République Tunisienne

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

Université de Tunis El Manar

Faculté des Sciences de Tunis

Département des Sciences de l'informatique

RAPPORT DE PROJET DE FIN D ETUDES

Présenté en vue de l'obtention du
Diplôme National de Licence en Ingénierie des Systèmes Informatiques
« Computer Engineering »
Spécialité : Systèmes Embarqués et Internet des Objets

Par

Malek Ben Slama
Yassine Hosni

Gestion des opérations des caisses

Organisme d'accueil : SNCFT

Encadré par : Mr. GARGOURI Khalil

Supervisé par : Mr.Mabrouk

Année Universitaire : 2022-202

Dédicaces

Je dédie ce modeste travail :

*A ma chère mère **Zeineb**,*
*A mon cher père **Moncef**,*
*A mes chers frères **Aziz** et **Aymen**,*
Pour leur amour, leur soutien, tous les sacrifices consentis et leurs précieux conseils, pour toute leur assistance et leur présence dans ma vie. J'espère qu'ils trouvent dans ce travail l'expression de ma gratitude.

*

A mes amis pour leurs encouragements et pour tous les bons moments que j'ai vécus avec eux. Je ne pourrai être que reconnaissant envers tout ce qu'ils ont fait pour moi.

**

A toutes celles et ceux qui m'ont aidé à franchir un horizon dans ma vie ...

Malek Ben Slama

Je dédie ce modeste travail :

À mes parents, pour leur amour inconditionnel, leur soutien indéfectible et les sacrifices consentis pour mon éducation. Chaque pas que j'ai franchi est le fruit de leur dévouement.

*

À mon frère, pour sa présence constante, son encouragement et son amitié précieuse. Ensemble, nous avons grandi et évolué, et je suis fière de partager ce moment avec lui.

**

À mes enseignants et mentors, dont la sagesse et les conseils éclairés ont façonné ma vision du monde. Leurs enseignements continueront d'inspirer mes pas futurs.

***

À mes amis, qui ont été des compagnons de route exceptionnels, partageant avec moi rires, joies et défis. Chacun d'entre eux a contribué à la richesse de mon expérience.

****

À toutes les personnes qui, de près ou de loin, ont apporté leur pierre à l'édifice de ce projet. Leur influence a été une source d'inspiration et de motivation.

*****

Que ce travail soit le reflet de ma gratitude envers chacun d'entre eux.

Yassine Hosni

Remerciements

Nos remerciements et notre gratitude vont à Monsieur MABROUK, notre encadrant professionnel, pour la confiance qu'il nous a accordée en acceptant de superviser notre stage auprès de son honorable entreprise et de nous avoir assuré toutes les conditions appropriées pour réussir ce projet de fin d'études.

On adresse, aussi, nos profonds et sincères remerciements à Monsieur Gargouri KHALIL, notre encadrant qui a accepté de diriger ce travail, pour ses indispensables conseils, son encouragement continu, et son temps précieux qu'il nous a accordé tout au long de la réalisation de ce projet de fin d'études.

Que les membres du jury trouvent aussi, ici, l'expression de nos remerciements pour l'honneur qu'ils nous font en acceptant d'évaluer ce travail.

Table des matières

Introduction Générale 1

1 Présentation du projet 2
1.1 Introduction.....2
1.2 Organisme d'accueil.....2
1.2.1 Principales missions : 2
1.2.2 Activités : 3
1.2.3 Ressources Humaines : 3
1.2.4 Fiche d'entreprise de SNCFT 4
1.3 Présentation du projet.....4
1.3.1 Cadre général du projet.....4
1.3.2 Problématique.....4
1.3.3 Etude de l'existant.....5
1.3.4 Solution proposée 5
1.4 Conclusion.....5

2 Spécification des besoins 6
2.1 Introduction.....6
2.2 Analyse des besoins.....6
2.2.1 Identification des acteurs.....6
2.2.2 Identification des besoins.....6
2.2.2.1 Besoins fonctionnels.....7
2.2.2.2 Besoins non fonctionnels.....7
2.2.3 Diagrammes des cas d'utilisation 7
2.2.3.1 Authentification.....10
2.2.3.2 Gestion des utilisateurs 11
2.2.3.3 Gestion des gares 12
2.2.3.4 Gestion des banques.....13
2.2.3.5 Gestion des caisses 14
2.2.3.6 Gestion des transactions 16
2.2.3.7 Gestion des opérations.....17
2.3 Conclusion.....18

3 Conception 19
3.1 Introduction.....19
3.2 Diagramme de cas d'utilisation général 19
3.3 Diagramme de classe général 20
3.4 Diagrammes de séquence.....22
3.4.0.1 Authentification.....22

3.4.0.2 Ajout de Gare 23
3.4.0.3 Suppression de Gare 24
3.4.0.4 Modification de Gare 25
3.4.0.5 Ajout de Banque 26
3.4.0.6 Suppression de Banque 27
3.4.0.7 Modification de Banque 28
3.4.0.8 Ajout de Caisse 29
3.4.0.9 Suppression de Caisse 30
3.4.0.10 Modification de Caisse 31
3.4.0.11 Ajout d'Utilisateur 32
3.4.0.12 Suppression d'Utilisateur 33
3.4.0.13 Modification d'Utilisateur 34
3.4.0.14 Ajout de Transaction 35
3.4.0.15 Suppression de Transaction 36
3.4.0.16 Modification de Transaction 37
3.4.0.17 Ajout d'Opération 38
3.4.0.18 Suppression d'Opération 39
3.4.0.19 Modification d'Opération 40
3.5 Conclusion 41

4 Réalisation **42**
4.1 Introduction 42
4.2 Environnement de développement 42
4.2.1 Environnement matériel : 42
4.2.1.1 Nos ordinateurs : 42
4.2.2 Environnement logiciel 42
4.2.2.1 Outils de conception 43
4.2.2.2 Outils de gestion de projet 43
4.2.2.3 Outils de test 43
4.2.2.4 Technologies utilisées 44
4.2.2.5 Environnement de développement 44
4.2.2.6 Environnement de développement de base de données 45
4.2.3 Choix des langages de programmation 45
4.2.4 ChatGPT 47
4.2.5 Bootstrap 47
4.3 Présentation de l'application 48
4.4 Tests de validation 62
4.5 Conclusion 75
Conclusion et perspectives 76

Netographie **76**

Annexes **77**

Table des figures

Figure 1.1 Logo SNCFT 4

Figure 2.1 Diagramme de cas d'utilisation global 9
Figure 2.2 Diagramme de cas d'utilisation « Authentification » 10
Figure 2.3 Diagramme de cas d'utilisation « Gestion des utilisateurs » 11
Figure 2.4 Diagramme de cas d'utilisation « Gestion des gares » 12
Figure 2.5 Diagramme de cas d'utilisation « Gestion des banques » 14
Figure 2.6 Diagramme de cas d'utilisation « Gestion des caisses » 15
Figure 2.7 Diagramme de cas d'utilisation « Gestion des transactions » 16
Figure 2.8 Diagramme de cas d'utilisation « Gestion des opérations » 17

Figure 3.1 Diagramme de classe général.....21
Figure 3.2 Diagramme de séquence « Authentification » 22
Figure 3.3 Diagramme de séquence « Ajout de Gare » 23
Figure 3.4 Diagramme de séquence « Suppression de Gare » 24
Figure 3.5 Diagramme de séquence « Modification de Gare » 25
Figure 3.6 Diagramme de séquence « Ajout de Banque » 26
Figure 3.7 Diagramme de séquence « Suppression de Banque » 27
Figure 3.8 Diagramme de séquence « Modification de Banque » 28
Figure 3.9 Diagramme de séquence « Ajout de Caisse » 29
Figure 3.10 Diagramme de séquence « Suppression de Caisse » 30
Figure 3.11 Diagramme de séquence « Modification de Caisse » 31
Figure 3.12 Diagramme de séquence « Ajout d'Utilisateur » 32
Figure 3.13 Diagramme de séquence « Suppression d'Utilisateur » 33
Figure 3.14 Diagramme de séquence « Modification d'Utilisateur » 34
Figure 3.15 Diagramme de séquence « Ajout de Transaction » 35
Figure 3.16 Diagramme de séquence « Suppression de Transaction » 36
Figure 3.17 Diagramme de séquence « Modification de Transaction » 37
Figure 3.18 Diagramme de séquence « Ajout d'Opération » 38
Figure 3.19 Diagramme de séquence « Suppression d'Opération » 39
Figure 3.20 Diagramme de séquence « Modification d'Opération » 40

Figure 4.1 StarUML 43
Figure 4.2 Logo Git 43
Figure 4.3 Logo Postman 43
Figure 4.4 Logo Angular 44
Figure 4.5 Logo Angular Material 44
Figure 4.6 Logo Spring Boot 44
Figure 4.7 Logo VSCode 45

Figure 4.8 Logo IntelliJ IDEA 45
Figure 4.9 Logo MySQL 45
Figure 4.10 Logo Java 46
Figure 4.11 Logo TypeScript 46
Figure 4.12 Logo HTML 46
Figure 4.13 Logo CSS 47
Figure 4.14 Logo ChatGPT 47
Figure 4.15 Logo Bootstrap 47
Figure 4.16 Page de démarrage 48
Figure 4.17 Interface Login 48
Figure 4.18 Interface de Login 49
Figure 4.19 Interface de Gare 49
Figure 4.20 Interface Add Gare 50
Figure 4.21 Interface Update Gare 50
Figure 4.22 Delete Gare 51
Figure 4.23 Interface Add Banque 51
Figure 4.24 Interface Update Banque 52
Figure 4.25 Delete Banque 52
Figure 4.26 Interface Add Caisse 53
Figure 4.27 Interface Update Caisse 53
Figure 4.28 Delete Caisse 54
Figure 4.29 Interface Add Utilisateur 54
Figure 4.30 Interface Update Utilisateur 55
Figure 4.31 Delete Utilisateur 55
Figure 4.32 Interface Add Transaction 56
Figure 4.33 Interface Update Transaction 56
Figure 4.34 Delete Transaction 57
Figure 4.35 Interface Add Operation 57
Figure 4.36 Interface Update Operation 58
Figure 4.37 Delete Operation 58
Figure 4.38 Interface Add Caissier 59
Figure 4.39 Interface ajout avec succès 59
Figure 4.40 Interface Login Caissier 60
Figure 4.41 Interface connecté avec caissier 60
Figure 4.42 Test d'authentification. 61
Figure 4.43 Test Logout 61
Figure 4.44 Test ajout d'utilisateur. 62
Figure 4.45 Test d'obtenir les utilisateurs. 62
Figure 4.46 Test modification d'utilisateur. 63
Figure 4.47 Test suppression d'utilisateur. 63
Figure 4.48 Test ajout de gare. 64
Figure 4.49 Test d'obtenir les gares. 64
Figure 4.50 Test modification de gare. 65
Figure 4.51 Test suppression d'une gare. 65
Figure 4.52 Test ajout de banque. 66
Figure 4.53 Test d'obtenir les banques. 66
Figure 4.54 Test modification de banque. 67
Figure 4.55 Test suppression d'une banque. 67

Figure 4.56 Test ajout de caisse. 68
Figure 4.57 Test d'obtenir les caisses. 68
Figure 4.58 Test modification de caisse. 69
Figure 4.59 Test suppression d'une caisse. 69
Figure 4.60 Test ajout de transaction. 70
Figure 4.61 Test d'obtenir les transactions. 70
Figure 4.62 Test modification de transaction. 71
Figure 4.63 Test suppression d'une transaction. 71
Figure 4.64 Test ajout d'opération. 72
Figure 4.65 Test d'obtenir les opérations. 72
Figure 4.66 Test modification d'opération. 73
Figure 4.67 Test suppression d'opération. 73
Figure 4.68 Logo JWT 77
Figure 4.69 Structure du jeton JWT 77

Liste des tableaux

Tableau 2.1 Classification des cas d'utilisation par acteur. 8
Tableau 2.2 Description relative au cas « Authentification » 10
Tableau 2.3 Description relative au cas "Gestion des utilisateurs" 12
Tableau 2.4 Description relative au cas "Gestion des gares" 13
Tableau 2.5 Description relative au cas "Gestion des banques" 14
Tableau 2.6 Description relative au cas "Gestion des caisses" 15
Tableau 2.7 Description relative au cas "Gestion des transactions" 16
Tableau 2.8 Description relative au cas "Gestion des opérations" 18

Tableau 4.1 Nos ordinateurs 42

Introduction Générale

À l'ère d'une mobilité croissante et d'un réseau ferroviaire en perpétuelle expansion, la gestion fluide et sécurisée des opérations de caisses est devenue un impératif stratégique pour la Société Nationale des Chemins de fer Tunisiens (SNCFT). Une gestion efficiente de ces opérations non seulement optimise les processus internes, mais assure également une expérience client optimale, garantissant ainsi la satisfaction et la fidélisation de la clientèle.

Au cours de ce stage, nous avons focalisé notre attention sur la conception et la mise en œuvre d'une solution complète de gestion des opérations de caisses. Celle-ci englobela gestion des gares, des caisses, des opérations financières, des transactions, du système bancaire et de l'authentification. En tirant profit des technologies les plus avancées, notamment la sécurité des transactions, la gestion des utilisateurs et la définition des droits d'accès, notre projet vise à optimiser et à sécuriser ces processus critiques.

L'objectif premier de cette initiative est de développer une plateforme intuitive et sécurisée qui permettra à la SNCFT de suivre, de contrôler et d'optimiser l'ensemble de ses opérations de caisses de manière efficace. En intégrant des fonctionnalités telles que le suivi des transactions en temps réel, la gestion des utilisateurs et la sécurisation des transactions financières, notre solution vise à fournir à la SNCFT les outils essentiels pour prendre des décisions éclairées et pour garantir la fluidité et la sécurité de ses opérations financières.

Nous sommes convaincus que cette solution de gestion des opérations de caisses représente une avancée majeure pour la SNCFT, contribuant ainsi à renforcer son positionnement en tant qu'acteur majeur du transport ferroviaire. Nous avons pour ambition d'optimiser les processus financiers, d'assurer la sécurité des transactions et de faciliter la gestion des utilisateurs et des droits d'accès, afin de répondre aux besoins spécifiques de la SNCFT dans un environnement en constante évolution.

Ce rapport détaillera l'approche méthodologique adoptée, le processus de développement de la solution, les fonctionnalités clés implémentées, ainsi que les résultats attendus de cette initiative stratégique.

Chapitre 1

Présentation du projet

1.1 Introduction

Dans ce premier chapitre introductif, nous présentons l'organisme d'accueil «SNCFT». Nous présentons par la suite, le cadre général du projet et les solutions existantes dans le but de dévoiler les objectifs de notre travail. Nous allons expliquer à la fin la méthode adoptée et les démarches suivies afin de réaliser ce projet.

1.2 Organisme d'accueil

SNCFT [1] La Société Nationale des Chemins de Fer Tunisiens est une entité publique à caractère non administratif dédiée à la gestion, à l'exploitation et à l'entretien du réseau ferroviaire tunisien. Fondée en 1956, elle dispose de la personnalité civile et de l'autonomie financière, et est placée sous la tutelle du Ministère du Transport. Elle incarne un maillon essentiel du secteur des transports en Tunisie, jouant un rôle central dans la mobilité nationale.

1.2.1 Principales missions :

Assurer, dans les meilleures conditions de sécurité et de ponctualité, les services de transport ferroviaire de voyageurs et de marchandises sur le réseau ferroviaire tunisien. Gérer et contrôler les biens qui lui sont confiés par l'État sous forme de concession.

L'ensemble de ces activités est géré par 5 unités d'affaires chargées du transport à savoir :

- Unité d'affaires du transport des voyageurs sur les grandes lignes (transport interurbain).
- Unité d'affaires du transport des voyageurs sur la banlieue sud de Tunis (Tunis–Borj Cédria–Erriadh).
- Unité d'affaires du transport des voyageurs sur la banlieue du sahel (Sousse Bab Jdid–Monastir–Mahdia).
- Unité d'affaires du transport de phosphate.
- Unité d'affaires du transport de fret (engrais et souffre, matériaux de construction, produits alimentaires, coke de pétrole, conteneurs maritimes, etc. . .).

Ces 4 unités d'affaires sont appuyées par d'autres unités prestataires de service à savoir :

- Direction Centrale de l'Exploitation du Réseau Ferroviaire Tunisien chargé de l'exploitation du réseau et des biens immobiliers.
- Unité d'Affaires de la Maintenance Industrielle chargée de la maintenance du matériel roulant.
- Direction Centrale du Domaine Ferroviaire.
- Direction Centrale de la Gestion et Finances.

1.2.2 Activités :

La Société Nationale des Chemins de fer Tunisiens (SNCFT) est principalement responsable de l'exploitation du réseau ferroviaire en Tunisie. Ses activités principales comprennent :

1. **Transport de Passagers :** La SNCFT offre des services de transport ferroviaire de passagers à travers tout le pays. Cela inclut des liaisons quotidiennes entre les principales villes tunisiennes, telles que Tunis, Sfax, Sousse, Gabès, et bien d'autres. Les trains sont un mode de transport populaire pour les déplacements nationaux et régionaux en Tunisie.
2. **Transport de Marchandises :** En plus du transport de passagers, la SNCFT propose des services de fret ferroviaire. Elle joue un rôle essentiel dans le transport de marchandises, y compris les produits industriels, agricoles et miniers, ainsi que les conteneurs. Le transport ferroviaire de marchandises est important pour le commerce intérieur et international de la Tunisie.
3. **Entretien et Modernisation des Voies Ferroviaires :** La SNCFT est responsable de l'entretien, de la modernisation et du développement du réseau ferroviaire national. Cela comprend la réparation des voies, la mise à niveau des gares, l'acquisition de nouveaux trains et l'amélioration de l'infrastructure ferroviaire.
4. **Sécurité Ferroviaire :** La sécurité des passagers et du fret est une priorité pour la SNCFT. Elle met en place des mesures de sécurité strictes pour assurer des voyages sûrs.
5. **Développement de Nouvelles Lignes :** La SNCFT travaille sur des projets de développement de nouvelles lignes ferroviaires pour étendre le réseau et faciliter le transport dans des régions mal desservies.
6. **Collaboration Internationale :** La SNCFT collabore avec d'autres organisations ferroviaires internationales pour échanger des connaissances et des meilleures pratiques dans le domaine ferroviaire.
7. **Promotion du Transport Durable :** La SNCFT s'engage à promouvoir le transport durable en tant que mode de transport respectueux de l'environnement. Elle encourage l'utilisation des trains pour réduire la congestion routière et les émissions de gaz à effet de serre.
8. **Développement Touristique :** La SNCFT a également contribué au développement du tourisme en Tunisie en offrant des services spéciaux pour les touristes, notamment des trains touristiques pour explorer les régions pittoresques du pays.

En résumé, la SNCFT joue un rôle clé dans la mobilité et le développement économique de la Tunisie en fournissant des services de transport de passagers et de marchandises efficaces et en investissant dans l'amélioration continue de son infrastructure ferroviaire.

1.2.3 Ressources Humaines :

Conscients de l'importance de leurs activités, sensibles à la satisfaction du client, fières de leurs appartenances, 4518 cheminots constituent le vrai capital de la SNCFT.

1.2.4 Fiche d'entreprise de SNCFT :

- **Taille de l'Entreprise :**

La SNCFT est une grande entreprise nationale en Tunisie, employant un nombre significatif de personnes. Le nombre exact d'employés peut varier au fil du temps en fonction des besoins opérationnels.

- **Catégorie de l'Entreprise :**

La SNCFT peut être classée dans la catégorie des entreprises de transport ferroviaire. Plus précisément, elle est le principal exploitant du réseau ferroviaire tunisien.

- **Adresse du Siège Social :**

Société Nationale des Chemins de fer Tunisiens (SNCFT) Adresse : Rue El Jazira, Tunis 1001, Tunisie.

Figure 1.1 – Logo SNCFT

1.3 Présentation du projet

1.3.1 Cadre général du projet

Ce projet intitulé « Gestion des opérations des caisses » est réalisé dans le cadre de la préparation d'un stage de fin d'études présenté à la faculté des sciences de Tunis (FST) en vue de l'obtention du diplôme national de licence en ingénierie des systèmes informatique et IOT pour l'année universitaire 2022-2023. Ce travail est réalisé au sein de l'entreprise « SNCFT » Il consiste à développer une Application Informatique pour l'amélioration du système d'information de la société.

1.3.2 Problématique

"Comment concevoir et développer efficacement une application informatique pour la gestion des opérations de caisse au sein de la Société Nationale des Chemins de Fer Tunisiens (SNCFT) tout en prenant en compte les besoins fonctionnels et opérationnels spécifiques de l'entreprise, ainsi que la transition depuis le système existant vers le nouveau système ?"

Cette problématique pose les bases du projet en mettant en avant les enjeux liés à la conception d'une application vitale pour l'entreprise et la nécessité de s'adapter à l'existant tout en intégrant de nouvelles fonctionnalités. Elle soulève des questions telles que la collecte des besoins, la transition depuis le système existant, l'interface avec d'autres logiciels et les meilleures pratiques en matière de développement logiciel.

Vous pouvez ensuite développer votre rapport en répondant à ces questions et en fournissant une solution complète à cette problématique tout au long du travail.

1.3.3 Etude de l'existant

La SNCFT dispose actuellement d'un système d'information réalisé depuis 1990 sous l'environnement DOS avec le langage de programmation Clipper 5.0 et qui a subi au cours du temps des modifications multiples. Ce système consiste en l'automatisation des opérations de caisses et bureaux mouvements de fonds afin de fournir un sous-produit au Bureau Comptabilité Financière.

1.3.4 Solution proposée

La solution que nous proposons représente une avancée significative dans l'optimisation des opérations des caissiers. Elle se concrétise par le déploiement d'une plateforme de gestion de caisses globale, minutieusement élaborée pour satisfaire de manière spécifique leurs exigences et besoins opérationnels.

Un élément déterminant de cette solution est l'intégration de logiciels de bureau à la fois contemporains et esthétiquement attrayants. Ces applications se démarquent par leur parfaite adaptation aux derniers systèmes d'exploitation en vigueur. Cette orientation résolument tournée vers la modernité vise à garantir une expérience utilisateur des plus fluides et intuitives, tout en assurant une performance optimale et une efficacité opérationnelle inégalée.

1.4 Conclusion

Ce premier chapitre résume l'étude réalisée lors des premiers jours du stage. Ceci nous a permis de bien comprendre l'environnement de notre société, les concepts de base de notre projet, étudier l'existant et ses défaillances et conclure par la proposition d'une solution. A ce stade il est désormais possible de passer au chapitre suivant, qui illustre une analyse complète des besoins.

Chapitre 2

Spécification des besoins

2.1 Introduction

Dans ce chapitre, nous allons présenter les besoins fonctionnels et non fonctionnels de l'application ainsi que la spécification formelle des besoins à travers des diagrammes de cas d'utilisation qui seront d'une grande utilité dans le processus de son développement.

2.2 Analyse des besoins

Cette section présente les différents acteurs présents dans notre plateforme, les besoins fonctionnels et non fonctionnels ainsi que le diagramme de cas d'utilisation global.

2.2.1 Identification des acteurs

Afin de bien comprendre l'environnement étudié, il faut décrire tout d'abord les acteurs.

En effet, un acteur est toute entité soit physique (personne) ou abstraite (logiciel, système) qui interagit avec le système pour répondre à un besoin spécifique.

Donc, dans notre projet on a deux acteurs :

1- Administrateur : C'est l'acteur qui a tous les privilèges, possédant les droits les plus élevés. Il a pour tâche d'administrer le système et assurer les fonctionnalités principales de l'application.

2- Utilisateur : C'est l'acteur principal qui utilise l'application pour la gestion de sa flotte automobile.

2.2.2 Identification des besoins

Cette étape constitue une phase de départ importante avant de commencer le développement et dans laquelle nous identifions les détails de l'ensemble des fonctionnalités et les besoins de notre plateforme. Nous distinguons deux types de besoins, les besoins fonctionnels qui représentent toutes les fonctionnalités primordiales spécifiées dans le cahier des charges et les besoins non fonctionnels qui sont des propriétés du système, et qui le caractérisent.

2.2.2.1 Besoins fonctionnels

- Authentification : Un administrateur qui a un compte doit s'authentifier pour pouvoir accéder à la plateforme. C'est une étape qui vise à vérifier l'identité de celui qui veut se connecter.

- Gestion des utilisateurs et des droits d'accès : Un administrateur peut gérer les utilisateurs en activant leurs comptes, en les supprimant, en modifiant leurs données ou leurs rôles en cas de nécessité et aussi en consultant la liste de tous les utilisateurs ainsi que celle des utilisateurs en attente et en les cherchant selon certains critères.

- Gestion des Gares : Un administrateur peut modifier les données de Gares.

- Gestion des Banques : Un administrateur peut modifier les données de Banques.

- Gestion des Caisses : Un administrateur peut modifier les données de Caisses.

- Gestion des Transactions : Un administrateur peut modifier les données de Tran-sactions.

- Gestion des Operations : l'administrateur et le caissier peuvent modifier les données d'Opération.

2.2.2.2 Besoins non fonctionnels

- La performance : La plateforme devra être performante en termes de temps de réponse et de temps de chargement de pages ce qui signifie qu'elle doit être élaborée avec un accès performant aux informations.

- L'extensibilité : La plateforme devra être extensible en offrant la possibilité d'ajouter ou de modifier quelques fonctionnalités vu que les besoins ne sont pas fixes.

- La sécurité : C'est l'un des critères les plus importants pour une application. En effet, la plateforme doit être hautement sécurisée et l'accès à travers l'authentification doit être limité aux personnes existant dans la base de données et ayant des comptes activés et avec des droits d'accès bien spécifiés grâce au jeton car les informations ne devront pas être accessibles à tout le monde.

- L'ergonomie : La présentation de la plateforme doit être homogène, claire et intuitive. Elle doit respecter les principes des Interfaces Homme/Machine ainsi que la charte graphique de l'entreprise.

- La convivialité : La plateforme doit être facile à manipuler, simple et compréhensible par tout type d'utilisateurs.

2.2.3 Diagrammes des cas d'utilisation

Un cas d'utilisation modélise un dialogue entre un acteur et le système. C'est la représentation d'une fonctionnalité offerte par le système. L'ensemble des cas d'utilisation forme toutes les façons possibles d'utilisation du système.

Les principaux objectifs des cas d'utilisation sont :

— Permettre de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
— Centrer l'expression des exigences du système sur ses utilisateurs.
— Se limiter aux préoccupations réelles des utilisateurs : ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.

— Identifier les utilisateurs du système (acteurs) et leurs interactions avec le système

Nous présentons, à partir de la figure 2.1 ci-après, le diagramme de cas d'utilisation pour identifier l'acteur et son interaction avec le système et permettre de structurer ses objectifs :

Acteurs	Cas d'utilisation
Administrateur	- Authentification - Gestion des Gares - Gestion des Utilisateurs - Gestion des Banques - Gestion des Caisses - Gestion des Transactions - Gestion des Operations
Caissier	- Authentification - Gestion des Operations

TABLE 2.1 – Classification des cas d'utilisation par acteur.

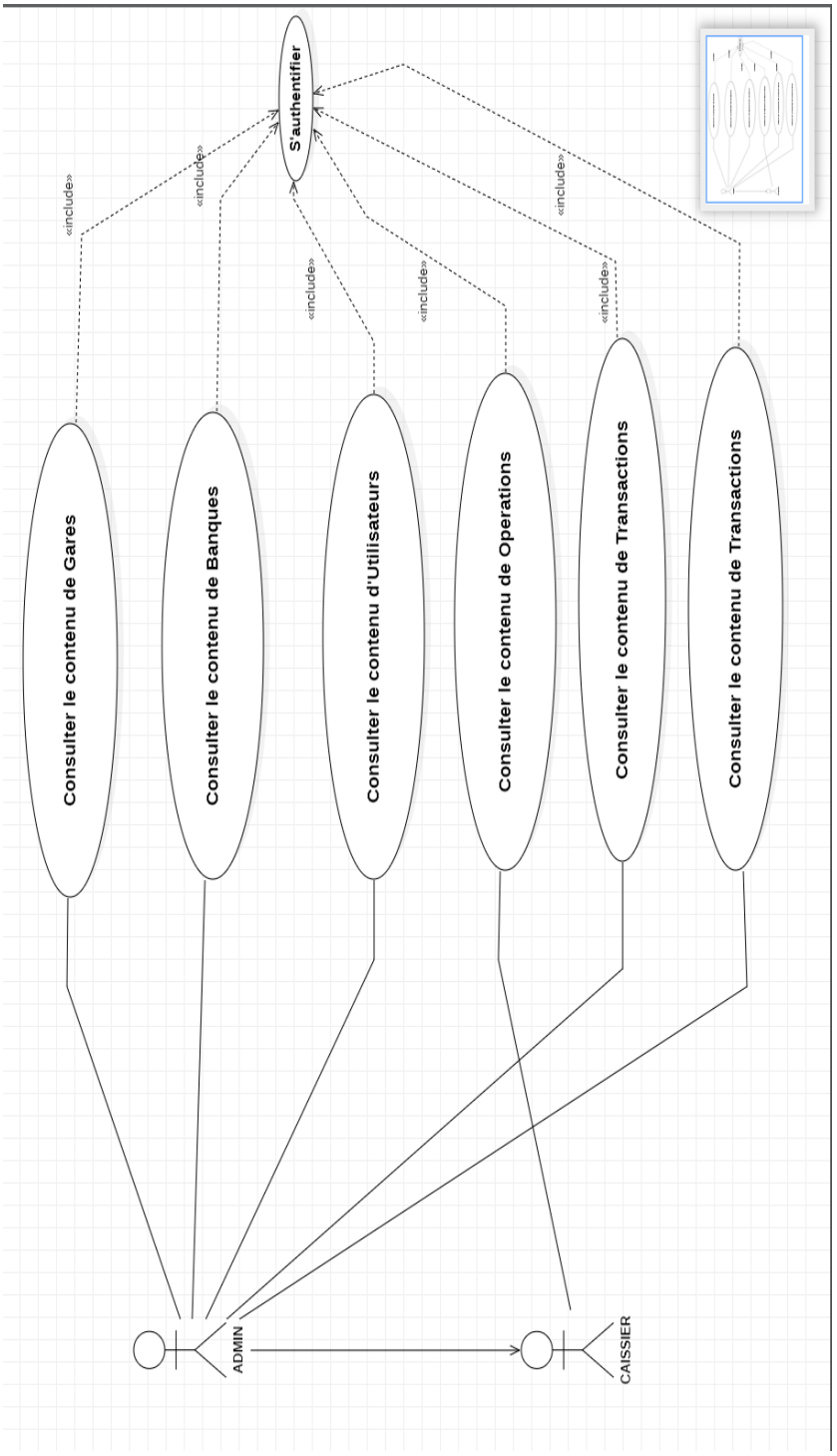


Figure 2.1 – Diagramme de cas d'utilisation global

2.2.3.1 Authentification

Ce diagramme de cas d'utilisation représente les interactions entre les acteurs et le système lors du processus d'authentification. Il décrit comment les utilisateurs interagissent avec le système pour s'authentifier et accéder aux fonctionnalités sécurisées de l'application.

- Raffinement du diagramme de cas d'utilisation « Authentification » :

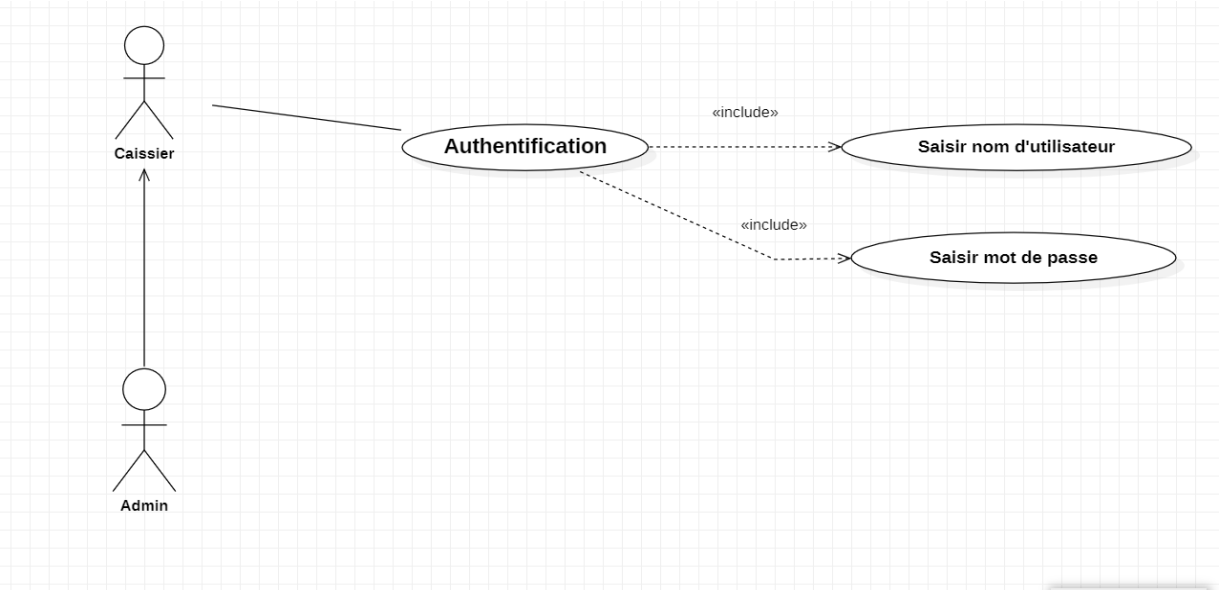


Figure 2.2 – Diagramme de cas d'utilisation « Authentification »

- Description relative au cas « Authentification » :

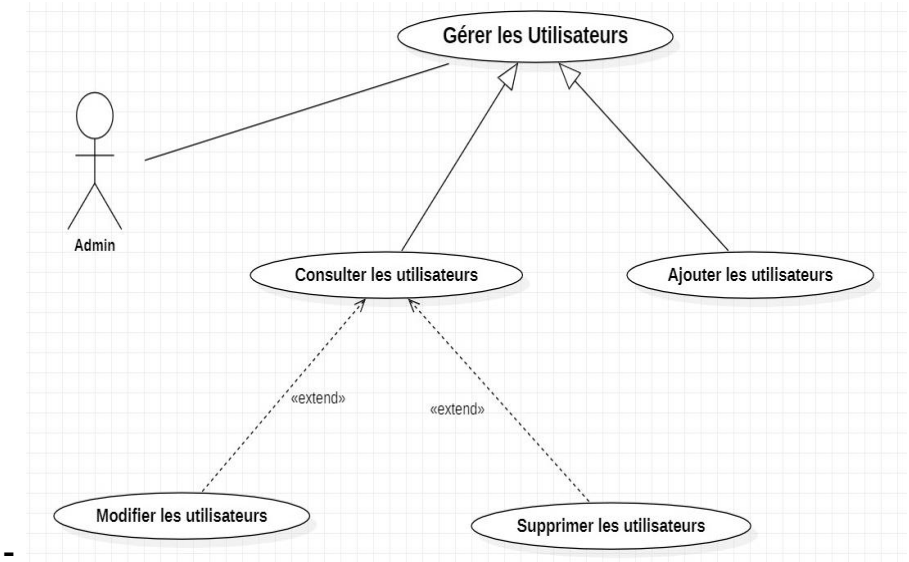
Acteur	L'utilisateur
Objectif	Permet aux utilisateurs d'application d'accéder à leur espace.
Pré-condition	— L'utilisateur doit exister dans la base de données et doit obligatoirement être connecté à Internet
Post-condition	Utilisateur authentifié
Scénario nominatif	— L'utilisateur saisit son email et son mot de passe. — L'utilisateur clique sur le bouton "Connexion" pour confirmer. — Vérifier si le champ est vide et donner un message d'erreur. — Les paramètres d'accès sont envoyés vers la base de données pour la vérification des données. — L'interface utilisateur apparaîtra.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche

TABLE 2.2 – Description relative au cas « Authentification »

2.2.3.2 Gestion des utilisateurs

Ce diagramme de cas d'utilisation représente les interactions entre les acteurs et le système pour la gestion des utilisateurs. Il décrit comment les administrateurs interagissent avec le système pour effectuer des opérations telles que la création, la modification, la suppression et la consultation des utilisateurs. Le diagramme met en évidence les fonctionnalités permettant aux administrateurs de gérer efficacement les utilisateur, y compris l'ajout des utilisateurs, modification des données et la suppression des utilisateurs.

- Raffinement du diagramme de cas d'utilisation « Gestion des utilisateurs



-

Figure 2.3 – Diagramme de cas d'utilisation « Gestion des utilisateurs »

- Description relative au cas « Gestion des utilisateurs » :

Acteur	L'Administrateur
Objectif	Permet à l'administrateur d'application de gérer les utilisateurs.
Pré-condition	— L'Administrateur doit exister dans la base de données.
Post-condition	— Utilisateur ajouté. — Données modifiées. — Utilisateur supprimé.
Scénario nominatif	— L'Administrateur doit s'authentifier. — L'Administrateur soit ajoute, supprime
Ajouter utilisateurs	— L'administrateur clique sur ajouter. — L'administrateur remplit le formulaire. — L'administrateur clique sur ajouter. — Un message de succès s'affiche.

Supprimer utilisateurs	— L'administrateur clique sur supprimer. — L'administrateur confirme la suppression. — Un message de succès s'affiche.
Modifier Données	— L'administrateur clique sur update — L'administrateur confirme la modification des données. — Un message de succès s'affiche.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche.

TABLe 2.3 – Description relative au cas "Gestion des utilisateurs"

2.2.3.3 Gestion des gares

Ce diagramme de cas d'utilisation représente les interactions entre les acteurs et le système pour la gestion des gares. Il décrit comment les administrateurs interagissent avec le système pour effectuer des opérations telles que la création, la modification, la suppression et la consultation des utilisateurs. Le diagramme met en évidence les fonctionnalités permettant aux administrateurs de gérer efficacement les gares, y compris l'ajout des gares, modification des données et la suppression des gares.

- Raffinement du diagramme de cas d'utilisation « Gestion des gares » :

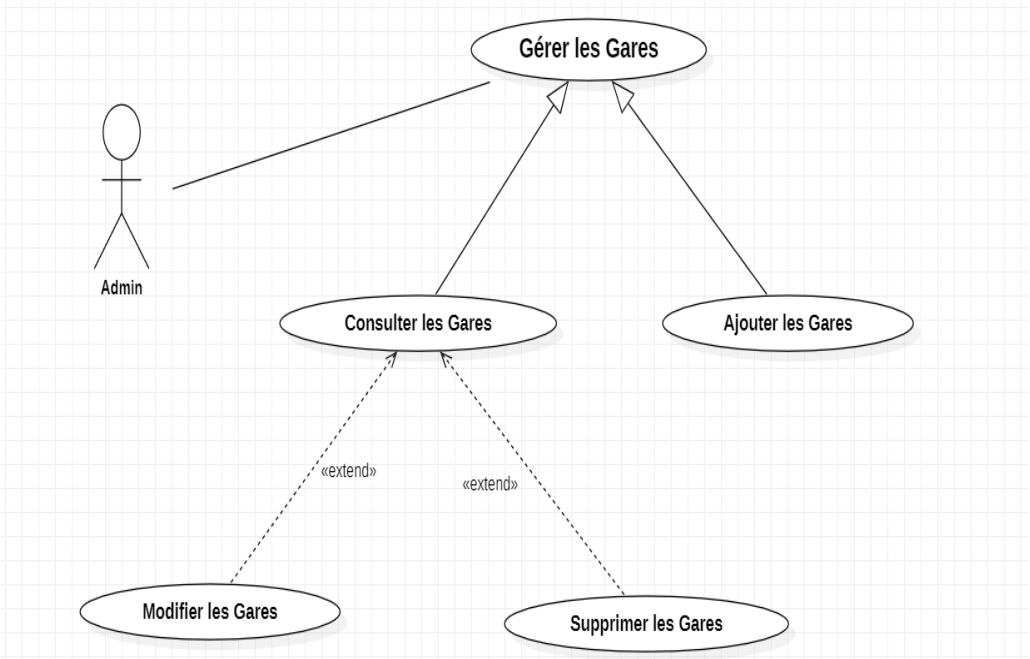


Figure 2.4 – Diagramme de cas d'utilisation « Gestion des gares »

- Description relative au cas « Gestion des gares » :

Acteur	L'Administrateur
Objectif	Permet à l'administrateur d'application de gérer les gares.
Pré-condition	— L'Administrateur doit exister dans la base de données
Post-condition	— Gare ajoutée. — Gare supprimée.
Scénario nominatif	— L'Administrateur doit s'authentifier. — L'Administrateur soit ajoute, supprime
Ajouter gares	— L'administrateur clique sur ajouter. — L'administrateur remplit le formulaire. — L'administrateur clique sur ajouter. — Un message de succès s'affiche.
Supprimer gares	— L'administrateur clique sur supprimer. — L'administrateur confirme la suppression. — Un message de succès s'affiche.
Modifier Données	— L'administrateur clique sur update — L'administrateur confirme la modification des données. — Un message de succès s'affiche.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche.

TABLE 2.4 – Description relative au cas "Gestion des gares"

2.2.3.4 Gestion des banques

Ce diagramme de cas d'utilisation représente les interactions entre les acteurs et le système pour la gestion des banques. Il décrit comment les administrateurs interagissent avec le système pour effectuer des opérations telles que la création, la modification, la suppression et la consultation des banques. Le diagramme met en évidence les fonctionnalités permettant aux administrateurs de gérer efficacement les banques, y compris l'ajout des banques, modification des données et la suppression des banques.

- Raffinement du diagramme de cas d'utilisation « Gestion des banques » :

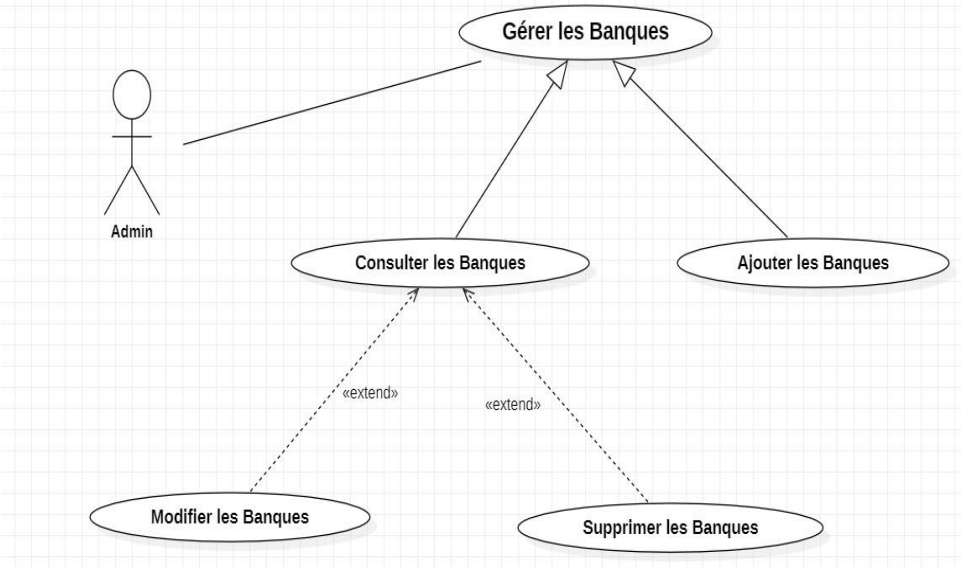


Figure 2.5 – Diagramme de cas d'utilisation « Gestion des banques »

- Description relative au cas « Gestion des banques » :

Acteur	L'Administrateur
Objectif	Permet à l'administrateur d'application de gérer les banques.
Pré-condition	— L'Administrateur doit exister dans la base de données
Post-condition	— Banque ajoutée. — Banque supprimée.
Scénario nominatif	— L'Administrateur doit s'authentifier. — L'Administrateur soit ajoute, supprime
Ajouter banques	— L'administrateur clique sur ajouter. — L'administrateur remplit le formulaire. — L'administrateur clique sur ajouter. — Un message de succès s'affiche.
Supprimer banques	— L'administrateur clique sur supprimer. — L'administrateur confirme la suppression. — Un message de succès s'affiche.
Modifier Données	— L'administrateur clique sur update — L'administrateur confirme la modification des données. — Un message de succès s'affiche.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche.

TABLE 2.5 – Description relative au cas "Gestion des banques"

2.2.3.5 Gestion des caisses

Ce diagramme de cas d'utilisation représente les interactions entre les acteurs et le système pour la gestion des caisses. Il décrit comment les administrateurs interagissent avec le système pour effectuer des opérations telles que la création, la modification, la suppression et la consultation des caisses. Le diagramme met en évidence les fonctionnalités permettant aux administrateurs de gérer efficacement les caisses, y compris l'ajout des caisses, modification des données et la suppression des caisses.

- Raffinement du diagramme de cas d'utilisation « Gestion des caisses » :

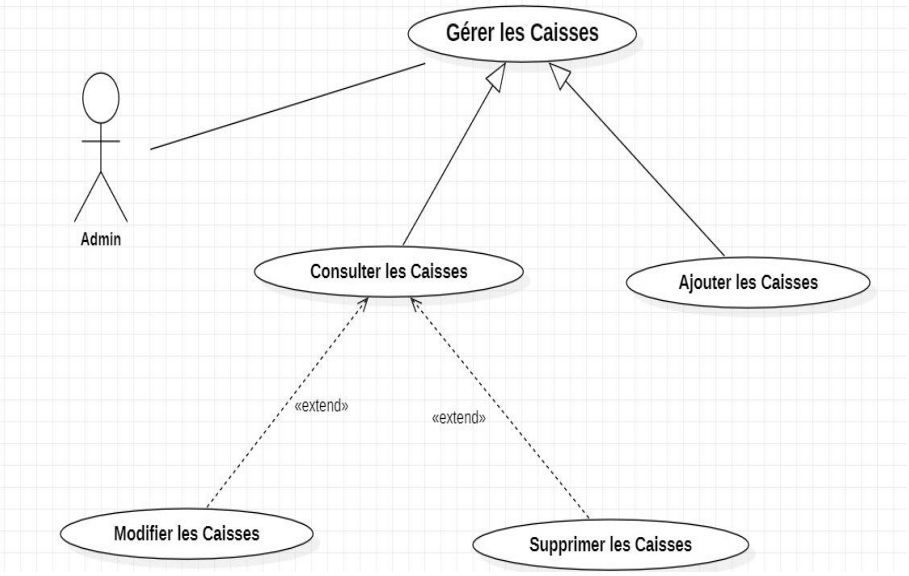


Figure 2.6 – Diagramme de cas d'utilisation « Gestion des caisses »

- Description relative au cas "Gestion des caisses

Acteur	L'Administrateur
Objectif	Permet à l'administrateur d'application de gérer les caisses.
Pré-condition	— L'Administrateur doit exister dans la base de données
Post-condition	— Caisse ajoutée. — Caisse supprimée.
Scénario nominatif	— L'Administrateur doit s'authentifier. — L'Administrateur soit ajoute, supprime
Ajouter caisses	— L'administrateur clique sur ajouter. — L'administrateur remplit le formulaire. — L'administrateur clique sur ajouter. — Un message de succès s'affiche.
Supprimer caisses	— L'administrateur clique sur supprimer. — L'administrateur confirme la suppression. — Un message de succès s'affiche.
Modifier Données	— L'administrateur clique sur update — L'administrateur confirme la modification des données. — Un message de succès s'affiche.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche.

TABLE 2.6 – Description relative au cas "Gestion des caisses"

2.2.3.6 Gestion des transactions

Ce diagramme de cas d'utilisation représente les interactions entre les acteurs et le système pour la gestion des transactions. Il décrit comment les administrateurs interagissent avec le système pour effectuer des opérations telles que la création, la modification, la suppression et la consultation des transactions. Le diagramme met en évidence les fonctionnalités permettant aux administrateurs de gérer efficacement les transactions, y compris l'ajout des transactions, modification des données et la suppression des transactions.

- Raffinement du diagramme de cas d'utilisation « Gestion des transactions» :

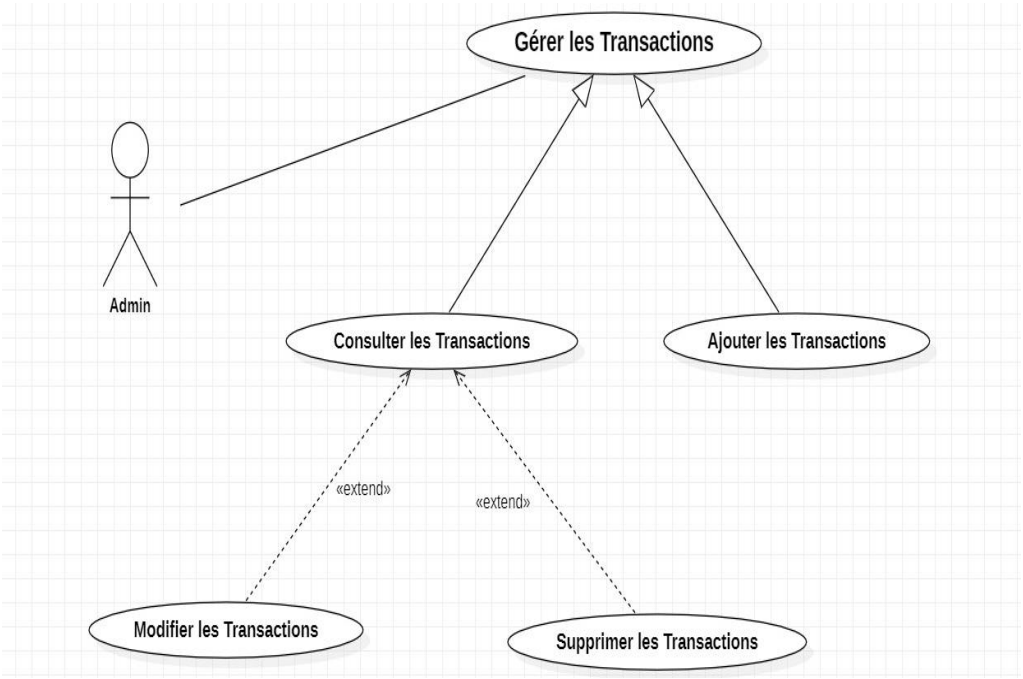


Figure 2.7 – Diagramme de cas d'utilisation « Gestion des transactions »

- Description relative au cas « Gestion des transactions » :

Acteur	L'Administrateur
Objectif	Permet à l'administrateur d'application de gérer les transactions.
Pré-condition	— L'Administrateur doit exister dans la base de données
Post-condition	— Transaction ajoutée. — Transaction supprimée.
Scénario nominatif	— L'Administrateur doit s'authentifier. — L'Administrateur soit ajoute, supprime
Ajouter transactions	— L'administrateur clique sur ajouter. — L'administrateur remplit le formulaire. — L'administrateur clique sur ajouter. — Un message de succès s'affiche.
Supprimer transactions	— L'administrateur clique sur supprimer. — L'administrateur confirme la suppression. — Un message de succès s'affiche.
Modifier Données	— L'administrateur clique sur update — L'administrateur confirme la modification des données. — Un message de succès s'affiche.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche.

TABLe 2.7 – Description relative au cas "Gestion des transactions"

2.2.3.7 Gestion des opérations

Ce diagramme de cas d'utilisation représente les interactions entre les acteurs et le système pour la gestion des opérations. Il décrit comment les administrateurs interagissent avec le système pour effectuer des opérations telles que la création, la modification, la suppression et la consultation des opérations. Le diagramme met en évidence les fonctionnalités permettant aux administrateurs de gérer efficacement les opérations, y compris l'ajout des opérations, modification des données et la suppression des opérations.

- **Raffinement du diagramme de cas d'utilisation « Gestion des opérations » :**

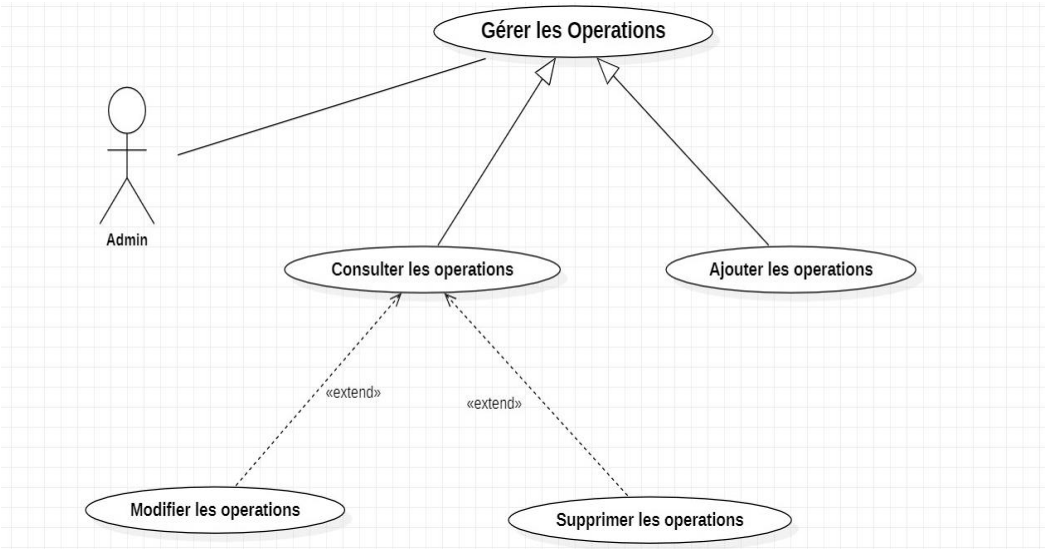


Figure 2.8 – Diagramme de cas d'utilisation « Gestion des opérations »

- Description relative au cas « Gestion des opérations » :

Acteur	L'Administrateur
Objectif	Permet à l'administrateur d'application de gérer les operations.
Pré-condition	— L'Administrateur doit exister dans la base de données
Post-condition	— Opération ajoutée. — Opération supprimée.
Scénario nominatif	— L'Administrateur doit s'authentifier. — L'Administrateur soit ajoute, supprime
Ajouter operations	— L'administrateur clique sur ajouter. — L'administrateur remplit le formulaire. — L'administrateur clique sur ajouter. — Un message de succès s'affiche.
Supprimer opérations	— L'administrateur clique sur supprimer. — L'administrateur confirme la suppression. — Un message de succès s'affiche.
Modifier Données	— L'administrateur clique sur update — L'administrateur confirme la modification des données. — Un message de succès s'affiche.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche.

TABLe 2.8 – Description relative au cas "Gestion des opérations"

2.3 Conclusion

Ce chapitre a donc permis de préciser les besoins fonctionnels que l'application développée doit offrir aux utilisateurs et les besoins non fonctionnels après l'identification des acteurs ainsi que les cas d'utilisation. Le chapitre suivant traitera de la conception du projet.

Chapitre 3

Conception

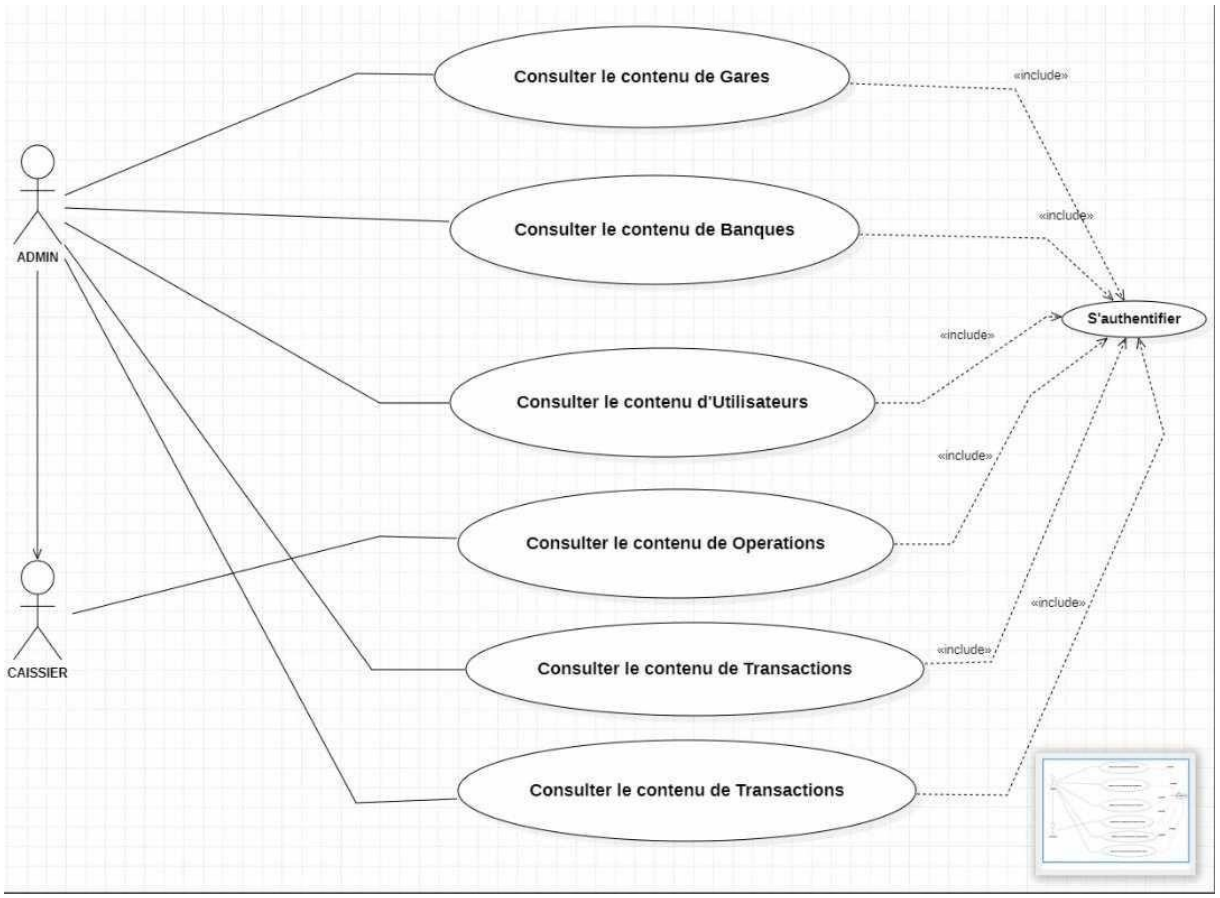
3.1 Introduction

Dans cette section, nous allons explorer en détail le processus de conception qui a été utilisé pour développer et créer notre solution. La conception est une étape essentielle qui permet de transformer une idée en réalité concrète.

3.2 Diagramme de cas d'utilisation général

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (Human ou Machine) et un system. Il est une entité significative de travail Dans un diagramme de cas d'utilisation il existe des acteurs (actors) qui interagissent avec des cas d'utilisation (use case). Les cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs du système. Une fois identifié et structuré ces besoins :

- Définissent le contour du système à modéliser.
- Permettent d'identifier les fonctionnalités principales ou les critiques du système.



3.3 Diagramme de classe général

Un diagramme de classe général est une représentation visuelle statique qui illustre la structure d'un système logiciel. Il met en avant les classes, les interfaces, les associations, les héritages, les dépendances, et d'autres éléments, ainsi que les relations entre eux. Ce type de diagramme est un outil essentiel pour la conception orientée objet et permet de visualiser comment les différentes parties d'un système interagissent.

Voici quelques éléments couramment présents dans un diagramme de classe général :

1. **Classes :** Représentent les entités fondamentales du système. Elles encapsulent les données et les méthodes qui agissent sur ces données.

2. **Attributs :** Définissent les propriétés ou les caractéristiques des classes. Ils représentent les données que chaque instance de la classe peut contenir.

3. **Méthodes :** Représentent les actions ou les opérations que les objets de la classe peuvent effectuer.

4. **Associations :** Décrit les relations entre les classes. Par exemple, une association entre les classes "Client" et "Commande" indique que chaque client peut avoir plusieurs commandes.

5. **Agrégation et Composition :** Montre les relations entre les agrégats (une partie fait partie d'un tout) et les compositions (une partie ne peut exister sans le tout).

6. **Héritage :** Indique qu'une classe hérite des attributs et des méthodes d'une autre classe. Cela représente la relation "est un type de".

7. **Interfaces :** Définit les contrats que les classes doivent implémenter. Elles spécifient les méthodes qu'une classe doit avoir.

8. **Dépendances :** Représentent les relations où une classe utilise une autre classe

pour accomplir une certaine tâche.

9. **Paquetages :** Organisent les classes en groupes logiques, ce qui facilite la gestion des systèmes complexes.

Ensemble, ces éléments permettent de visualiser la structure et les interactions entre les composants d'un système. Cela facilite la compréhension et la communication entre les membres de l'équipe de développement.

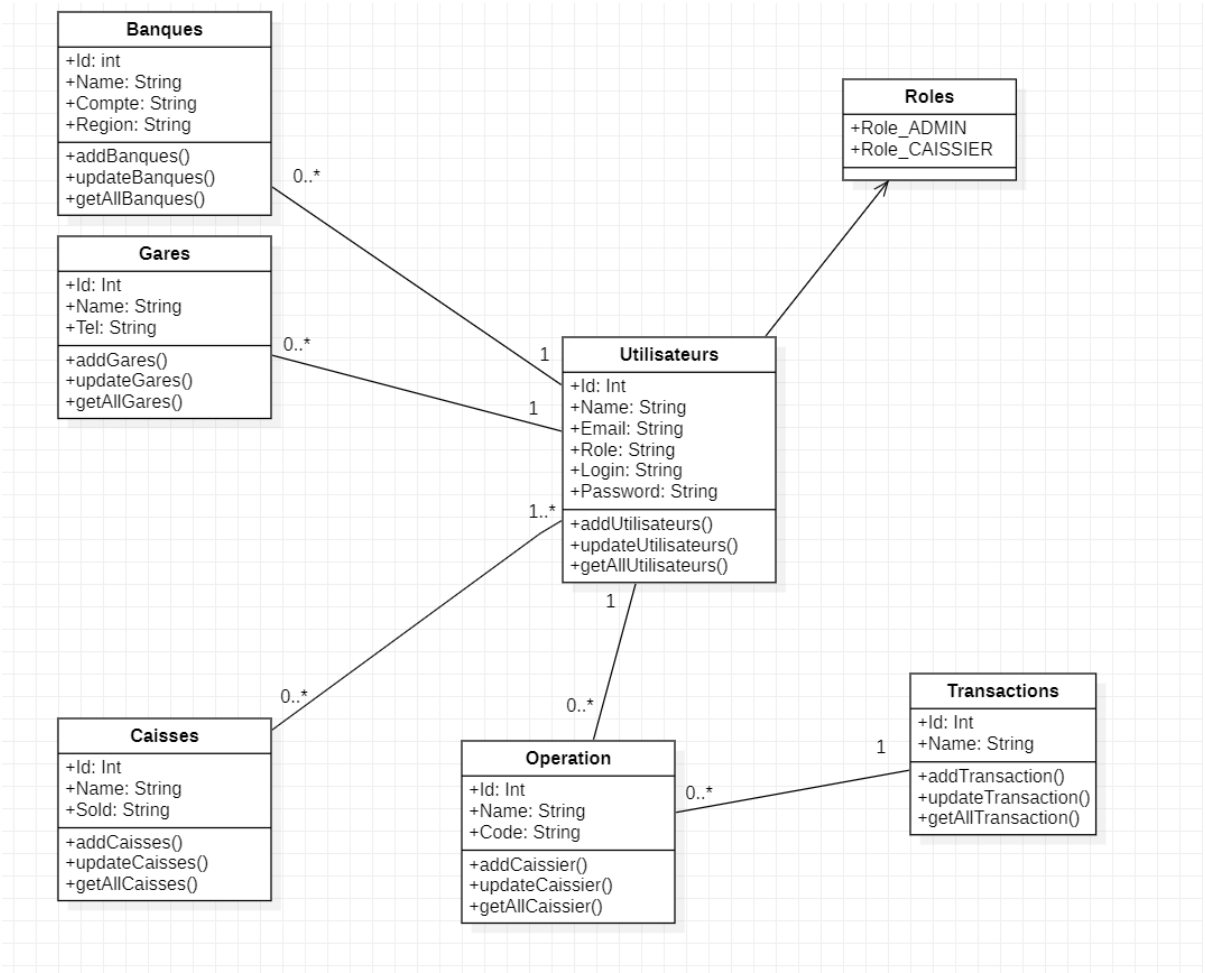


Figure 3.1 – Diagramme de classe général.

3.4 Diagrammes de séquence

3.4.0.1 Authentification

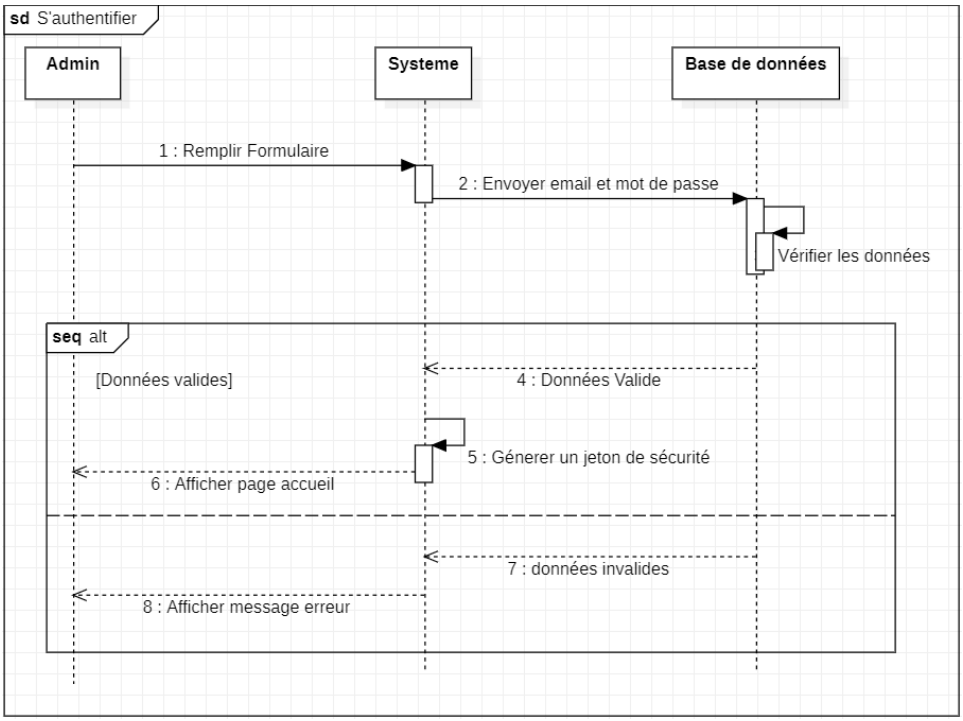


Figure 3.2 – Diagramme de séquence « Authentification »

Description du diagramme de séquence "Authentification" :

1. L'utilisateur lance le processus d'authentification en entrant son nom d'utilisateur et son mot de passe.
2. Le client envoie une demande au serveur avec les informations d'identification de l'utilisateur.
3. Le serveur reçoit la demande et vérifie les informations d'identification de l'utilisateur.
4. Si les données d'identification ne sont pas valides, le serveur renvoie un résultat invalide.
5. L'interface affiche un message d'erreur spécifique indiquant que les données d'identification sont incorrectes.
6. Le serveur vérifie l'existence du compte.
7. Si les informations d'identification sont valides et que le compte existe, le serveur renvoie un résultat valide contenant un jeton.
8. L'interface affiche le tableau de bord.
9. Si le compte n'existe pas, le serveur renvoie un résultat indiquant que le compte n'est pas valide.
10. L'interface affiche un message d'erreur spécifique indiquant que le compte n'existe pas.

3.4.0.2 Ajout de Gare

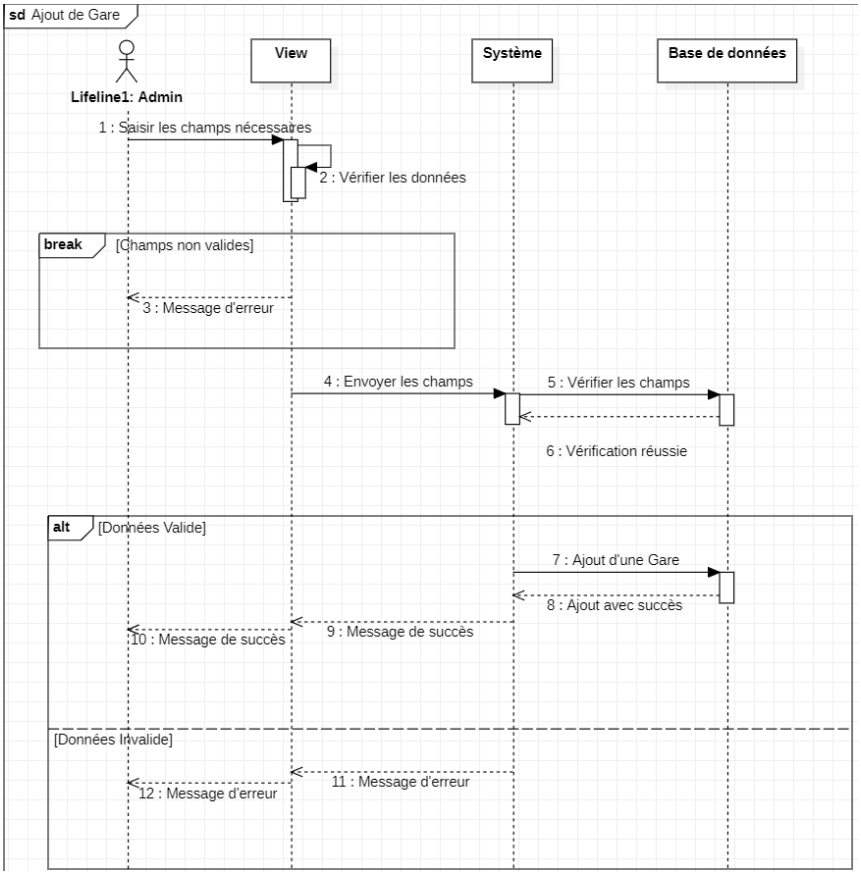


Figure 3.3 – Diagramme de séquence « Ajout de Gare »

Description du diagramme de séquence "Ajout de Gare" :

1. L'administrateur accède à l'interface de gestion des Gares.
2. L'administrateur sélectionne l'option d'ajout d'un Gare.
3. L'interface affiche un formulaire pour saisir les informations de Gare.
4. L'administrateur remplit le formulaire avec les informations de Gare.
5. L'interface envoie une demande d'ajout de Gare au contrôleur.
6. Le contrôleur valide les informations de Gare.
7. Si les informations sont invalides :
   (a) Le contrôleur envoie un message d'erreur à l'interface.
   (b) L'interface affiche un message d'erreur indiquant que les informations de Gare sont invalides.
8. Si les informations sont valides :
   (a) Le contrôleur ajoute la Gare à la base de données.
   (b) Le contrôleur envoie une confirmation d'ajout à l'interface.
   (c) L'interface affiche un message de succès indiquant que la Gare a été ajouté avec succès.

3.4.0.3 Suppression de Gare

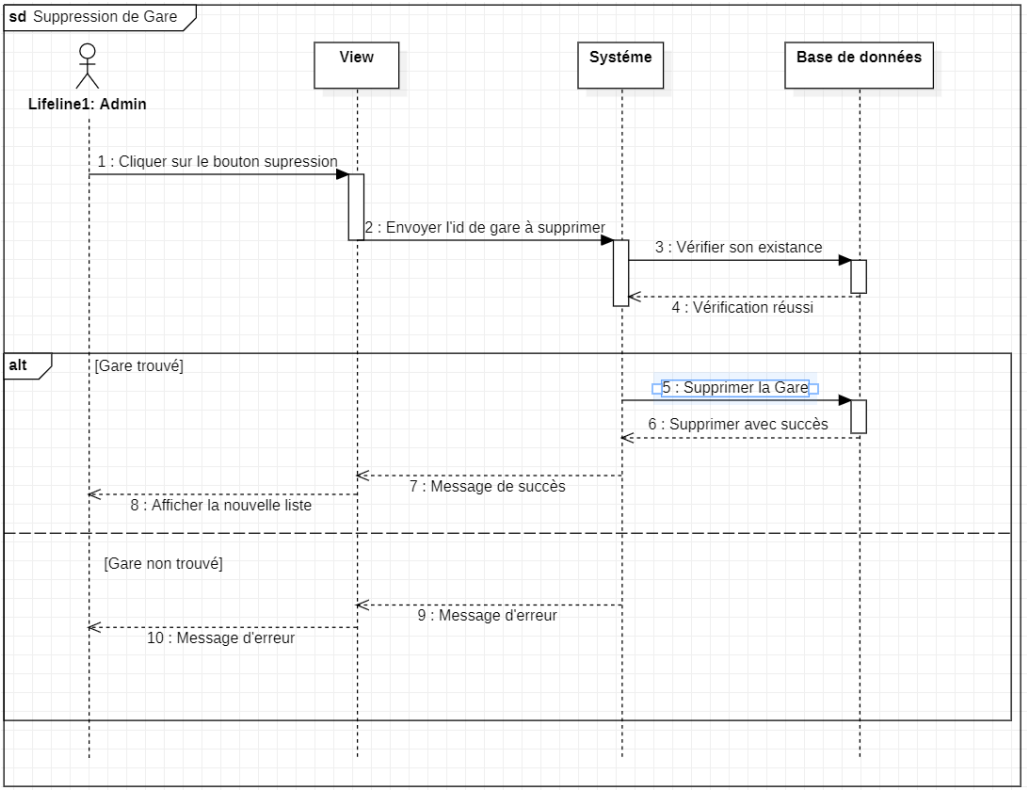


Figure 3.4 – Diagramme de séquence « Suppression de Gare »

Description du diagramme de séquence "Suppression de Gare" :

1. L'admin demande l'interface de gestion de Gares.
2. L'interface alors s'affiche.
3. L'interface d'administration envoie une demande de suppression au contrôleur.
4. Le contrôleur reçoit la demande et vérifie l'existence de Gare dans la base de données.
5. Le contrôleur supprime la Gare de la base de données.
6. Le contrôleur renvoie une confirmation de suppression à l'interface d'administration.
7. L'interface d'administration affiche un message de succès à l'administrateur.
8. Si la Gare ne confirme pas la suppression, elle s'annule.

3.4.0.4 Modification de Gare

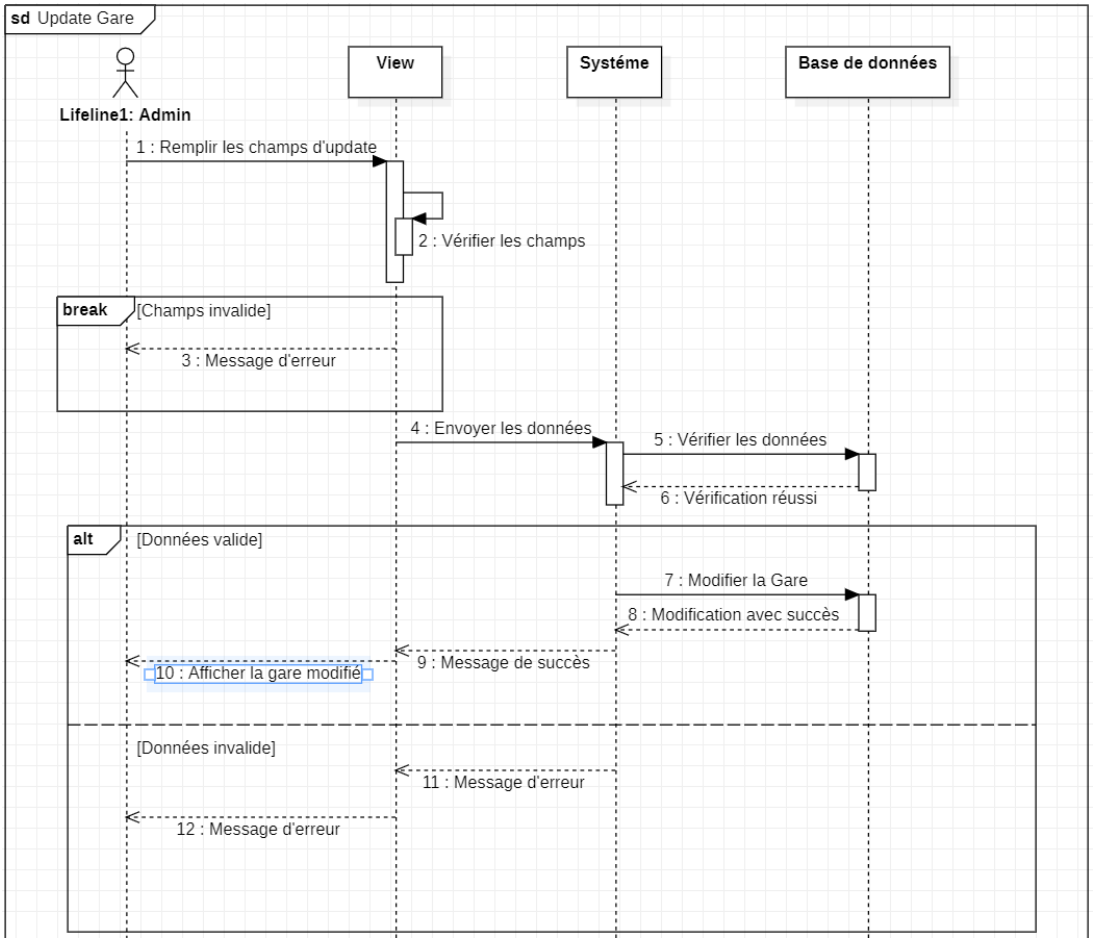


Figure 3.5 – Diagramme de séquence « Modification de Gare »

Description du diagramme de séquence "Modification de Gare" :

1. L'Admin remplis le champ de contrôle.
2. La View vérifie les champs qui sont déjà remplis par l'admin.
3. Si les champs remplis sont incorrects, un message d'erreur va apparaître et la View demande de remplir à nouveau les champs.
4. La View envoie les données au système.
5. Le système vérifie les données avec l'envoie d'une requête vers la base de données.
6. Si les données sont correctes, Le Système envoie une requête de modification vers la base de données.
7. Le produit est modifié avec succès et un message est reçu par l'admin.
8. Sinon (les données sont incorrectes) Le système envoie un message d'erreur à l'ad-min.

3.4.0.5 Ajout de Banque

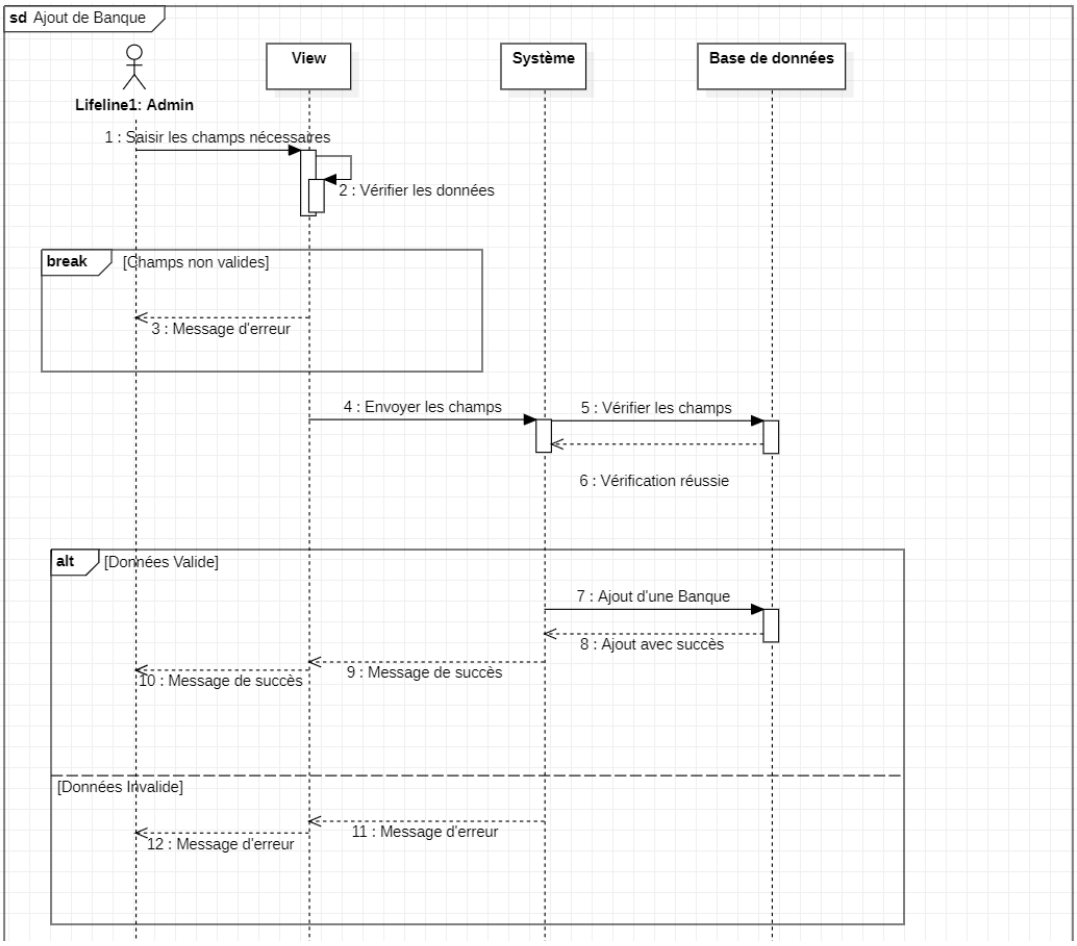


Figure 3.6 – Diagramme de séquence « Ajout de Banque »

Description du diagramme de séquence "Ajout de Banque" :

1. L'administrateur accède à l'interface de gestion des Banques.
2. L'administrateur sélectionne l'option d'ajout d'un Banque.
3. L'interface affiche un formulaire pour saisir les informations de Banque.
4. L'administrateur remplit le formulaire avec les informations de Banque.
5. L'interface envoie une demande d'ajout de Banque au contrôleur.
6. Le contrôleur valide les informations de Banque.
7. Si les informations sont invalides :
   (a) Le contrôleur envoie un message d'erreur à l'interface.
   (b) L'interface affiche un message d'erreur indiquant que les informations de Banque sont invalides.
8. Si les informations sont valides :
   (a) Le contrôleur ajoute la Banque à la base de données.
   (b) Le contrôleur envoie une confirmation d'ajout à l'interface.
   (c) L'interface affiche un message de succès indiquant que la Banque a été ajouté avec succès.

3.4.0.6 Suppression de Banque

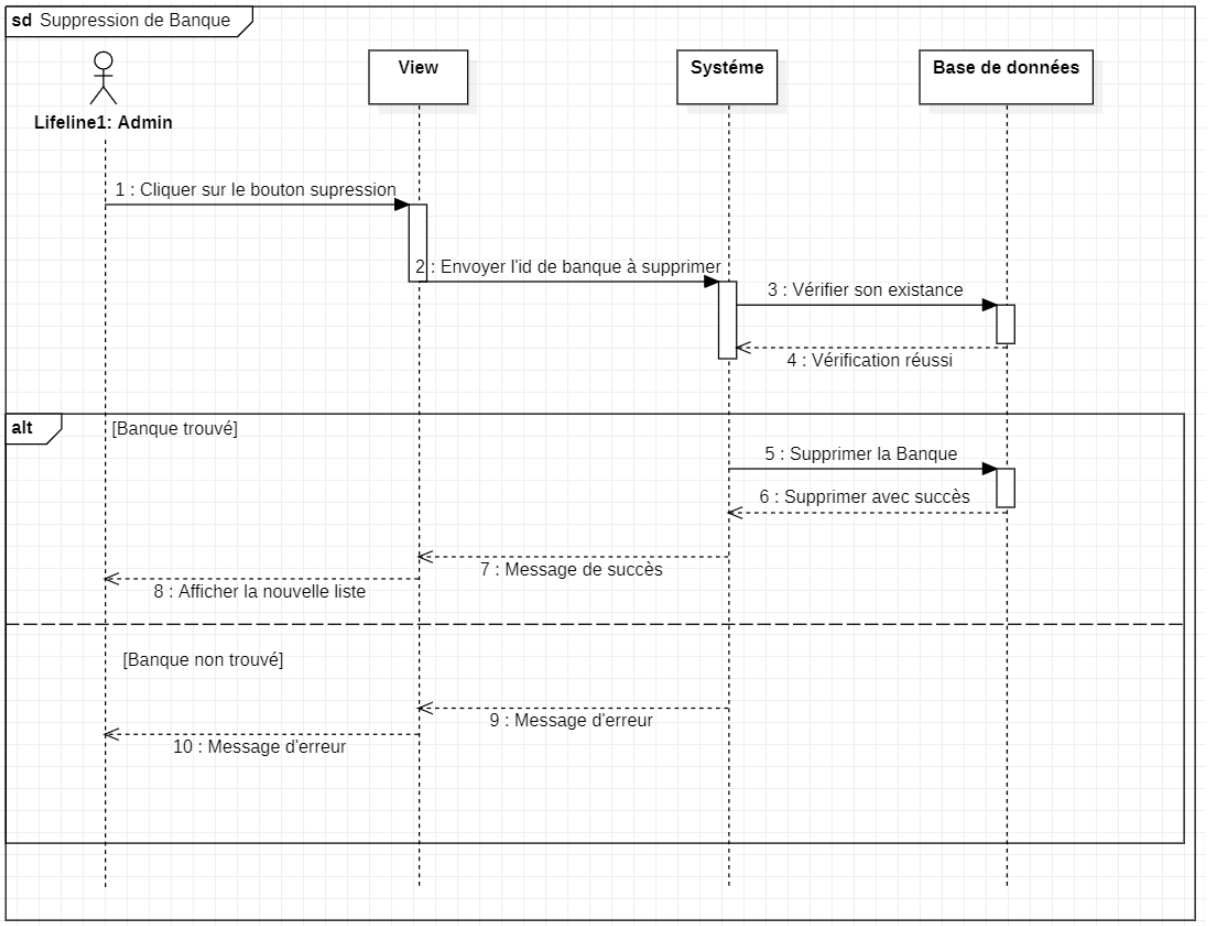


Figure 3.7 – Diagramme de séquence « Suppression de Banque »

Description du diagramme de séquence "Suppression de Banque" :

1. L'admin demande l'interface de gestion de Banques.
2. L'interface alors s'affiche.
3. L'interface d'administration envoie une demande de suppression au contrôleur.
4. Le contrôleur reçoit la demande et vérifie l'existence de Banque dans la base de données.
5. Le contrôleur supprime la Banque de la base de données.
6. Le contrôleur renvoie une confirmation de suppression à l'interface d'administration.
7. L'interface d'administration affiche un message de succès à l'administrateur.
8. Si la Banque ne confirme pas la suppression, elle s'annule.

3.4.0.7 Modification de Banque

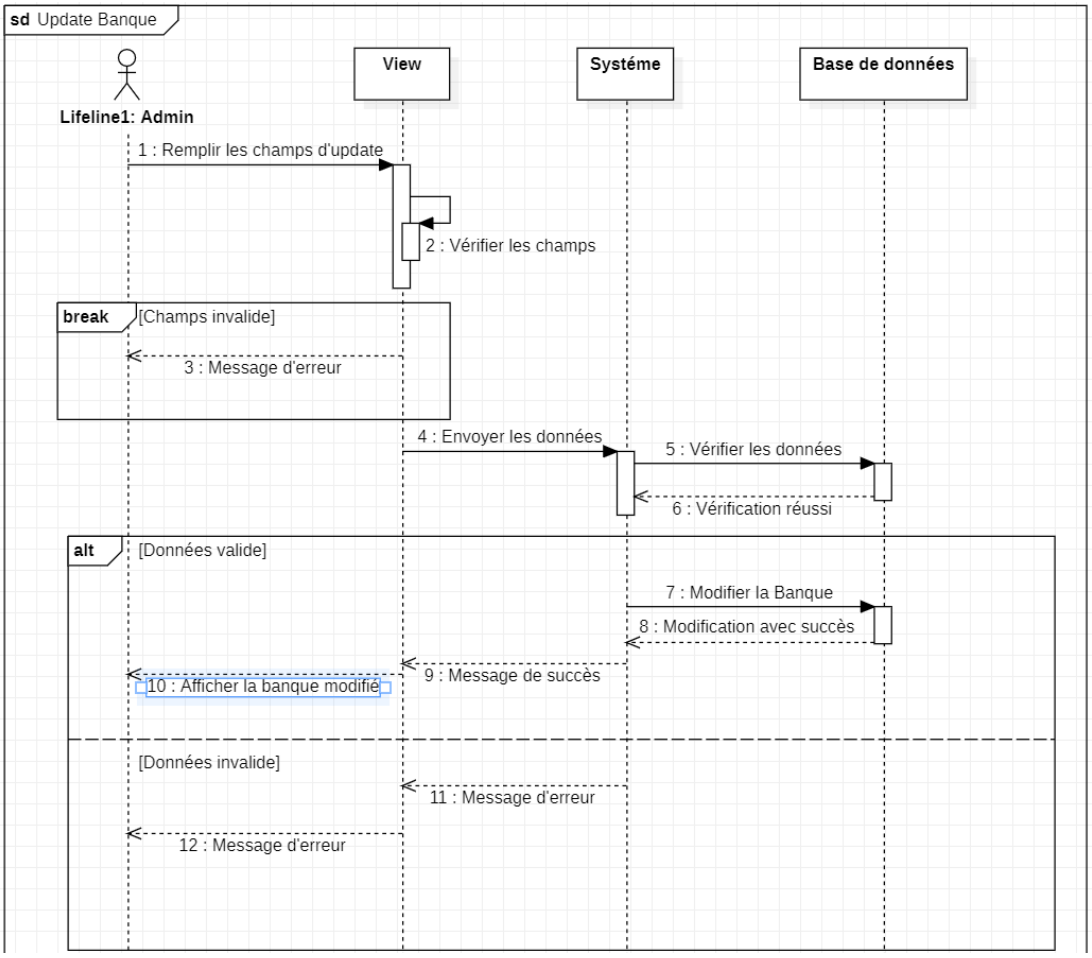


Figure 3.8 – Diagramme de séquence « Modification de Banque »

Description du diagramme de séquence "Modification de Banque" :

1. L'Admin remplis le champ de contrôle.
2. La View vérifie les champs qui sont déjà remplis par l'admin.
3. Si les champs remplis sont incorrects, un message d'erreur va apparaître et la View demande de remplir à nouveau les champs.
4. La View envoie les données au système.
5. Le Système vérifie les données avec l'envoie d'une requête vers la base de données.
6. Si les données sont correctes, Le Système envoie une requête de modification vers la base de données.
7. La Banque est modifiée avec succès et un message est reçu par l'admin.
8. Sinon (les données sont incorrectes) Le système envoie un message d'erreur à l'ad-min.

3.4.0.8 Ajout de Caisse

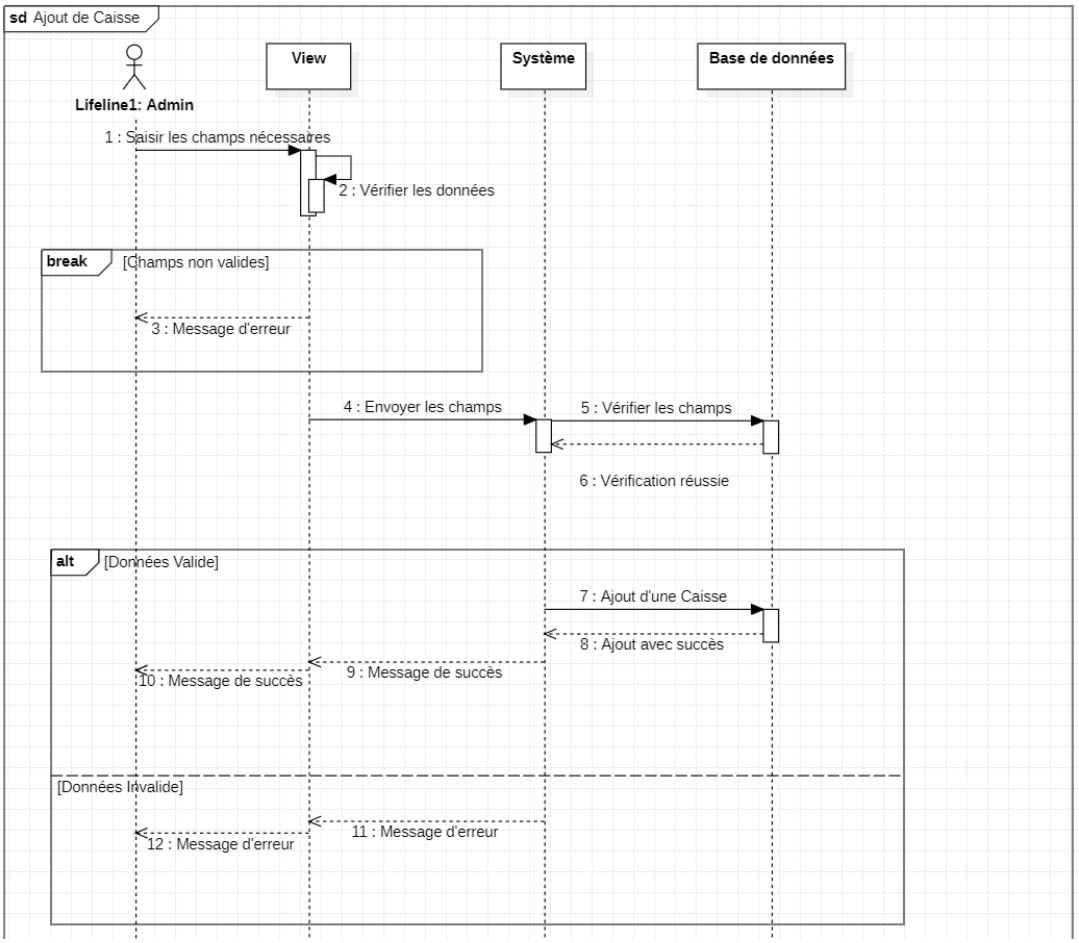


Figure 3.9 – Diagramme de séquence « Ajout de Caisse »

Description du diagramme de séquence "Ajout de Caisse" :

1. L'administrateur accède à l'interface de gestion des Caisse.
2. L'administrateur sélectionne l'option d'ajout d'une Caisse.
3. L'interface affiche un formulaire pour saisir les informations de Caisse.
4. L'administrateur remplit le formulaire avec les informations de Caisse.
5. L'interface envoie une demande d'ajout de Caisse au contrôleur.
6. Le contrôleur valide les informations de Caisse.
7. Si les informations sont invalides :
   (a) Le contrôleur envoie un message d'erreur à l'interface.
   (b) L'interface affiche un message d'erreur indiquant que les informations de Caisse sont invalides.
8. Si les informations sont valides :
   (a) Le contrôleur ajoute la Caisse à la base de données.
   (b) Le contrôleur envoie une confirmation d'ajout à l'interface.
   (c) L'interface affiche un message de succès indiquant que la Caisse a été ajouté avec succès.

3.4.0.9 Suppression de Caisse

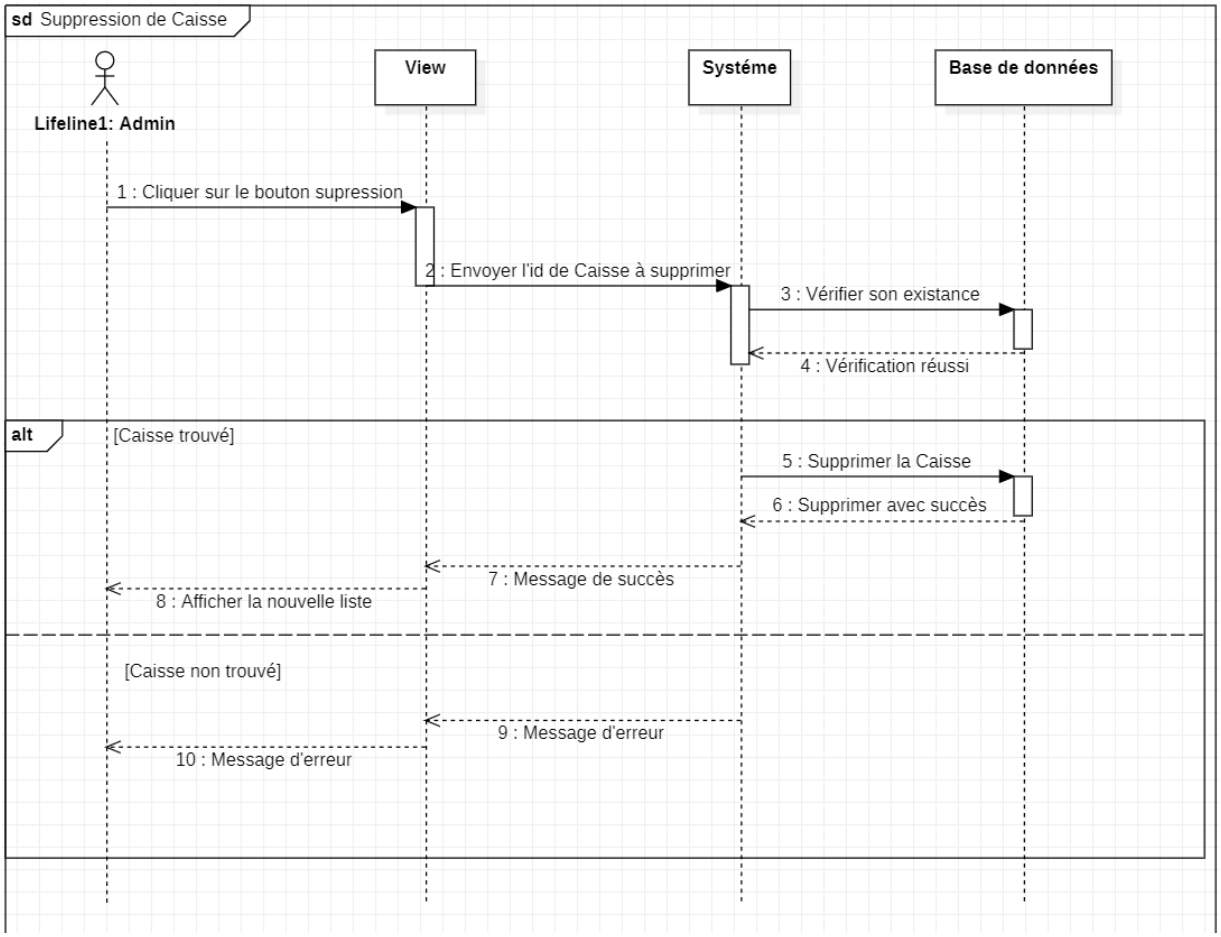


Figure 3.10 – Diagramme de séquence « Suppression de Caisse »

Description du diagramme de séquence "Suppression de Caisse" :

1. L'admin demande l'interface de gestion de Caisse.
2. L'interface alors s'affiche.
3. L'interface d'administration envoie une demande de suppression au contrôleur.
4. Le contrôleur reçoit la demande et vérifie l'existence de Caisse dans la base de données.
5. Le contrôleur supprime la Caisse de la base de données.
6. Le contrôleur renvoie une confirmation de suppression à l'interface d'administration.
7. L'interface d'administration affiche un message de succès à l'administrateur.
8. Si la Caisse ne confirme pas la suppression, elle s'annule.

3.4.0.10 Modification de Caisse

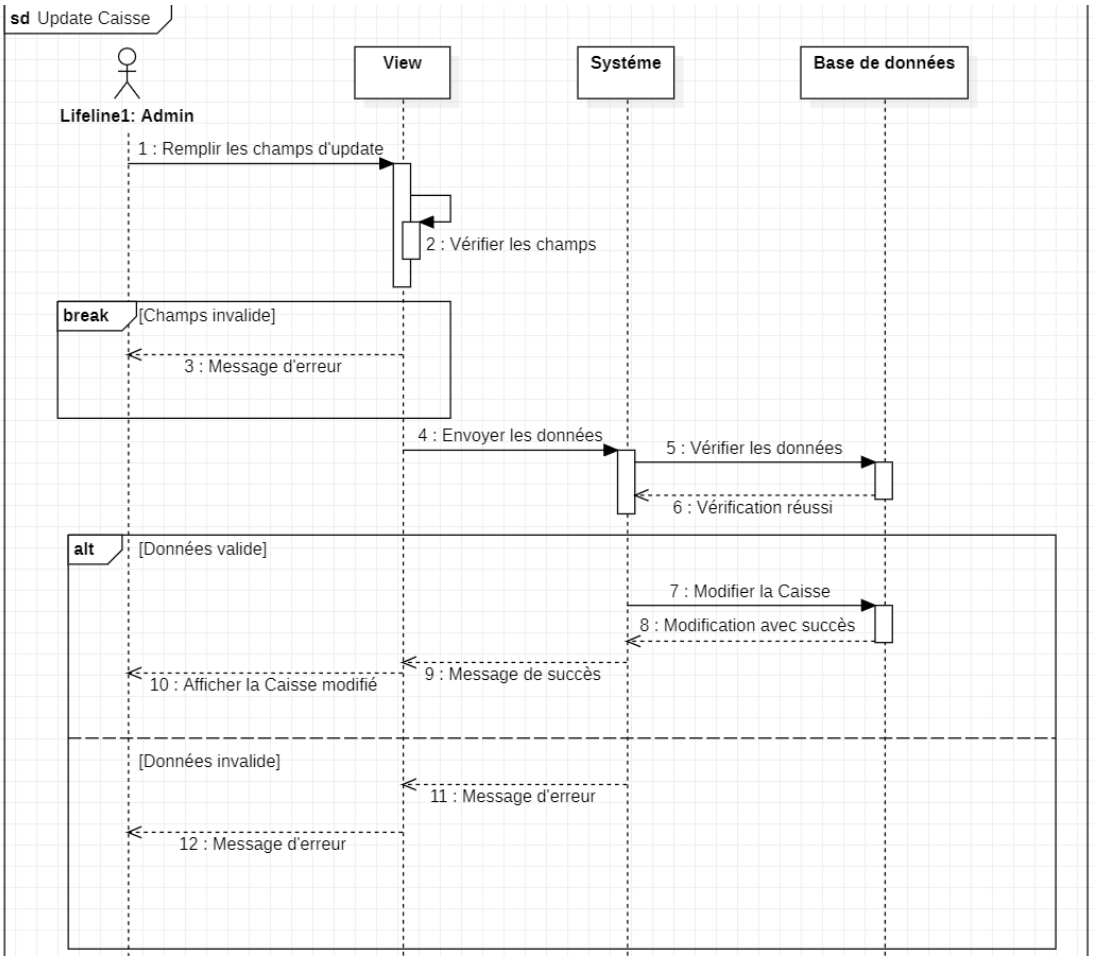


Figure 3.11 – Diagramme de séquence « Modification de Caisse »

Description du diagramme de séquence "Modification de Caisse" :

1. L'Admin remplis le champ de contrôle.
2. La View vérifie les champs qui sont déjà remplis par l'admin.
3. Si les champs remplis sont incorrects, un message d'erreur va apparaître et la View demande de remplir à nouveau les champs.
4. La View envoie les données au système.
5. Le Système vérifie les données avec l'envoie d'une requêtée vers la base de données.
6. Si les données sont correctes, Le Système envoie une requête de modification vers la base de données.
7. La Caisse est modifiée avec succès et un message est reçu par l'admin.
8. Sinon (les données sont incorrectes) Le système envoie un message d'erreur à l'ad-min.

3.4.0.11 Ajout d'Utilisateur

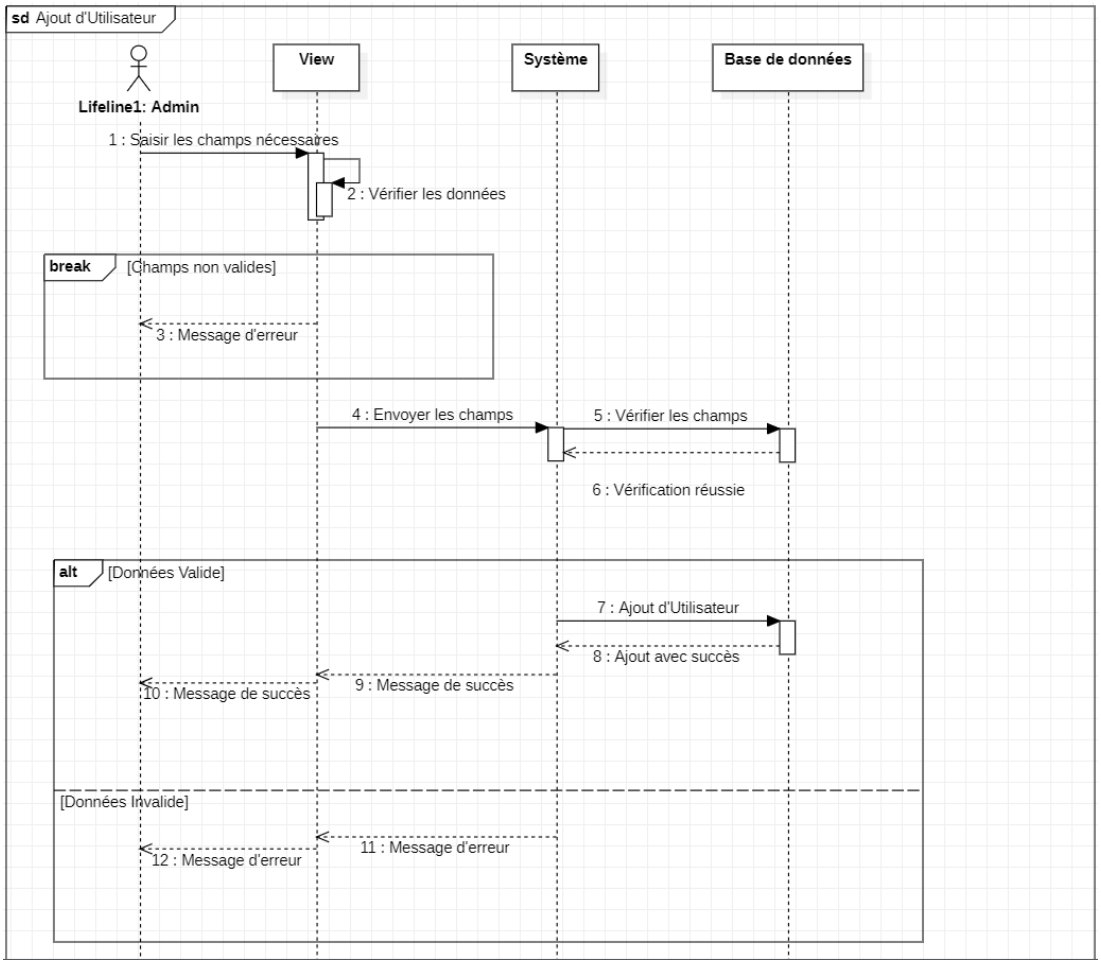


Figure 3.12 – Diagramme de séquence « Ajout d'Utilisateur »

Description du diagramme de séquence "Ajout d'Utilisateur" :

1. L'administrateur accède à l'interface de gestion des Utilisateurs.
2. L'administrateur sélectionne l'option d'ajout d'un Utilisateur.
3. L'interface affiche un formulaire pour saisir les informations d'Utilisateur.
4. L'administrateur remplit le formulaire avec les informations d'Utilisateur.
5. L'interface envoie une demande d'ajout d'Utilisateur au contrôleur.
6. Le contrôleur valide les informations d'Utilisateur.
7. Si les informations sont invalides :
   (a) Le contrôleur envoie un message d'erreur à l'interface.
   (b) L'interface affiche un message d'erreur indiquant que les informations d'Utilisateur sont invalides.
8. Si les informations sont valides :
   (a) Le contrôleur ajoute l'Utilisateur à la base de données.
   (b) Le contrôleur envoie une confirmation d'ajout à l'interface.
   (c) L'interface affiche un message de succès indiquant que l'Utilisateur a été ajouté avec succès.

3.4.0.12 Suppression d'Utilisateur

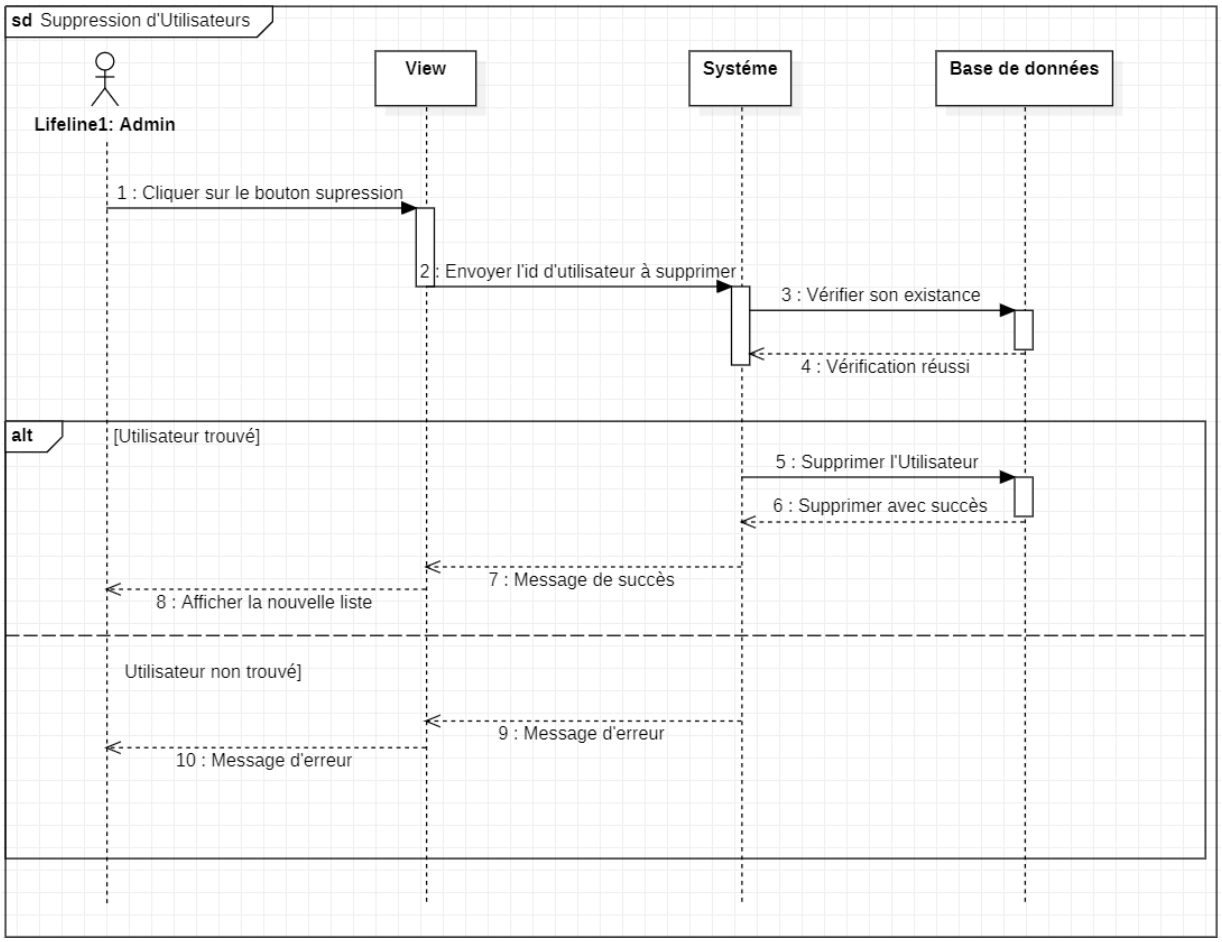


Figure 3.13 – Diagramme de séquence « Suppression d'Utilisateur »

Description du diagramme de séquence "Suppression d'Utilisateur" :

1. L'admin demande l'interface de gestion d'Utilisateur.
2. L'interface alors s'affiche.
3. L'interface d'administration envoie une demande de suppression au contrôleur.
4. Le contrôleur reçoit la demande et vérifie l'existence d'Utilisateur dans la base de données.
5. Le contrôleur supprime l'Utilisateur de la base de données.
6. Le contrôleur renvoie une confirmation de suppression à l'interface d'administration.
7. L'interface d'administration affiche un message de succès à l'administrateur.
8. Si l'Utilisateur ne confirme pas la suppression, elle s'annule.

3.4.0.13 Modification d'Utilisateur

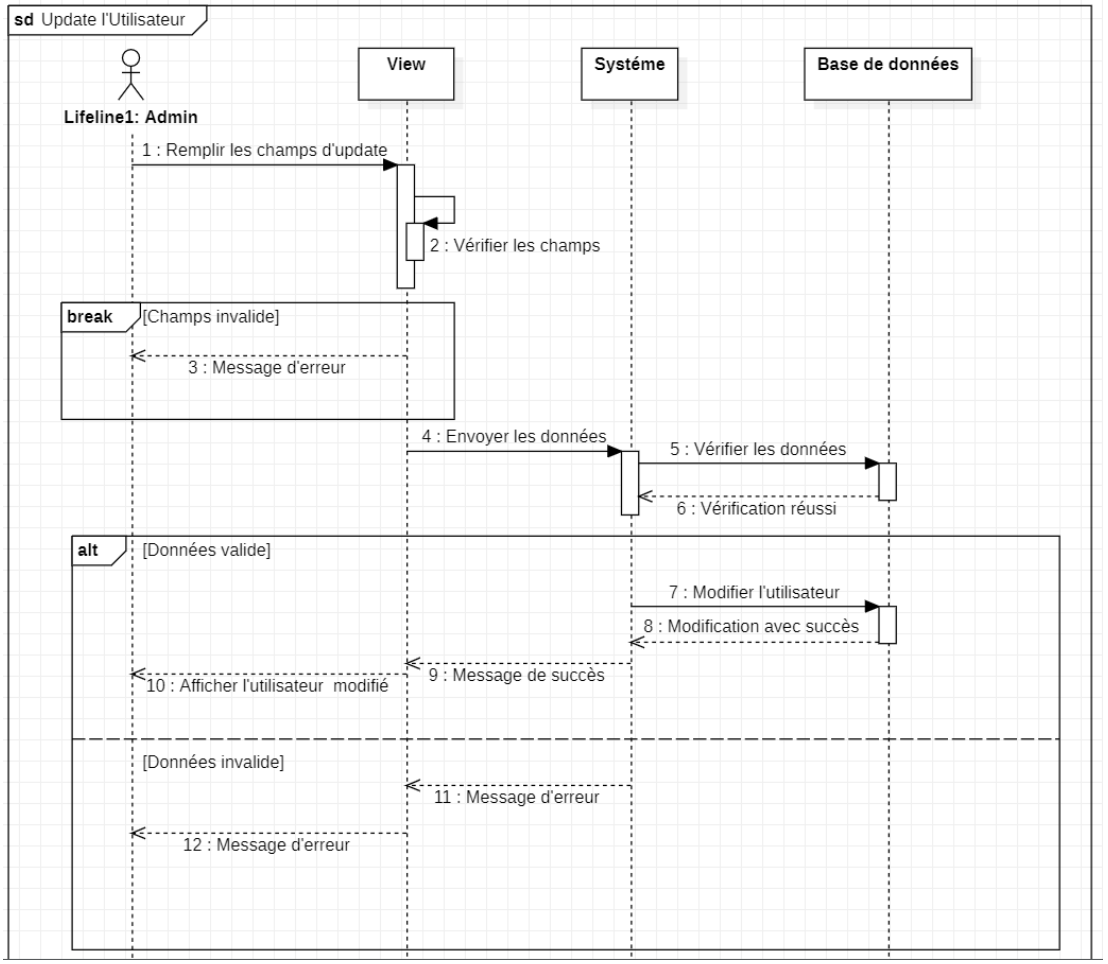


Figure 3.14 – Diagramme de séquence « Modification d'Utilisateur »

Description du diagramme de séquence "Modification d'Utilisateur" :

1. L'Admin remplis le champ de contrôle.
2. La View vérifie les champs qui sont déjà remplis par l'admin.
3. Si les champs remplis sont incorrects, un message d'erreur va apparaître et la View demande de remplir à nouveau les champs.
4. La View envoie les données au système.
5. Le Système vérifie les données avec l'envoie d'une requête vers la base de données.
6. Si les données sont correctes, Le Système envoie une requête de modification vers la base de données.
7. L'Utilisateur est modifié avec succès et un message est reçu par l'admin.
8. Sinon (les données sont incorrectes) Le système envoie un message d'erreur à l'ad-min.

3.4.0.14 Ajout de Transaction

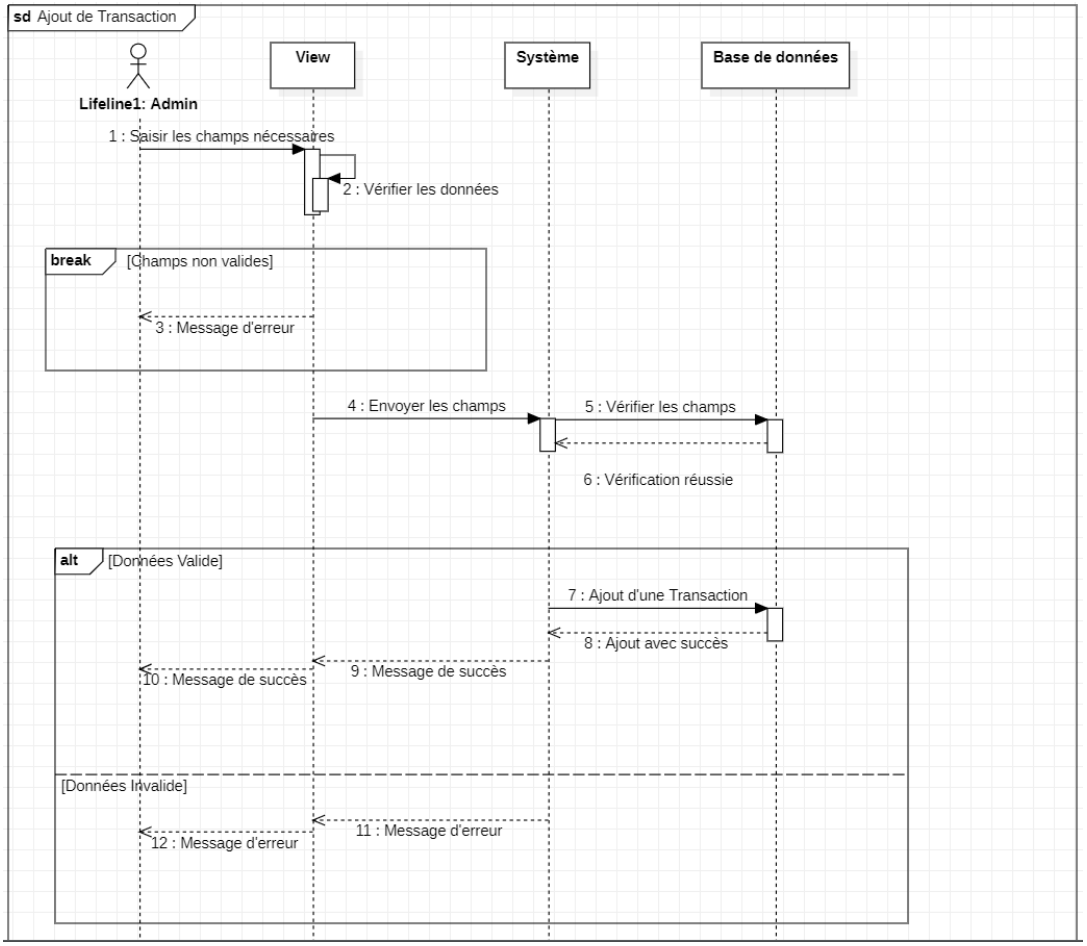


Figure 3.15 – Diagramme de séquence « Ajout de Transaction »

Description du diagramme de séquence "Ajout de Transaction" :

1. L'administrateur accède à l'interface de gestion des Transactions.
2. L'administrateur sélectionne l'option d'ajout d'une Transaction.
3. L'interface affiche un formulaire pour saisir les informations de Transaction.
4. L'administrateur remplit le formulaire avec les informations de Transaction.
5. L'interface envoie une demande d'ajout de Transaction au contrôleur.
6. Le contrôleur valide les informations de Transaction.
7. Si les informations sont invalides :
   (a) Le contrôleur envoie un message d'erreur à l'interface.
   (b) L'interface affiche un message d'erreur indiquant que les informations de Transaction sont invalides.
8. Si les informations sont valides :
   (a) Le contrôleur ajoute la Transaction à la base de données.
   (b) Le contrôleur envoie une confirmation d'ajout à l'interface.
   (c) L'interface affiche un message de succès indiquant que la Transaction a été ajouté avec succès.

3.4.0.15 Suppression de Transaction

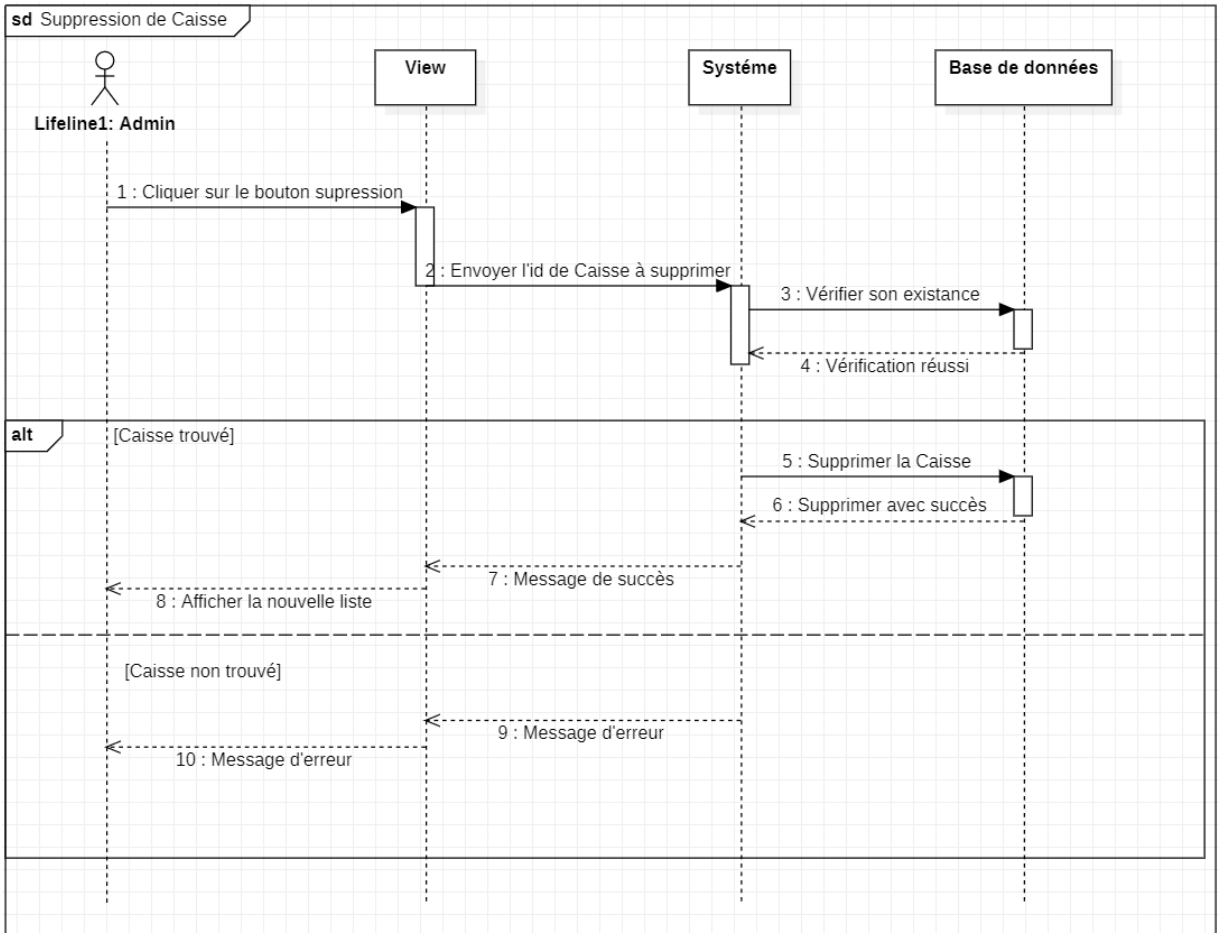


Figure 3.16 – Diagramme de séquence « Suppression de Transaction »

Description du diagramme de séquence "Suppression de Transaction" :

1. L'admin demande l'interface de gestion de Transaction.
2. L'interface alors s'affiche.
3. L'interface d'administration envoie une demande de suppression au contrôleur.
4. Le contrôleur reçoit la demande et vérifie l'existence de Transaction dans la base de données.
5. Le contrôleur supprime la Transaction de la base de données.
6. Le contrôleur renvoie une confirmation de suppression à l'interface d'administration.
7. L'interface d'administration affiche un message de succès à l'administrateur.
8. Si la Transaction ne confirme pas la suppression, elle s'annule.

3.4.0.16 Modification de Transaction

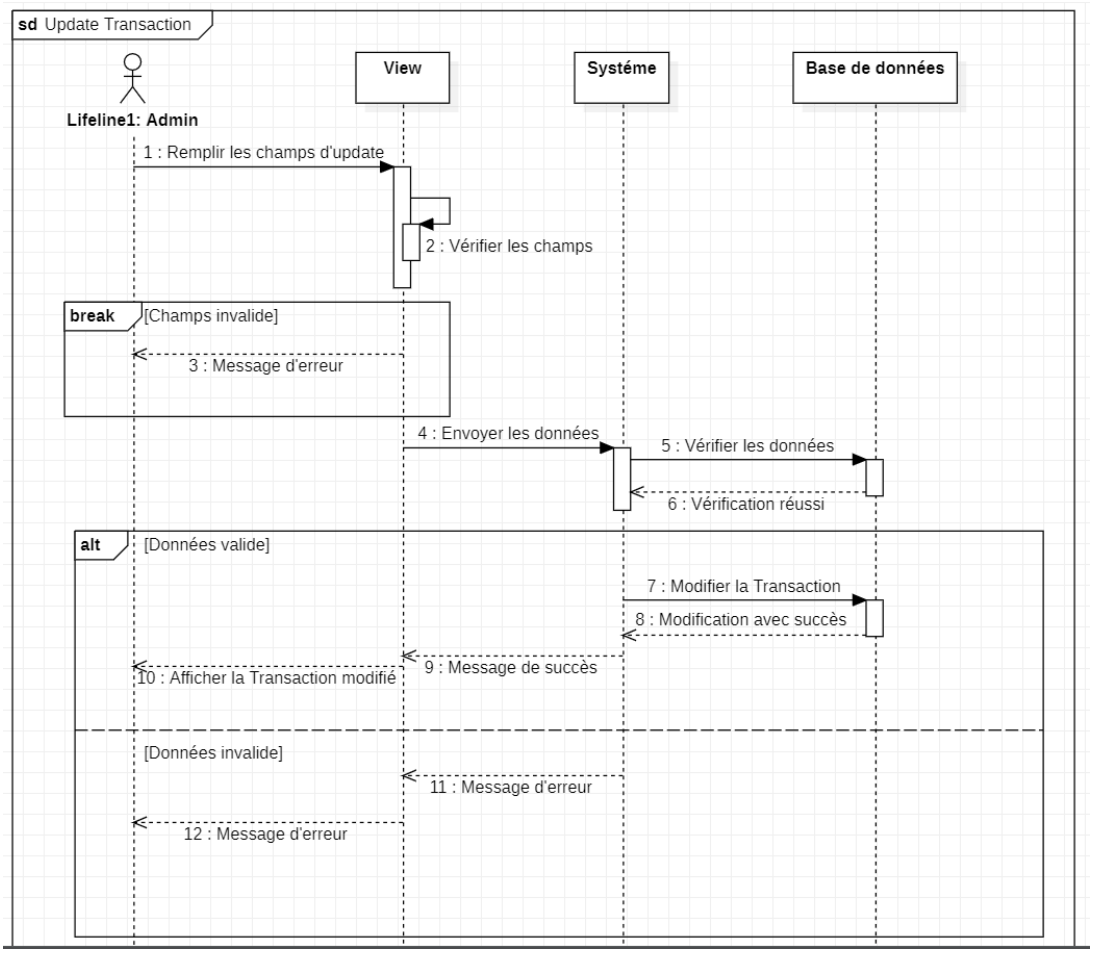


Figure 3.17 – Diagramme de séquence « Modification de Transaction »

Description du diagramme de séquence "Modification de Transaction" :

1. L'Admin remplis le champ de contrôle.
2. La View vérifie les champs qui sont déjà remplis par l'admin.
3. Si les champs remplis sont incorrects, un message d'erreur va apparaître et la View demande de remplir à nouveau les champs.
4. La View envoie les données au système.
5. Le Système vérifie les données avec l'envoie d'une requête vers la base de données.
6. Si les données sont correctes, Le Système envoie une requête de modification vers la base de données.
7. La Transaction est modifiée avec succès et un message est reçu par l'admin.
8. Sinon (les données sont incorrectes) Le système envoie un message d'erreur à l'ad-min.

3.4.0.17 Ajout d'Opération

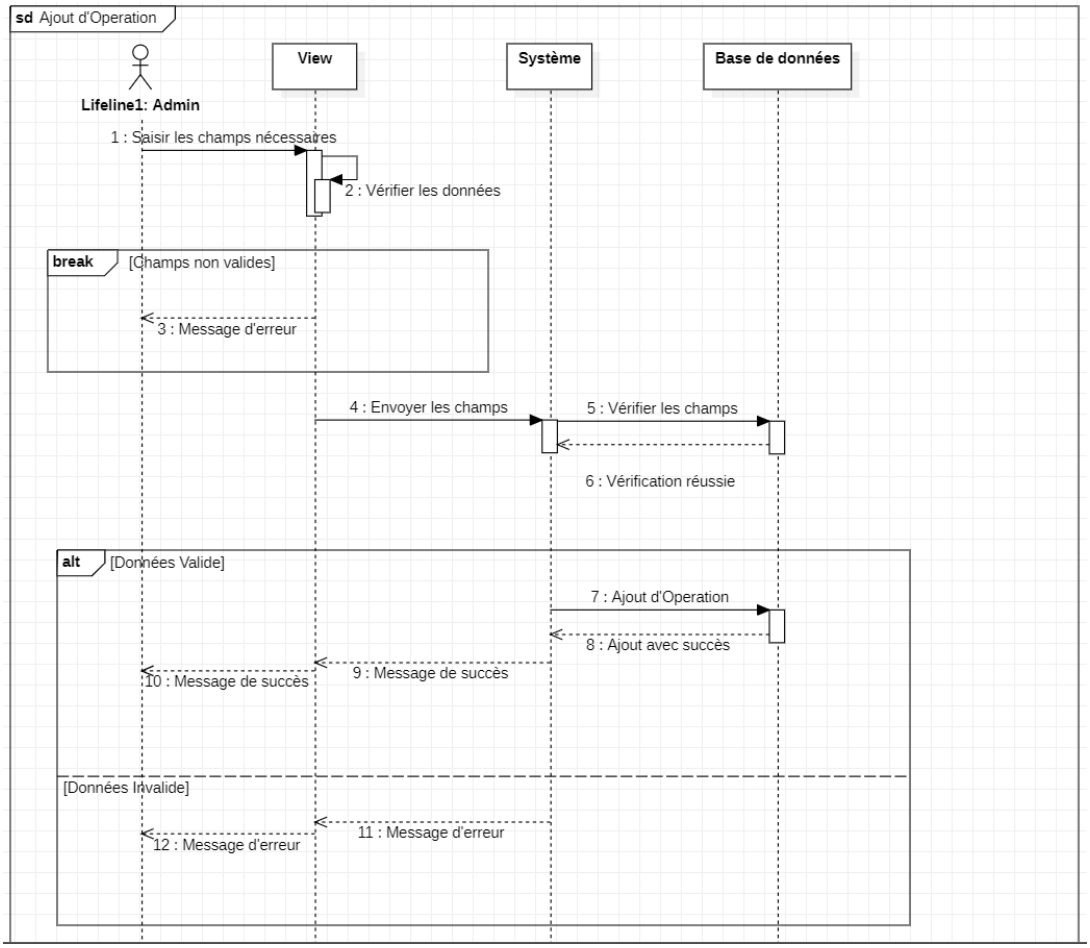


Figure 3.18 – Diagramme de séquence « Ajout d'Opération »

Description du diagramme de séquence "Ajout d'Opération" :

1. L'administrateur accède à l'interface de gestion des Operations.
2. L'administrateur sélectionne l'option d'ajout d'une Opération.
3. L'interface affiche un formulaire pour saisir les informations d'Opération.
4. L'administrateur remplit le formulaire avec les informations d'Opération.
5. L'interface envoie une demande d'ajout d'Opération au contrôleur.
6. Le contrôleur valide les informations d'Opération.
7. Si les informations sont invalides :
   (a) Le contrôleur envoie un message d'erreur à l'interface.
   (b) L'interface affiche un message d'erreur indiquant que les informations d'Operation sont invalides.
8. Si les informations sont valides :
   (a) Le contrôleur ajoute l'Opération à la base de données.
   (b) Le contrôleur envoie une confirmation d'ajout à l'interface.
   (c) L'interface affiche un message de succès indiquant que l'Opération a été ajouté avec succès.

3.4.0.18 Suppression d'Opération

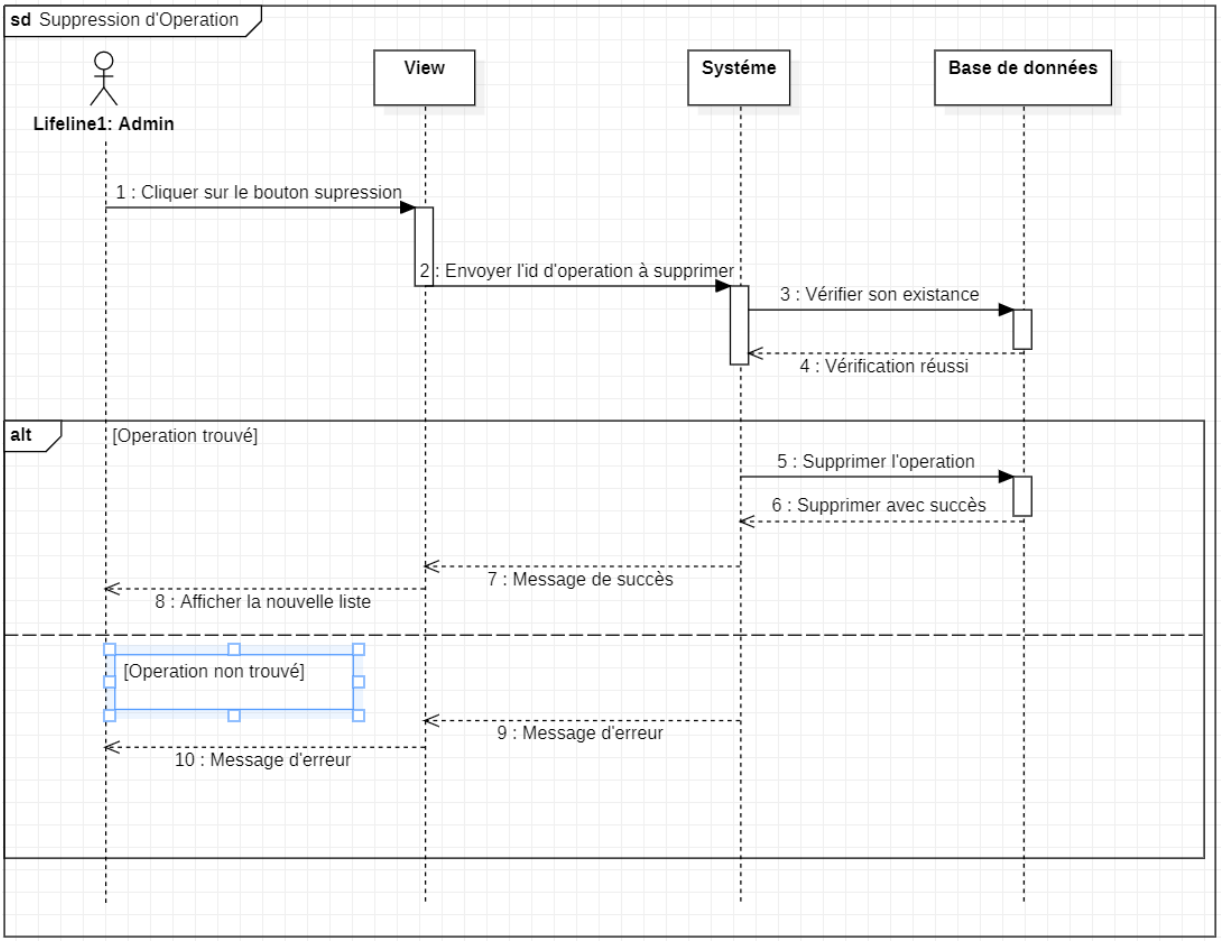


Figure 3.19 – Diagramme de séquence « Suppression d'Opération »

Description du diagramme de séquence "Suppression d'Opération" :

1. L'admin demande l'interface de gestion d'Opération.
2. L'interface alors s'affiche.
3. L'interface d'administration envoie une demande de suppression au contrôleur.
4. Le contrôleur reçoit la demande et vérifie l'existence d'Opération dans la base de données.
5. Le contrôleur supprime l'Opération de la base de données.
6. Le contrôleur renvoie une confirmation de suppression à l'interface d'administration.
7. L'interface d'administration affiche un message de succès à l'administrateur.
8. Si l'Opération ne confirme pas la suppression, elle s'annule.

3.4.0.19 Modification d'Opération

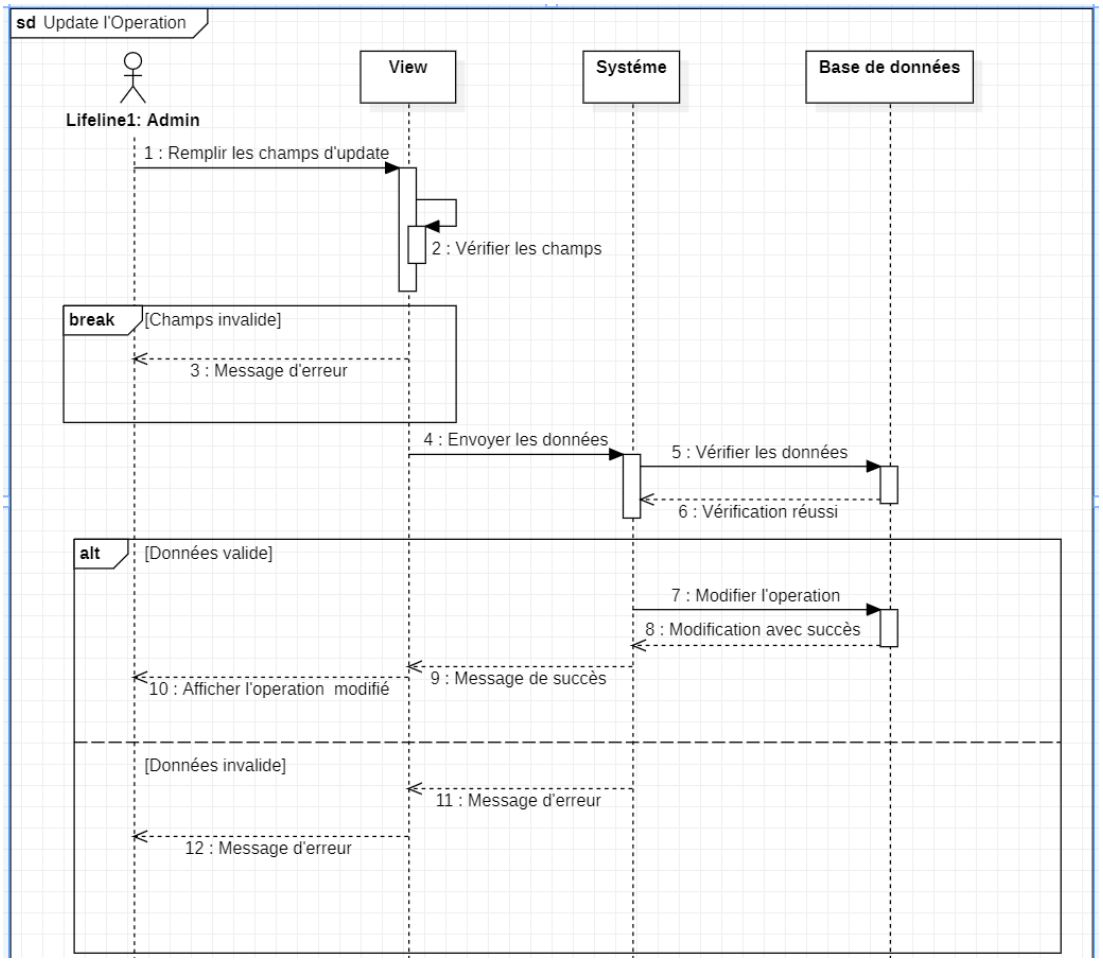


Figure 3.20 – Diagramme de séquence « Modification d'Opération »

Description du diagramme de séquence "Modification d'Opération" :

1. L'Admin remplis le champ de contrôle.
2. La View vérifie les champs qui sont déjà remplis par l'admin.
3. Si les champs remplis sont incorrects, un message d'erreur va apparaître et la View demande de remplir à nouveau les champs.
4. La View envoie les données au système.
5. Le Système vérifie les données avec l'envoie d'une requête vers la base de données.
6. Si les données sont correctes, Le Système envoie une requête de modification vers la base de données.
7. L'Opération est modifiée avec succès et un message est reçu par l'admin sinon (les données sont incorrectes) Le système envoie un message d'erreur au admin.

3.5 Conclusion

En conclusion, ce chapitre a présenté l'architecture générale, le diagramme de classes et les diagrammes de séquence. Ces éléments ont permis de comprendre la structure de l'application, d'organiser les entités et de représenter les interactions. Ils constitueront une base solide pour la réalisation de la solution dans le prochain chapitre.

Chapitre 4

Réalisation

4.1 Introduction

Dans ce chapitre de réalisation, nous passerons de la conception à l'implémentation concrète du système. Nous développerons les fonctionnalités spécifiées en utilisant les modélais et les concepts définis précédemment. L'objectif est de créer un système fonctionnel répondant aux exigences identifiées.

4.2 Environnement de développement

L'environnement de développement se décompose en deux parties : une partie matérielle et une partie logicielle.

4.2.1 Environnement matériel :

4.2.1.1 Nos ordinateurs :

Propriétaire	**Malek Ben Slama**	**Yassine Hosni**
Pc Portable	DELL	MSI
Processeur	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz	Intel I7 10eme
RAM	12GO	16GO
Stockage	512GO (SSD)	512GO (SSD) 1TB (HDD)
Carte graphique	Nvidia GTX 1660Ti	Nvidia GTX 1650Ti
OS	Windows 11 Home	Windows 10

TABLe 4.1 – Nos ordinateurs

4.2.2 Environnement logiciel

Après avoir présenté l'environnement matériel mis à notre disposition dans le cadre de réalisation de notre solution, nous abordons dans cette partie l'ensemble des outils de conception, les technologies utilisées, l'environnement de développement ainsi que les outils de tests.

4.2.2.1 Outils de conception

- StarUML [2] UML, ou Unified Modeling Language, est un langage graphique utilisé pour concevoir, représenter et spécifier visuellement les systèmes logiciels. Il offre des notations standardisées pour décrire la structure, le comportement, et les interactions entre les composants d'une application. UML est largement utilisé dans l'industrie du logiciel pour faciliter la conception et la communication entre les membres d'une équipe de développement.

Figure 4.1 – StarUML

4.2.2.2 Outils de gestion de projet

- Git [3] est un système de contrôle de version fondamental pour les développeurs. Il permet de suivre les changements, collaborer et revenir en arrière en cas d'erreur. C'est un outil incontournable pour gérer des projets de toutes tailles, de petites applications à des entreprises majeures, grâce à sa vitesse et sa fiabilité.

Figure 4.2 – Logo Git

4.2.2.3 Outils de test

- Postman [4] est un outil qui permet de tester les API que nous avons crée. Iljoue le rôle d'un client qui consomme les API REST et effectue des requêtes HTTP.

Figure 4.3 – Logo Postman

4.2.2.4 Technologies utilisées

- Angular [5] est un Framework de développement en frontend coté client basé sur le langage TypeScript et sur le pattern MVVM qui est le descendant de MVC. Il est robuste et il permet le développement des interfaces client à page unique.

Figure 4.4 – Logo Angular

- Angular Material [6] est une bibliothèque de composants d'interface utilisateur (UI) que les développeurs peuvent utiliser dans leurs projets Angular pour accélérer le développement d'interfaces utilisateur élégantes et cohérentes.

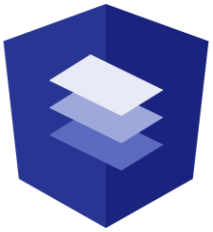


Figure 4.5 – Logo Angular Material

- Spring Boot [7] Spring Boot est un Framework de développement d'applications Java qui simplifie considérablement le processus de création d'applications robustes et évolutives. Il offre une approche convention-over-configuration, ce qui signifie que de nombreuses configurations par défaut sont fournies, permettant aux développeurs de se concentrer davantage sur le développement de fonctionnalités.

Figure 4.6 – Logo Spring Boot

4.2.2.5 Environnement de développement

- Visual Studio Code [8] est un éditeur de code qui supporte divers langages de programmation et qui contient plusieurs fonctionnalités lui permettant d'être extrêmement facile et intéressant à utiliser. On l'a utilisé pour développer en Angular et ExpressJS.

Figure 4.7 – Logo VSCode

- IntelliJ IDEA [9] un environnement de développement intégré (IDE) développé par JetBrains, destiné principalement à la programmation en Java. Il offre un large éventail de fonctionnalités et d'outils pour aider les développeurs à créer, déboguer et gérer des applications Java de manière efficace.

Figure 4.8 – Logo IntelliJ IDEA

4.2.2.6 Environnement de développement de base de données

- MySQL [10] est un système de gestion de base de données relationnelle open-source largement utilisé. Il a été initialement développé par MySQL AB, qui a été acquise par Oracle Corporation. MySQL est populaire en raison de sa simplicité, de sa stabilité et de sa performance, ce qui en fait un choix courant pour de nombreuses applications, notamment les sites web, les applications d'entreprise et les projets open source.

Figure 4.9 – Logo MySQL

4.2.3 Choix des langages de programmation

Java [11] est un langage de programmation polyvalent et orienté objet qui a été développé par Sun Microsystems (acquise par Oracle Corporation) dans les années 1990.

Il est célèbre pour sa portabilité, ce qui signifie que les programmes Java peuvent être exécutés sur différentes plateformes, telles que Windows, MacOs et divers systèmes Unix, sans nécessiter de modifications significatives.

Figure 4.10 – Logo Java

TypeScript [12] est un langage de programmation libre et open source développée par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript.

Figure 4.11 – Logo TypeScript

HTML [13] est un langage de balisage. Il est utilisé pour structurer et organiser le contenu d'une page web en utilisant des balises (éléments) pour décrire les différents types de contenu comme les titres, les paragraphes, les listes, les images, les liens, etc.

Figure 4.12 – Logo HTML

CSS [14] est un langage de style. Il est utilisé pour définir l'apparence et la mise en page des éléments HTML. CSS permet de spécifier des propriétés telles que la couleur, la taille, la police, la position, etc., pour créer un design visuellement attrayant.

Figure 4.13 – Logo CSS

4.2.4 ChatGPT

ChatGPT [15] est un prototype d'agent conversationnel utilisant l'intelligence artificielle, développé par OpenAI et spécialisé dans le dialogue.

Nous avons utilisé ChatGPT, un puissant modèle de langage développé par OpenAI, pour nous aider dans la rédaction de notre rapport et pour obtenir des exemples de code fonctionnel pour le développement de notre application web.

Figure 4.14 – Logo ChatGPT

4.2.5 Bootstrap

Bootstrap [16] est un Framework de développement web qui offre des outils et des styles prédéfinis pour créer rapidement des sites web modernes et réactifs. Il utilise du HTML, CSS et du JavaScript pour faciliter la conception et la mise en page des pages web. Bootstrap est largement utilisé pour sa simplicité d'utilisation et sa compatibilité avec différents navigateurs.

Figure 4.15 – Logo Bootstrap

4.3 Présentation de l'application

Figure 4.16 – Page de démarrage

La Figure 4.16 présente l'interface de la page de démarrage. Cette interface accueille l'utilisateur et lui donne la possibilité de connecter en tant qu'admin.

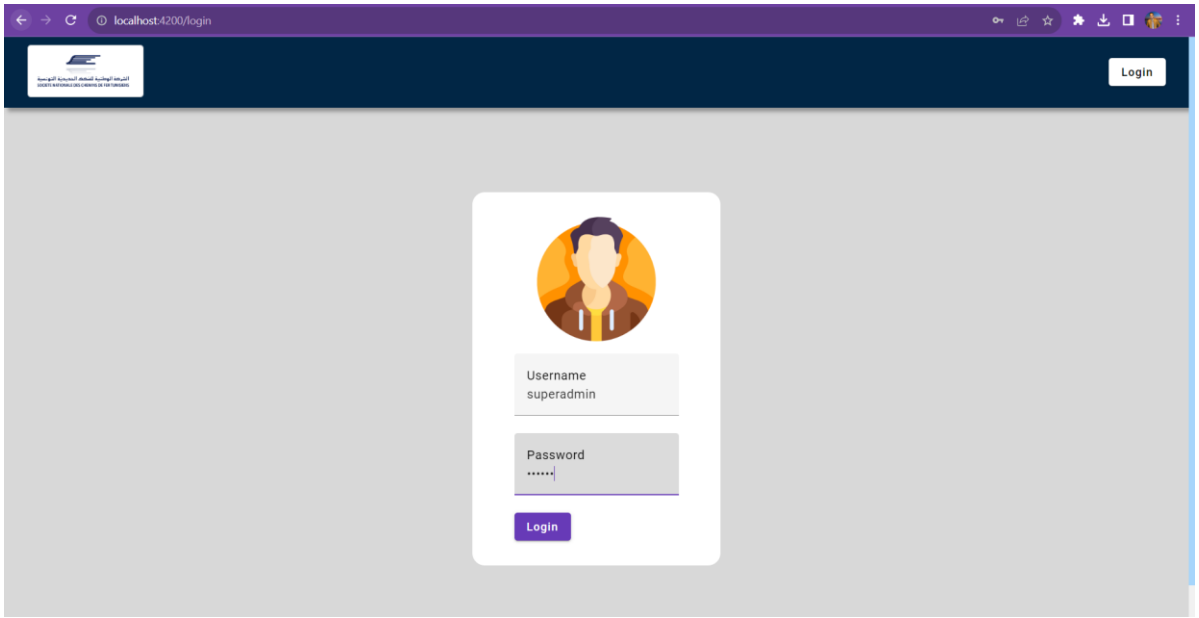


Figure 4.17 – Interface Login

La Figure 4.17 présente l'interface Login. Cette interface est utilisée pour s'authentifier et accéder à l'application.

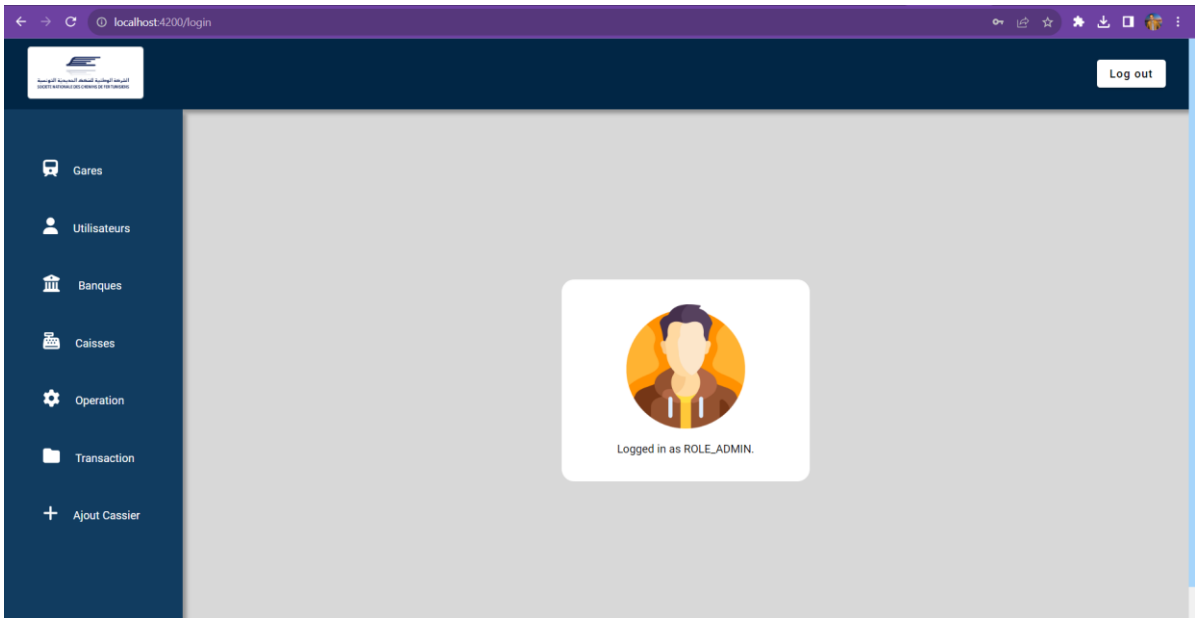


Figure 4.18 – Interface de Login

La Figure 4.17 présente l'interface Login. Cette interface lors l'accès à l'application est accordé.

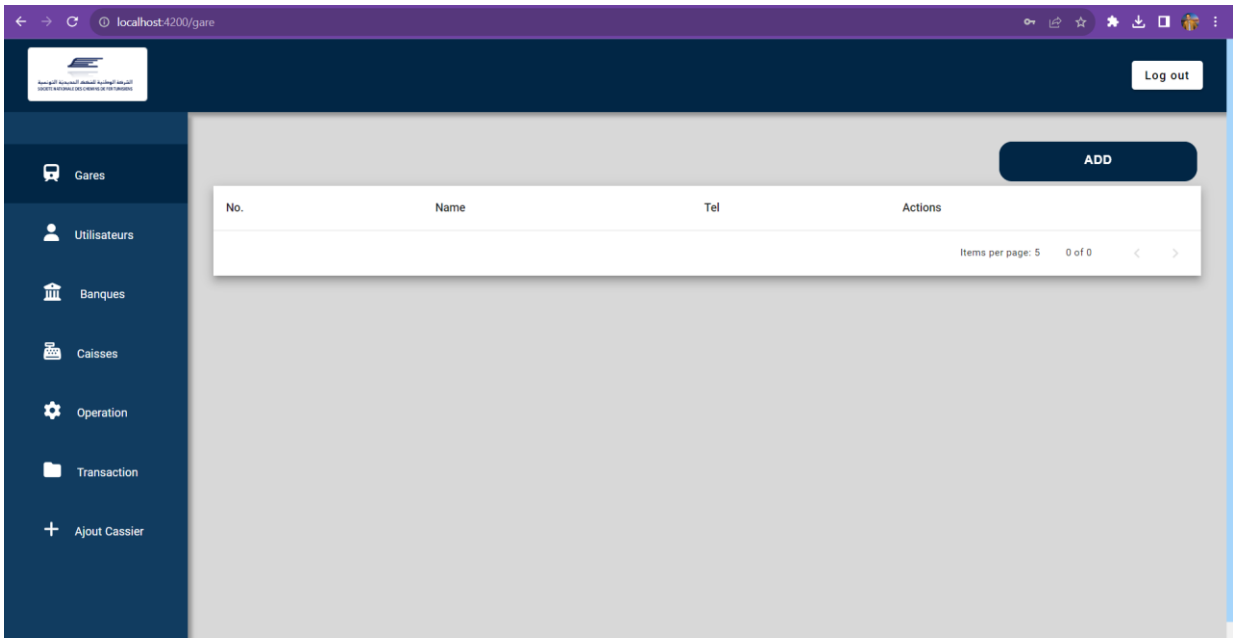


Figure 4.19 – Interface de Gare

La Figure 4.19 présente l'interface du Gare. Cette interface est utilisée pour consulter les informations de Gares.

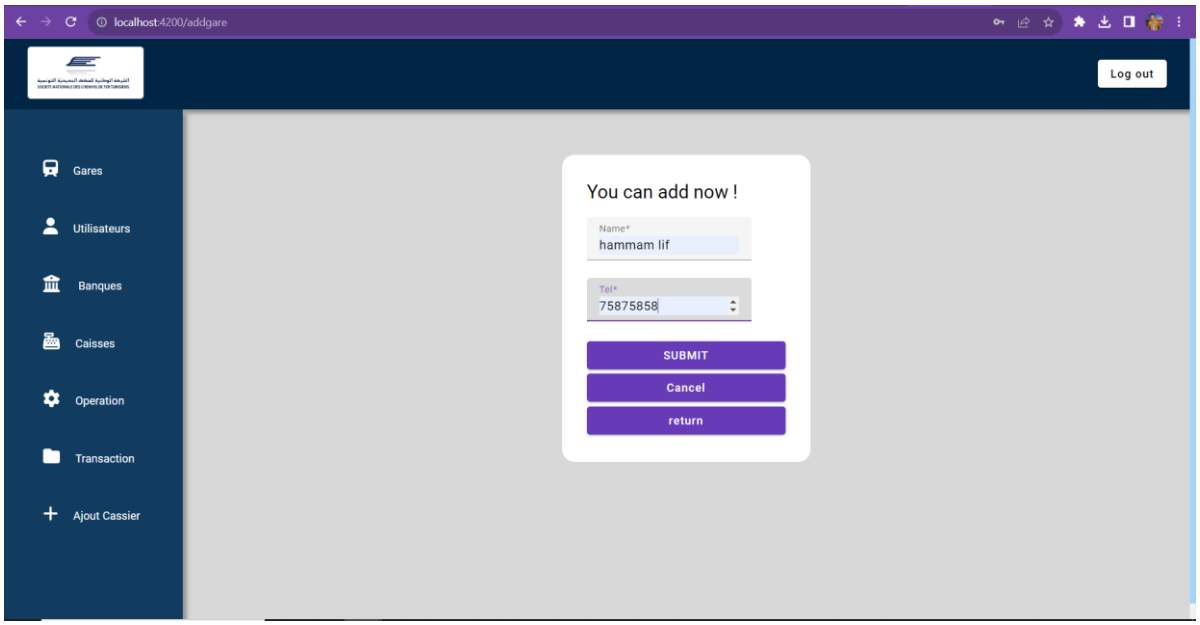


Figure 4.20 – Interface Add Gare

La Figure 4.20 montre l'interface Add Gare. Cette interface affiche comment on add une gare.

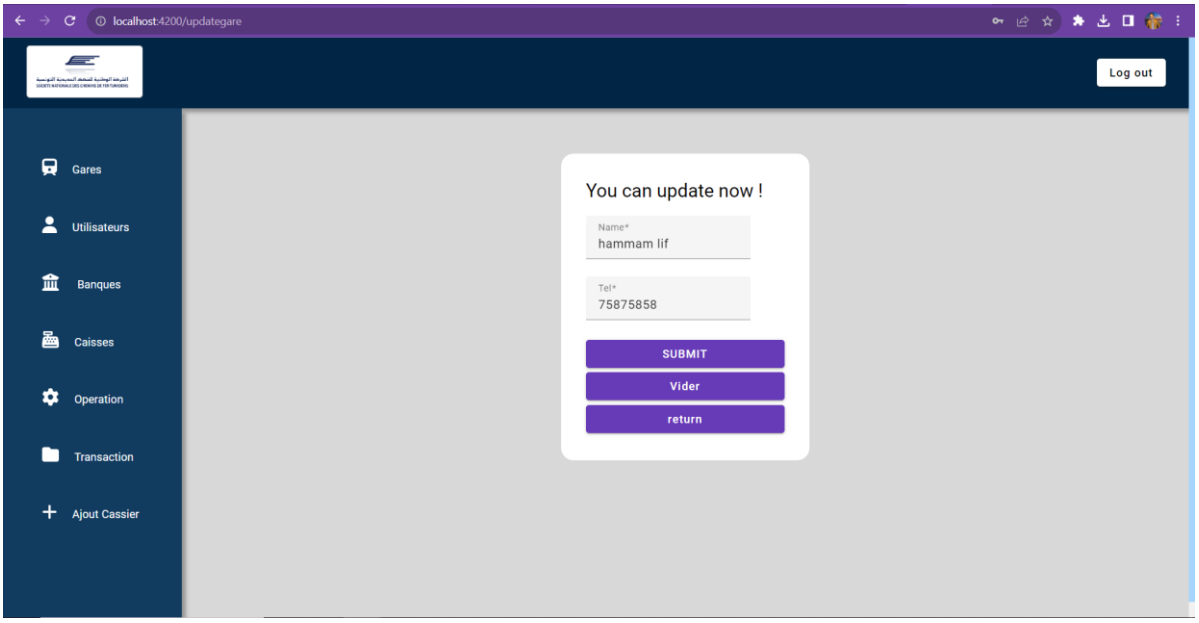


Figure 4.21 – Interface Update Gare

La Figure 4.21 montre l'interface Update Gare. Cette interface affiche comment on update une gare.

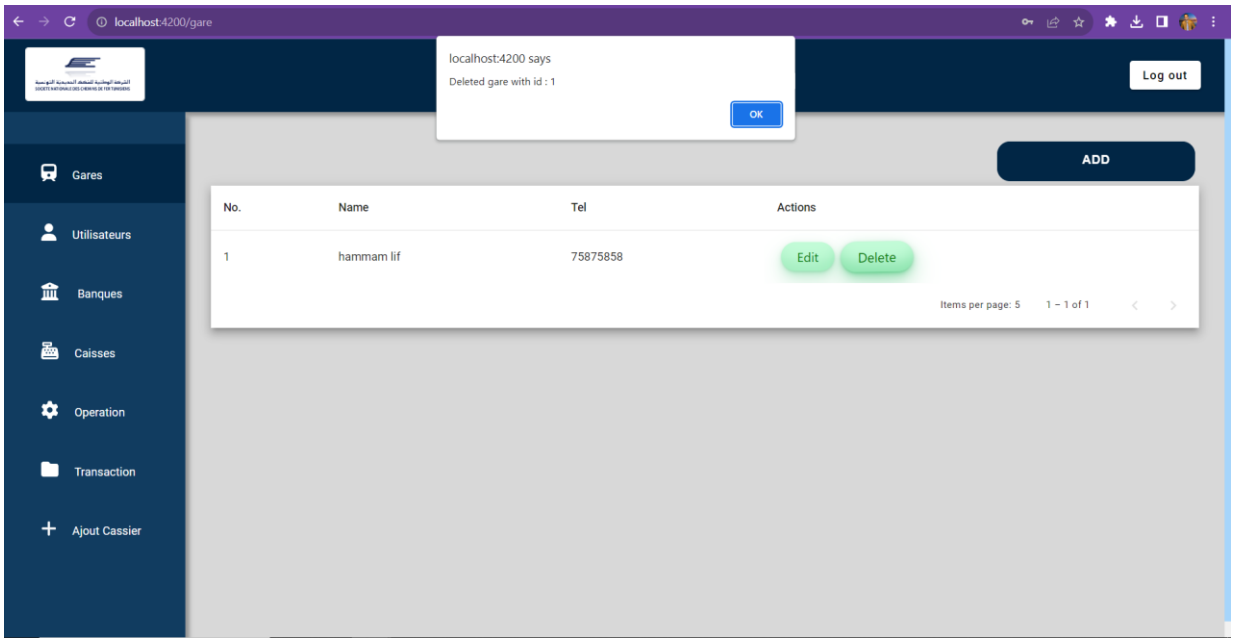


Figure 4.22 – Delete Gare

La Figure 4.22 montre la suppression de Gare. S'affiche comment on supprime une gare.

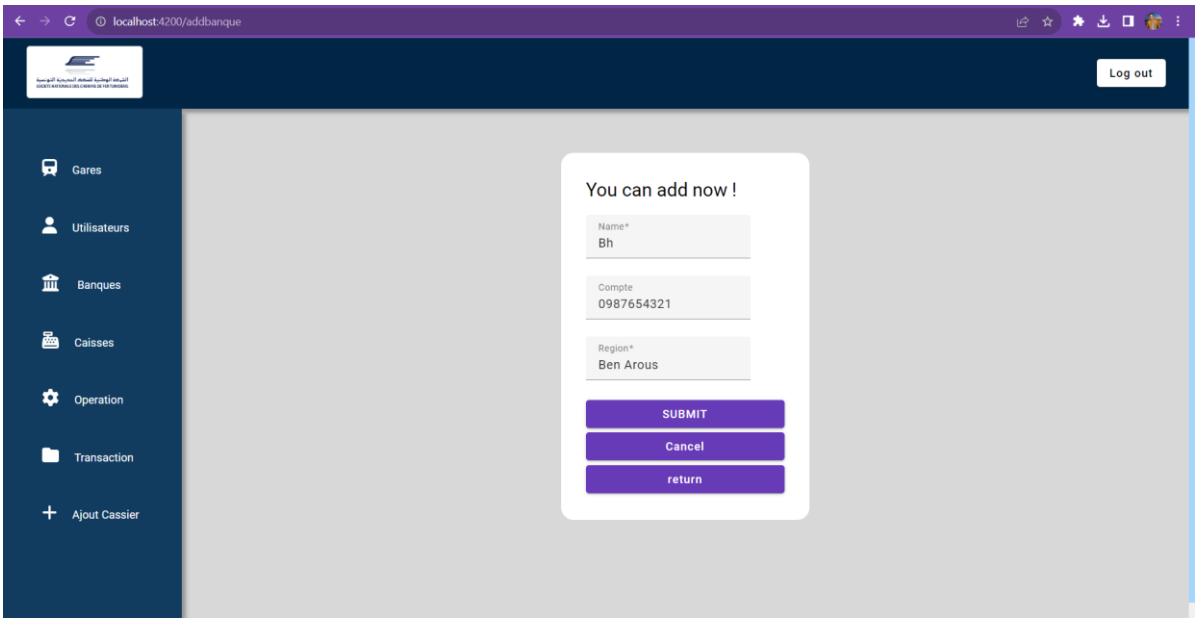


Figure 4.23 – Interface Add Banque

La Figure 4.23 montre l'interface Add Banque. Cette interface affiche comment on add une banque.

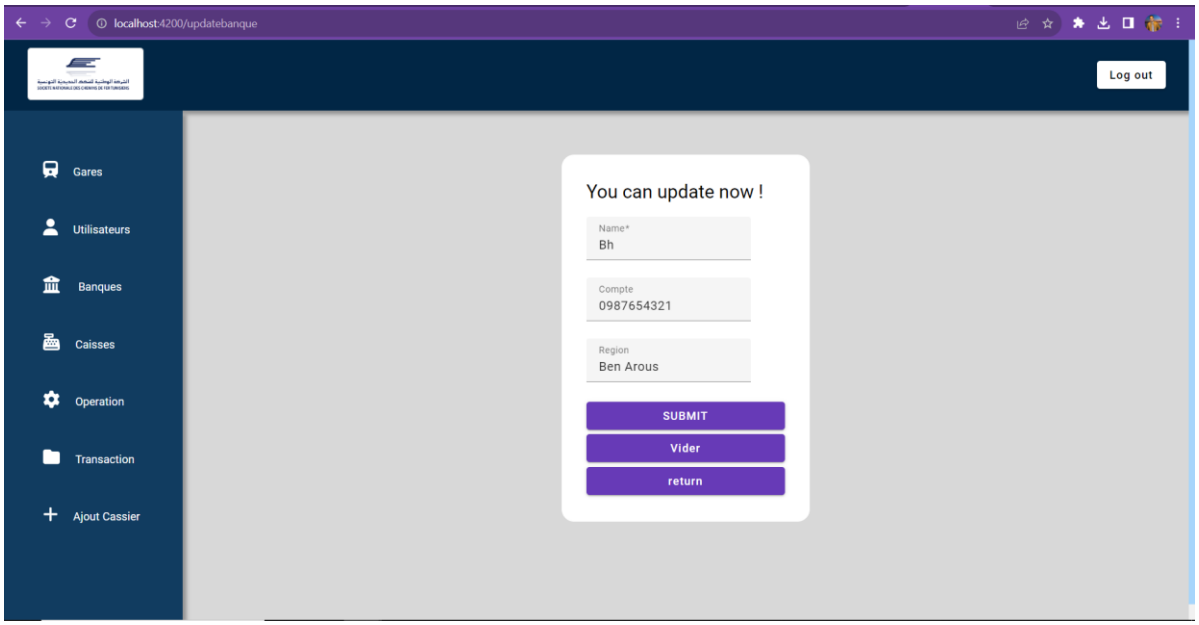


Figure 4.24 – Interface Update Banque

La Figure 4.24 montre l'interface Update Banque. Cette interface affiche comment on update une banque.

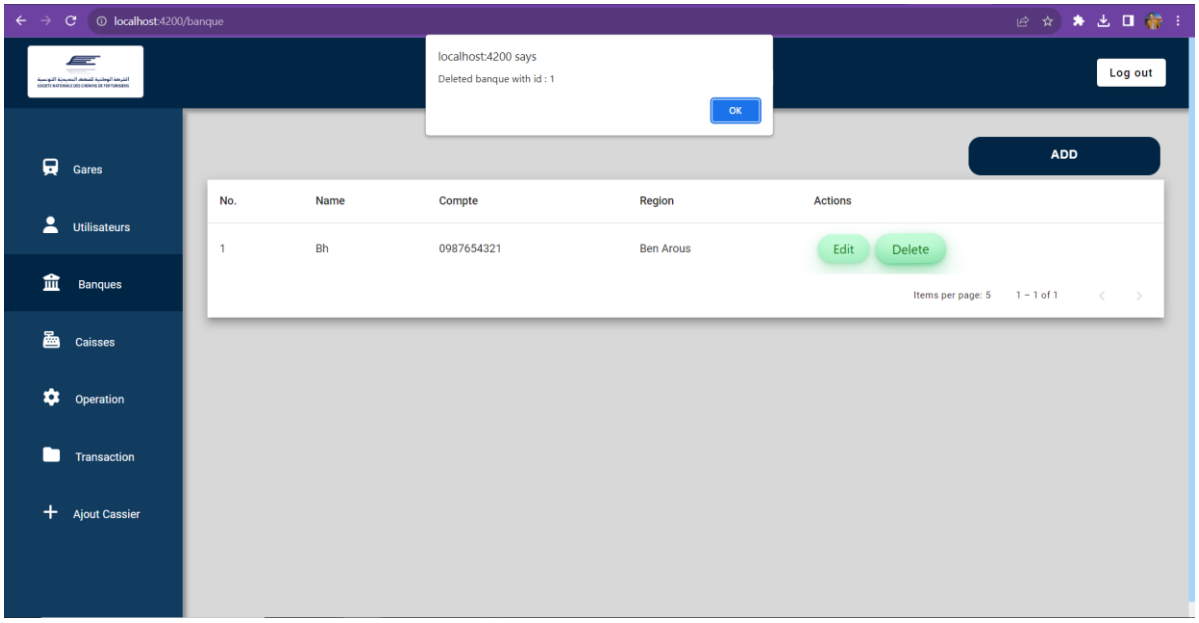


Figure 4.25 – Delete Banque

La Figure 4.25 montre la Suppression de Banque. S'affiche comment on supprime une banque.

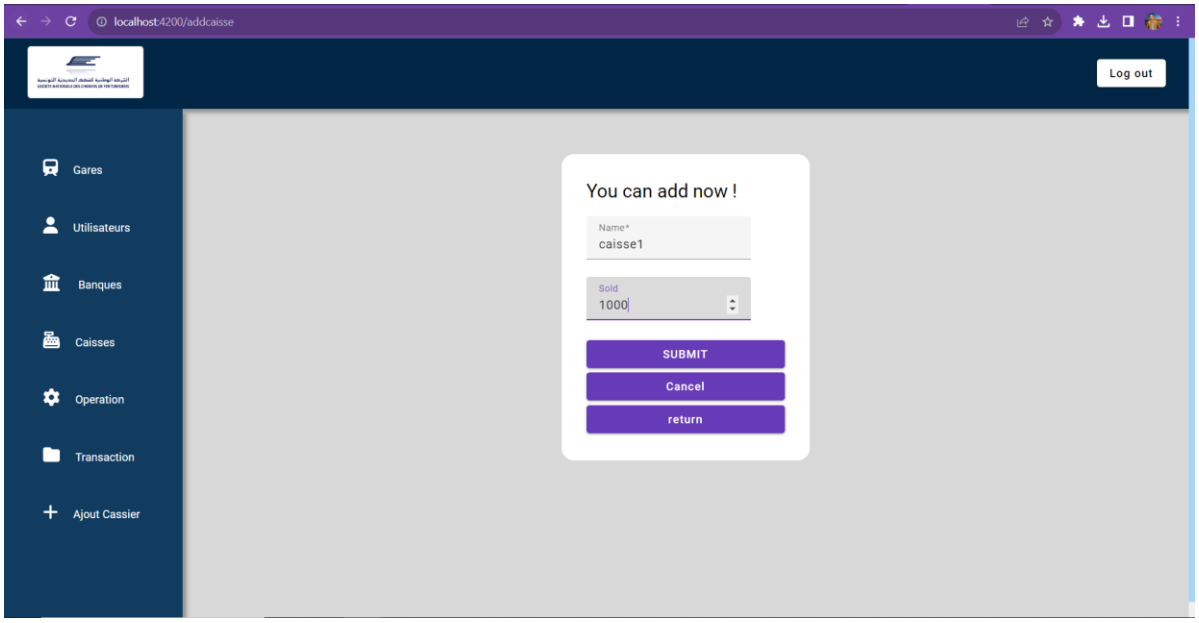


Figure 4.26 – Interface Add Caisse

La Figure 4.26 montre l'interface Add Caisse. Cette interface affiche comment on add une caisse.

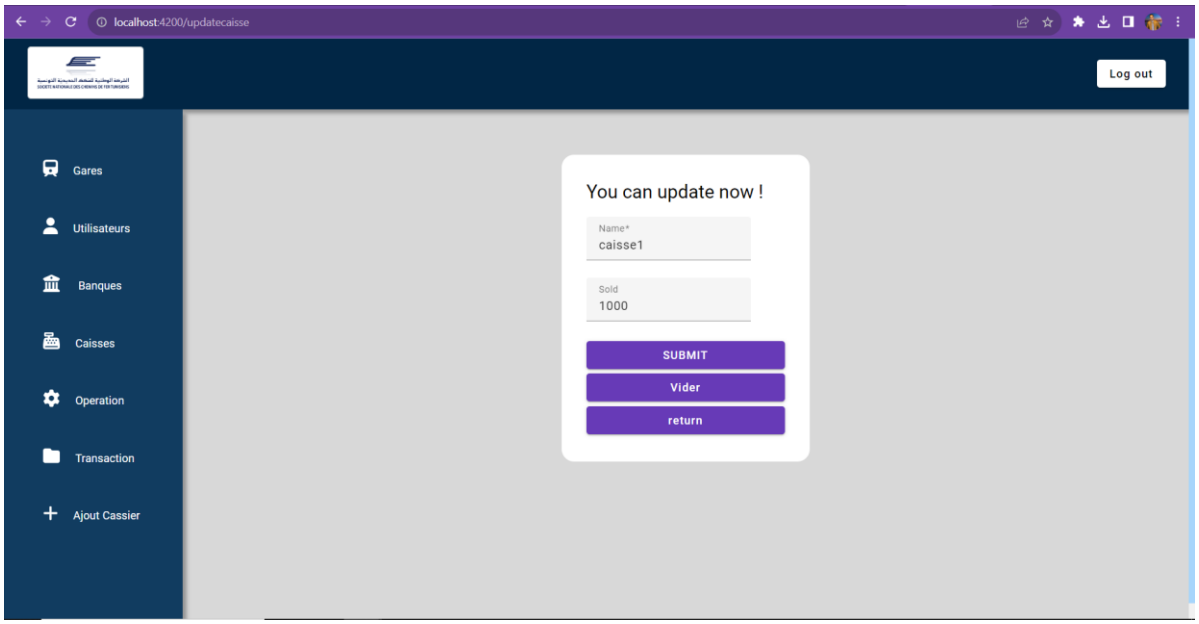


Figure 4.27 – Interface Update Caisse

La Figure 4.27 montre l'interface Update Caisse. Cette interface affiche comment on update une caisse.

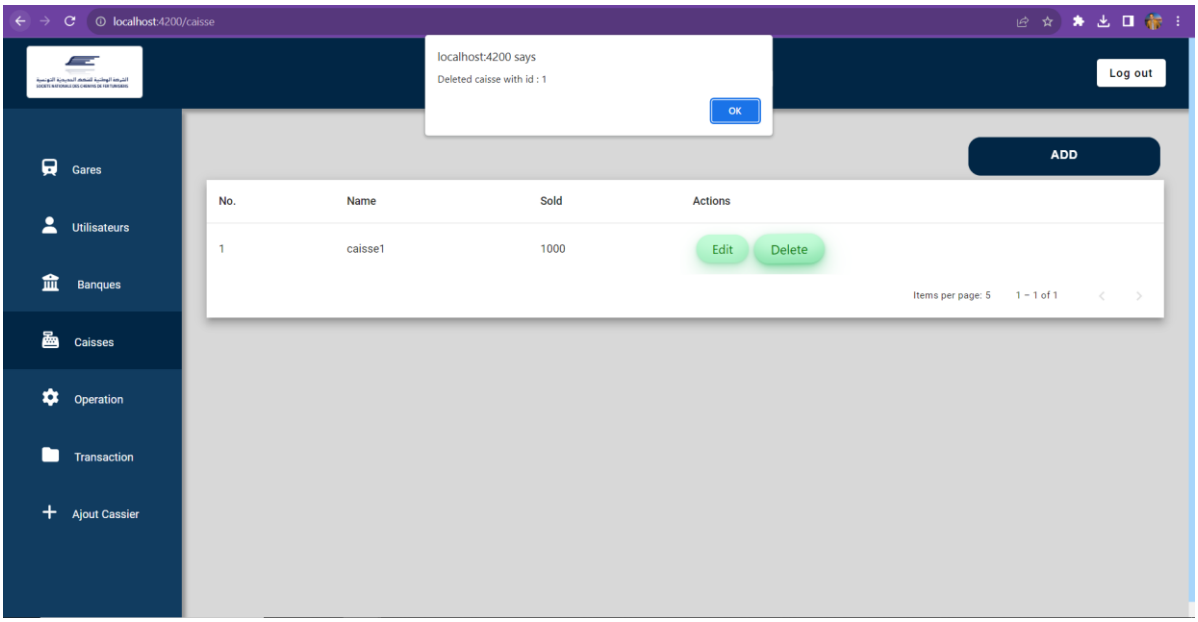


Figure 4.28 – Delete Caisse

La Figure 4.28 montre la Suppression de Caisse. S'affiche comment on supprime une caisse.

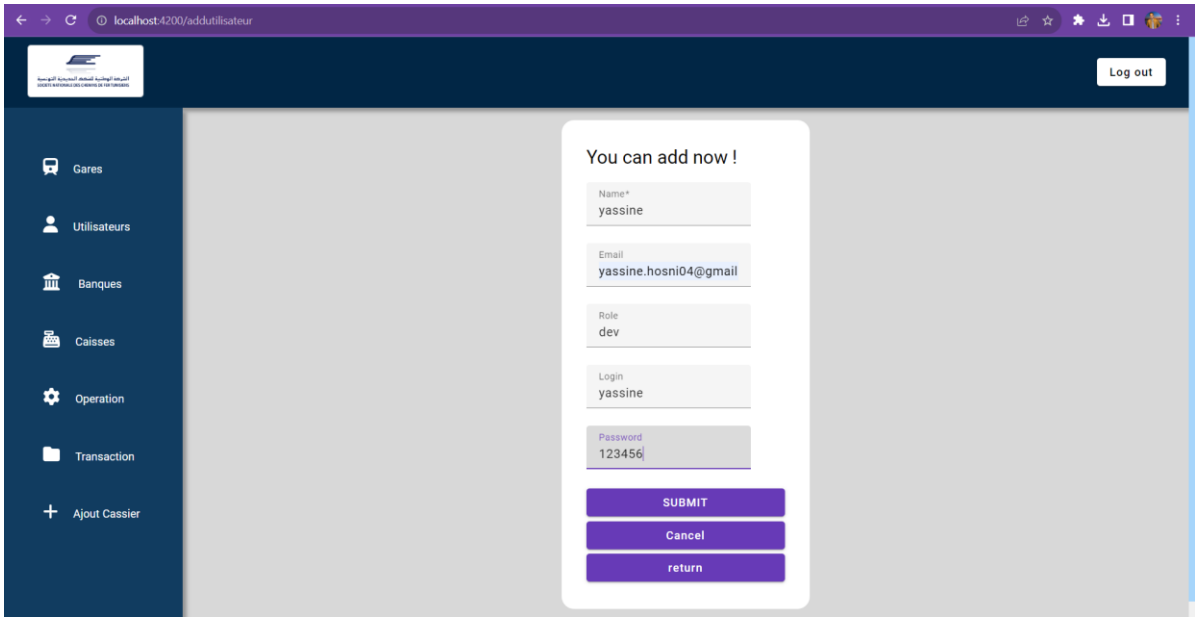


Figure 4.29 – Interface Add Utilisateur

La Figure 4.29 montre l'interface Add Utilisateur. Cette interface affiche comment on add un utilisateur.

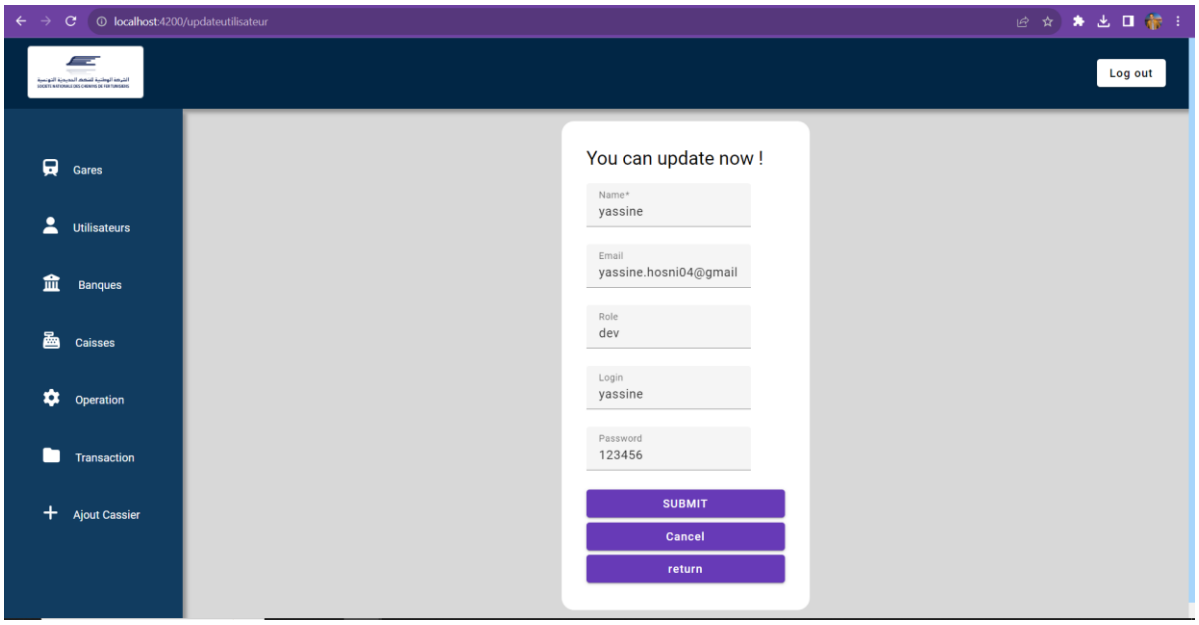


Figure 4.30 – Interface Update Utilisateur

La Figure 4.30 montre l'interface Update Utilisateur. Cette interface affiche comment on update un utilisateur.

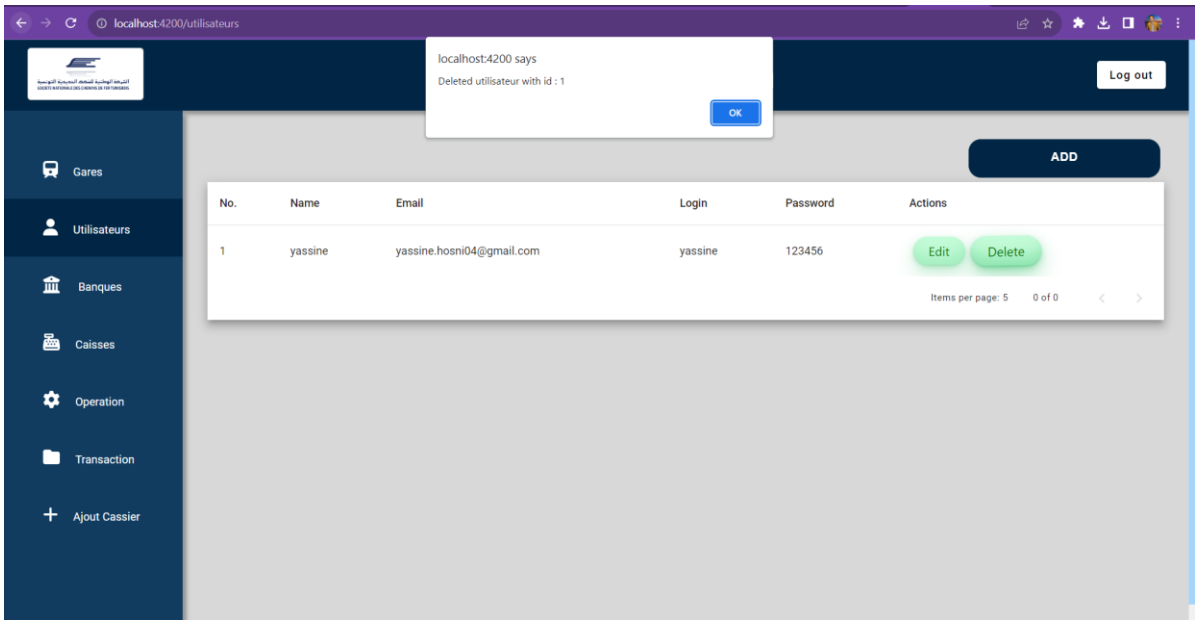


Figure 4.31 – Delete Utilisateur

La Figure 4.31 montre la Suppression de Utilisateur. S'affiche comment on supprime un utilisateur.

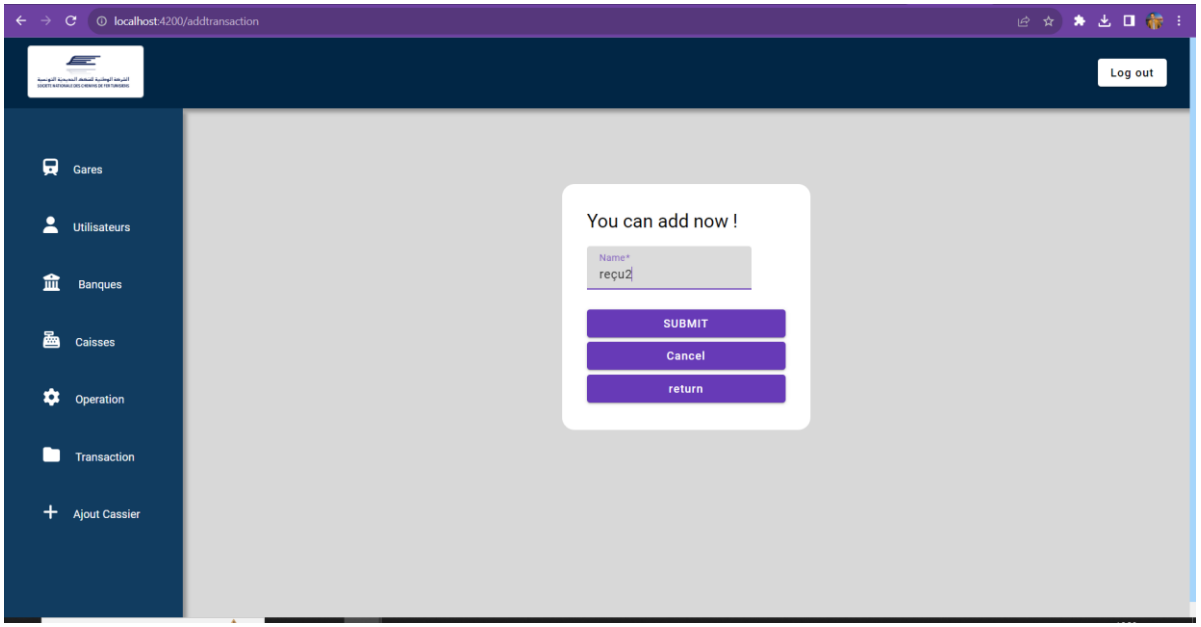


Figure 4.32 – Interface Add Transaction

La Figure 4.32 montre l'interface Add Transaction. Cette interface affiche comment on add une transaction.

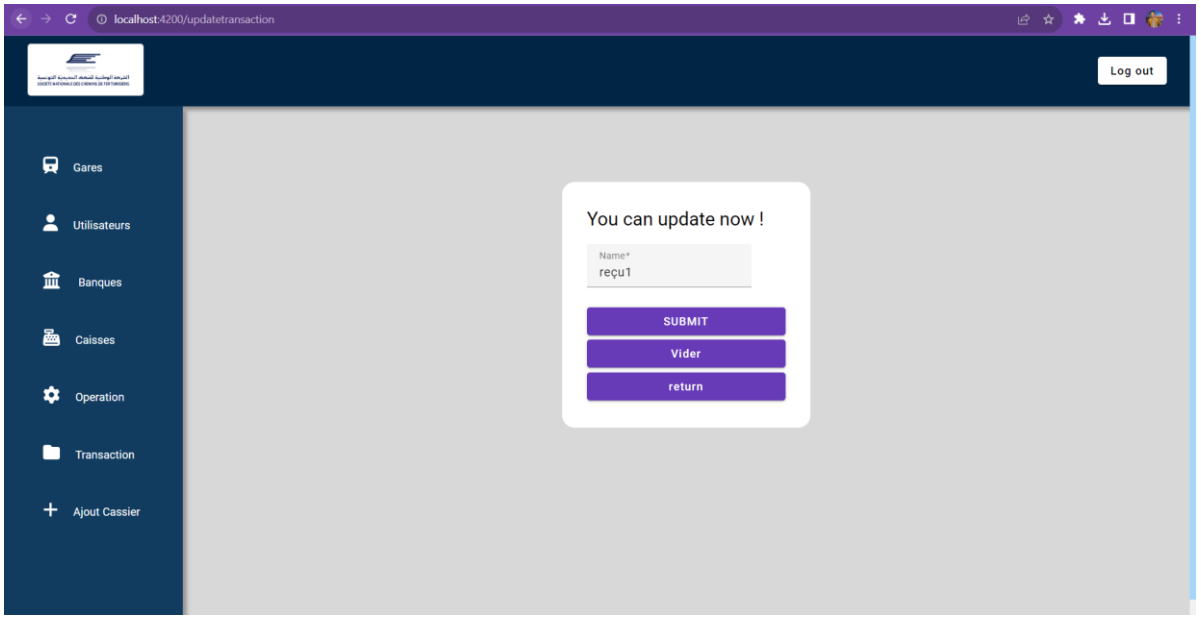


Figure 4.33 – Interface Update Transaction

La Figure 4.33 montre l'interface Update Transaction. Cette interface affiche comment on update une transaction.

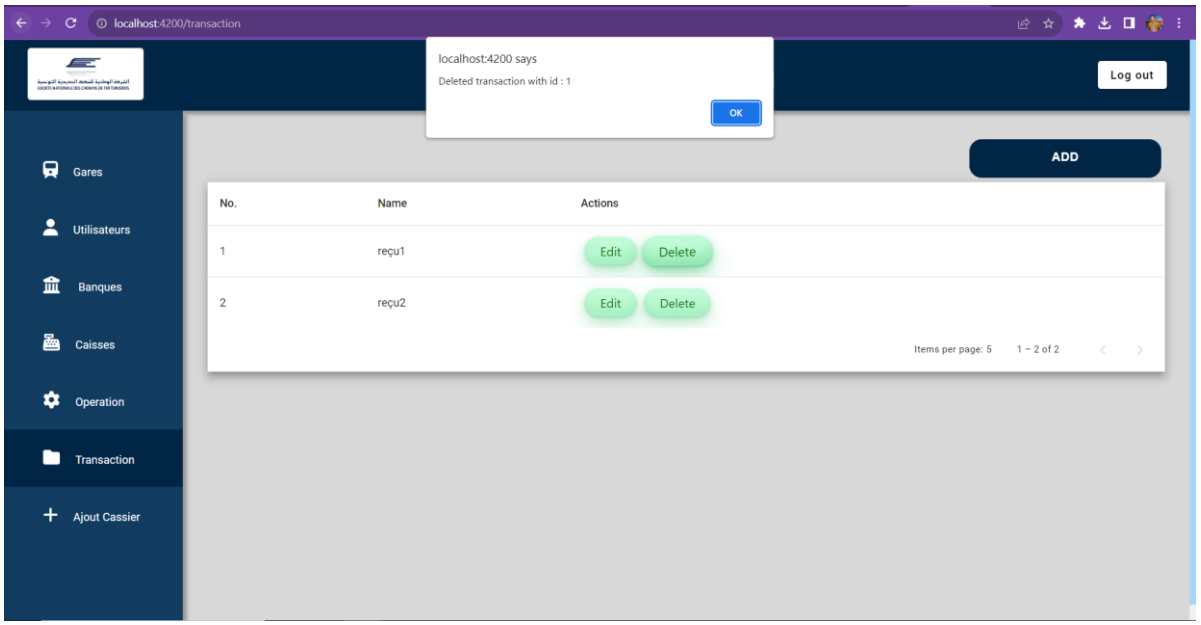


Figure 4.34 – Delete Transaction

La Figure 4.34 montre la Suppression de Transaction. S'affiche comment on supprime une transaction.

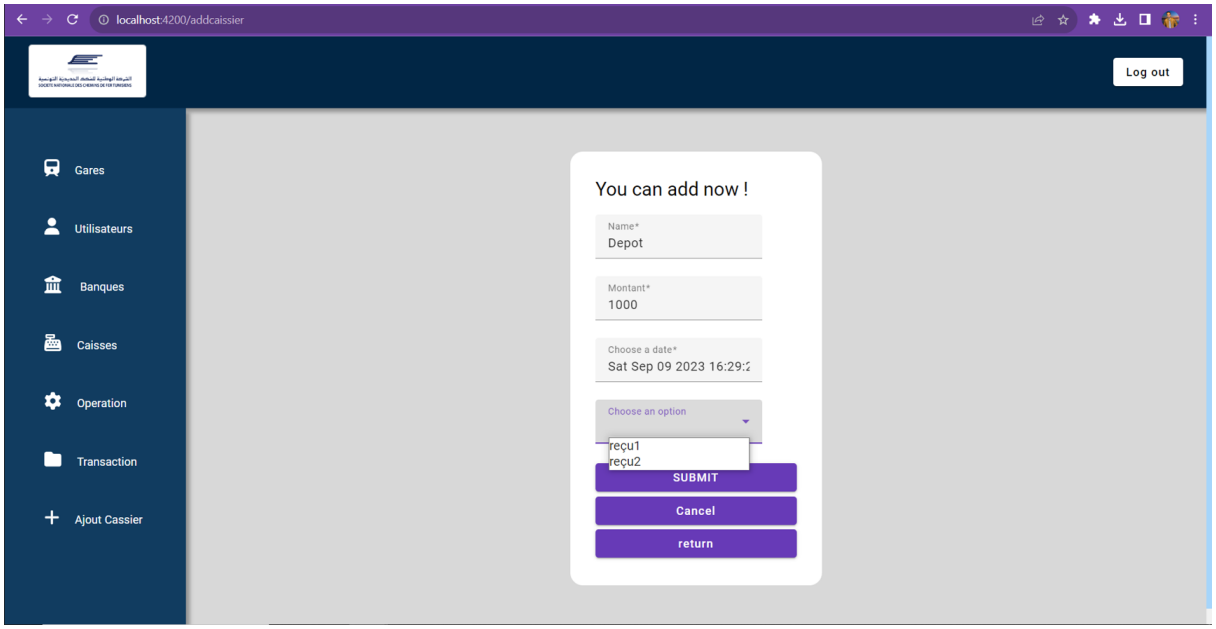


Figure 4.35 – Interface Add Operation

La Figure 4.35 montre l'interface Add Operation. Cette interface affiche comment on add une opération.

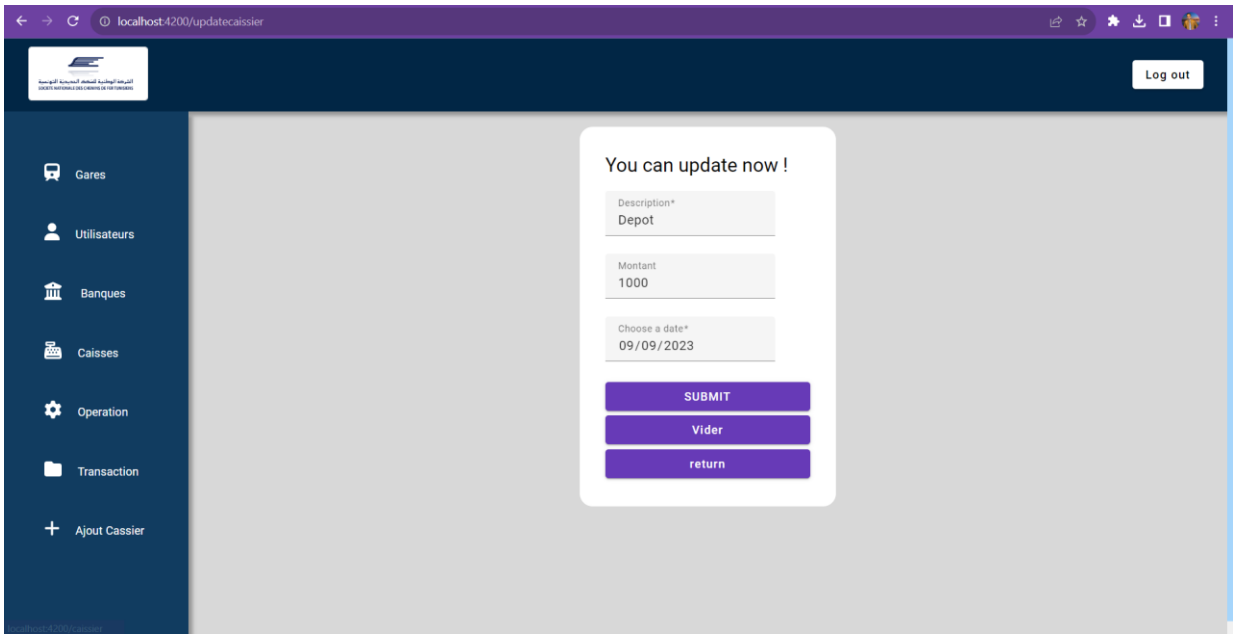


Figure 4.36 – Interface Update Operation

La Figure 4.36 montre l'interface Update Operation. Cette interface affiche comment on update une opération.

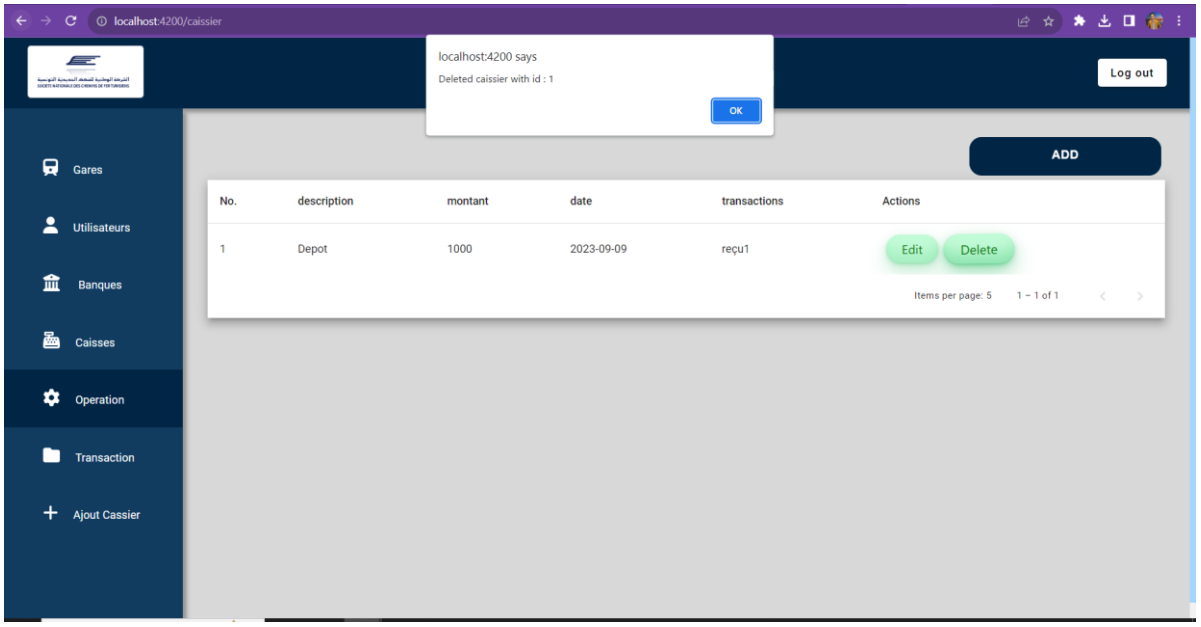


Figure 4.37 – Delete Operation

La Figure 4.37 montre la Suppression de Operation. S'affiche comment on supprime une opération.

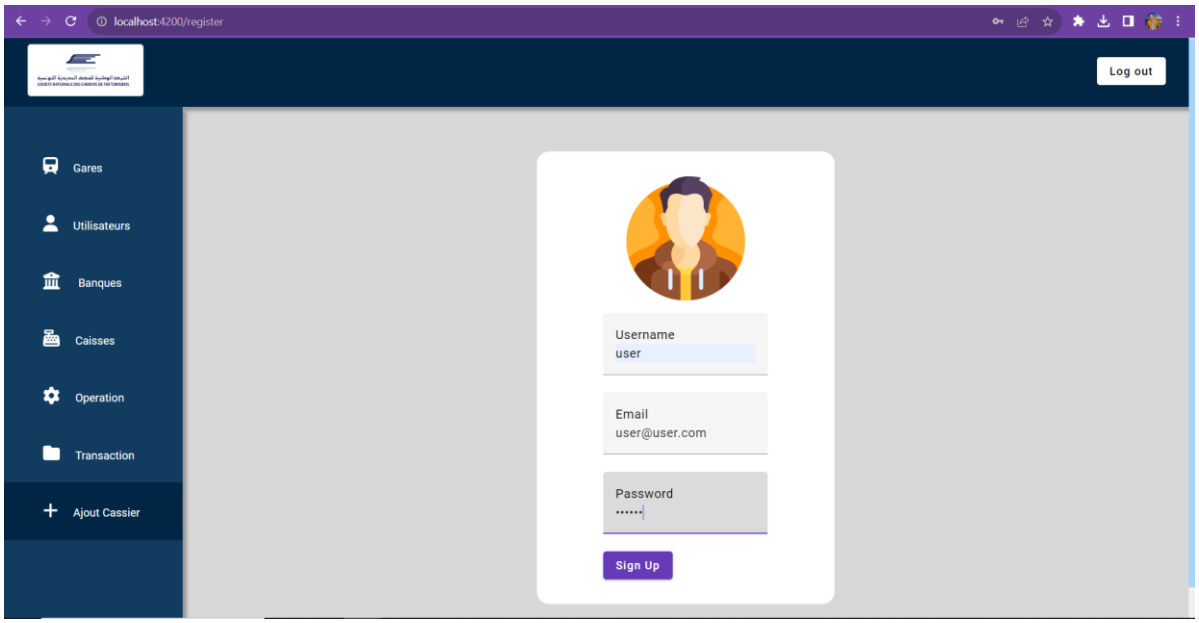


Figure 4.38 – Interface Add Caissier

La Figure 4.38 montre l'interface Add Caissier. Cette interface affiche comment on add un caissier.

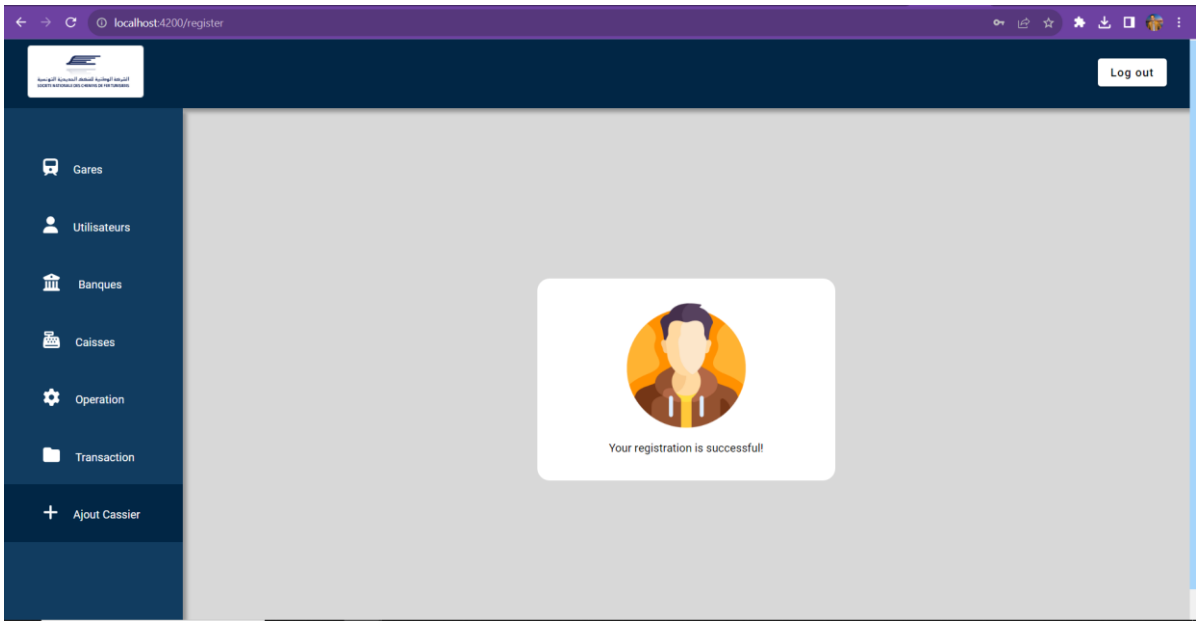


Figure 4.39 – Interface ajout avec succès

La Figure 4.39 montre l'interface d'ajout avec succès.

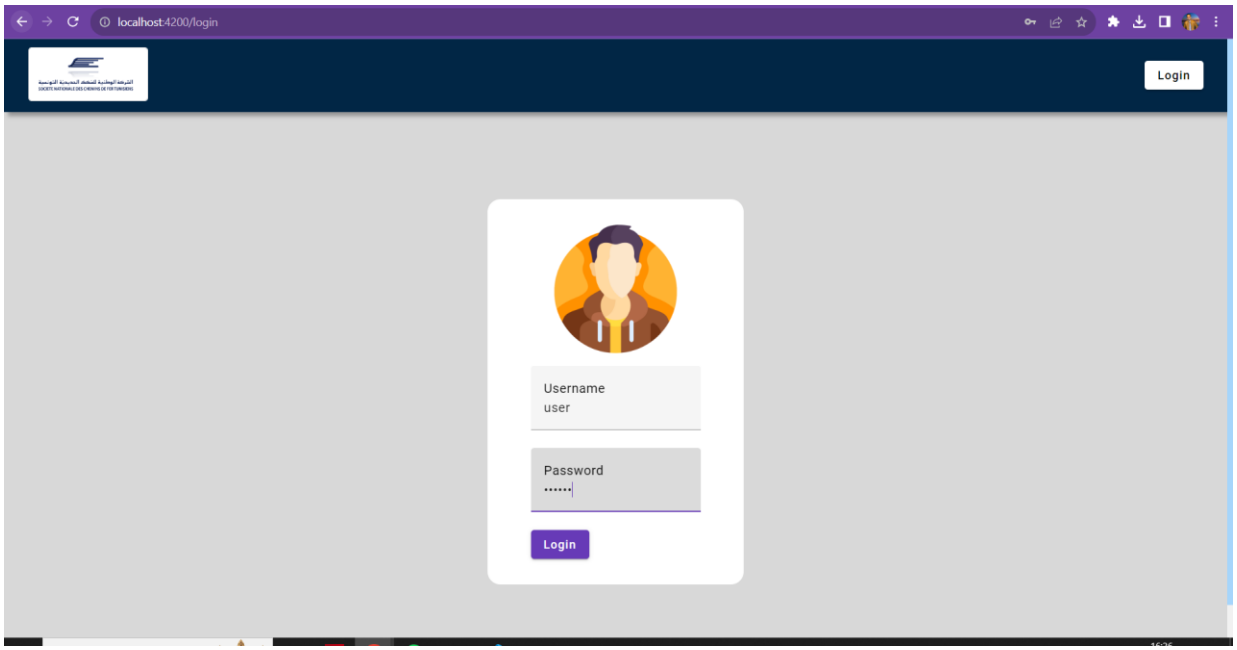


Figure 4.40 – Interface Login Caissier

La Figure 4.40 présente l'interface Login. Cette interface est utilisée pour s'authentifier et accéder à l'application.

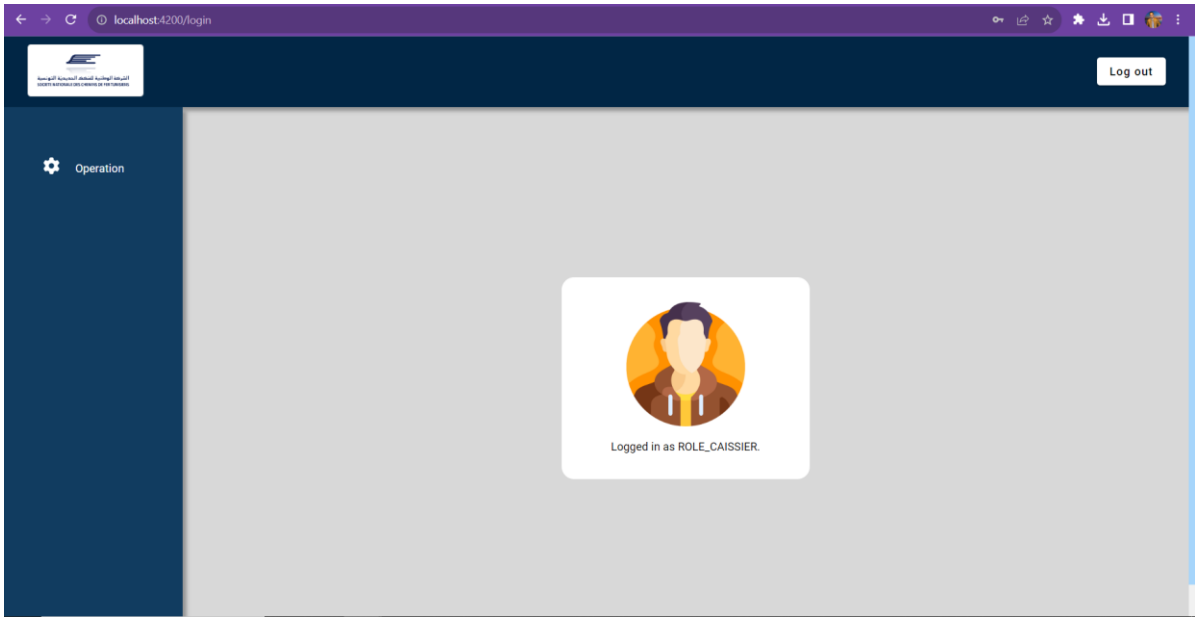


Figure 4.41 – Interface connecté avec caissier

4.1 Tests de validation

Cette section met en œuvre certains des tests de validation que nous avons réalisés avec Postman afin de vérifier le bon fonctionnement de l'API (SpringBoot). Les autres tests figurent dans les annexes.

Test Authentification

Dans ce test d'authentification utilisant Postman, nous avons vérifié le processus d'authentification et nous nous sommes assurés qu'il renvoie un jeton JWT (présenté en annexes). En simulant la requête d'authentification, nous avons validé la génération réussie d'un jeton, qui a donné accès aux ressources sécurisées du système.

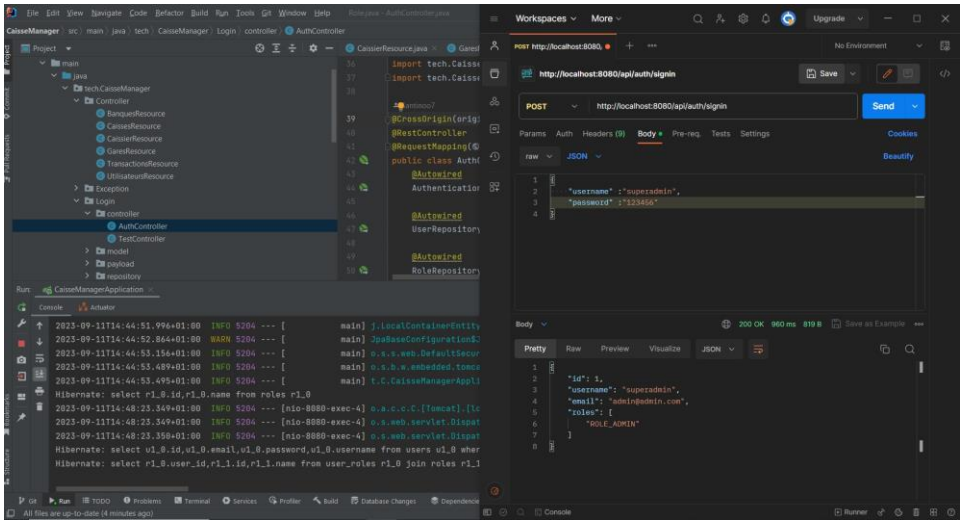


Figure 4.42 – Test d'authentification.

Test Logout

Dans ce test, nous avons pu vérifier Logout

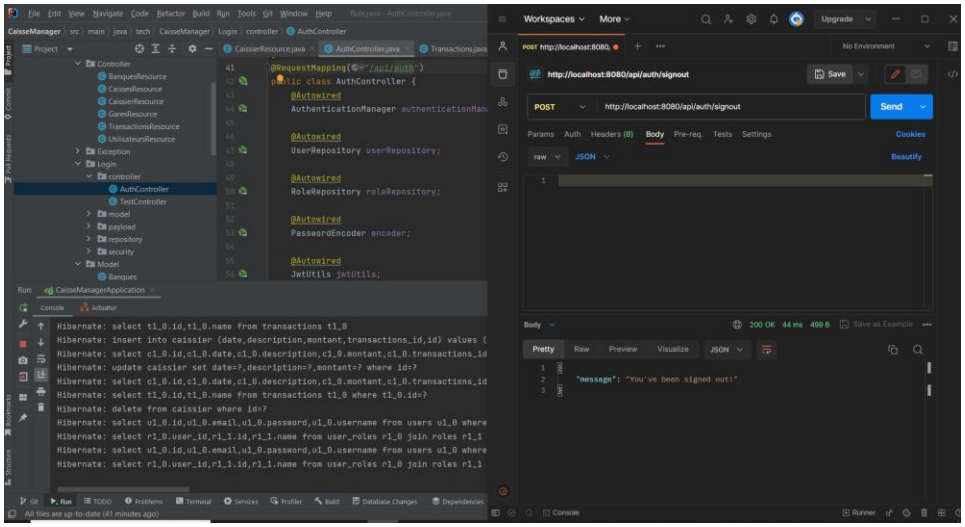


Figure 4.43 – Test Logout

Test ajout d'utilisateur

Cette figure représente un cas de test spécifique à l'ajout d'utilisateur. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'ajout d'un nouvel utilisateur dans le système.

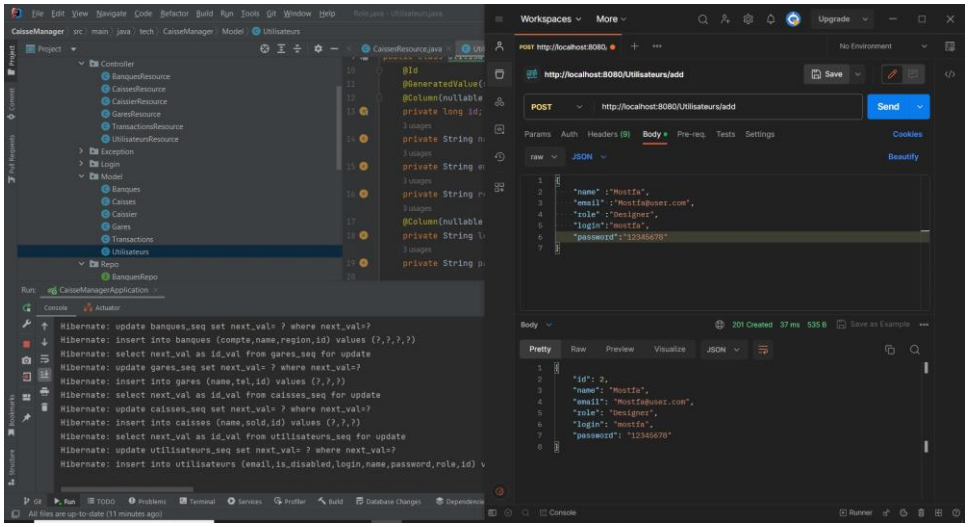


Figure 4.44 – Test ajout d'utilisateur.

Test d'obtenir les utilisateurs

Cette figure représente un cas de test spécifique à l'obtenir d'utilisateurs. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'obtenir la liste d'utilisateurs dans le système.

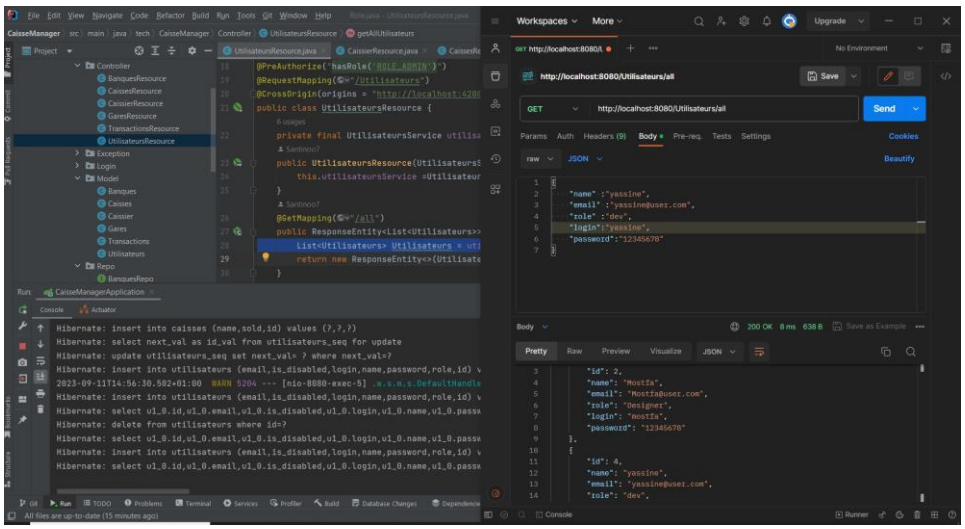


Figure 4.45 – Test d'obtenir les utilisateurs.

Test modification d'utilisateur

Cette figure représente un cas de test spécifique à la modification de données d'utilisateur. L'objectif de ce test a été de vérifier le bon fonctionnement du processus de modification d'un utilisateur dans le système.

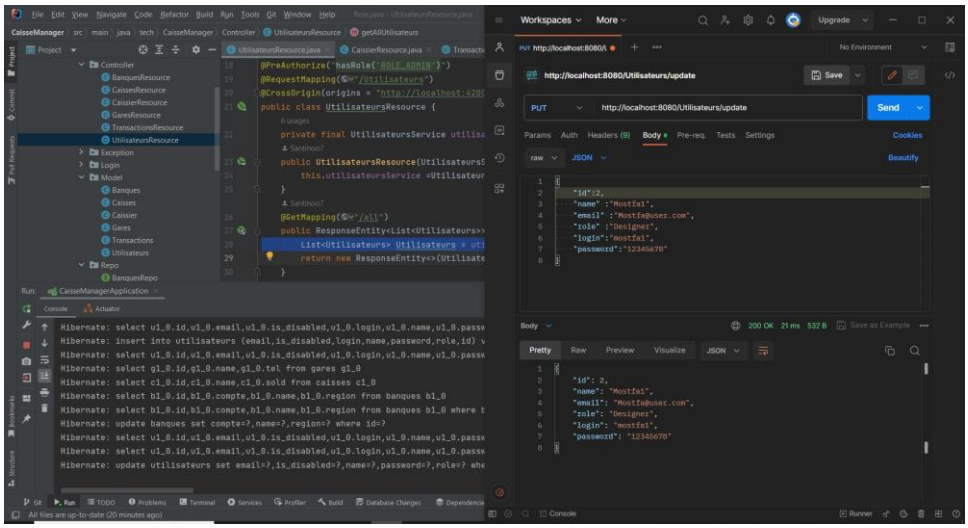


Figure 4.46 – Test modification d'utilisateur.

Test suppression d'utilisateur

Cette figure représente un cas de test spécifique à la suppression d'utilisateur. Ce test a permis de vérifier le bon fonctionnement du processus de suppression d'un utilisateur existant dans le système.

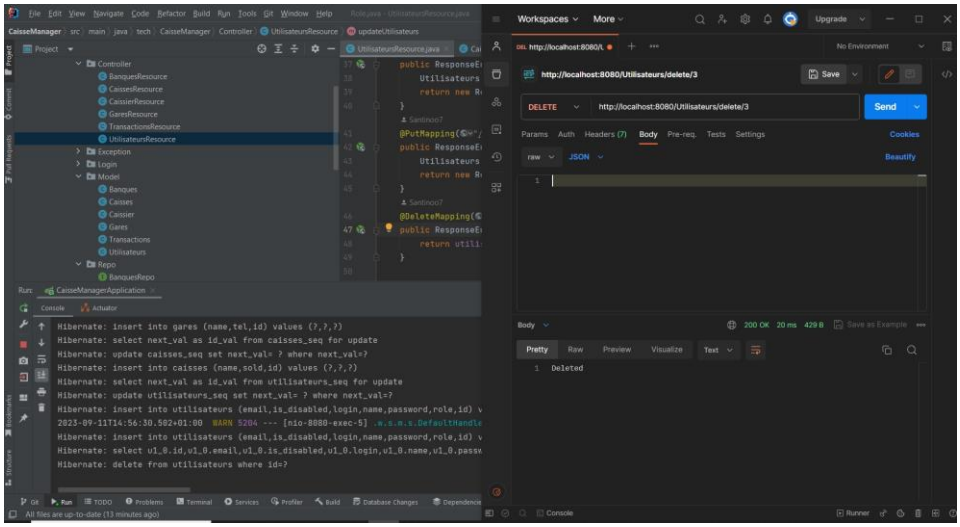


Figure 4.47 – Test suppression d'utilisateur.

Test ajout de gare

Cette figure représente un cas de test spécifique à l'ajout de gare. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'ajout d'une nouvelle gare dans le système.

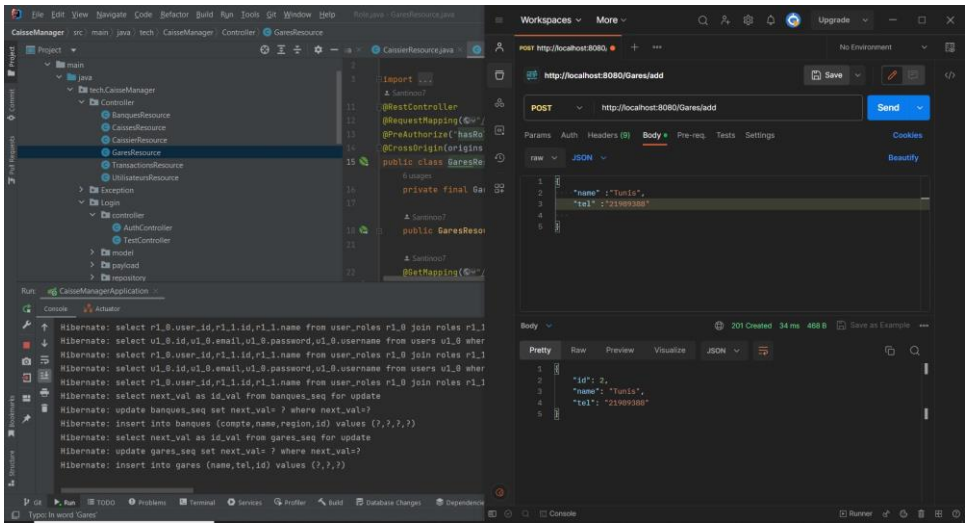


Figure 4.48 – Test ajout de gare.

Test d'obtenir les gares

Cette figure représente un cas de test spécifique à l'obtenir des gares. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'obtenir la liste des gares dans le système.

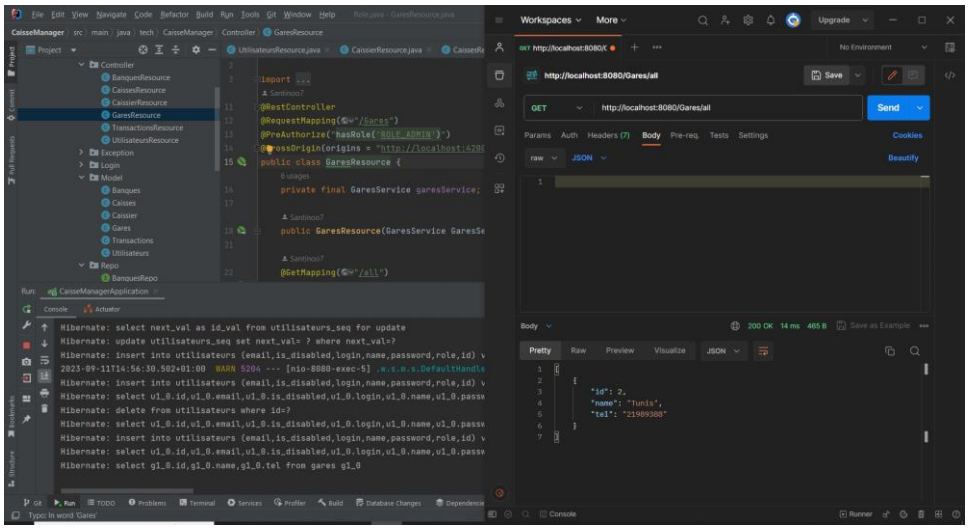


Figure 4. 49 – Test d'obtenir les gares.

Test modification de gare

Cette figure représente un cas de test spécifique à la modification de données de gare. L'objectif de ce test a été de vérifier le bon fonctionnement du processus de modification d'une gare dans le système.

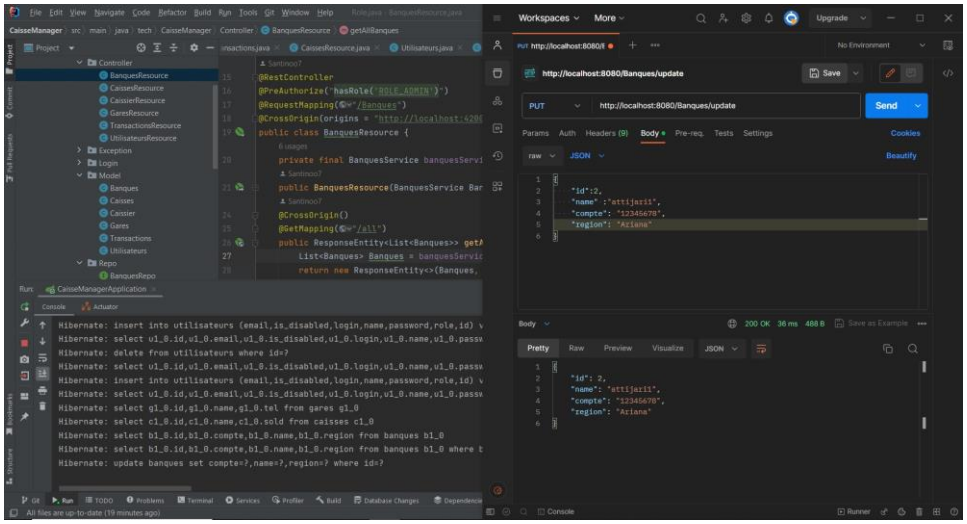


Figure 4.50 – Test modification de gare.

Test suppression de gare

Cette figure représente un cas de test spécifique à la suppression d'une gare. Ce test a permis de vérifier le bon fonctionnement du processus de suppression d'une gare existant dans le système.

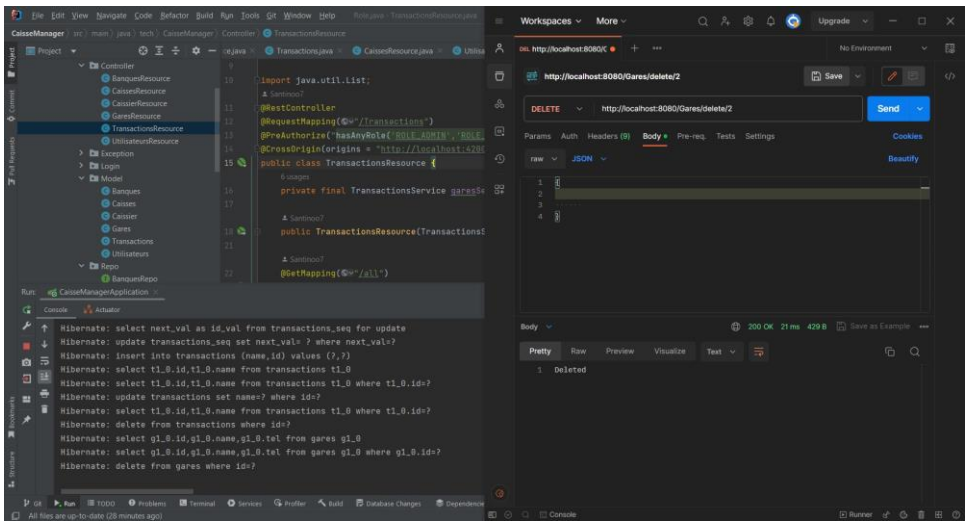


Figure 4.51 – Test suppression d'une gare.

Test ajout de banque

Cette figure représente un cas de test spécifique à l'ajout de banque. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'ajout d'une nouvelle banque dans le système.

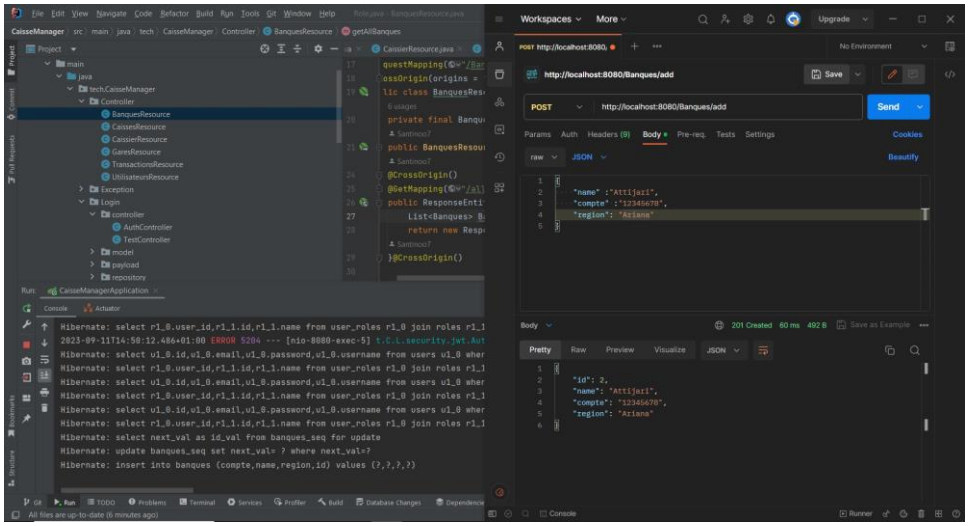


Figure 4.52 – Test ajout de banque.

Test d'obtenir les banques

Cette figure représente un cas de test spécifique à l'obtenir des banques. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'obtenir la liste des banques dans le système.

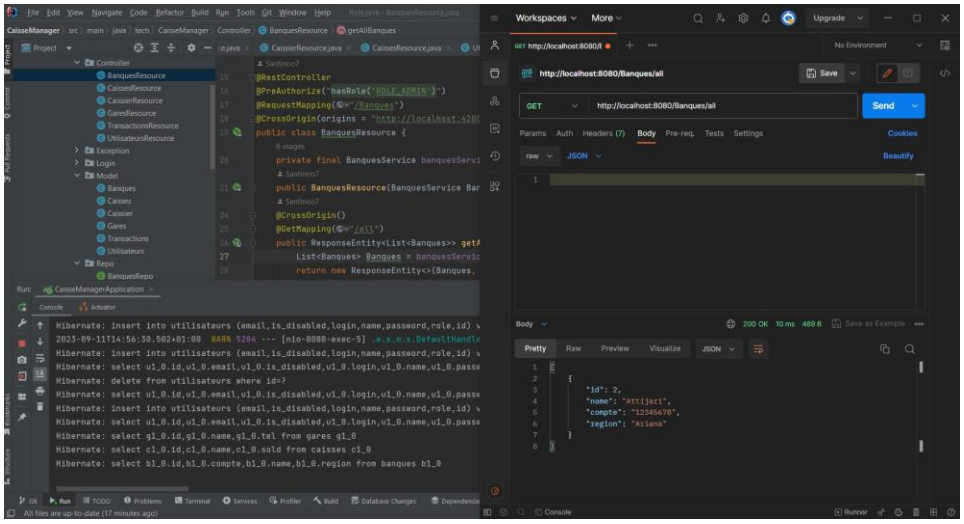


Figure 4.53 – Test d'obtenir les banques.

Test modification de banque

Cette figure représente un cas de test spécifique à la modification de données de banque. L'objectif de ce test a été de vérifier le bon fonctionnement du processus de modification d'une gare dans le système.

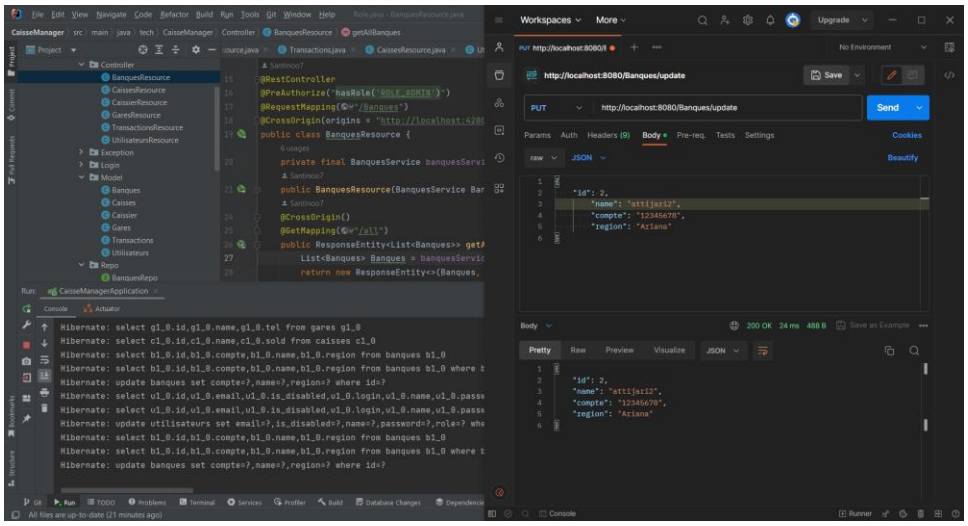


Figure 4.54 – Test modification de banque.

Test suppression de banque

Cette figure représente un cas de test spécifique à la suppression d'une banque. Ce test a permis de vérifier le bon fonctionnement du processus de suppression d'une gare existant dans le système.

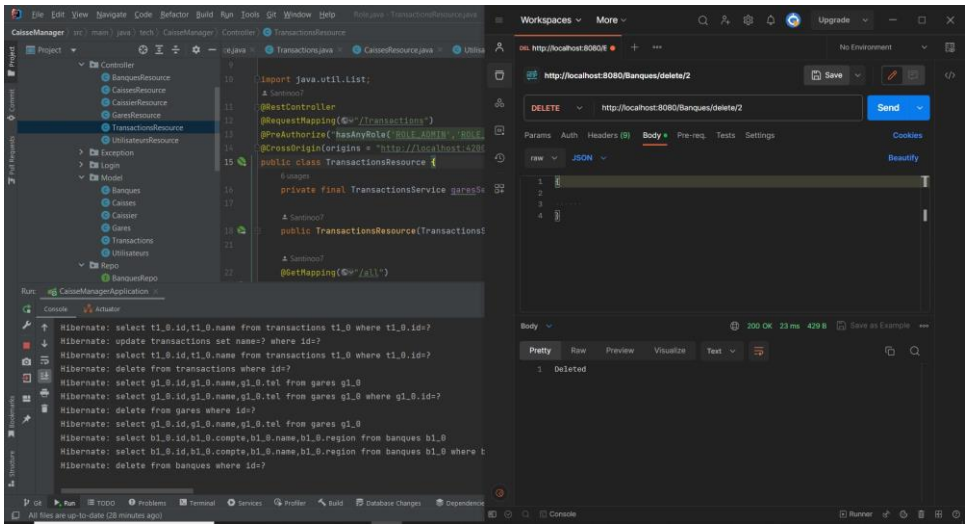


Figure 4.55 – Test suppression d'une banque.

Test ajout de caisse

Cette figure représente un cas de test spécifique à l'ajout de caisse. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'ajout d'une nouvelle caisse dans le système.

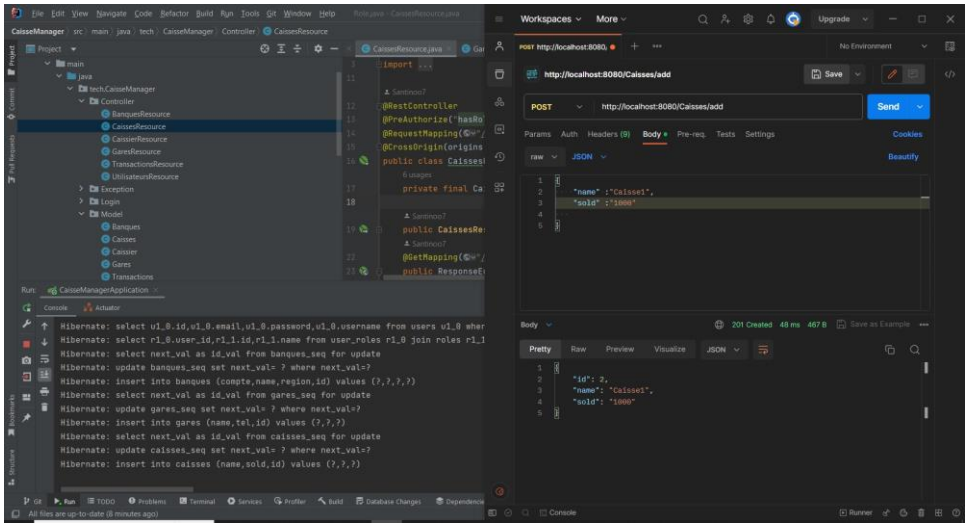


Figure 4.56 – Test ajout de caisse.

Test d'obtenir les caisses

Cette figure représente un cas de test spécifique à l'obtenir des caisses. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'obtenir la liste des caisses dans le système.

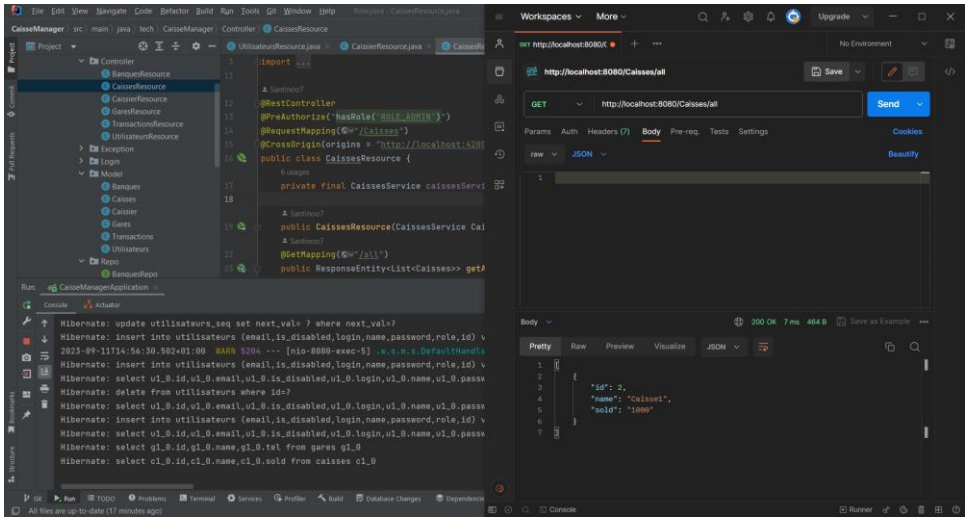


Figure 4.57 – Test d'obtenir les caisses.

Test modification de caisse

Cette figure représente un cas de test spécifique à la modification de données de caisse. L'objectif de ce test a été de vérifier le bon fonctionnement du processus de modification d'une caisse dans le système.

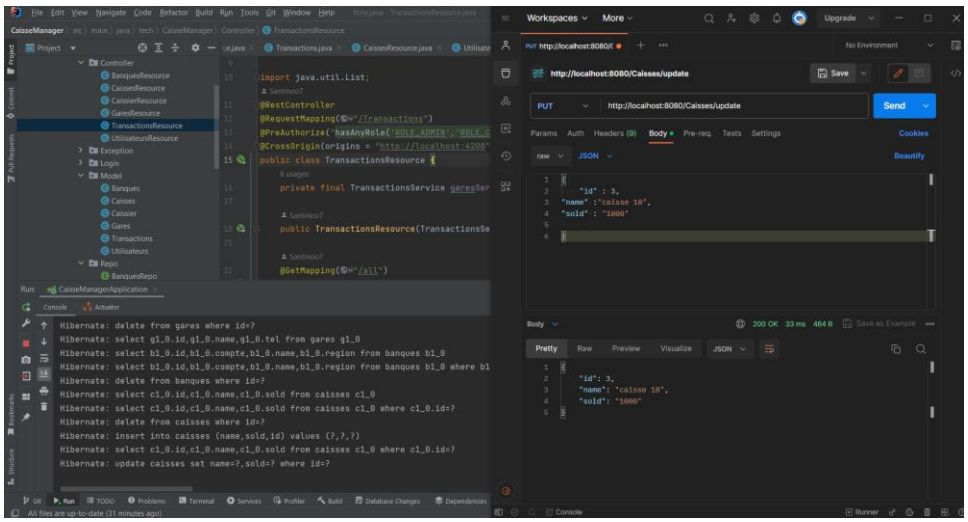


Figure 4.58 – Test modification de caisse.

Test suppression de caisse

Cette figure représente un cas de test spécifique à la suppression d'une caisse. Ce test a permis de vérifier le bon fonctionnement du processus de suppression d'une caisse existant dans le système.

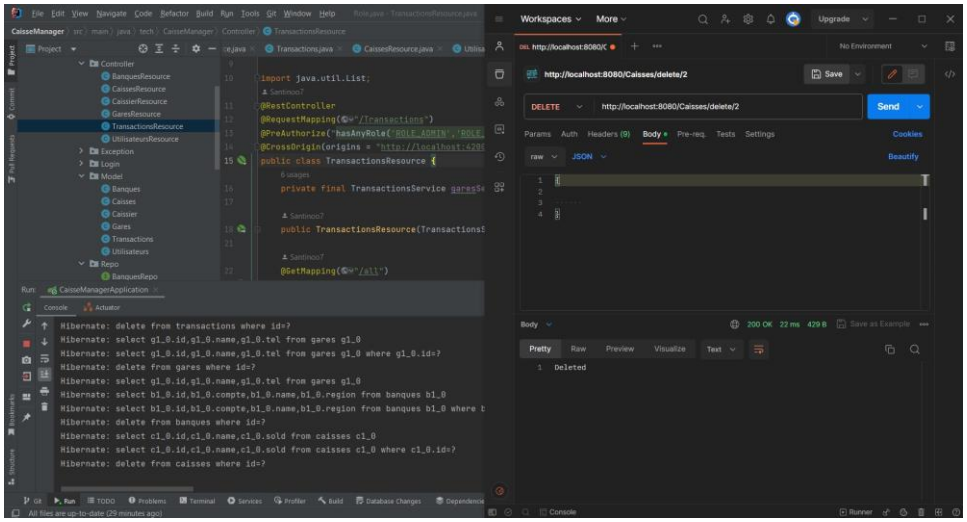


Figure 4. 59 – Test suppression d'une caisse.

Test ajout de transaction

Cette figure représente un cas de test spécifique à l'ajout de transaction. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'ajout d'une nouvelle transaction dans le système.

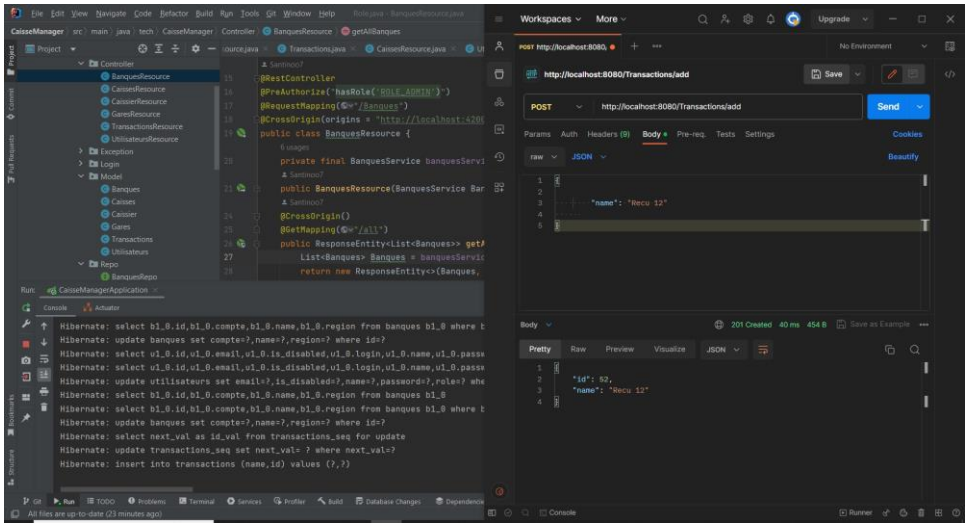


Figure 4.60 – Test ajout de transaction.

Test d'obtenir les transactions

Cette figure représente un cas de test spécifique à l'obtenir des transactions. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'obtenir la liste des transactions dans le système.

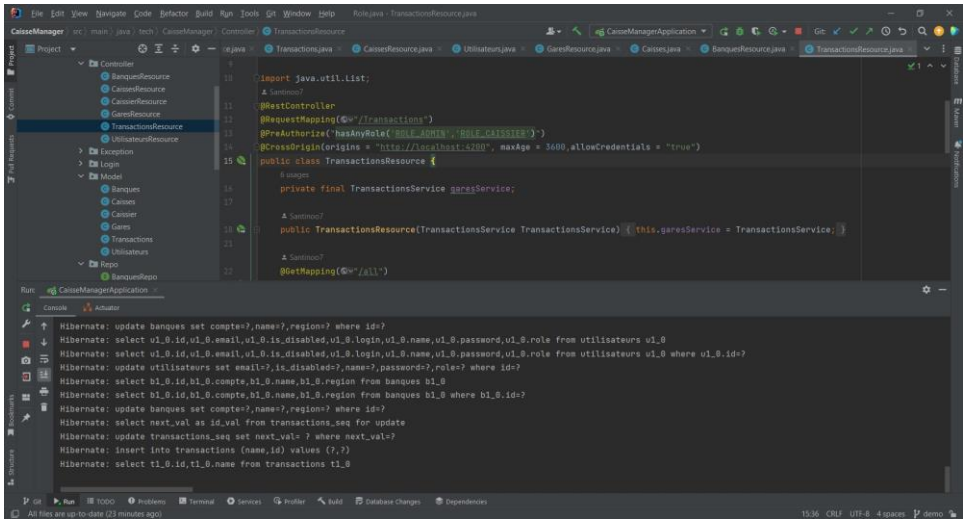


Figure 4.61 – Test d'obtenir les transactions.

Test modification de transaction

Cette figure représente un cas de test spécifique à la modification de données de transaction. L'objectif de ce test a été de vérifier le bon fonctionnement du processus de modification d'une transaction dans le système.

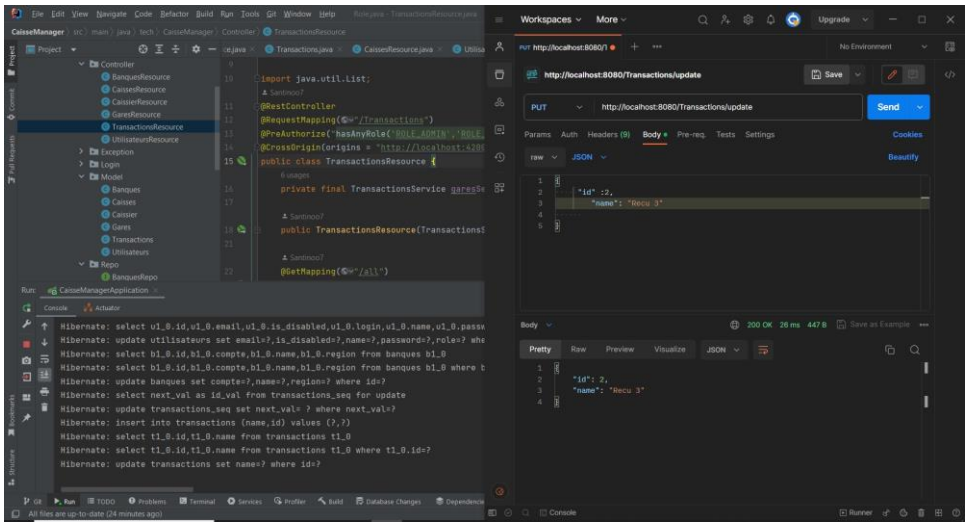


Figure 4.62 – Test modification de transaction.

Test suppression de transaction

Cette figure représente un cas de test spécifique à la suppression d'une transaction. Ce test a permis de vérifier le bon fonctionnement du processus de suppression d'une transaction existant dans le système.

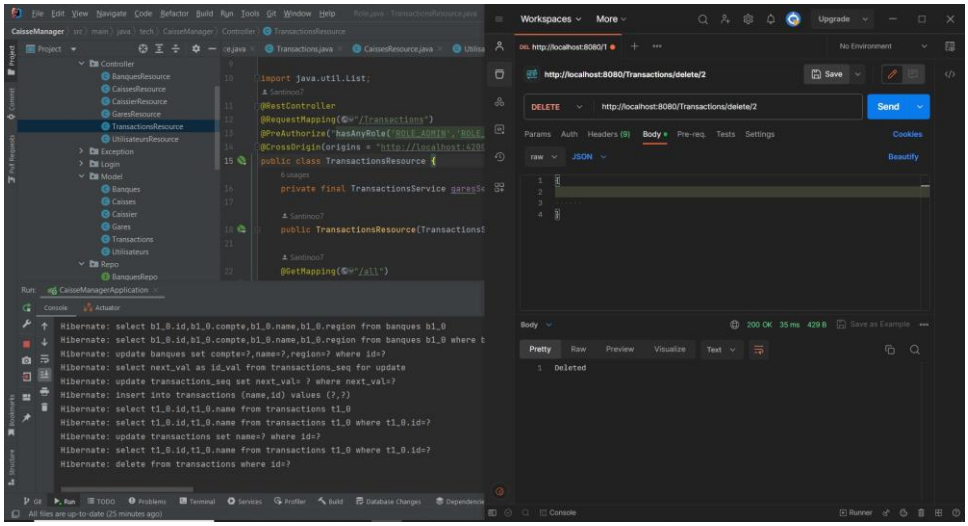


Figure 4.63 – Test suppression d'une transaction.

Test ajout d'opération

Cette figure représente un cas de test spécifique à l'ajout d'opération. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'ajout d'une nouvelle opération dans le système.

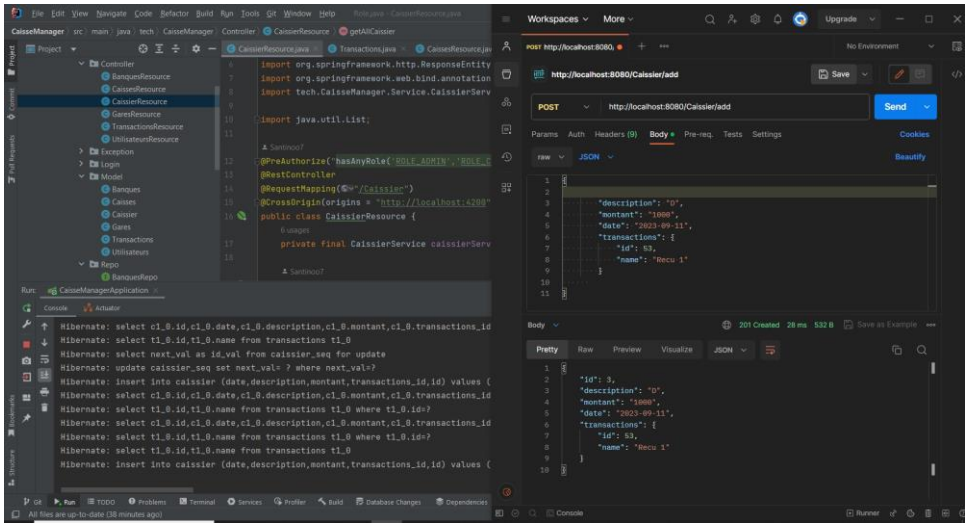


Figure 4.64 – Test ajout d'opération.

Test d'obtenir les opérations

Cette figure représente un cas de test spécifique à l'obtenir des opérations. L'objectif de ce test a été de vérifier le bon fonctionnement du processus d'obtenir la liste des opérations dans le système.

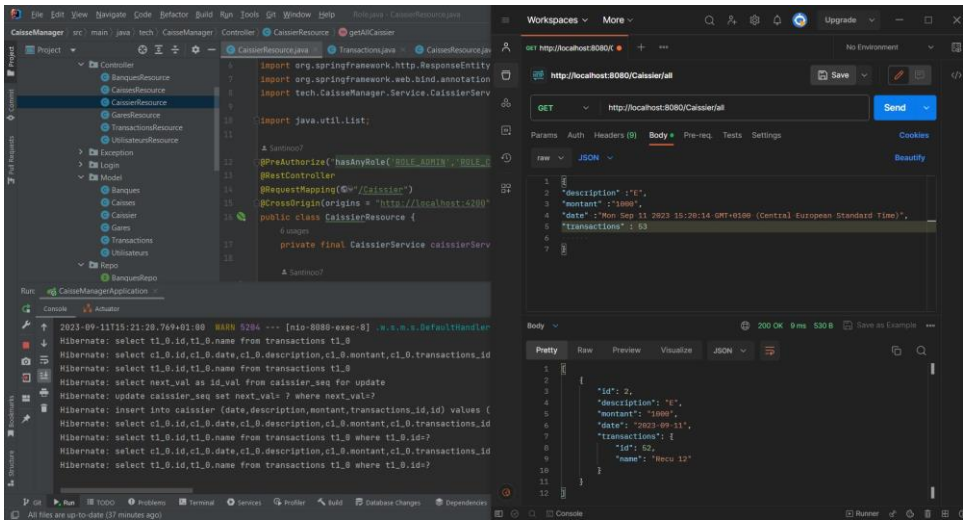


Figure 4.65 – Test d'obtenir les opérations.

Test modification d'opération

Cette figure représente un cas de test spécifique à la modification de données d'opération. L'objectif de ce test a été de vérifier le bon fonctionnement du processus de modification d'une opération dans le système.

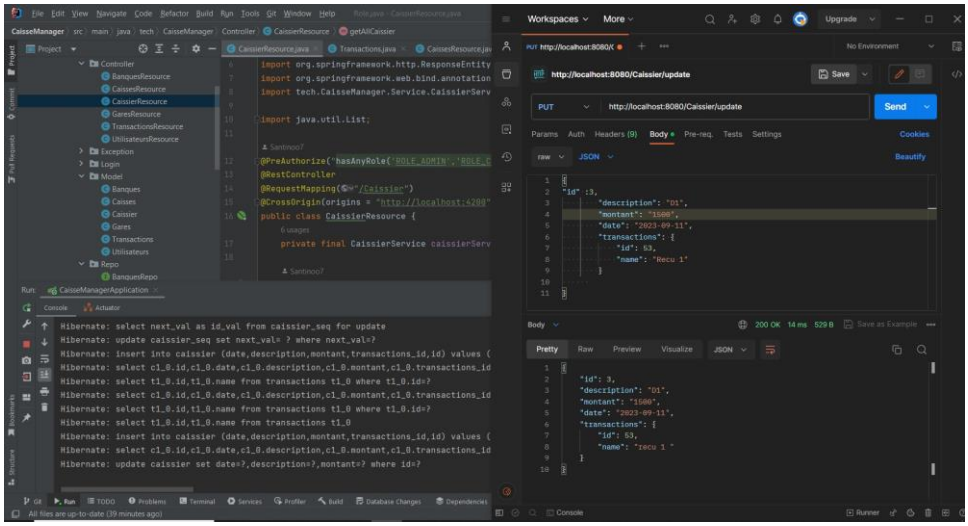


Figure 4.66 – Test modification d'opération.

Test suppression d'opération

Cette figure représente un cas de test spécifique à la suppression d'une opération. Ce test a permis de vérifier le bon fonctionnement du processus de suppression d'une opération existant dans le système.

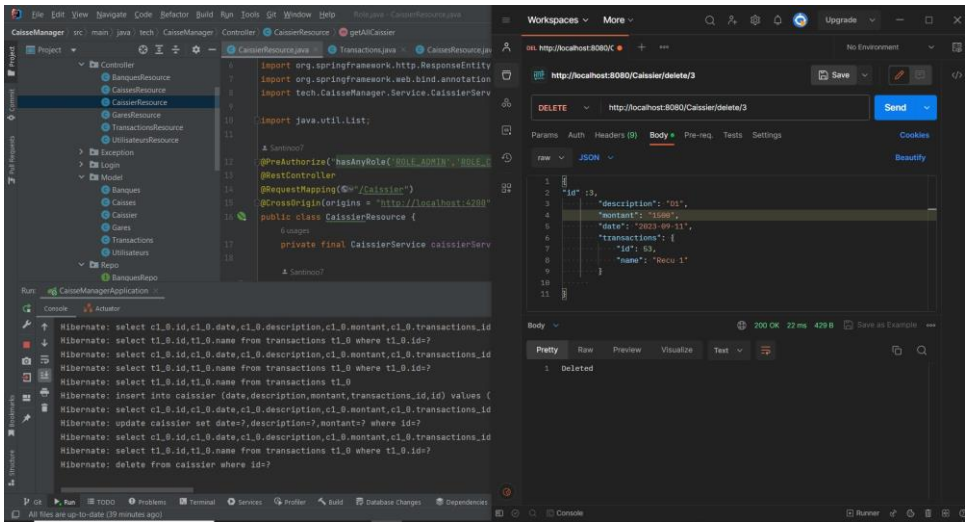


Figure 4.67 – Test suppression d'opération.

4.2 Conclusion

A ce stade, notre application est entièrement réalisée, et nous avons donc réussi à produire une solution offrant toutes les fonctionnalités telles que nous les avions identifiées lors de l'analyse des besoins du client. Elle répond de façon opérationnelle à la gestion de la flotte automobile et prête à son déploiement dans un environnement de production.

Conclusion et perspectives

En conclusion, le projet de "Gestion des Opérations de Caisses" visait à concevoir et développer une solution innovante pour optimiser la gestion des opérations liées aux caisses. Notre équipe s'est penchée sur la mise en place d'un système robuste et efficace.

Pour atteindre cet objectif, nous avons utilisé un ensemble de technologies modernes. Le backend a été construit avec Spring Boot, offrant ainsi une architecture solide et hautement flexible pour le traitement des opérations serveur. L'utilisation d'Angular comme Framework frontend a permis la création d'une interface utilisateur dynamique et interactive. L'intégration d'Angular Materials a joué un rôle clé dans la création d'une interface à la fois conviviale et attrayante.

Dans le processus de développement, nous avons également utilisé des outils tels que Postman pour le test des API, MySQL pour la gestion de base de données, et UML pour la modélisation et la conception. Le code a été écrit en Java et TypeScript, avec Visual Studio Code et IntelliJ comme environnements de développement intégrés (IDE).

Au cours du projet, nous avons fait face à quelques défis, notamment l'intégration de certaines bibliothèques et modules. Cependant, grâce à la communauté active des développeurs et aux documentations complètes disponibles, en particulier pour Spring Boot et Angular, nous avons pu surmonter ces obstacles avec succès.

Nous espérons que notre solution apportera une contribution significative au domaine de la gestion des opérations de caisses. Nous encourageons également d'autres développeurs à utiliser notre travail comme source d'inspiration pour créer des applications encore plus innovantes et conviviales.

Enfin, nous tenons à exprimer notre gratitude envers tous ceux qui ont contribué à la réussite de ce projet.

Netographie

[1] *SNCFT*. URL : https://sncft.com.tn/.

[2] *StarUML*. URL : https://staruml.io/.

[3] *Git*. URL : https://git-scm.com/.

[4] *Postman*. URL : https://www.postman.com/company/about-postman/.

[5] *AngularJS*. URL : https://angular.io/.

[6] *Angular Material*. URL : https://material.angular.io/.

[7] *Spring Boot*. URL : https://spring.io/projects/spring-boot/.

[8] *Visual Studio Code*. URL : https://code.visualstudio.com/.

[9] *IntelliJ IDEA*. URL : https://www.jetbrains.com/idea/.

[10] *MySQL*. URL : https://www.mongodb.com/.

[11] *Java*. URL : https://https://www.java.com/fr/.

[12] *TypeScript*. URL : https://www.typescriptlang.org/.

[13] *HTML*. URL : https://developer.mozilla.org/fr/docs/Web/HTML/.

[14] *CSS*. URL : https://developer.mozilla.org/fr/docs/Web/CSS/.

[15] *ChatGPT*. URL : https://chat.openai.com/.

[16] *Bootstrap*. URL : https://getbootstrap.com.

[17] *JWT*. URL : https://jwt.io/.

Annexes

Annexe 1. - Jeton JWT :

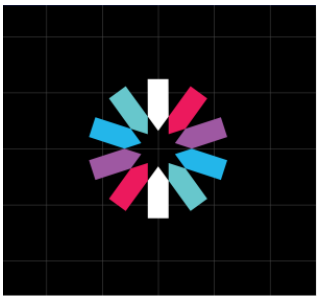


Figure 4.68 – Logo JWT

JWT [17] signifie JSON Web Token. C'est une norme ouverte pour transmettre de manière sécurisée des informations entre différentes parties sous la forme d'un objet JSON. Les JWT sont couramment utilisés pour l'authentification et l'autorisation dans les applications web et les APIs.

Figure 4.69 – Structure du jeton JWT

Un jeton se compose de trois parties :

- Un en-tête (header), utilisé pour décrire le jeton. Il s'agit d'un objet JSON.
- Une charge utile (payload) qui représente les informations embarquées dans le jeton. Il s'agit également d'un objet JSON.
- Une signature numérique.

ملخص

الفكرة التي قمنا بتطويرها تمثل تقدمًا كبيرًا في تحسين عمليات الصرافين. إنها تتجسد من خلال نشر منصة شاملة لإدارة النقدية، صممت بعناية فائقة لتلبية متطلباتهم التشغيلية واحتياجاتهم. في صميم هذا الحل، تكمن تكامل تطبيقات الكمبيوتر الحديثة والجذابة من الناحية الجمالية. تتميز هذه التطبيقات بتكييفها السلس مع أحدث أنظمة التشغيل القائمة. تهدف هذه النهج الحديث بحزم إلى ضمان تجربة مستخدم سلسة وبديهية، مع ضمان الأداء الأمثل وكفاءة التشغيل الفائقة. يمثل هذا النهج الابتكاري قفزة حقيقية في كفاءة التعامل مع المشكل المطروح، ويعد بإدارة المعاملات بشكل أسهل وأكثر كفاءة.

RESUME

La solution que nous avons développée marque une avancée significative dans l'optimisation des opérations des caissiers. Elle se concrétise par le déploiement d'une plateforme de gestion de caisses globale, minutieusement élaborée pour répondre de manière spécifique à leurs exigences et besoins opérationnels.

Au cœur de cette solution se trouve l'intégration de logiciels de bureau à la fois contemporains et esthétiquement attrayants. Ces applications se distinguent par leur adaptation parfaite aux derniers systèmes d'exploitation en vigueur. Cette orientation résolument tournée vers la modernité vise à garantir une expérience utilisateur des plus fluides et intuitives, tout en assurant une performance optimale et une efficacité opérationnelle inégalée. Cette approche novatrice représente un véritable bond en avant dans l'efficacité des opérations de caisse, promettant une productivité accrue et une gestion plus fluide et efficiente des transactions.

ABSTRACT

The solution we have developed marks a significant advancement in optimizing cashier operations. It materializes through the deployment of a comprehensive cash management platform, meticulously crafted to specifically address their operational requirements and needs.

At the core of this solution lies the integration of both contemporary and aesthetically appealing desktop software. These applications stand out for their seamless adaptation to the latest operating systems in use. This resolutely modern-oriented approach aims to ensure the smoothest and most intuitive user experience while guaranteeing optimal performance and unparalleled operational efficiency. This innovative approach represents a true leap forward in cash handling efficiency, promising increased productivity and smoother, more efficient transaction management.