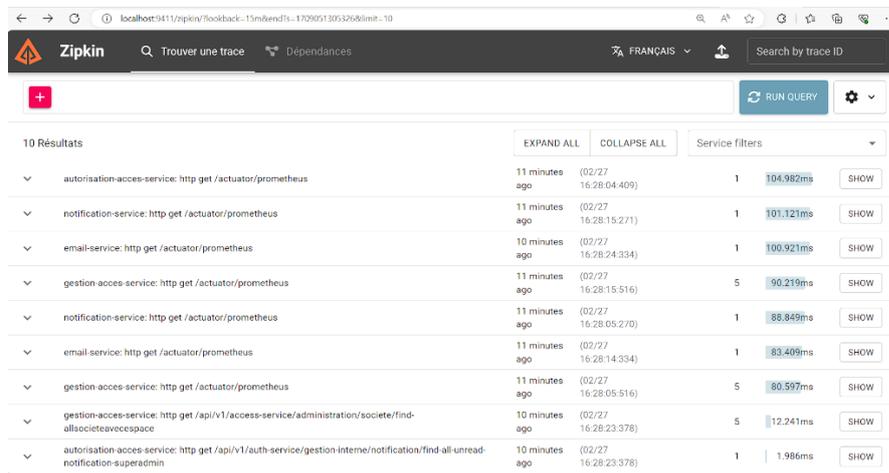


Figure 7-7 Zipkin 2/3

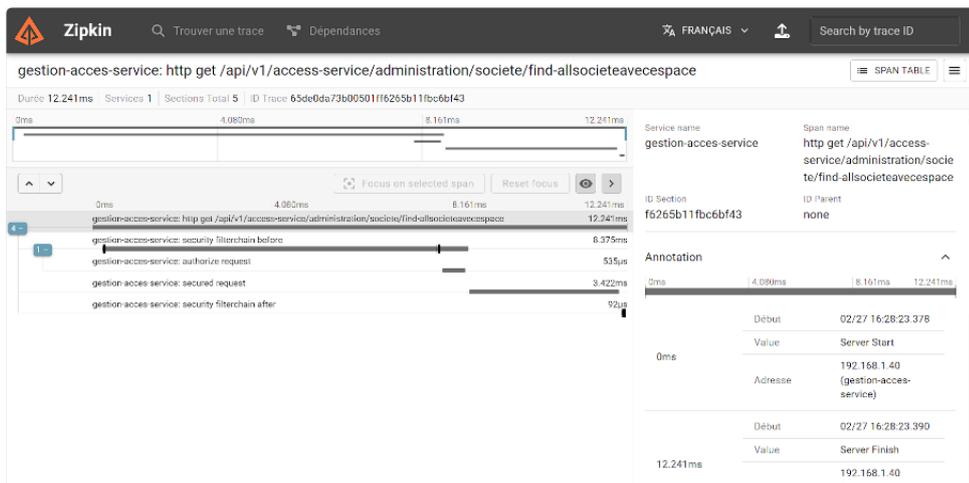


Figure 7-8 Zipkin 3/3

- Grafana

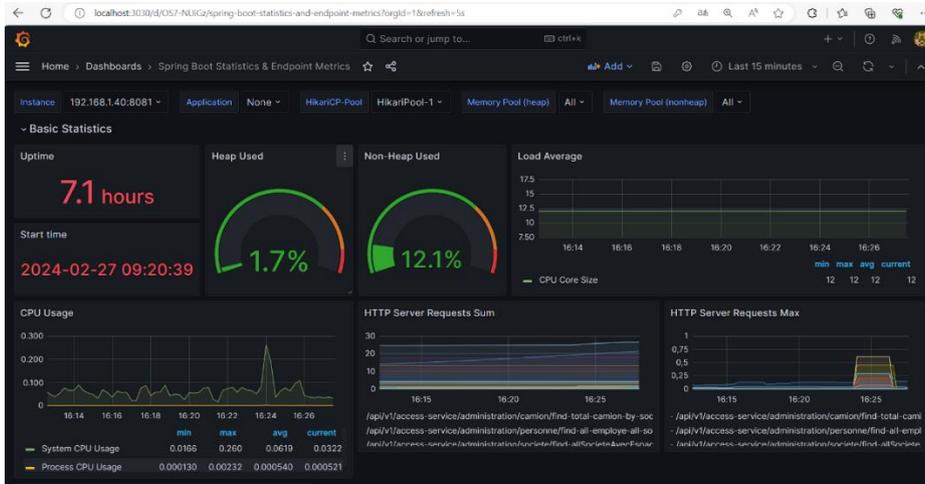


Figure 7-9 Tableau de bord grafana

- Spring boot admin

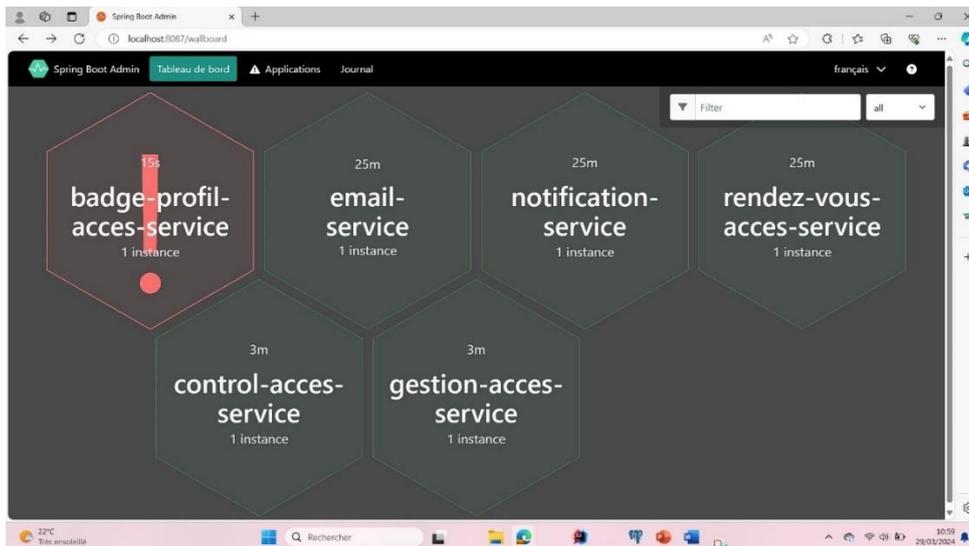


Figure 7-10 Spring boot admin

# Bibliographie

- [1] visiotechsecurity, « Différents types de contrôles d'accès : besoins et technologies », 11 août 2017.
- [2] D.-H. Shin, J. Jung, et B.-H. Chang, « The psychology behind QR codes: User experience perspective », *Comput. Hum. Behav.*, vol. 28, n° 4, p. 1417-1426, juill. 2012, doi: 10.1016/j.chb.2012.03.004.
- [3] S. Tiwari, « An Introduction to QR Code Technology », in *2016 International Conference on Information Technology (ICIT)*, déc. 2016, p. 39-44. doi: 10.1109/ICIT.2016.021.
- [4] A. N. Nambiar, « RFID Technology: A Review of its Applications », 2009.
- [5] Y. AMOUSSASANOGO, « RÉALISATION À L'AIDE D'UNE PLATEFORME NI PXI-E D'UN LECTEUR RFID RECONFIGURABLE POUR DES APPLICATIONS EN MILIEUX HOSTILES ». novembre 2016.
- [6] L. Mohamed, M. Souheil, C. Yamina, U. M. Larbi, et B. M'hidi, « Etude et réalisation d'un système de contrôle d'accès sécurisé à base de la technologie RFID », 2020.
- [7] P. Perrot, « La reconnaissance faciale : une pluralité de méthodes, une pluralité d'usages en sécurité ». septembre 2022.
- [8] I. BENCHENNANE, « Etude et mise au point d'un procédé biométrique multimodale pour la reconnaissance des individus », 2015.
- [9] P.-A. Muller, *Modélisation objet avec UML*. Paris : Eyrolles, 2000.
- [10] V. Messenger Rota, *Gestion de projet: vers les méthodes agiles*. in Architecte logiciel. Paris: Eyrolles, 2007.
- [11] P. Roques, *UML 2: modéliser une application Web*, 4e éd. in Les cahiers du programmeur. Paris: Eyrolles, 2008.
- [12] F. KAMNANG NGANSOP, *Mise en place d'un logiciel pour le suivi d'anomalies industrielles*. 2017.
- [13] Sophnoui, « UP : Unified Process. » 2020.
- [14] D. G. Frédéric, « Méthodologie des systèmes d'information - UML », 2001.
- [15] Project managing institute et Agile Alliance, Éd., *Guide pratique agile*. Newton Square (Pa.): PMI, Project management institute, global standard, 2017.
- [16] K. Schwaber et J. Sutherland, « The Definitive Guide to Scrum: The Rules of the Game ». novembre 2020.
- [17] G. Booch, *Le guide de l'utilisateur UML*. Paris : Eyrolles, 2000.

- [18] B. P. Bergeron, *Developing serious games*. Hingham, Mass. : Charles River Media, 2006.
- [19] M. Fowler, *UML 2.0*. Paris : CampusPress, 2004.
- [20] J. Ingeno, *Software Architect's Handbook*. 2018.
- [21] R. Pethuru, A. Raman, et H. Subramanian, *Architectural Patterns*. 2017.
- [22] D. Conan, C. Taconet, et C. Bac, « Conception et programmation orientées objet ». octobre 2015.
- [23] L. Debrauwer et F. V. der Heyde, *UML 2: Initiation, exemples et exercices corrigés*. Ediciones ENI, 2008.
- [24] P. Roques, *UML 2 Par la Pratique*. 2006.
- [25] IBM, « Diagrammes de cas d'utilisation - Documentation IBM ». 6 mars 2021.
- [26] H. Aloulou, « Dérivation de diagrammes de séquence UML compactes à partir de traces d'exécution en se basant des heuristiques ». juillet 2015.
- [27] H. Grati, « Rétro-ingénierie des diagrammes de séquence par visualisation interactive », juill. 2010.
- [28] Y. Boudria, R. Chial, et A. Belghiat, « Une Approche Automatique de Transformation des Diagrammes d'activité UML vers OWL-S », Thesis, University of Jijel, 2020.
- [29] T. Pender, *UML Bible*. 2003.
- [30] P. G. D. Group, « PostgreSQL ». 28 février 2024.
- [31] R. Krimi, « GitHub C'est Quoi et Comment L'utiliser ? » 2023.