e

Mémoire du Projet de Fin d’Année

Spécialité**: Génie informatique**

Thème**:**

**Conception et réalisation d'une plateforme des enquêtes de satisfaction.**

**Soutenu par : Sous la direction de :**

**TANAFAAT Abdessamad M. AKSASSE Hamid,** Tuteur pédagogique

**ROUCHDI Ilias M. Aymen BOUAMEUR,** Tuteur professionnel

Soutenu le ../../2024, devant la commission du jury :

**M. AKSASSE Hamid**

**Insérer M Prénom NOM**

**Insérer M Prénom NOM**

Année Universitaire 2023-2024

**Dédicace**

Nous dédions ce travail aux personnes les plus chères à nos cœurs :

À nos chers parents, qui nous ont donné tant de soutien et d'encouragement, symbolisant pour nous le sacrifice et la source d'où naît la lumière qui éclaire notre vie.

À nos frères et sœurs, nous vous remercions pour votre amour inconditionnel. Que Dieu vous garde. Nous vous aimons et vous souhaitons une vie pleine de succès et de réussite.

À nos amis et amies qui représentent pour nous le sens de la sincérité et de la fidélité.

À nos professeurs qui nous ont guidés sur le chemin du succès. Leurs enseignements et leurs conseils nous ont permis de mieux apprécier la formation dispensée au sein de l'école.

**Abdessamad, Ilias**

**Remerciements**

Nous tenons à exprimer notre profonde gratitude et notre respectueuse reconnaissance à notre professeur encadrant, Monsieur **AKSASSE Hamid**, pour sa bonne volonté d'accepter de nous encadrer, pour tout le temps qu'il nous a accordé et pour tous ses précieux et judicieux conseils qu'il nous a prodigués.

Nous ne pouvons pas oublier également de remercier Monsieur **Ayman**, notre encadrant au sein de l'entreprise **MAJOR MEDIA**, qui a toujours été présent pour orienter et stimuler nos recherches avec une patience d'or et une attention énorme. Avec lui, nous avons eu des discussions enrichissantes tout au long de notre stage.

Ensuite, nous tenons à remercier l'ensemble du **corps administratif et professoral de L’École Nationale des Sciences Appliquées d'Agadir** pour leur contribution, de près ou de loin, à notre formation dans l'établissement.

Enfin, nous tenons à exprimer notre gratitude envers toutes les personnes qui, bien que non mentionnées dans ce rapport, ont été d'un très grand appui dans l'accomplissement de notre travail.

**Résumé**

L'objectif principal de notre projet était de développer une plateforme en ligne dédiée à la gestion des enquêtes de satisfaction pour les entreprises. Cette plateforme offre deux parties : une partie dédiée à l'administration et une autre réservée aux propriétaires d'entreprises.

Du côté de l'administration, les administrateurs ont la responsabilité de gérer les utilisateurs, les rôles et les privilèges, assurant ainsi une attribution efficace des autorisations et des tâches au sein de la plateforme. De plus, ils peuvent créer des modèles par défaut à utiliser par le propriétaire, offrant ainsi une structure prête à l'emploi pour les enquêtes.

Quant aux propriétaires d'entreprises, ils peuvent utiliser ces modèles pour créer, gérer et analyser des enquêtes personnalisées, facilitant ainsi le processus de collecte de feedback client et de prise de décision basée sur les données. Grâce à des fonctionnalités telles que les codes QR et les programmes de fidélité, les clients peuvent facilement accéder aux enquêtes, exprimer leur satisfaction et participer à programmes de fidélité d’une manière rapide.

En somme, notre projet vise à simplifier le processus de collecte de feedback client et à promouvoir l'amélioration continue des services offerts par les entreprises.

**Mots-clés :** Plateforme en ligne, Gestion des enquêtes, Satisfaction client, Codes QR, Programmes de fidélité, Administrateurs spécifiques, Modèles par défaut.

**Abstract**

The main objective of our project was to develop an online platform dedicated to the management of satisfaction surveys for companies. This platform offers two distinct parts: a part dedicated to administration and another reserved for business owners.

On the administration side, administrators are responsible for managing users, roles and privileges, ensuring efficient assignment of permissions and tasks within the platform. Additionally, they can create default templates for the owner to use, providing a ready-made structure for surveys.

As for business owners, they can use these templates to create, manage and analyze custom surveys, facilitating the process of collecting customer feedback and making data-driven decisions. With features like QR codes and loyalty programs, customers can easily access surveys, express satisfaction, and participate in loyalty programs in a quick manner.

In short, our project aims to simplify the process of collecting customer feedback and to promote the continuous improvement of the services offered by companies.

**Keywords**: Online platform, Survey management, Customer satisfaction, QR codes, Loyalty programs, Specific administrators, Default templates.

**Liste des abréviations**

|  |  |
| --- | --- |
| **Abréviation** | **Désignation** |
| **QR** | Quick Response (code) |
| **REST** | Representational State Transfer |
| **NPM** | Node Package Manager |
| **API** | Application Programming Interface |
| **JWT** | JSON Web Token |
| **CSS** | Cascading Style Sheets |
| **SQL** | Structured Query Language |
| **POSTMAN** | Outil de test d'API |
| **IDEA** | Integrated Development Environment |
| **UI** | User Interface (Interface Utilisateur) |
| **UX** | User Experience (Expérience Utilisateur) |
| **SEO** | Search Engine Optimization |
| **SEA** | Search Engine Advertising |
| **SARL** | Société à Responsabilité Limitée |
| **NPS** | Net Promoter Score |
| **SCRUM** | Méthodologie de développement Agile |
| **JSON** | JavaScript Object Notation |
| **HTTP** | Hypertext Transfer Protocol |
| **URI** | Uniform Resource Identifier |
| **UML** | Unified Modeling Language |
| **JPA** | Java Persistence API |
| **HMAC** | Hash-based Message Authentication Code |
| **RSA** | Rivest-Shamir-Adleman (cryptographie asymétrique) |
| **ECDSA** | Elliptic Curve Digital Signature Algorithm |
| **DOM** | Document Object Model |
| **HTML** | Hypertext Markup Language |
|  |  |
|  |  |

Table des matières

[Table des matières 8](#_Toc164716776)

[Liste des figures 11](#_Toc164716777)

[Liste des tables 14](#_Toc164716778)

[**Introduction générale** 15](#_Toc164716779)

[**Chapitre 1** 16](#_Toc164716780)

[**ETUDE PRÉALABLE** 16](#_Toc164716781)

[1.1 Présentation de l’entreprise d’accueil : 17](#_Toc164716782)

[1.1.1 Définition de l’entreprise 17](#_Toc164716783)

[1.1.2 Services proposés 18](#_Toc164716784)

[1.1.3 Fiche technique 18](#_Toc164716785)

[1.2 Présentation du projet : 18](#_Toc164716786)

[**1.2.1** **Contexte du projet** 19](#_Toc164716787)

[**1.2.2** **Objectif du projet** 19](#_Toc164716788)

[1.2.3 Problématique 19](#_Toc164716789)

[1.2.4 Solution proposée 19](#_Toc164716790)

[1.2.5 Description du projet 20](#_Toc164716791)

[1.3.1 Gestion de projet 23](#_Toc164716792)

[1.3.2 Méthodologie de travail 23](#_Toc164716793)

[1.3.3 Diagramme de Gant Prévisionnel 27](#_Toc164716794)

[**1.4.1** **Architecture de la partie Back end** 28](#_Toc164716795)

[**1.4.2** **Architecture de la partie Front end** 29](#_Toc164716796)

[**1.4.3** **Les Web services REST** 29](#_Toc164716797)

[**Chapitre 2** 31](#_Toc164716798)

[**Sprint 0 : MISE EN ŒUVRE DU PROJET** 31](#_Toc164716799)

[2.1 Identification des acteurs 32](#_Toc164716800)

[2.1 Analyse des besoins 33](#_Toc164716801)

[2.2.1 Les besoins fonctionnels 33](#_Toc164716802)

[2.2.2 Les besoins non fonctionnels 36](#_Toc164716803)

[2.3 Backlog de produit 37](#_Toc164716804)

[2.4 Planification des releases 39](#_Toc164716805)

[2.5 Diagramme de cas d’utilisation global 40](#_Toc164716806)

[**Chapitre 3** 43](#_Toc164716807)

[**ETUDE TECHNOLOGIQUE** 43](#_Toc164716808)

[3.1 Outils de Modélisation 44](#_Toc164716809)

[3.2 Outils de Développement Back end 44](#_Toc164716810)

[3.3 Outils de Développement Front end 46](#_Toc164716811)

[3.4 Outils de gestion de version 47](#_Toc164716812)

[3.5 Base de données 48](#_Toc164716813)

[3.6 Framework de gestion des dépendances NPM 49](#_Toc164716814)

[3.7 Logiciel de Test : Postman 49](#_Toc164716815)

[3.8 Environnements de développement intégrés 49](#_Toc164716816)

[**Chapitre 4** 52](#_Toc164716817)

[**RELEASE 1** 52](#_Toc164716818)

[4.1 Organisation des Sprints 53](#_Toc164716819)

[4.2 Backlog des sprints pour la release 1 53](#_Toc164716820)

[4.3 Sprint 1 : Gestion des administrateurs 55](#_Toc164716821)

[4.4 Sprint 2 : rôles et des privilèges 66](#_Toc164716822)

[**Chapitre 5** 77](#_Toc164716823)

[**RELEASE 2** 77](#_Toc164716824)

[5.1 Organisation des Sprints 78](#_Toc164716825)

[5.2 Backlog des sprints pour la release 2 78](#_Toc164716826)

[5.3 Sprint 1 : propriétaires et entreprises 80](#_Toc164716827)

[5.4 Sprint 2 : paramétrage de l'application 86](#_Toc164716828)

[**Chapitre 6** 92](#_Toc164716829)

[**RELEASE 3** 92](#_Toc164716830)

[6.1 Organisation des Sprints 93](#_Toc164716831)

[6.2 Backlog des sprints pour la release 3 93](#_Toc164716832)

[6.3 Sprint 1 : Programmes de fidélité 95](#_Toc164716833)

[6.4 Sprint 2 : Analyse des statistiques 99](#_Toc164716834)

[**Annexes** 107](#_Toc164716835)

[**Annexe A : UML** 107](#_Toc164716836)

[**Annexe B : Diagramme de GANTT** 107](#_Toc164716837)

[**Annexe C : Diagramme de GANTT** 107](#_Toc164716838)

[**Annexe A** 108](#_Toc164716839)

[**UML** 108](#_Toc164716840)

[**Annexe B** 110](#_Toc164716841)

[**Diagramme de GANTT** 110](#_Toc164716842)

Liste des figures

[*Figure 1 : Situation géographique* 17](#_Toc164711232)

[*Figure 2 : trois contraintes de projet* 23](#_Toc164711233)

[*Figure 3: la méthodologie agile* 24](#_Toc164711234)

[*Figure 4 : Méthode Agile Scrum* 26](#_Toc164711235)

[*Figure 5: diagramme de Gantt prévisionnel.* 27](#_Toc164711236)

[*Figure 6: Architecture globale de l'application* 28](#_Toc164711237)

[*Figure 7: Hiérarchie des acteurs* 33](#_Toc164711238)

[*Figure 8: List Trello* 40](#_Toc164711239)

[*Figure 9:Diagramme de cas d'utilisation général* 41](#_Toc164711240)

[*Figure 10: Logo Astah* 44](#_Toc164711241)

[*Figure 11:Logo du la plateforme Spring* 44](#_Toc164711242)

[*Figure 12:Logo spring boot* 45](#_Toc164711243)

[*Figure 13: Logo Spring Security* 45](#_Toc164711244)

[*Figure 14: Logo Spring Data* 46](#_Toc164711245)

[*Figure 15:Logo JWT* 46](#_Toc164711246)

[*Figure 16: Logo React* 47](#_Toc164711247)

[*Figure 17: Logo Axios* 47](#_Toc164711248)

[*Figure 18: Tailwind CSS Logo* 47](#_Toc164711249)

[*Figure 19: Logo GitHub* 48](#_Toc164711250)

[*Figure 20:Logo Git* 48](#_Toc164711251)

[*Figure 21: Logo MySQL* 48](#_Toc164711252)

[*Figure 22: Logo NPM* 49](#_Toc164711253)

[*Figure 23: Logo POSTMAN* 49](#_Toc164711254)

[*Figure 24: Logo Intellij IDEA* 50](#_Toc164711255)

[*Figure 25: Logo Visual Studio Code* 50](#_Toc164711256)

[*Figure 26: Diagramme de cas d'utilisation du sprint 1 : Gestion des administrateurs* 56](#_Toc164711257)

[*Figure 27: Diagramme des classes du Sprint 1* 58](#_Toc164711258)

[*Figure 28: Diagramme de séquence :S'authentifier* 59](#_Toc164711259)

[*Figure 29: Diagramme de séquence : Créer Administrateur* 60](#_Toc164711260)

[*Figure 30: Interface du résultat de la Release 1 :* 61](#_Toc164711261)

[*Figure 31:Interface authentification* 61](#_Toc164711262)

[*Figure 32: Interface authentification en mode sombre* 61](#_Toc164711263)

[*Figure 33: Interface authentification avec un login et mot de passe invalide* 62](#_Toc164711264)

[*Figure 34: Interface Réinitialiser mot de passe.* 62](#_Toc164711265)

[*Figure 35: Interface saisir le nouveau mot de passe* 63](#_Toc164711266)

[*Figure 36: Lien pour Réinitialiser mot de passe* 63](#_Toc164711267)

[*Figure 37: Interface Modifier profile* 64](#_Toc164711268)

[*Figure 38: Interface Modifier Profile* 64](#_Toc164711269)

[*Figure 39: Interface pour la table des administrateurs* 65](#_Toc164711270)

[*Figure 40: Interface créer les administrateurs* 65](#_Toc164711271)

[*Figure 41: Interface créer les administrateurs* 66](#_Toc164711272)

[*Figure 42: Diagramme de cas d'utilisation du sprint 1 : Gestion des rôles et des privilèges* 67](#_Toc164711273)

[*Figure 43: Diagramme des classes du Sprint 2* 68](#_Toc164711274)

[*Figure 44: Diagramme de séquence : Ajouter Rôle* 69](#_Toc164711275)

[*Figure 45: Diagramme de séquence : Créer privilège* 70](#_Toc164711276)

[*Figure 46: Diagramme de séquence : Attribuer rôle à l’administrateur* 71](#_Toc164711277)

[*Figure 47: Diagramme de séquence : Attribuer des privilèges à un rôle :* 72](#_Toc164711278)

[*Figure 48: Interface créer privilège* 73](#_Toc164711279)

[*Figure 49: Interface créer rôle* 74](#_Toc164711280)

[*Figure 50: Interface créer rôle* 74](#_Toc164711281)

[*Figure 51: Interface créer rôle* 74](#_Toc164711282)

[*Figure 52: Interface créer rôle* 75](#_Toc164711283)

[*Figure 53: Interface Attribuer rôle à un utilisateur* 75](#_Toc164711284)

[*Figure 54: Interface Attribuer rôle à un utilisateur* 76](#_Toc164711285)

[*Figure 55: Diagramme de cas d'utilisation du sprint 1 : Gestion des propriétaires et des entreprises* 81](#_Toc164711286)

[*Figure 56: Diagramme des classes du Sprint 1* 83](#_Toc164711287)

[*Figure 57: Diagramme de séquence : Créer une catégorie d'entreprise* 84](#_Toc164711288)

[*Figure 58: Diagramme de séquence : Activer Compte de propriétaire* 85](#_Toc164711289)

[*Figure 59: Diagramme de cas d'utilisation du sprint 2 : Configuration et paramétrage de l'application* 86](#_Toc164711290)

[*Figure 60: Diagramme des classes du Sprint 2* 88](#_Toc164711291)

[*Figure 61: Diagramme de séquence : Ajouter langue* 89](#_Toc164711292)

[*Figure 62: Diagramme de séquence : Ajouter Modèle* 90](#_Toc164711293)

[*Figure 63: Diagramme de cas d'utilisation du sprint 1 : Gestion des Programmes de fidélité* 95](#_Toc164711294)

[*Figure 64: Diagramme des classes du Sprint 1* 97](#_Toc164711295)

[*Figure 65: Diagramme de séquence : Ajouter programme de fidélité* 98](#_Toc164711296)

[*Figure 66: Diagramme de cas d'utilisation du sprint 2 : Analyse des statistiques* 99](#_Toc164711297)

[*Figure 67: Diagramme des classes du Sprint 2* 101](#_Toc164711298)

[*Figure 68: diagramme de séquence : Écrire commentaire* 102](#_Toc164711299)

Liste des tables

[*Tableau 1: Fiche technique de l‘entreprise* 18](#_Toc164639046)

[*Tableau 2: Backlog de produit* 37](#_Toc164639047)

[*Tableau 3: Planification des releases* 39](#_Toc164639048)

[*Tableau 4: Organisation des sprints de la release 1* 53](#_Toc164639049)

[*Tableau 5 : Backlog des sprints pour release 1* 54](#_Toc164639050)

[*Tableau 6: Description Textuelle : Authentification* 57](#_Toc164639051)

[*Tableau 7: Description Textuelle : Gestion des rôles et privilèges* 68](#_Toc164639052)

[*Tableau 8: Organisation des sprints de la release 2* 78](#_Toc164639053)

[*Tableau 9:Backlog des sprints pour release 2* 80](#_Toc164639054)

[*Tableau 10: Description Textuelle : Gestion des propriétaires et des entreprises* 82](#_Toc164639055)

[*Tableau 11: Description Textuelle : Configuration et paramétrage de l'application* 88](#_Toc164639056)

[*Tableau 12: Organisation des sprints de la release 3* 93](#_Toc164639057)

[*Tableau 13: Backlog des sprints pour release 3* 94](#_Toc164639058)

[*Tableau 14: Description Textuelle : Gestion des Programmes de fidélité* 97](#_Toc164639059)

[*Tableau 15: Description Textuelle : Analyse des statistiques* 101](#_Toc164639060)

**Introduction générale**

La satisfaction client est cruciale pour toute entreprise qui souhaite se développer. Elle reflète la perception des clients sur la qualité des produits ou services d'une entreprise, ainsi que sur leur expérience globale. En effet, les clients satisfaits sont plus susceptibles de rester fidèles à une marque, de recommander ses produits ou services, et ainsi de contribuer à sa croissance sur le long terme.

Ainsi, dans le cadre de notre projet, nous visons à élaborer une solution informatique dédiée à la gestion des enquêtes de satisfaction pour les entreprises. Cette solution permettra aux entreprises de toutes tailles et de tous secteurs d'activité de créer, gérer et analyser des enquêtes personnalisées afin d'obtenir des retours clients précieux. En fournissant aux entreprises les outils nécessaires pour recueillir les opinions de leurs clients de manière structurée et systématique, notre solution vise à améliorer la qualité des produits ou services proposés, à renforcer la fidélité client et à stimuler la croissance de l'entreprise.

Ce rapport détaillera l'ensemble du processus de développement du système de gestion des enquêtes de satisfaction. Nous aborderons les différentes phases du projet, de la conception initiale à la mise en œuvre finale, en passant par le développement, les tests et la validation. Nous mettrons également l'accent sur les différents choix techniques effectués.

**Chapitre 1**

**ETUDE PRÉALABLE**

Dans ce chapitre, nous présenterons de manière générale l'objectif et l'environnement du stage. Tout d'abord, nous aborderons l'entreprise "Major Media" au sein de laquelle le projet a été réalisé. Ensuite, nous exposerons la problématique à résoudre ainsi que la solution que nous proposons. Enfin, nous fournirons une description du projet dans son ensemble.

# Présentation de l’entreprise d’accueil :

## Définition de l’entreprise

Major Media est une agence spécialisée dans le marketing digital, le développement informatique et le design graphique.  
 Son objectif principal est de renforcer la présence en ligne de ses clients et de garantir leur succès en leur fournissant une gamme variée de services. Cela inclut la création de sites web, la gestion des réseaux sociaux, ainsi que le développement d'applications sur mesure. En mettant l'accent sur la transformation digitale et la création d'expériences uniques pour les clients, Major Media s'efforce de répondre aux besoins spécifiques de chaque entreprise et de les aider à atteindre leurs objectifs en ligne.

Voici la situation géographique de l'entreprise

******

*Figure 1 : Situation géographique*

## Services proposés

Major Media offre à ses clients une gamme diversifiée de services pour répondre à leurs besoins spécifiques. Ces services comprennent :

* **Développement informatique** : création de sites web et d'applications mobiles sur mesure.
* **Graphic design :** Conception de publicités sociales, UI/UX
* **Digital marketing** : Publicités sociales, SEO/SEA

## Fiche technique

|  |  |
| --- | --- |
| **Nom des dirigeants** | ARTUR CARDOSO, BIRAME FAYE |
| **Forme juridique** | SARL |
| **Site web** | https://majormedia.ma |
| **Adresse** | Bureau N°45, immeuble yasmine, entrée A, 4 -ème étage, avenue moukawama, Agadir 80000 |
| **Année de création** | 2020 |
| **E-mail** | [Contact@majormedia.com](mailto:Contact@majormedia.com) |
| **Téléphone** | +212 80 85 77 710 |

*Tableau 1: Fiche technique de l‘entreprise*

# Présentation du projet :

* + 1. **Contexte du projet**

Le projet de fin d'année consiste à concevoir une application de gestion des enquêtes de satisfaction dans le cadre de notre cursus d'ingénierie à l'École Nationale des Sciences Appliquées d’Agadir. Notre équipe se concentre spécifiquement sur le développement de la partie administrative de cette application.

Nous travaillons sur la création d'une application permettant de gérer et d'administrer efficacement l'ensemble du processus de gestion des enquêtes de satisfaction.

* + 1. **Objectif du projet**

L'objectif du projet est de développer une application de gestion des enquêtes de satisfaction afin de permettre aux entreprises de collecter, gérer et analyser les retours clients de manière efficace, ce qui les aidera à améliorer la qualité de leurs produits ou services et à renforcer leur relation client.

### Problématique

Les entreprises ont adopté l'utilisation des méthodes traditionnelles telles que les questionnaires papier, les e-mails et les formulaires en ligne simples pour collecter les retours clients et les analyser. Cependant, ces approches présentent des limitations importantes. Par exemple, les questionnaires papier et les formulaires en ligne peuvent être limités en termes de personnalisation et d'analyse de données. De plus, la collecte manuelle des réponses peut être sujette à des erreurs humaines

### Solution proposée

La solution envisagée consiste à proposer le développement d'une application web dédié à la gestion de la satisfaction client.

Cette application offre aux propriétaires d'entreprises la possibilité de créer des enquêtes sur mesure en sélectionnant parmi une gamme de modèles préétablis ou en en créant de nouveaux.

En automatisant le processus de collecte des réponses et d'analyse des données, elle permet de gagner du temps et des ressources précieuses. De plus, grâce à des méthodes de distribution variées telles que les codes QR, les liens partageables ou les e-mails, les enquêtes peuvent être facilement accessibles aux clients, facilitant ainsi la collecte de leurs réponses de manière efficace et transparente.

Grâce à cette solution, les entreprises pourront évaluer la satisfaction de leurs clients de manière rapide et efficace, ce qui contribuera à améliorer la qualité de leurs services et à renforcer la fidélité client.

### Description du projet

Le projet aura comme objectif de développer une plateforme de gestion d'enquêtes de satisfaction et de programmes de fidélité. Cette plateforme vise à fournir aux propriétaires d'entreprises un outil complet pour collecter des retours clients, analyser les données et mettre en œuvre des programmes de fidélité efficaces.

Dans ce cadre, la plateforme permettrait :

* Les clients de sélectionner leur langue préférée dès le début de l'enquête et fournir des informations de base telles que leur nom, leur âge, etc. pour une expérience personnalisée.
* Les clients d’exprimer leur satisfaction sur les différents services à l'aide d'émoticônes, offrant un retour visuel instantané.
* Les clients de Participer de manière optionnelle à un programme de fidélité pour bénéficier d'avantages supplémentaires tels que des réductions, des récompenses spéciales ou des privilèges exclusifs.
* Les clients d’évaluer globalement leur expérience avec l'entreprise en utilisant l’Indice de Recommandation Net (NPS), où ils attribuent une note de 0 à 10 selon leur propension à recommander l'entreprise à d'autres personnes.

En fonction de leurs réponses, ils sont classés en promoteurs, passifs ou détracteurs, permettant ainsi à l'entreprise de mesurer sa satisfaction client et d'identifier les domaines à améliorer.

* Les propriétaires d'entreprise de créer et gérer les détails de leur entreprise, y compris les informations clés telles que les coordonnées, les heures d'ouverture et les descriptions.
* Les propriétaires d'entreprise de créer des enquêtes en sélectionnant parmi une gamme de modèles préétablis ou en en créant de nouvelles adaptées à leurs besoins spécifiques.
* Les propriétaires d'entreprise de Gérer les programmes de fidélité tels que Client Ambassador, Mystery Box, Coupon code et Raffle, favorisant ainsi l'engagement et la fidélité des clients
* Les propriétaires d’entreprise de générer des QR codes associés à leurs enquêtes pour une collecte de données facile.
* Les administrateurs supérieurs de créer des comptes d’administrateurs, ce qui permet une répartition efficace des tâches dans la plateforme.
* Les administrateurs supérieurs de gérer les utilisateurs de la plateforme en attribuant des rôles et des autorisations, garantissant ainsi la sécurité et la confidentialité des données.
* Les administrateurs de consulter les listes des entreprises enregistrées, notamment en visualisant, et en archivant les informations, permettant ainsi une gestion efficace.

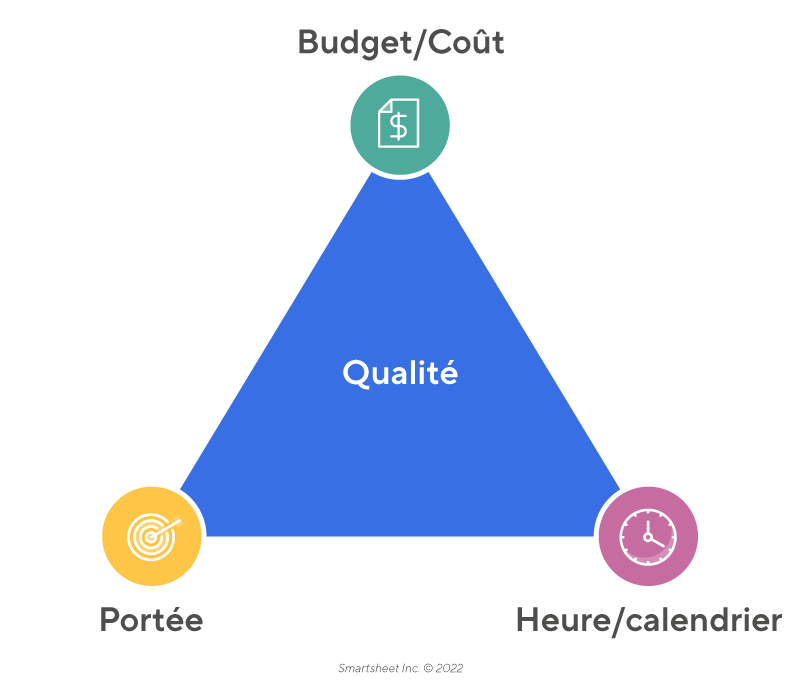
Dans la configuration et le paramétrage de l'application, l'administrateur a la possibilité de :

* Créer une gamme de modèles par défaut pour les enquêtes de satisfaction, offrant ainsi une sélection préétablie prête à être utilisée par les propriétaires.
* Ajouter et modifier les catégories d'entreprises et leurs types, permettant aux propriétaires de choisir facilement la catégorie et le type appropriés lors de la création des enquêtes.
* Ajouter et modifier les zones de service et leurs types, ce qui donne aux propriétaires la flexibilité d'inclure des zones de service spécifiques dans les enquêtes de satisfaction.
* Créer des programmes de fidélité prédéfinis, offrant ainsi une sélection de programmes à utiliser par les propriétaires dans le cadre de leurs activités de fidélisation.

* 1. **Méthodologie de travail**

### Gestion de projet

La gestion du projet est primordiale et complexe puisqu’elle sert à évoluer les environnements qui amène à une évolution constante. Par conséquent, nous devons avoir plusieurs compétences c'est à dire la maitrise des techniques du gestion de projet, la gestion d'équipe, le bon contact et l'échange avec le client et finalement bien comprendre les besoins et les spécificités du projet. L'objectif de la gestion de projet est de pouvoir le terminer dans les délais toute en respectant le budget alloué. Pour atteindre l'objectif, on doit bien comprendre et prendre en considération les trois principales contraintes qui constituent le projet.



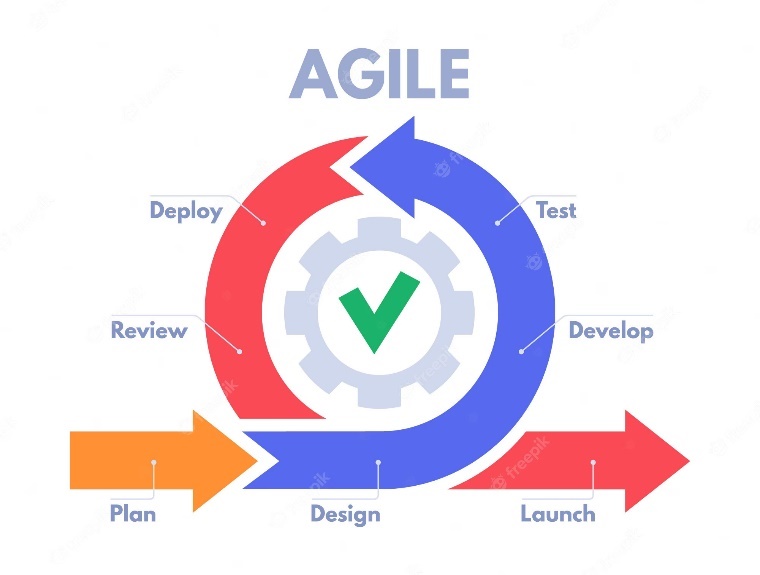
*Figure 2 : trois contraintes de projet*

### Méthodologie de travail

Le choix d'une méthodologie de développement de logiciel est un processus critique dans le cycle de vie d'un projet de développement.

C'est la structure de planification et de contrôle de la procédure de création d'un système d'information. Le choix dépend de la nature du projet lui-même et de sa taille et de sa complexité.

Il est recommandé d'opter pour une méthodologie agile si la spécification et les exigences du projet ne sont pas complètement définies au début et devraient être modifiées ultérieurement.



*Figure 3: la méthodologie agile*

Agile est une méthodologie utilisée pour gérer un projet en le décomposant en plusieurs étapes. Elle favorise la collaboration continue entre les parties prenantes et encourage l'amélioration itérative à chaque étape du processus.Top of Form

**La Méthodologie Scrum**

La plus célèbre des méthodologies de gestion de projets déclinées de la méthode Agile relève de la “Scrum”, autrement dit la “mêlée” dans le langage rugby. Le responsable de projet s’appelle ainsi le “SCRUM Master”.

Cette approche s’organise autour de cycles courts, qu’on appelle communément des itérations. En langage Scrum, une itération se nomme un “sprint”.

À chaque nouveau sprint, l’équipe projet se rassemble pour lister les tâches à exécuter. Cette liste s’appelle le “sprint Backlog”.

Les exigences pour chaque sprint sont définies avant le début du sprint, et les résultats sont évalués à la fin de chaque cycle de sprint.

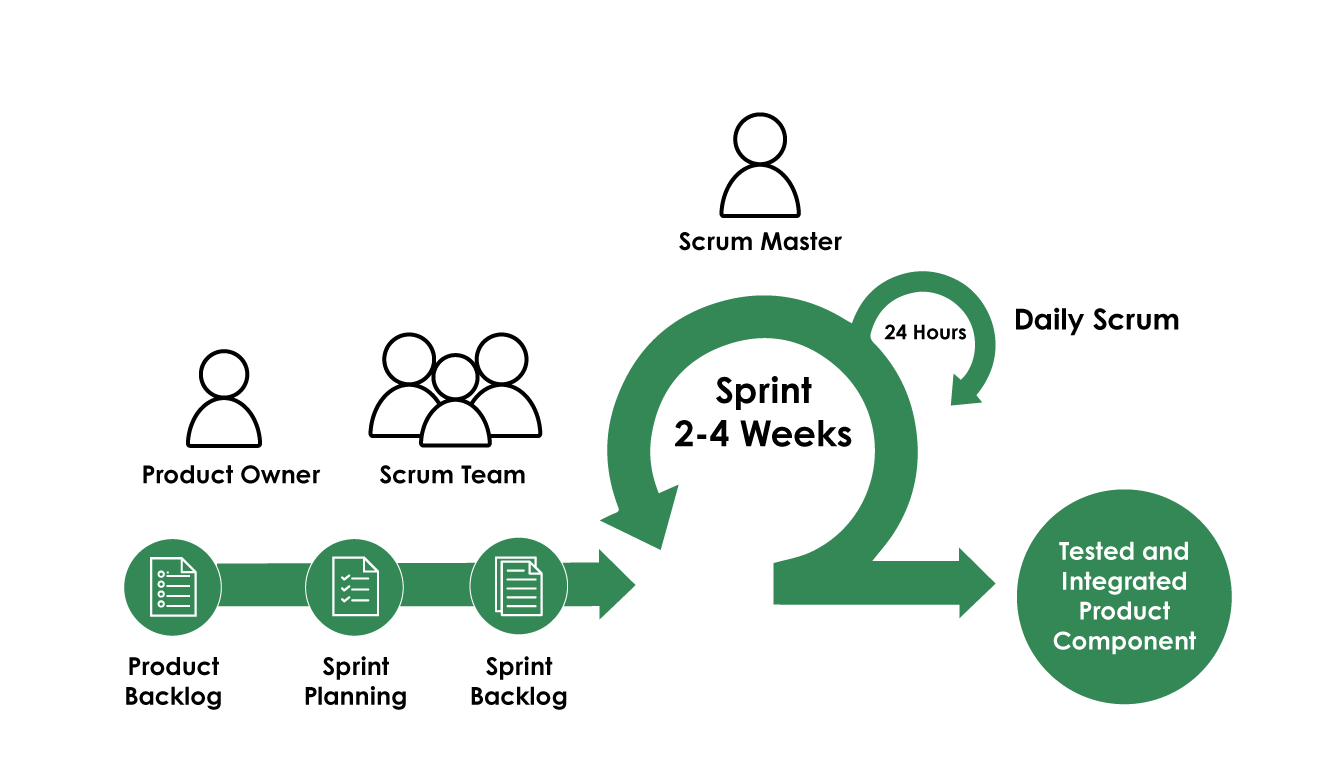
Après validation du sprint, l'équipe peut alors entamer un nouveau cycle. Cette approche offre un environnement plus flexible qui permet aux développeurs de réagir rapidement aux problèmes émergents.

Scrum se compose de trois rôles principaux :

* Le Product Owner : cette personne détient l'autorité, la vision et la disponibilité nécessaires. Elle est chargée de maintenir une communication continue des exigences avec l'équipe de développement.
* Scrum Master : Le Scrum master agit en tant que facilitateur pour le propriétaire du produit et l'équipe. Son rôle consiste à éliminer tous les obstacles qui empêchent l'équipe d'atteindre ses objectifs.
* L'équipe Scrum : L'équipe est composée des développeurs du projet. Cette équipe compte entre trois et neuf membres délestés, idéalement dans une pièce protégée des distractions extérieures. L'équipe comprend généralement des architectes, des analystes, des testeurs, des experts en assurance qualité et des UI designers.

Mr **Aymen BOUAMEUR** est à la fois le Scrum Master et le Product Owner de notre projet et nous sommes l’équipe de développement.

Dans la méthodologie agile Scrum, l'objectif est de fournir un cadre de travail clair et précis par le biais d'itérations courtes, facilitant ainsi la réalisation de projets complexes.



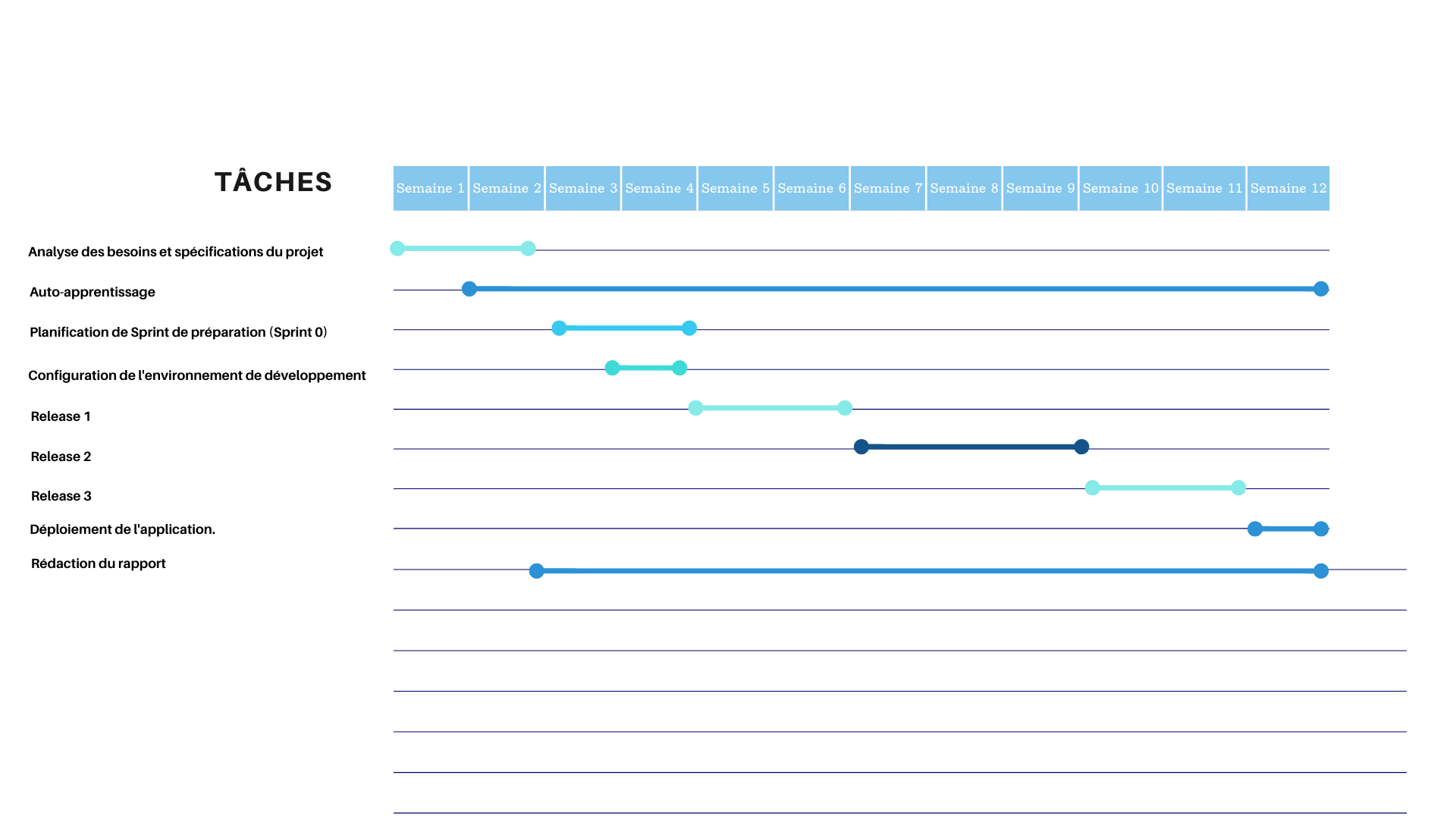
*Figure 4 : Méthode Agile Scrum*

La méthode Scrum utilise un vocabulaire spécifique pour décrire les différents rôles, événements impliqués dans le processus de développement logiciel. Parmi les termes clés, on trouve :

* **Backlog Produit :** Une liste organisée de toutes les fonctionnalités, exigences et tâches essentielles pour la réalisation du projet.
* **Sprint Backlog :** Une liste des tâches à accomplir pendant le sprint en cours.
* **Revue de Sprint :** Une réunion où l'équipe présente le travail accompli durant le sprint et recueille les commentaires des parties prenantes.
* **Rétrospective de Sprint :** Une réunion où l'équipe évalue le sprint passé pour identifier ce qui a bien fonctionné et ce qui doit être amélioré pour les futurs sprints.
* **User Stories :** Une technique de description des besoins d'un utilisateur ou d'un client du point de vue fonctionnel.
* **Sprint :** Une itération courte et limitée dans le temps, généralement de 1 à 4 semaines.
* **Daily Scrum (ou stand-up meetings) :** Une réunion quotidienne courte de l'équipe de développement, d'une durée maximale de 15 minutes.
* **Épics :** Une grande fonctionnalité ou un ensemble de fonctionnalités nécessitant plusieurs sprints pour être développé.

### Diagramme de Gant Prévisionnel

Le diagramme de Gantt prévisionnel offre une vue chronologique planifiée des différentes tâches et étapes du projet, permettant une gestion efficace du temps et des ressources.

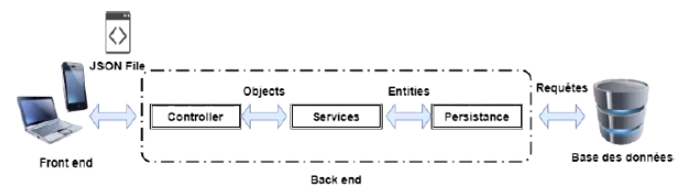
 Voici la figure représentant le diagramme Gantt prévisionnel de notre projet :

*Figure 5: diagramme de Gantt prévisionnel.*

Le diagramme de Gantt réel du projet sera partagé dans **l'annexe B**.

* 1. **Architecture de l’application**

La figure suivante résume l’architecture global de notre application



*Figure 6: Architecture globale de l'application*

**Front end** : Cette couche représente les interfaces de l'application, comprenant les éléments avec lesquels l'utilisateur peut interagir.

**Back end :** Cette partie contient les fonctionnalités de l'application et leur implémentation. Elle n'est pas visible pour les utilisateurs de l'application mais garantit le fonctionnement dynamique de celle-ci.

**Base de données :** Elle permet aux utilisateurs de gérer les données de l'application, y compris la consultation, la saisie, la modification ou la suppression, tout en assurant les droits d'accès appropriés.

* + 1. **Architecture de la partie Back end**

**Contrôleur :** Les contrôleurs représentent le point d'entrée de communication avec le Frontend. Ils gèrent les requêtes envoyées depuis le Frontend et renvoient des réponses exploitables avec les interfaces. Le rôle du contrôleur est d'envoyer les réponses après chaque action effectuée dans la partie Frontend. Dans notre application, les réponses sont au format JSON.

**Services :** Cette partie concerne la manipulation des données renvoyées de la couche persistante et l'implémentation de la logique métier de notre application.

**Persistante :** Le rôle de cette couche est d'interagir directement avec la base de données.

* + 1. **Architecture de la partie Front end**

**Components :** sont des éléments réutilisables qui représentent des parties de l'interface utilisateur, comme les boutons, les formulaires, ou les cartes.

**Pages :** différentes vues de l'application. Chaque page est généralement composée de plusieurs composants et est responsable de l'affichage et de la gestion des données spécifiques à cette vue.

**Utilitaires (Utils) :** Les utilitaires regroupent des fonctions ou des modules qui fournissent des fonctionnalités génériques et des méthodes de traitement de données utilisées à plusieurs endroits dans l'application. Ces utilitaires sont des fonctions de formatage de texte, des outils de manipulation de dates, etc.

**Fonctionnalités (Features) :** Les fonctionnalités regroupent les différents modules fonctionnels de l'application.

* + 1. **Les Web services REST**

Les API REST, également connues sous le nom de services web RESTful, sont utilisées pour faciliter la communication entre la partie backend et la partie frontend d'une application. Elles reposent sur des échanges via des requêtes HTTP envoyées vers des URI de ressources spécifiques, générant ainsi une réponse. Le corps de la réponse peut être formaté dans différents types de données, mais dans notre application, toutes nos API REST renvoient les réponses au format JSON, offrant ainsi une structure standardisée et facilement exploitable par les clients frontend.

**Conclusion**

Dans ce chapitre, nous avons exploré divers aspects du projet. Nous avons commencé par présenter l'entreprise impliquée, puis nous avons exposé le défi à surmonter ainsi que la solution envisagée pour y faire face. En outre, nous avons examiné en détail l'architecture de l'application, mettant en lumière les différentes couches et composants qui la composent. Enfin, nous avons discuté de la méthodologie de travail que nous avons sélectionnée pour guider le développement de ce projet, en mettant particulièrement l'accent sur notre choix d'adopter la méthode Scrum pour sa flexibilité et son adaptabilité aux besoins changeants du projet.

**Chapitre 2**

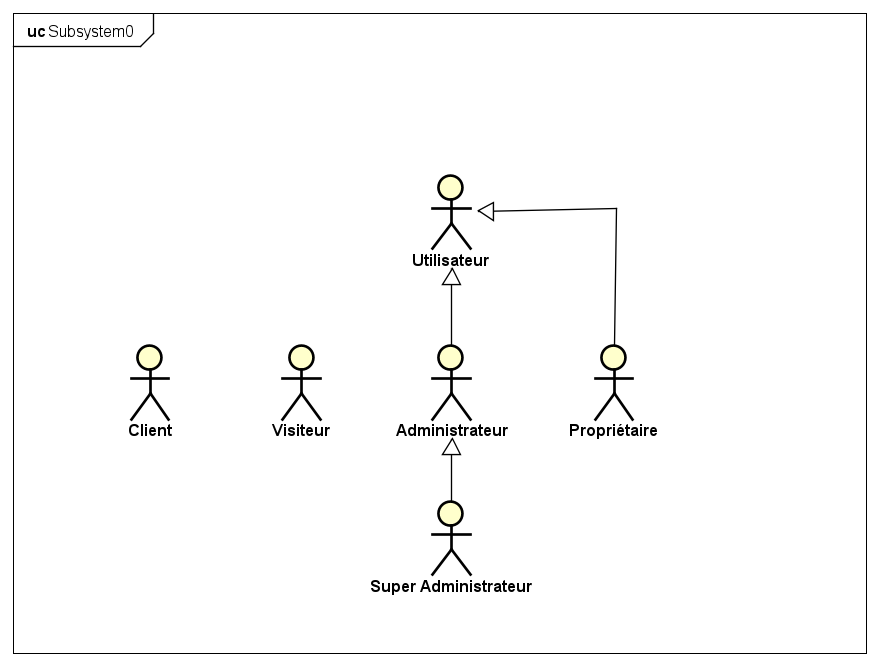
**Sprint 0 :** **MISE EN ŒUVRE DU PROJET**

Après avoir initialement présenté le préalable de notre système, il est crucial d'examiner attentivement ses besoins. Cette étape implique la définition des services offerts par notre plateforme. Nous allons détailler les besoins et les exigences fonctionnelles à travers un Backlog Produit et des diagrammes UML. En parallèle, nous élaborerons un plan de projet pour organiser efficacement les différentes étapes à venir.

# Identification des acteurs

Nous détaillons dans cette partie les acteurs qui interagissent dans notre application. Un acteur peut être un utilisateur ou un dispositif matériel qui interagit directement avec le système étudié :

* **Visiteur :** Accède au site vitrine de la plateforme sans s'identifier. Le but est de lui permettre de se familiariser avec la plateforme et ses fonctionnalités avant de s'inscrire en tant que propriétaire.
* **Client :** Utilise l'application pour exprimer leur satisfaction, participer aux programmes de fidélité et évaluer leur expérience globale.
* **Propriétaire d’entreprise :** Cet acteur peut créer et gérer les détails de son entreprise, ainsi que les enquêtes de satisfaction et les programmes de fidélité. De plus, il peut créer des campagnes marketing.
* **Super Administrateur :** Cetacteur estresponsable de la gestion globale du système. Ses principales fonctions incluent la création de comptes administrateurs, l'attribution de rôles et de privilèges aux utilisateurs du système, ainsi que la configuration des paramètres globaux pour assurer un fonctionnement optimal de la plateforme.
* **Administrateur :** il est chargé de la gestion des comptes des propriétaires. En outre, il est habilité à configurer les paramètres globaux de l'application et à définir les configurations par défaut, notamment en créant des modèles d'enquêtes, en ajoutant et en modifiant les catégories d'entreprises ainsi que les zones de service.
* **Utilisateur :** est une généralisation des utilisateurs de l’application qui peut être, Propriétaire, Administrateur, Super-Administrateur.



*Figure 7: Hiérarchie des acteurs*

# Analyse des besoins

Dans cette partie, nous nous intéressons aux besoins des utilisateurs dans notre application à travers les spécifications fonctionnelles et non fonctionnelles pour aboutir à une application de bonne qualité qui répond aux besoins du client afin de dégager notre backlog de produit.

# 2.2.1 Les besoins fonctionnels

Cette partie vise à recenser les besoins des utilisateurs de notre application afin d'en obtenir une compréhension approfondie. Parmi ces besoins, nous identifions :

**Le super Administrateur**

* S’authentifier.
* Créer des comptes d’administrateurs pour une répartition efficace des tâches.
* Gérer les utilisateurs de la plateforme en attribuant des rôles et des autorisations.

**L’administrateur**

* S’authentifier
* Accepter l’inscription des propriétaires à l’application après avoir vérifié leurs Coordonnées pour pouvoir passer à l’espace de travail.
* Consulter les listes des entreprises : L’admin peut modifier, archiver, supprimer et rechercher les listes des entreprises.
* Gérer les comptes des propriétaires sur la plateforme en assurant la modification et la gestion de leurs informations et paramètres.
* Configurer et paramétrer l'application afin de personnaliser les réglages et les options selon les besoins spécifiques du propriétaire d'entreprise. Cette étape comprend :
* L'ajout et la modification de modèles par défaut pour les enquêtes de satisfaction.
* L'ajout et la modification des catégories d'entreprises ainsi que de leurs types.
* L'ajout et la modification des zones de service et de leurs types.
* L’ajout des programmes de fidélité prédéfinis, offrant ainsi une sélection de programmes à utiliser par les propriétaires dans le cadre de leurs activités de fidélisation.

**Le propriétaire d’entreprise**

* Créer et gérer les détails de leur entreprise.
* Créer des enquêtes personnalisées en choisissant parmi une gamme de modèles fournis par défaut.
* Gérer les programmes de fidélité tels que Client Ambassador, Mystery Box, Coupon code et Raffle.
* Générer des QR codes associés à leurs enquêtes pour une collecte de données facile.

**Le client**

* Sélectionner leur langue préférée au début de l'enquête.
* Fournir des informations de base telles que leur nom, leur âge, etc.
* Exprimer leur satisfaction à l'égard des différents services proposés par l'entreprise en sélectionnant des émoticônes correspondant à leur niveau de satisfaction.
* Participer de manière optionnelle à un Programme de Fidélité pour des avantages supplémentaires tels que des réductions, des récompenses spéciales ou des privilèges exclusifs en échange de leur fidélité.
* Évaluer globalement leur expérience avec l'entreprise en utilisant l'Indice de Recommandation Net (NPS) en donnant une note sur une échelle de 0 à 10, selon leur propension à recommander l'entreprise à d'autres personnes. Cette évaluation permet de mesurer la satisfaction client et d'identifier les domaines à améliorer.

**Le visiteur**

* Accéder au site vitrine de la plateforme, il peut naviguer sur les différentes pages et sections de la plateforme sans avoir besoin de s'identifier. Le but de cette expérience est de permettre au visiteur de se familiariser avec la plateforme et ses fonctionnalités avant de décider de s'inscrire ou de devenir un propriétaire enregistré.

# 2.2.2 Les besoins non fonctionnels

Les besoins non fonctionnels sont d'une importance particulière car ils influent indirectement sur les résultats et les performances des utilisateurs. Les conditions suivantes doivent être satisfaites :

**-** **Fiabilité :** L'application doit fonctionner de manière cohérente, sans erreurs et être satisfaisante pour l'utilisateur.

**-** **Ergonomie et Interface utilisateur :** L'application doit offrir une expérience utilisateur simple et intuitive, facilitant la navigation entre les pages avec une mise en texte et des couleurs appropriées.

**- Sécurité :** La sécurité de l'application est une exigence primordiale. C'est l'une des exigences principales du client qui veut toujours sécuriser ses ressources et ses données. Ce besoin sera principalement garanti par :

* L'obligation d'authentification pour accéder à la plateforme.
* L'administrateur est responsable d'approuver, activer et désactiver les compte clients et propriétaires.
* La manipulation des mots de passe est faite avec le principe cryptage / décryptage.
* La génération automatique d'un Token de communication suite à l'étape de l'authentification et ce Token sera envoyé au serveur avec chaque appel à une ressource pour qu'il puisse identifier le demandeur.

**- Possibilité de maintenance et de réutilisation :** L'architecture du système doit être claire et standardisée pour permettre une maintenance efficace et une réutilisation des composants.

**- Compatibilité et portabilité :** L'application doit être compatible avec tous les navigateurs et modes d'utilisation, assurant ainsi sa fiabilité quel que soit l'environnement d'exécution.

# Backlog de produit

Le backlog de produit est établi préalablement au début du développement des releases, en réponse aux besoins des utilisateurs. Son objectif principal est de planifier les releases et les sprints, ainsi que de définir le contenu des sprints pour lancer efficacement le travail de développement.

*Tableau 2: Backlog de produit*

|  |  |  |  |
| --- | --- | --- | --- |
| **Backlog produit** | | | |
| **ID** | **User Story** | **Priorité** | **Release** |
| **1.1** | En tant utilisateur, je veux m'authentifier afin d’accéder à l’application. | 1 | 1 |
| **1.2** | En tant qu’utilisateur, je souhaite pouvoir réinitialiser mon mot de passe. | 1 | 1 |
| **1.3** | En tant qu’utilisateur, je veux modifier mon profil, afin de mettre à jour mes informations personnelles et adapter mon compte selon mes besoins et préférences. | 4 | 1 |
| **1.4** | En tant que super administrateur, je veux gérer les comptes des administrateurs de la plateforme afin de garantir un contrôle efficace sur les administrateurs. | 2 | 1 |
| **1.5** | En tant que super administrateur, je veux gérer les rôles et les privilèges des utilisateurs afin de garantir une répartition efficace des tâches. | 2 | 1 |
| **2.1** | En tant qu'administrateur, je veux gérer la liste des entreprises afin de maintenir une base de données à jour et complète. | 3 | 2 |
| **2.2** | En tant qu'administrateur, je veux gérer les propriétaires, afin de maintenir une liste précise et à jour des propriétaires. | 3 | 2 |
| **2.3** | En tant qu'administrateur, je veux configurer les paramètres globaux de l'application afin d’adapter l'application aux besoins spécifiques des entreprises. | 1 | 2 |
| **2.4** | En tant que propriétaire, je veux m'inscrire sur la plateforme afin d’accéder aux fonctionnalités disponibles. | 3 | 2 |
| **3.1** | En tant que propriétaire d'entreprise, je veux créer et gérer des campagnes d'enquêtes de satisfaction afin de recueillir les retours des clients. | 3 | 3 |
| **3.2** | En tant que propriétaire d'entreprise, je veux gérer les détails de mon entreprise afin de maintenir une présentation précise. | 3 | 3 |
| **3.3** | En tant que propriétaire d'entreprise, je veux gérer les programmes de fidélité offerts à mes clients afin de renforcer l'engagement et la fidélité des clients. | 3 | 3 |
| **3.4** | En tant que propriétaire d'entreprise, je veux générer des codes QR pour mes campagnes afin de faciliter la participation des clients à des enquêtes. | 3 | 3 |
| **3.5** | En tant que propriétaire d'entreprise, je veux consulter les statistiques et les rapports sur la satisfaction afin d’évaluer les performances de mon entreprise et identifier les domaines à améliorer. | 3 | 3 |
| **3.6** | En tant que client, je veux scanner des codes QR afin de participer à des enquêtes afin de participer à des enquêtes. | 3 | 3 |
| **3.7** | En tant que client, je veux exprimer ma satisfaction afin de partager mon expérience avec les propriétaires. | 3 | 3 |
| **3.8** | En tant que client, je veux participer à des programmes de fidélité afin de bénéficier de récompenses et d'avantages supplémentaires. | 3 | 3 |

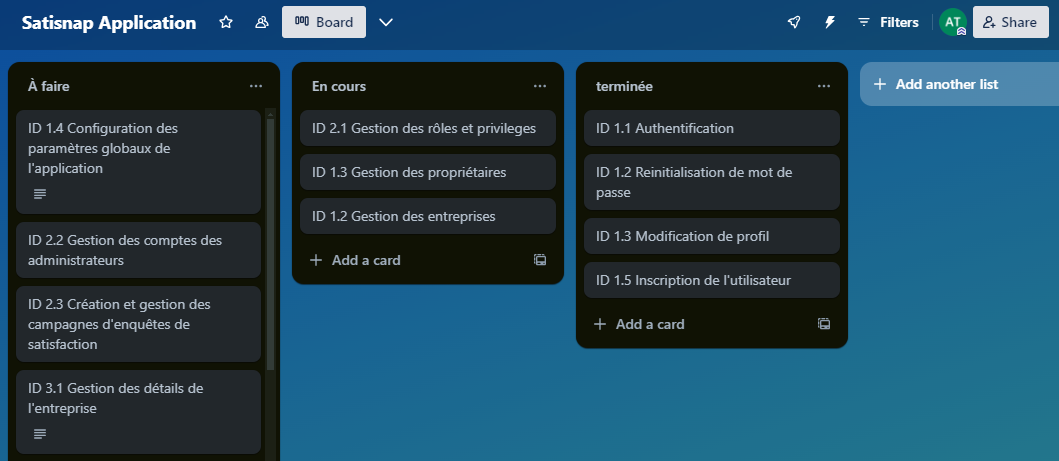
# Planification des releases

Dans le tableau ci-dessous, nous présentons le planning des différentes releases du projet. Chaque release comprend un ensemble de fonctionnalités réparties pour être réalisées en deux sprints, chaque sprint étant fixé à une semaine.

|  |  |  |
| --- | --- | --- |
|  | **Sprint 1** | **Sprint 2** |
| **Release 1** | Gérer les comptes administrateurs. | Gérer les rôles et privilèges. |
| **Release 2** | Gérer les comptes propriétaires et les entreprises. | Configurer et paramétrer l'application. |
| **Release 3** | Gestion des programmes de fidélité | Analyse et gestion des statistiques |

*Tableau 3: Planification des releases*

Après avoir organisé toutes les user stories, nous utilisons un outil de gestion de projet en ligne, appelé "Trello", pour travailler efficacement sur ces user stories dans le délai donné. Nous avons structuré notre travail en trois listes, où nous stockons et suivons nos tâches. Vous pouvez voir une capture d'écran de nos listes sur Trello ci-dessous :



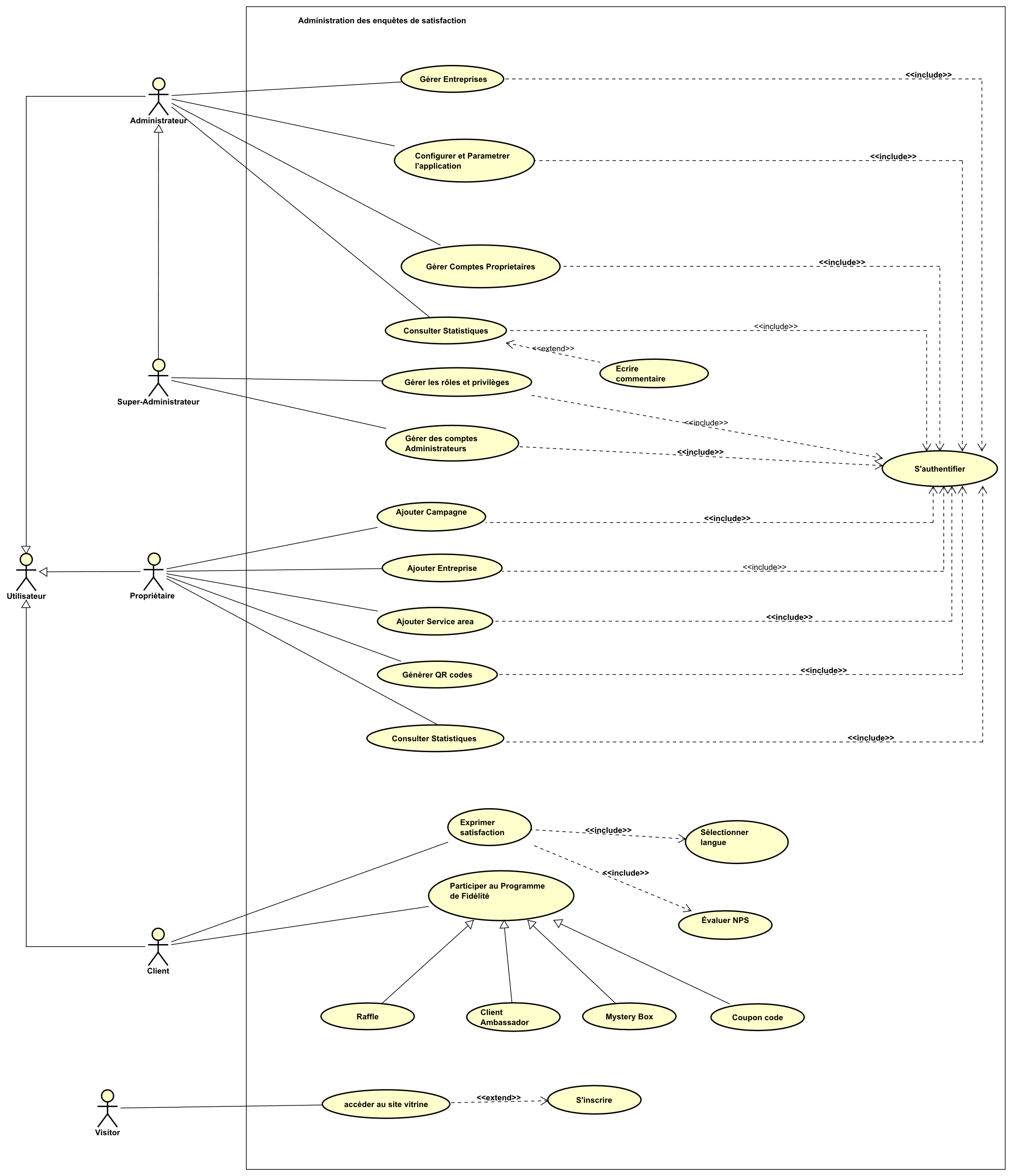
*Figure 8: List Trello*

# Diagramme de cas d’utilisation global

Le format de représentation UML (Unified Modeling Language) fournit le formalisme nécessaire pour afficher clairement et normativement les participants, le système et l'interaction entre les deux. Nous utilisons un diagramme de cas d'utilisation pour illustrer les différentes actions que les acteurs peuvent effectuer sur notre plateforme de gestion d'enquêtes de satisfaction. Ce diagramme nous permet de visualiser les fonctionnalités offertes aux propriétaires d'entreprises, aux clients et aux administrateurs, ainsi que les interactions entre eux.

Dans la figure ci-dessous :

Nous pouvons clairement distinguer la répartition des fonctionnalités entre les différents acteurs. L'authentification est une étape essentielle pour accéder aux fonctionnalités autorisées, peu importe la catégorie d'utilisateur. Il est toutefois à noter qu'un visiteur a la possibilité de devenir un propriétaire en s'inscrivant sur la plateforme.



*Figure 9:Diagramme de cas d'utilisation général*

**Conclusion**

Après avoir défini les besoins fonctionnels et non fonctionnels, la prochaine étape consistera à décomposer les éléments du backlog produit en différentes tâches réalisables.

**Chapitre 3**

**ETUDE TECHNOLOGIQUE**

Dans ce chapitre d'étude technique, nous explorerons les technologies et les Frameworks utilisés dans le développement des différents aspects de notre application. Nous examinerons en détail les choix technologiques afin de fournir un aperçu clair de l'architecture et des composants de notre système.

# Outils de Modélisation

La modélisation est le processus de création d'une représentation abstraite d'un système réel, utilisée pour concevoir, visualiser et spécifier des systèmes logiciels avant leur réalisation. Les modèles peuvent prendre différentes formes, telles que des diagrammes, des schémas, des cartes, et bien plus encore.

* 1. **Astah**

****Astah est un outil de modélisation UML créé par la compagnie japonaise Change Vision. Il fonctionne dans l'environnement d'exécution Java et est utilisé pour concevoir des diagrammes de classes, de séquence, d'activité et des cas d'utilisation.

*Figure 10: Logo Astah*

# Outils de Développement Back end

Le "backend" désigne la partie invisible d'une application logicielle située du côté du serveur. Il constitue le moteur de l'application, assurant le traitement des données, la gestion de la logique métier, l'interaction avec la base de données et la sécurisation des opérations, tout en exposant des API pour permettre aux clients d'interagir avec l'application de manière contrôlée.

* 1. **Plateforme Spring**La plateforme Spring est un écosystème de projets open source qui fournit une infrastructure complète pour le développement d'applications Java. Elle offre des solutions pour divers aspects du développement logiciel, y compris la gestion de la configuration, l'implémentation de la couche

*Figure 11:Logo du la plateforme Spring*

d'accès aux données, la gestion des transactions, la sécurisation des applications, et bien plus encore. En intégrant différentes bibliothèques et composants, Spring facilite le développement d'applications robustes, évolutives et hautement performantes.

* 1. **Spring Boot**

Spring Boot est un Framework Java qui accélère et simplifie le développement d'applications Web et de micro services avec Spring Framework en fournissant une configuration minimale.

*Figure 12:Logo spring boot*

Il offre des fonctionnalités intégrées telles que l'inversion de contrôle (IoC), l'injection de dépendances (DI), la configuration automatique et la création de serveurs web embarqués. En réduisant la configuration nécessaire. Spring Boot permet aux développeurs de se concentrer davantage sur la logique métier de leurs applications, offrant ainsi une approche plus conventionnelle et une productivité accrue.

* 1. **Spring Security**

Spring Security est un Framework de sécurité puissant pour les applications Java. Il fournit des fonctionnalités pour l'authentification, l'autorisation, la protection contre les attaques, la gestion des sessions, la gestion des rôles et bien plus encore.

*Figure 13: Logo Spring Security*

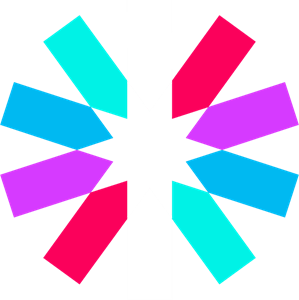
* 1. **Spring Data JPA**

Spring Data JPA est un sous-projet de Spring Data, qui simplifie l'utilisation de Java Persistance API (JPA) pour gérer la persistance des données. Il permet d’écrire des repositories de données avec des méthodes de requête simples et déclaratives, sans avoir à implémenter manuellement les requêtes SQL.

*Figure 14: Logo Spring Data*

Cela permet un développement plus rapide et plus efficace en réduisant la quantité de code nécessaire pour interagir avec la base de données.

* 1. **JWT**

JSON Web Token (JWT) est un format de token d'authentification basé sur JSON, fournissant un moyen compact et autonome de transférer en toute sécurité des données entre différentes parties sous forme d'objets JSON.   
 Ces données peuvent être vérifiées et validées grâce à une signature numérique. Les JWT peuvent être signés à l'aide d'un secret utilisant l'algorithme HMAC, ou avec une paire de clés publique/privée utilisant RSA ou ECDSA.

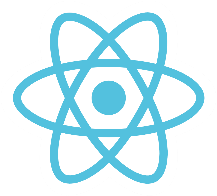
*Figure 15:Logo JWT*

Les JWT sont souvent utilisés dans les systèmes d'authentification stateless et dans les applications Web modernes pour gérer les sessions utilisateur et les autorisations d'accès de manière sécurisée et efficace.

# Outils de Développement Front end

Le "front end" ou interface utilisateur désigne la partie visible et interactive d'une application logicielle avec laquelle les utilisateurs interagissent directement. C'est la partie de l'application que les utilisateurs voient, utilisent et avec laquelle ils interagissent pour effectuer des actions ou consommer du contenu.

* 1. **React**

****React est une bibliothèque JavaScript développée par Meta et utilisée pour créer des interfaces utilisateur interactives et dynamiques pour les applications web et mobiles**.**

Il permet de diviser l'interface utilisateur en composants réutilisables, ce qui facilite la construction et la maintenance des applications. Son approche déclarative et sa gestion efficace du DOM (Document Object Model) permettent de créer des applications réactives avec des mises à jour d'interface utilisateur rapides et performantes.

*Figure 16: Logo React*

* + 1. **Axios**

Axios est une bibliothèque JavaScript populaire utilisée pour effectuer des requêtes HTTP depuis les navigateurs web et depuis Node.js. Elle offre une syntaxe simple et intuitive, basée sur des promesses, ce qui la rend facile à utiliser pour effectuer des appels à des API distantes.

*Figure 17: Logo Axios*

* + 1. **Tailwind CSS**

****Tailwind CSS est un Framework CSS open source qui permet de créer des interfaces utilisateur modernes et esthétiques de manière rapide et efficace.

Il se concentre sur l'utilisation de classes utilitaires pour styliser les éléments HTML.

*Figure 18: Tailwind CSS Logo*

# Outils de gestion de version

1. **GitHub**

 GitHub est une plateforme de développement collaboratif basée sur le système de contrôle de version Git, développée en Ruby on Rails et Erlang par Chris Wanstrath, PJ Hyett et Tom Preston-Werner.

Elle offre une gamme complète d'outils permettant d'héberger, de réviser et de gérer des projets de développement logiciel.

*Figure 19: Logo GitHub*

1. **Git**

Git est un système de contrôle de version distribué qui permet de suivre les modifications apportées au code source d'un projet.

Il offre un historique complet des modifications, permettant aux développeurs de travailler en collaboration, de gérer les différentes versions du code et de fusionner les modifications de manière efficace.

*Figure 20:Logo Git*

# Base de données

* + 1. **MySQL**

MySQL est un système de gestion de base de données relationnelle open source largement utilisé dans le développement d'applications web et d'entreprise.****

Il offre une gestion efficace des données structurées, permettant de stocker, d'organiser et de récupérer des informations de manière fiable et sécurisée.

*Figure 21: Logo MySQL*

# Framework de gestion des dépendances NPM

npm (Node Package Manager) - utilisé principalement pour les projets React, est un gestionnaire de paquets JavaScript qui facilite l'installation, la gestion et la mise à jour des dépendances nécessaires au développement d'applications React

Pour les projets React, npm est souvent utilisé pour installer des bibliothèques et des outils tels que React, React DOM, Redux, Axios, et bien d'autres, qui sont des éléments essentiels pour construire des interfaces utilisateur modernes et interactives.

*Figure 22: Logo NPM*

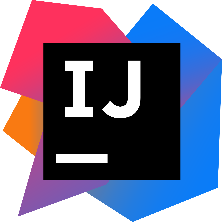
# Logiciel de Test : Postman

Postman est un outil de développement d'API qui permet aux développeurs de créer, tester, documenter et déboguer des API efficacement.  
 Il offre une interface pour envoyer des requêtes HTTP, inspecter les réponses, automatiser les tests et partager des collections d'API avec d'autres membres de l'équipe.

*Figure 23: Logo POSTMAN*

# Environnements de développement intégrés

* + 1. **IntelliJ IDEA**

**** IntelliJ IDEA est un environnement de développement intégré (IDE) développé par JetBrains, conçu pour faciliter le développement de logiciels en utilisant Java et d'autres langages de programmation tels que Kotlin, Groovy et Scala

*Figure 24: Logo Intellij IDEA*

* + 1. **Visual Studio Code**

Visual Studio Code est un éditeur de code source développé par Microsoft. Il est utilisé pour le développement d'applications web et d'autres types de logiciels pour Windows, Linux et macOS.

C'est un outil léger, rapide et polyvalent, doté de nombreuses fonctionnalités telles que la coloration syntaxique, l'achèvement automatique du code, le débogage intégré, le contrôle de version, les extensions personnalisables, et bien plus encore.

*Figure 25: Logo Visual Studio Code*

**Conclusion**

Ce chapitre de l'étude technologique a présenté une gamme d'outils et de technologies essentiels que nous utiliserons dans le développement de notre projet.

**Chapitre 4**

**RELEASE 1**

Dans ce chapitre consacré à la "Release 1", nous aborderons les bases de notre application, en mettant l'accent sur la gestion des comptes administrateurs ainsi que sur la gestion des rôles et des privilèges. Cette release marque une étape cruciale dans le développement de notre plateforme, car elle établit les fondements nécessaires à son bon fonctionnement.

# Organisation des Sprints

Il est crucial de planifier les sprints qui seront réalisés tout au long de la release. Pour la release 1, l'accent est mis sur la gestion des comptes administrateurs et la gestion des rôles et des privilèges. Cette phase implique principalement la mise en place d'un système permettant de gérer efficacement les comptes des administrateurs, ainsi que de définir et attribuer les rôles et privilèges appropriés à chaque utilisateur.

|  |  |
| --- | --- |
| **Release 1** | |
| **Sprint 1** | **Sprint 2** |
| Gestion des comptes administrateurs | Gestion des rôles et des privilèges |

*Tableau 4: Organisation des sprints de la release 1*

# Backlog des sprints pour la release 1

À partir de l'identification des sprints, nous allons spécifier de les user stories ainsi que les taches à réaliser durant cette release. Le tableau suivant présente les deux sprints de notre release est décrit le temps estimé pour la mise en œuvre :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Backlog des sprints** | | | | |
| **ID** | **Sprint** | **User Story** | **Estimation (En jours)** | **Tâche à réaliser** |
| **1.1** | Gestion des comptes administrateurs | * En tant que super administrateurs, je veux m’authentifier. * En tant qu’utilisateur, je souhaite pouvoir réinitialiser mon mot de passe. * En tant qu’utilisateur, je veux modifier mon profil, afin de mettre à jour mes informations personnelles et adapter mon compte selon mes besoins et préférences. * En tant que super administrateur, je veux gérer les comptes des administrateurs afin de garantir un contrôle efficace. | 14 jours | * Créer l’interface pour l’authentification. * Créer l’interface et le service permettant de réinitialiser le mot de passe. * Créer l’interface et le service permettant de visualiser et de gérer les comptes des administrateurs. * Créer l’interface pour la gestion des rôles et des privilèges. * Implémenter la logique backend permettant d'attribuer des rôles et des privilèges aux utilisateurs. * Faire les tests. |
| **1.2** | Gestion des rôles et des privilèges | * En tant que super administrateur, je veux gérer les rôles et les privilèges des utilisateurs afin de garantir une répartition efficace des tâches. |

*Tableau 5 : Backlog des sprints pour release 1*

Gérer = Ajout, mis à jour, recherche et consultation.

# Sprint 1 : Gestion des administrateurs

1. **Analyse et conception**

Le format de représentation UML offre une manière claire et normative de visualiser les participants, le système et leurs interactions.

Pour donner une vision globale de la conception du système, différents diagrammes sont conçus, notamment les diagrammes de cas d'utilisation, de classes et de séquence.  
 Dans le cadre de chaque sprint, nous présentons des diagrammes spécifiques, tels que le diagramme de cas d'utilisation du sprint 1, qui illustre le comportement fonctionnel du système.

* + - 1. **Diagramme de cas d’utilisation du Sprint 1**

Les diagrammes de cas d'utilisation sont des outils UML utilisés pour offrir une perspective globale sur le comportement des fonctionnalités du système logiciel et ses interactions avec divers intervenants externes. Dans cette section, nous introduirons le diagramme de cas d'utilisation du premier sprint afin de présenter une vue d'ensemble du comportement fonctionnel de notre système.



*Figure 26: Diagramme de cas d'utilisation du sprint 1 : Gestion des administrateurs*

* + - 1. **Description textuelle du sprint 1**

La description textuelle des cas d'utilisation offre une représentation narrative des interactions entre les acteurs et le système, décrivant la séquence temporelle des actions à effectuer.

Elles servent à clarifier le processus de fonctionnement et à indiquer les contraintes à prendre en compte lors du développement logiciel.

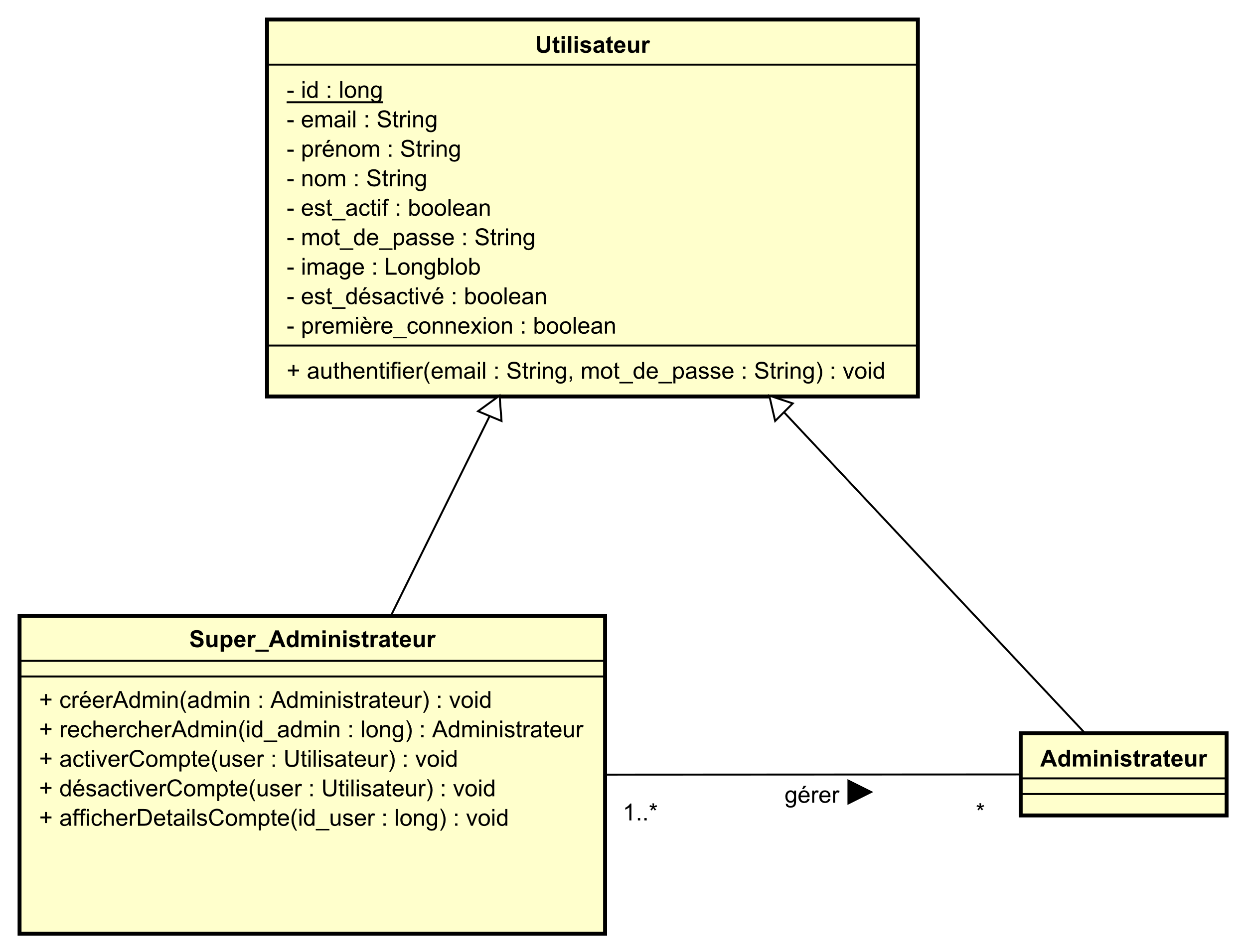
Dans le cadre du sprint 1, nous détaillons les cas d'utilisation suivants :

|  |  |
| --- | --- |
| **Cas d’utilisation** | Authentification |
| **Acteurs** | Super Administrateur |
| **Précondition** | L'utilisateur doit être créé dans la base de données et connaître ses identifiants. |
| **Postcondition** | Le super administrateur doit se connecter au système pour Ajouter, Modifier, Rechercher, Consulter, Activer, Désactiver un compte administrateur. |
| **Scenario nominal** | * L'utilisateur saisit son nom d'utilisateur et son mot de passe. * Le système vérifie les informations d'identification. * Si les informations sont valides, l'utilisateur est redirigé vers son tableau de bord. |
| **Scenario alternatif** | - Le super administrateur n’a pas saisi les bons identifiants. - Le super administrateur n'existe pas dans la base de données. - Un message d'erreur est renvoyé à l'utilisateur, l'informant de l'erreur. |

*Tableau 6: Description Textuelle : Authentification*

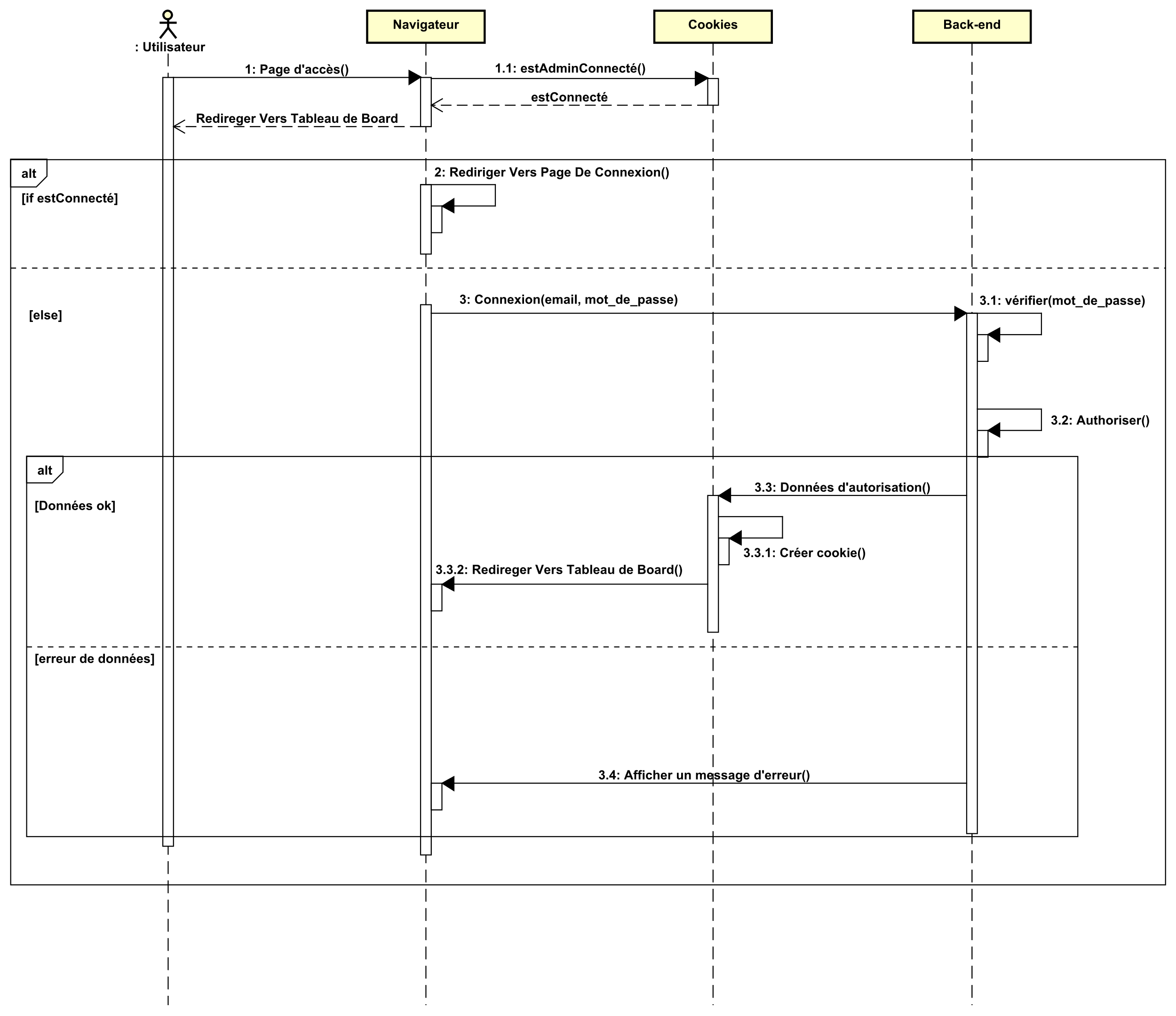
* + - 1. **Diagramme des classes du sprint 1**

Le diagramme de classes, faisant partie de la conception UML, offre une représentation statique des différents éléments du système logiciel et de leurs relations possibles. Les classes représentent des concepts abstraits, tels que les utilisateurs, les administrateurs ou les super administrateurs impliqués dans le système. L'application développée définit ensuite différentes catégories d'objets dans le diagramme de classes, permettant ainsi de visualiser la structure et les interactions entre les différents éléments du système.

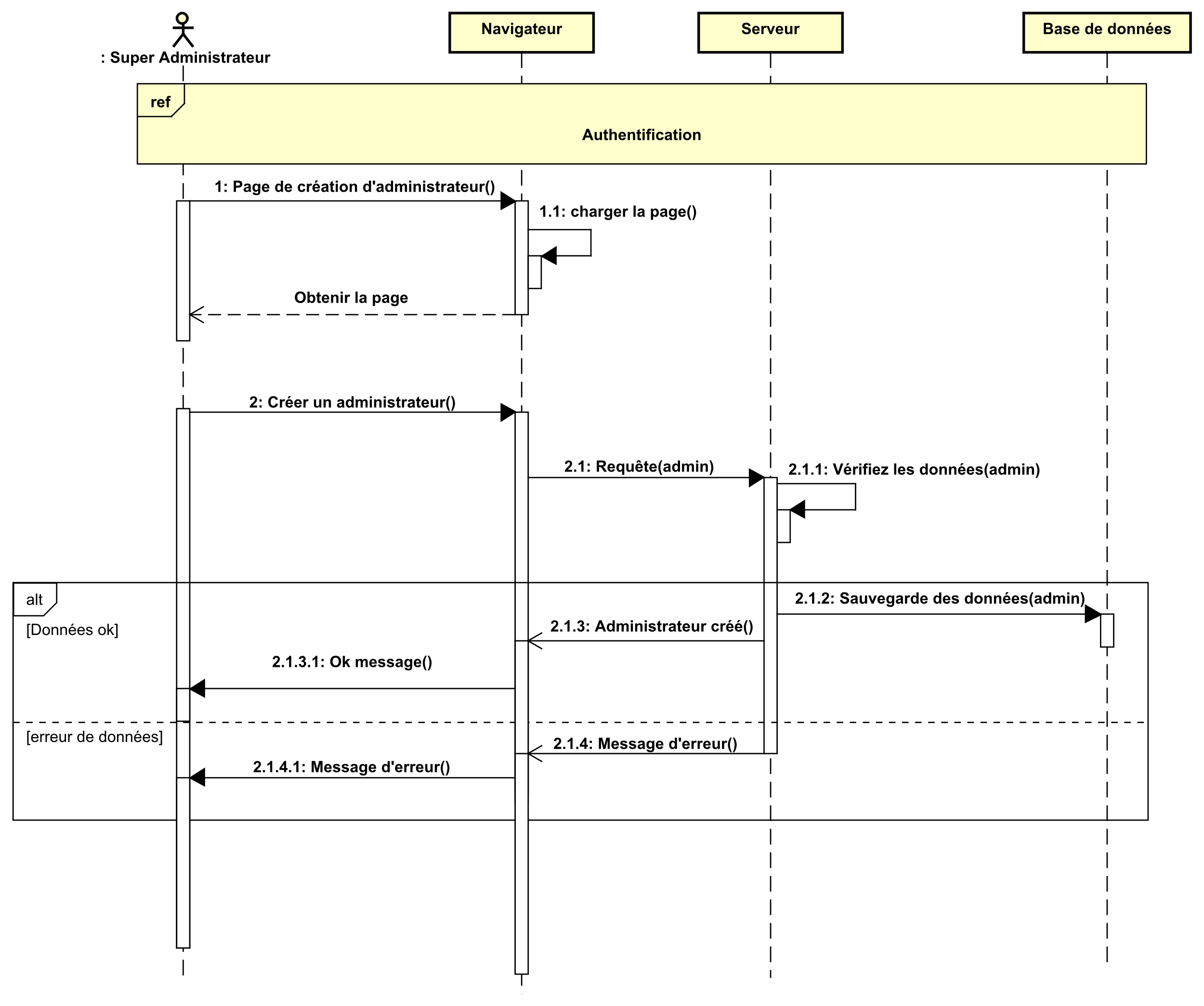


*Figure 27: Diagramme des classes du Sprint 1*

* + - 1. **Diagramme de séquence du sprint 1**

  
 Dans le cadre de la méthodologie UML, le diagramme de séquence représente visuellement les interactions chronologiques entre les acteurs et le système, illustrant les différents scénarios d'utilisation.   
Ces schémas se divisent en trois composantes : une section pour le navigateur, les cookies, et le backend pour le traitement des requêtes côté serveur.

*Figure 28: Diagramme de séquence :S'authentifier*

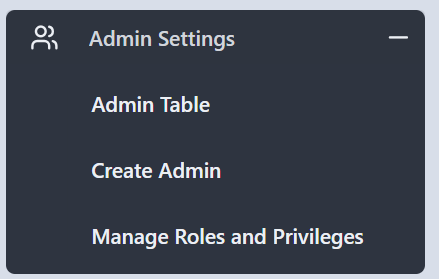


*Figure 29: Diagramme de séquence : Créer Administrateur*

* + - 1. **Réalisation, test et validation**

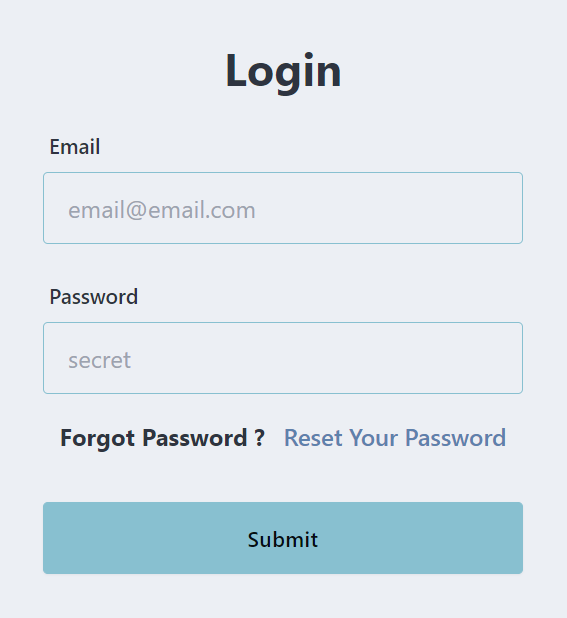
Dans cette section, nous examinerons quelques interfaces de notre produit final qui sont spécifiques à ce sprint. Pour chaque cas d'utilisation, nous explorerons tous les scénarios envisageables.

**Interface du résultat de release 1 :**



*Figure 30: Interface du résultat de la Release 1 :*

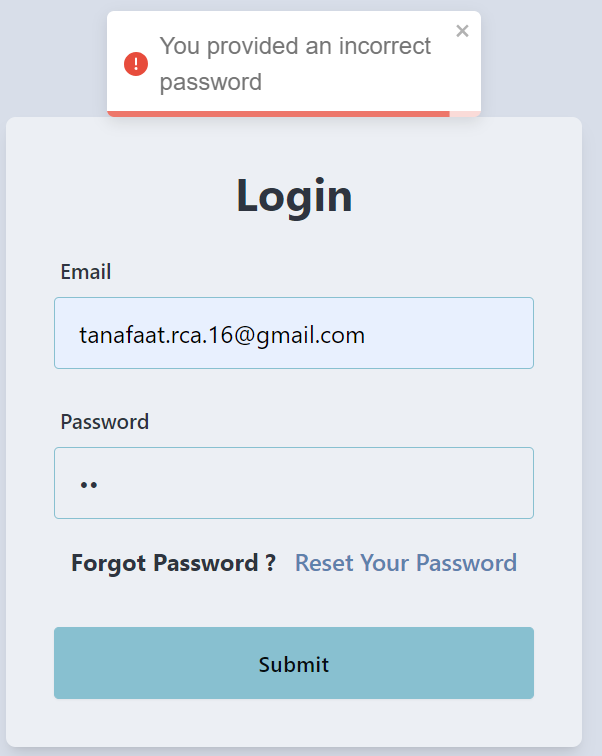
**Interface d’authentification en deux modes :**

**** 

*Figure 31:Interface authentification*

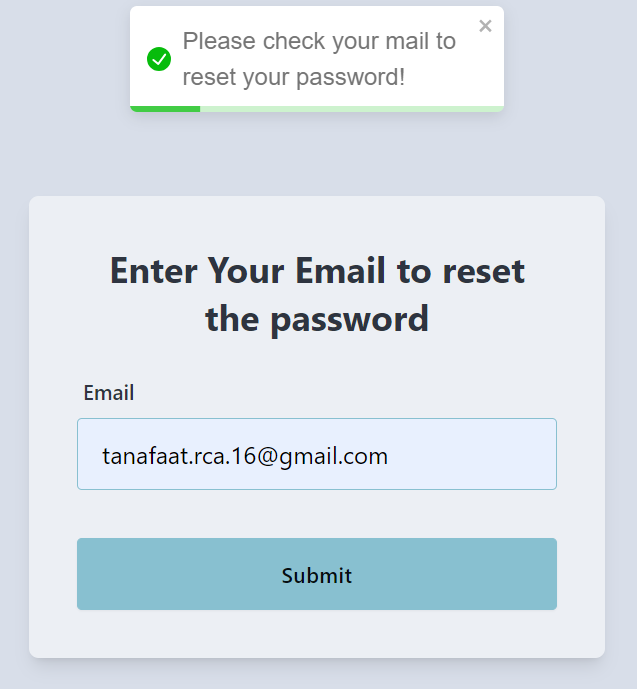
*Figure 32: Interface authentification en mode sombre*

**Interface d’authentification avec un login et mot de passe invalide :**

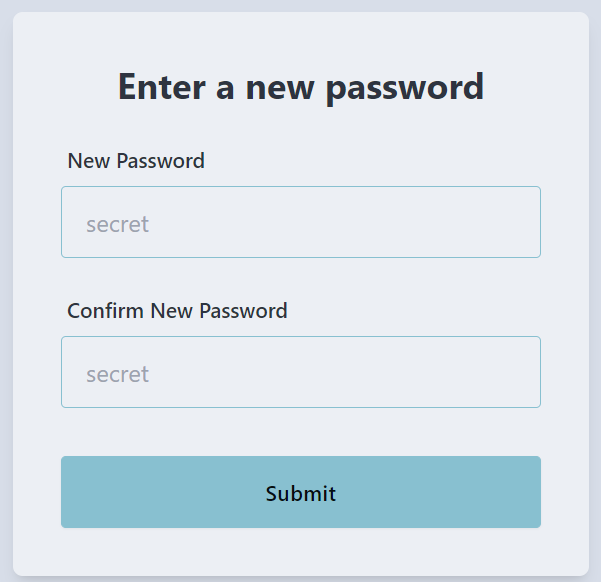
****

*Figure 33: Interface authentification avec un login et mot de passe invalide*

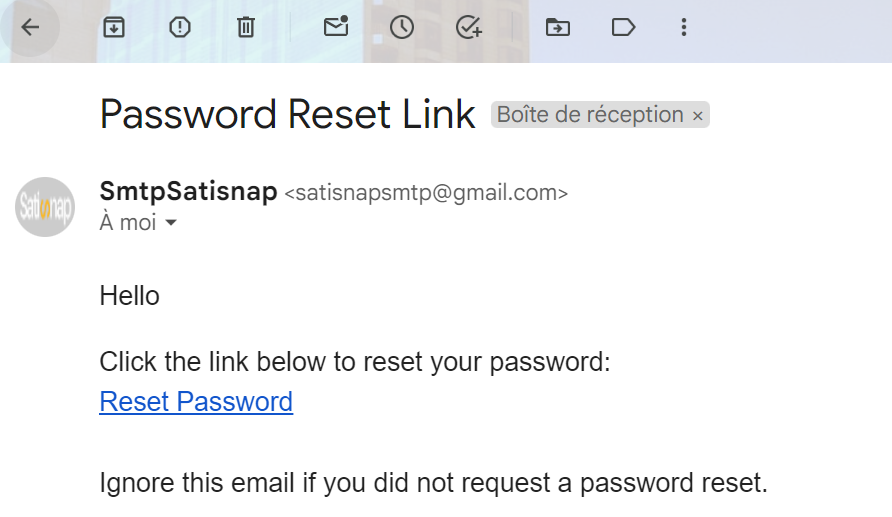
**Interface Réinitialiser mot de passe :**

****

*Figure 34: Interface Réinitialiser mot de passe.*

****

*Figure 35: Interface saisir le nouveau mot de passe*

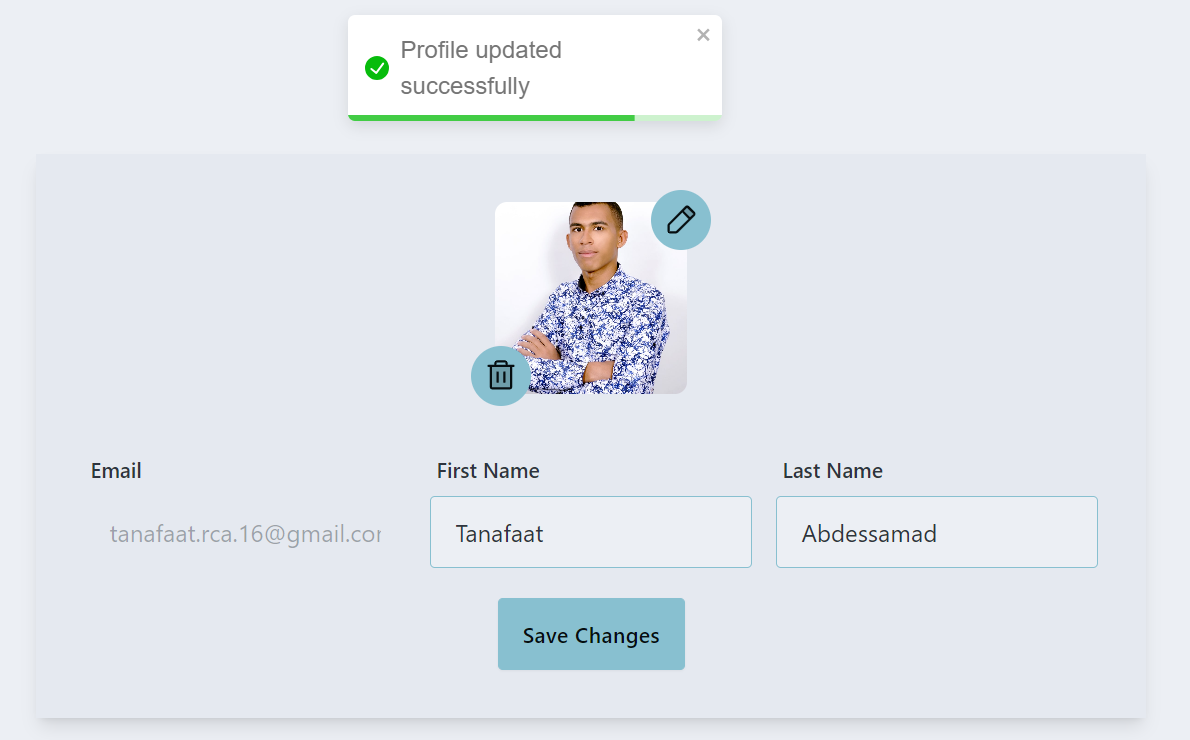


*Figure 36: Lien pour Réinitialiser mot de passe*

**Interface Modifier le profile :**

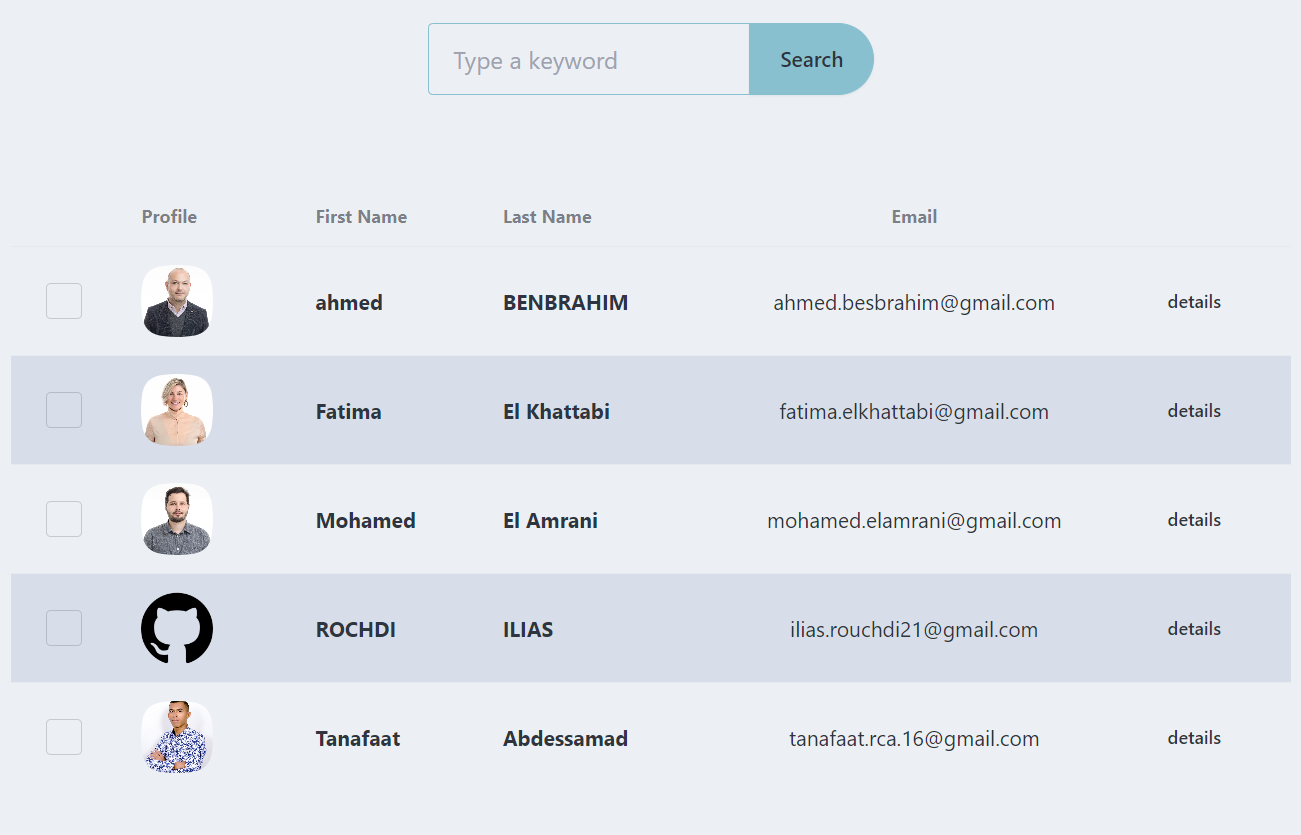
****

*Figure 37: Interface Modifier profile*



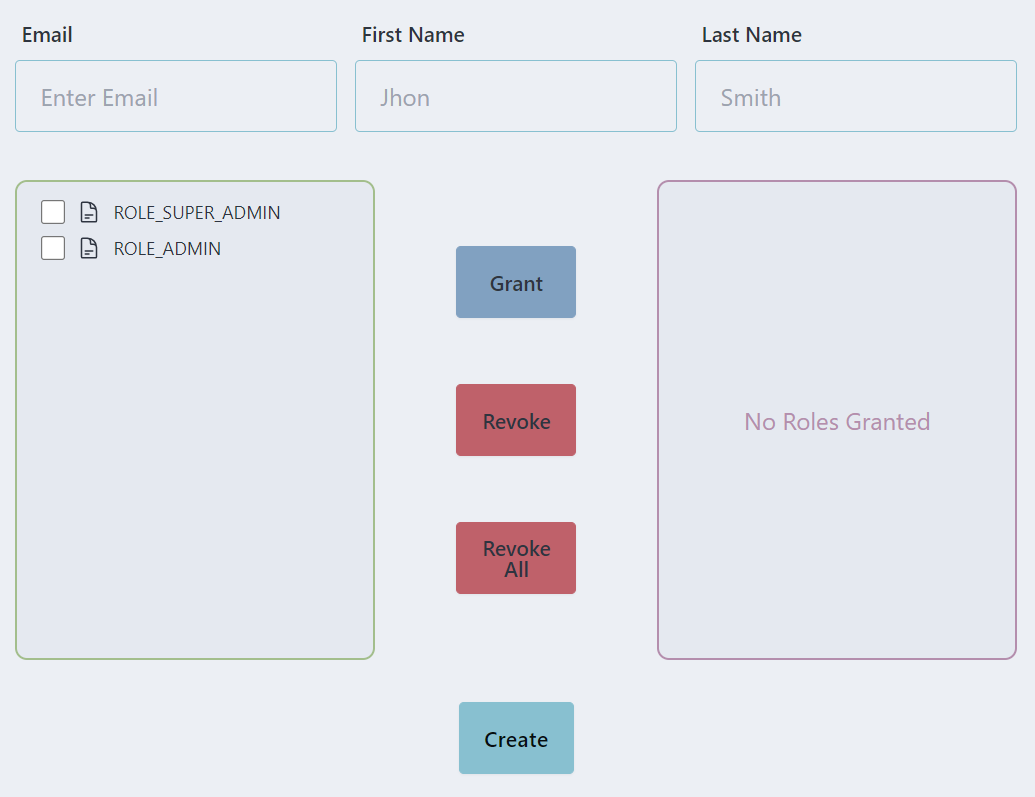
*Figure 38: Interface Modifier Profile*

**Interface pour la table des administrateurs :**

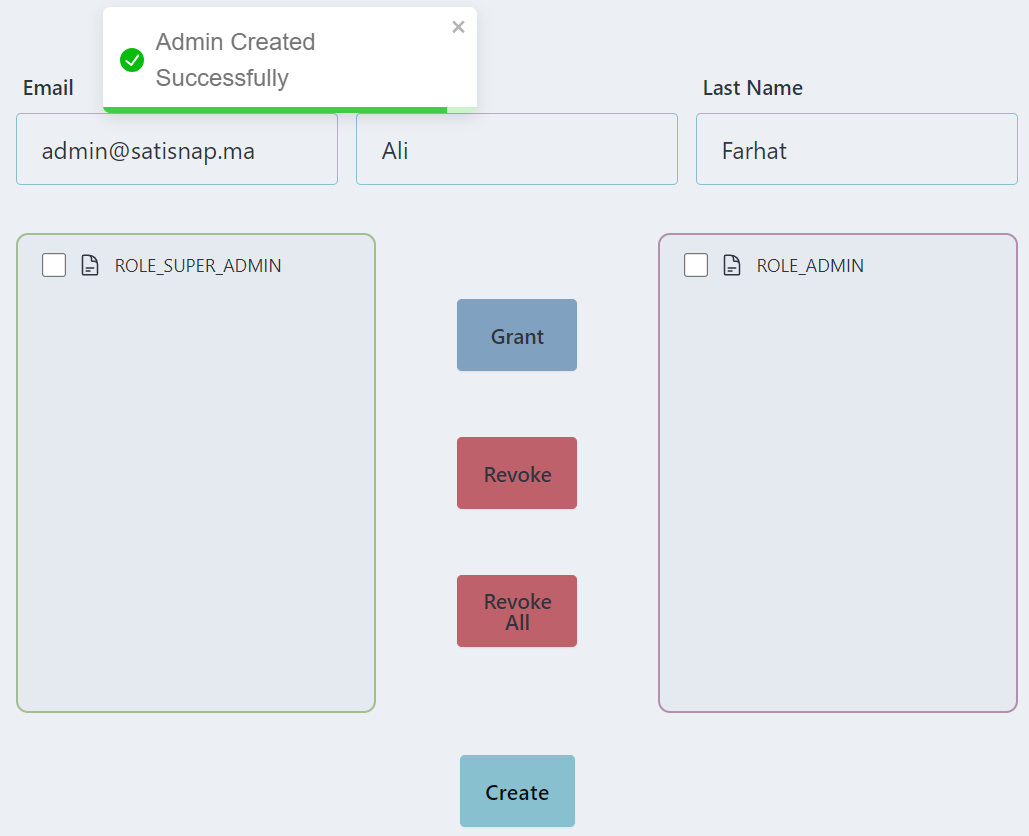
****

*Figure 39: Interface pour la table des administrateurs*

**Interface créer les administrateurs :**



*Figure 40: Interface créer les administrateurs*



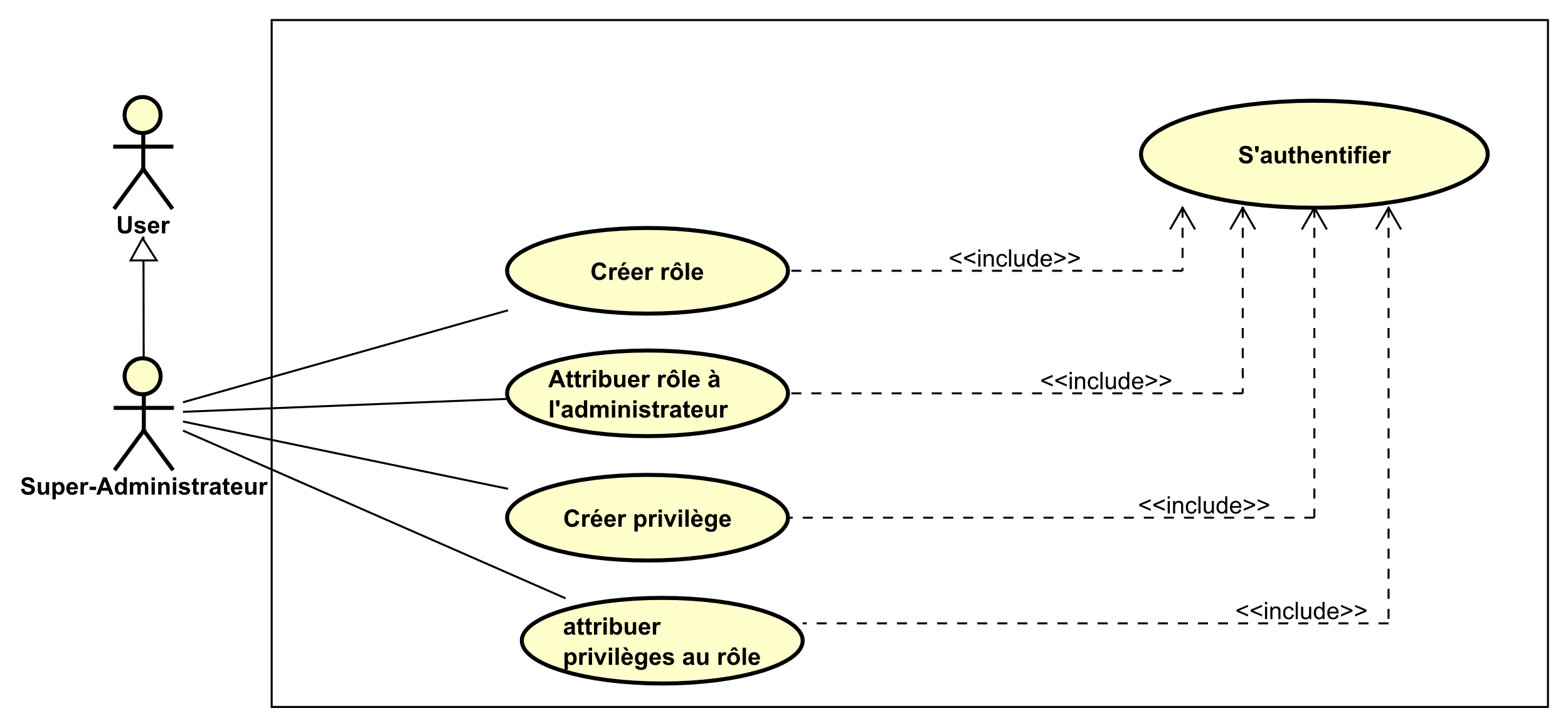
*Figure 41: Interface créer les administrateurs*

# Sprint 2 : rôles et des privilèges

|  |  |
| --- | --- |
| **Release 1** | |
| **Sprint 1** | **Sprint 2** |
| Gestion des comptes administrateurs | Gestion des rôles et des privilèges |

1. **Analyse et conception**
   * + 1. **Diagramme de cas d’utilisation du Sprint 2**

Dans cette section, nous allons présenter le diagramme de cas d'utilisation global du deuxième sprint.



*Figure 42: Diagramme de cas d'utilisation du sprint 1 : Gestion des rôles et des privilèges*

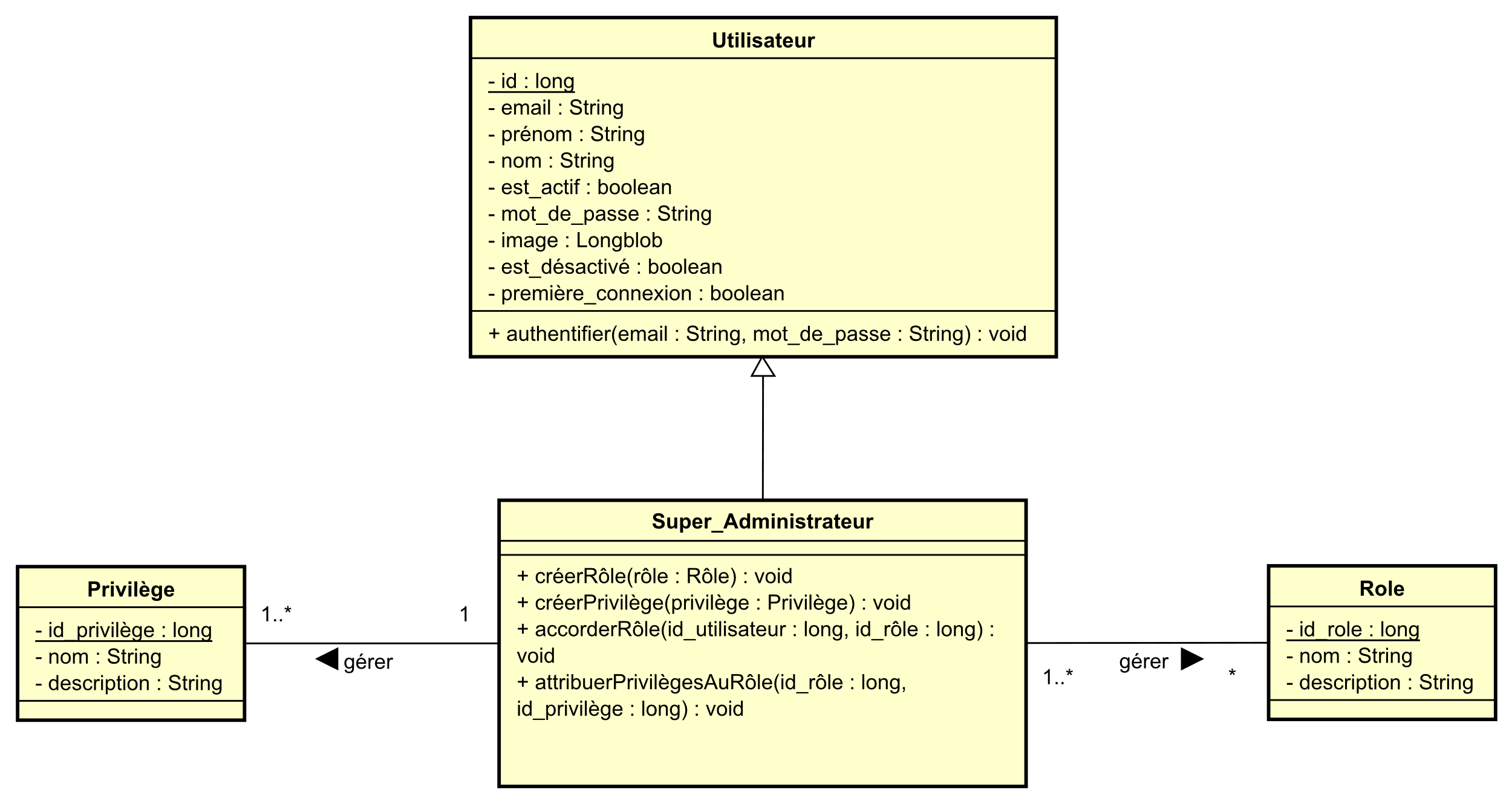
* + - 1. **Description textuelle du Sprint 2**

Dans le cadre du sprint 2, nous détaillons les cas d'utilisation suivants :

|  |  |
| --- | --- |
| **Cas d’utilisation** | Gestion des rôles et des privilèges |
| **Acteurs** | Super Administrateur |
| **Précondition** | L'utilisateur est connecté en tant que super administrateur. |
| **Postcondition** | Les rôles et des privilèges sont enregistrées avec succès.  Les rôles et privilèges sont attribués à l’administrateur avec succès. |
| **Scenario nominal** | * Le système enregistre les modifications dans la base de donnes. * Le système envoie un message informant le super administrateur que les modifications ont été terminée avec succès. |
| **Scenario alternatif** | * Le super administrateur saisit des données déjà existant dans la base de données. * Un message d'erreur est renvoyé à l'utilisateur, l'informant de l'erreur. |

*Tableau 7: Description Textuelle : Gestion des rôles et privilèges*

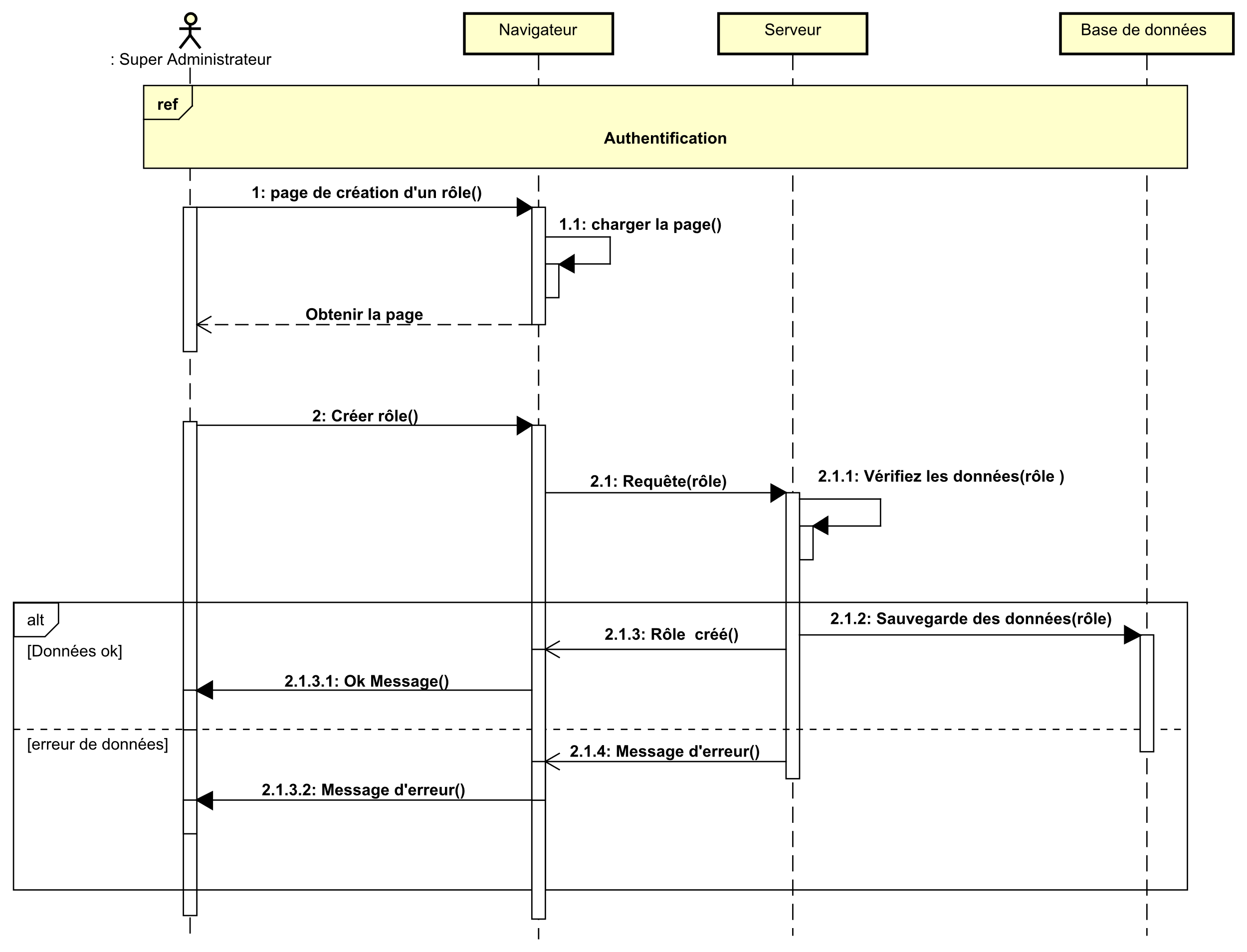
* + - 1. **Diagramme des classes du Sprint 2**

La figure illustre le diagramme des classes du sprint 2.

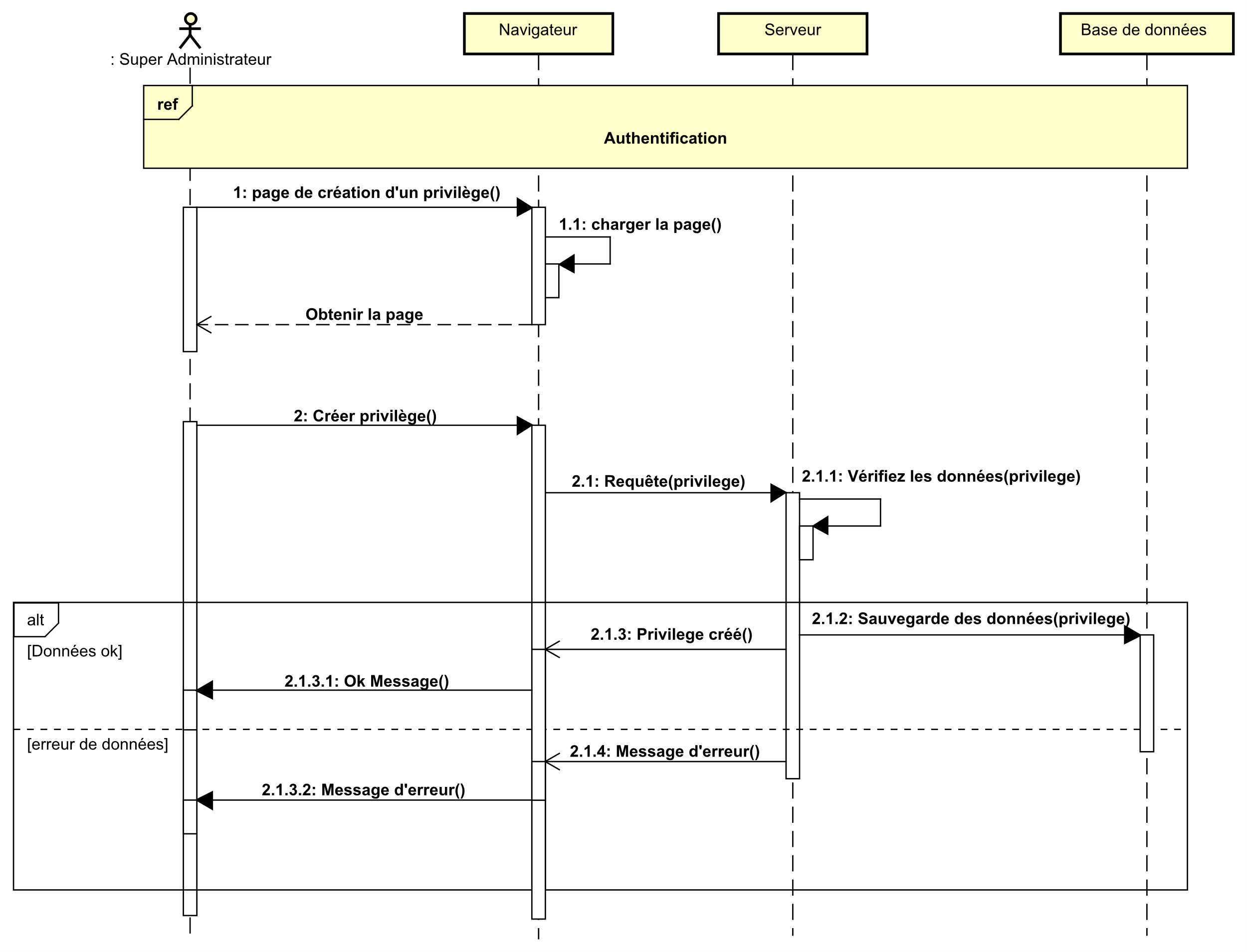
*Figure 43: Diagramme des classes du Sprint 2*

* + - 1. **Diagramme de séquence du Sprint 2**

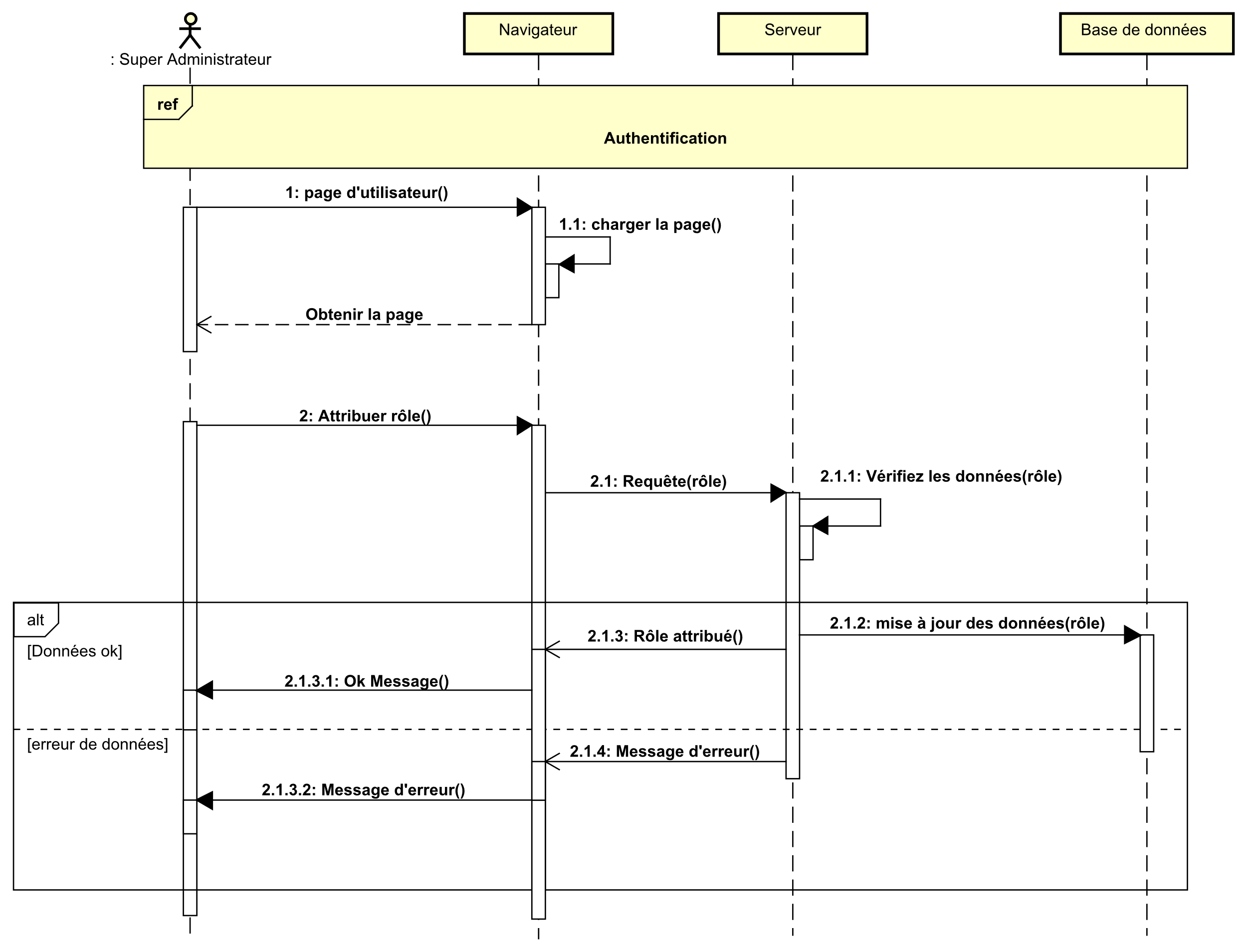
La figure illustre le diagramme de séquence relative au cas d’utilisation " Ajouter Rôle " :

****

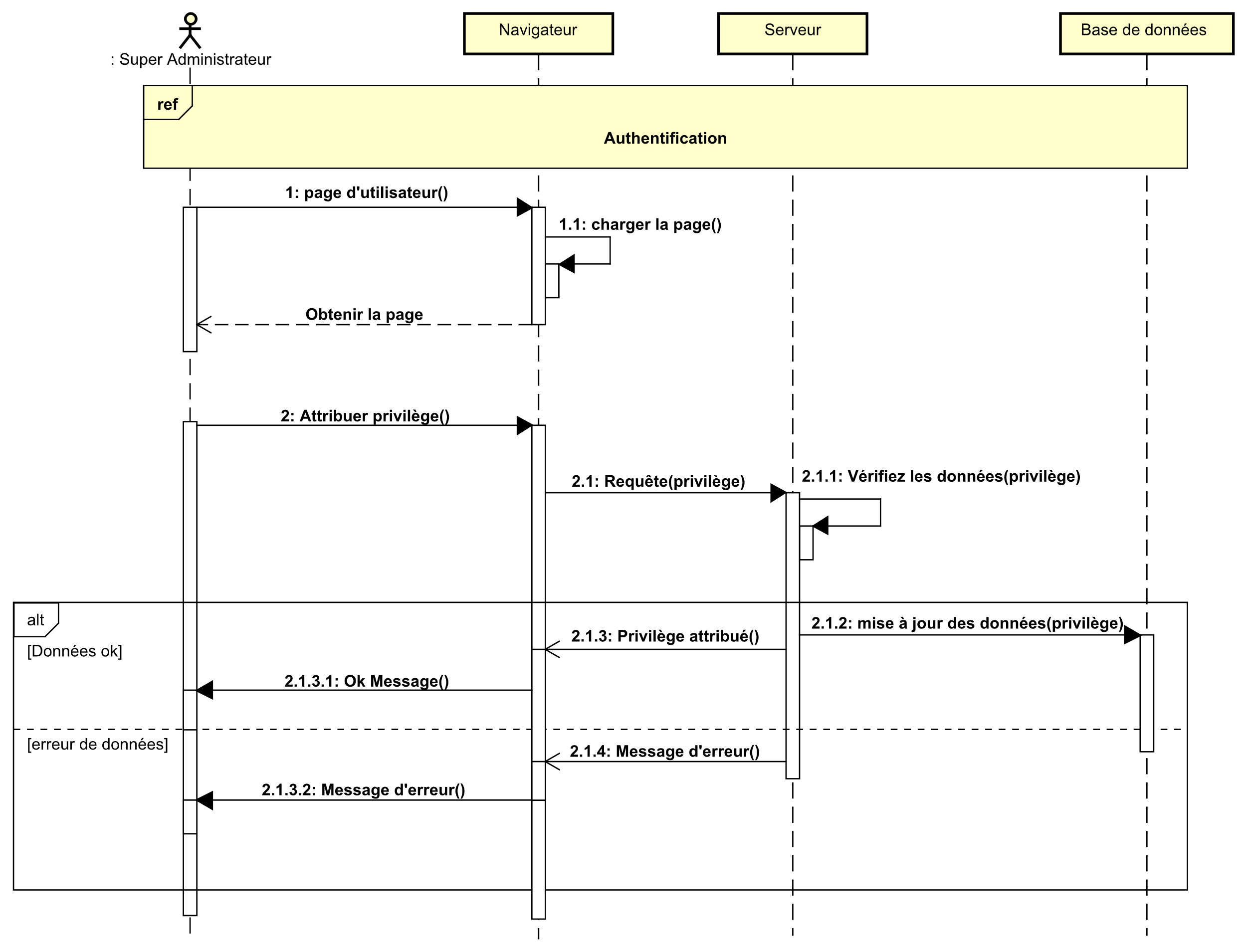
*Figure 44: Diagramme de séquence : Ajouter Rôle*

La figure illustre le diagramme de séquence relative au cas d’utilisation "Ajouter Privilège " :   


*Figure 45: Diagramme de séquence : Créer privilège*

La figure illustre le diagramme de séquence relative au cas d’utilisation "Attribuer un rôle à l’administrateur" :  
****

*Figure 46: Diagramme de séquence : Attribuer rôle à l’administrateur*

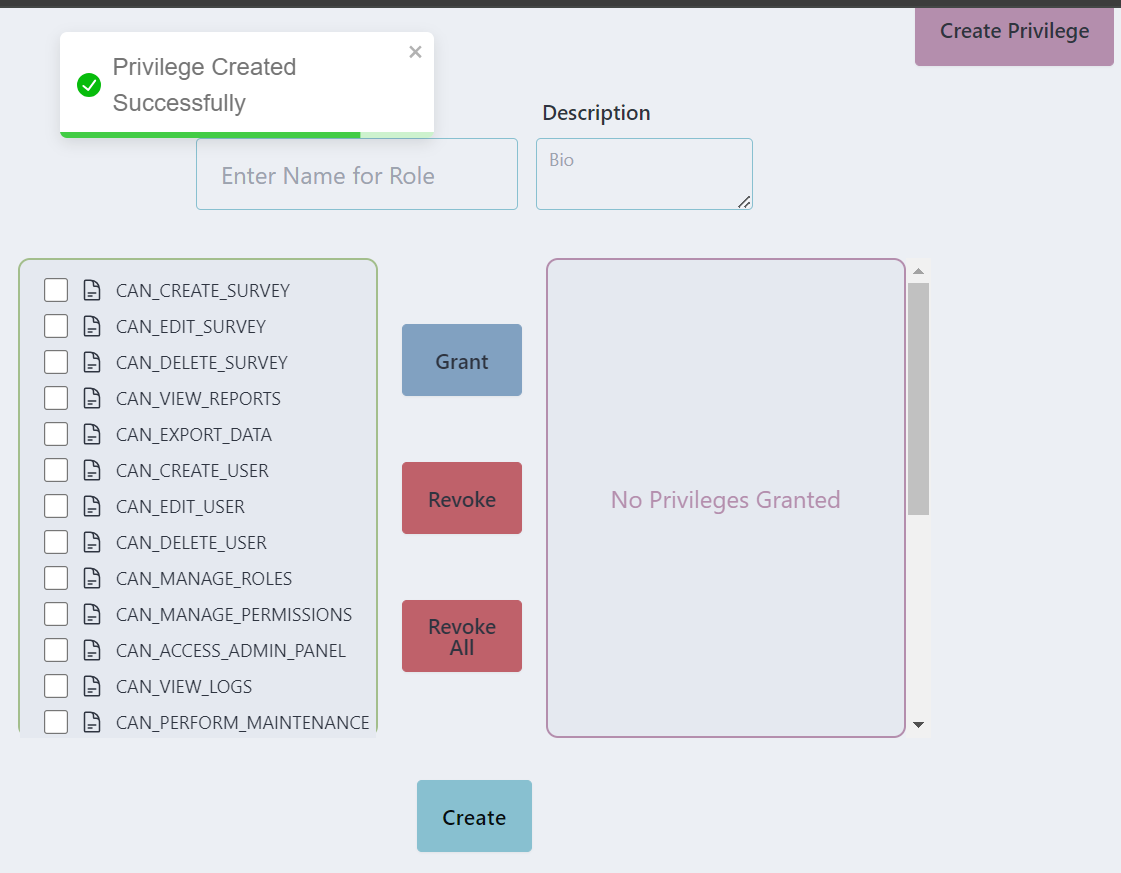
La figure illustre le diagramme de séquence relative au cas d’utilisation "Attribuer des privilèges à un rôle" :

*Figure 47: Diagramme de séquence : Attribuer des privilèges à un rôle :*

* + - 1. **Réalisation, test et validation**

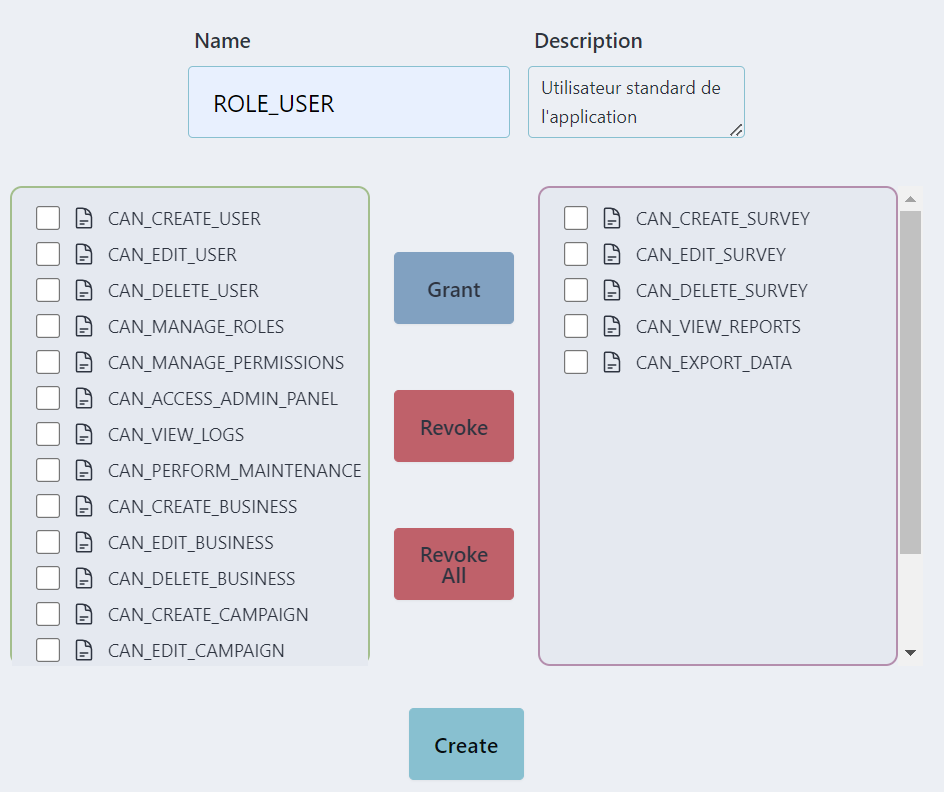
Dans cette section, nous examinerons quelques interfaces de notre produit final qui sont spécifiques à ce sprint. Pour chaque cas d'utilisation, nous explorerons tous les scénarios envisageables.

**Interface Créer privilège :**



*Figure 48: Interface créer privilège*

**Interface Créer rôle et Attribuer privilèges à un rôle :**

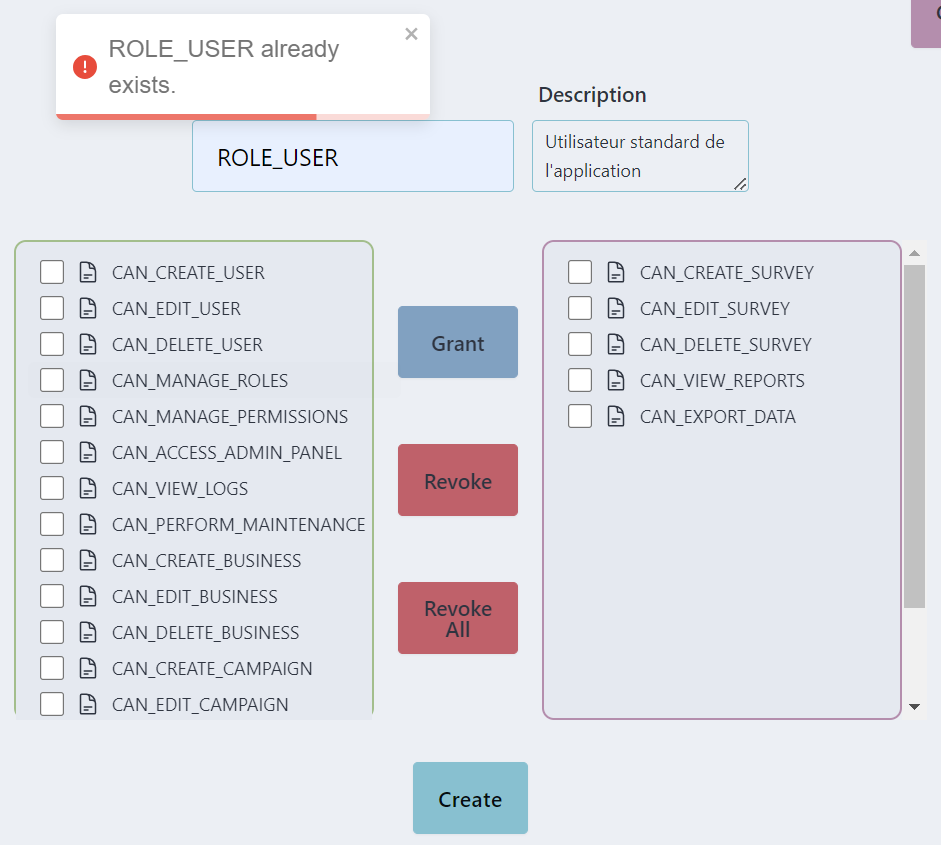
****

*Figure 49: Interface créer rôle*



*Figure 50: Interface créer rôle*

*Figure 51: Interface créer rôle*

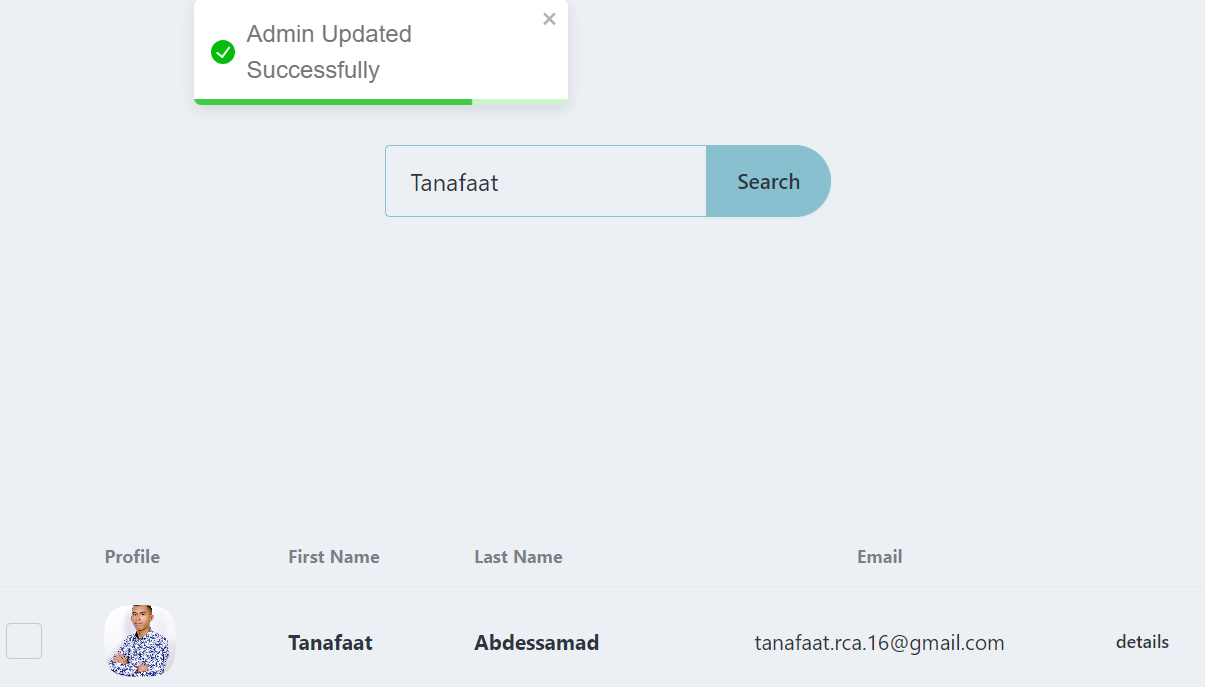
****

*Figure 52: Interface créer rôle*

**Interface Attribuer les rôles à un utilisateur :**



*Figure 53: Interface Attribuer rôle à un utilisateur*

******

*Figure 54: Interface Attribuer rôle à un utilisateur*

**Conclusion**

La mise en œuvre de cette release nous dote d'outils permettant aux administrateurs de gérer efficacement les comptes et les privilèges des utilisateurs. Dans la prochaine édition, nous présenterons la suite de l'application, axée sur la gestion des propriétaires d'entreprises et la configuration de l'application.

**Chapitre 5**

**RELEASE 2**

Dans ce chapitre portant sur la release 2, notre focalisation se portera sur la gestion des comptes des propriétaires, ainsi que sur la configuration et le paramétrage de l'application. Nous examinerons en détail les fonctionnalités associées à la gestion de ces comptes, ainsi que la mise en place de paramètres par défaut, directement accessibles aux propriétaires pour une utilisation immédiate et adaptée à leurs besoins.

# Organisation des Sprints

Cette version est planifiée sur deux sprints. Chaque release est prévue sur deux semaines, assurant une durée fixe tout au long du projet. Le tableau ci-dessous illustre l'organisation des sprints pour la release 2.

|  |  |
| --- | --- |
| **Release 2** | |
| **Sprint 1** | **Sprint 2** |
| Gestion des comptes propriétaires et des entreprises. | Configuration et le paramétrage de l'application |

*Tableau 8: Organisation des sprints de la release 2*

# Backlog des sprints pour la release 2

Le tableau ci-dessous présente le backlog des sprints pour notre deuxième release, qui comprend deux sprints répartis comme suit :

* Sprint 1 : Gestion des comptes propriétaires et des entreprises
* Sprint 2 : Configuration et paramétrage de l'application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Backlog des sprints** | | | | |
| **ID** | **Sprint** | **User Story** | **Estimation (En jours)** | **Tâche à réaliser** |
| **2.1** | Gestion des comptes propriétaires et des entreprises | * En tant qu'administrateur, je veux gérer les propriétaires, afin de maintenir une liste précise et à jour des propriétaires. * En tant qu'administrateur, je veux gérer la liste des entreprises afin de maintenir une base de données à jour et complète. * En tant qu’administrateur, je veux activer et désactiver les comptes des propriétaires afin de gérer efficacement leur accès à la plateforme. | 14 jours | * Créer l’interface et le service d'administration des propriétaires. * Créer l'interface et le service pour l'activation et la désactivation des comptes des propriétaires. * Faire les tests |
| **2.2** | Configuration et paramétrage de l'application | * En tant qu'administrateur, je veux configurer les paramètres globaux afin d’adapter l'application aux besoins spécifiques des propriétaires. | * Créer l’interface et le service d'administration des paramètres globaux. * Créer l’interface et le service d'ajout des catégories d'entreprises par défaut. * Créer l'interface et le service d'ajout de modèles graphiques pour la création d'enquêtes de satisfaction par défaut. * Faire les tests. |

*Tableau 9:Backlog des sprints pour release 2*

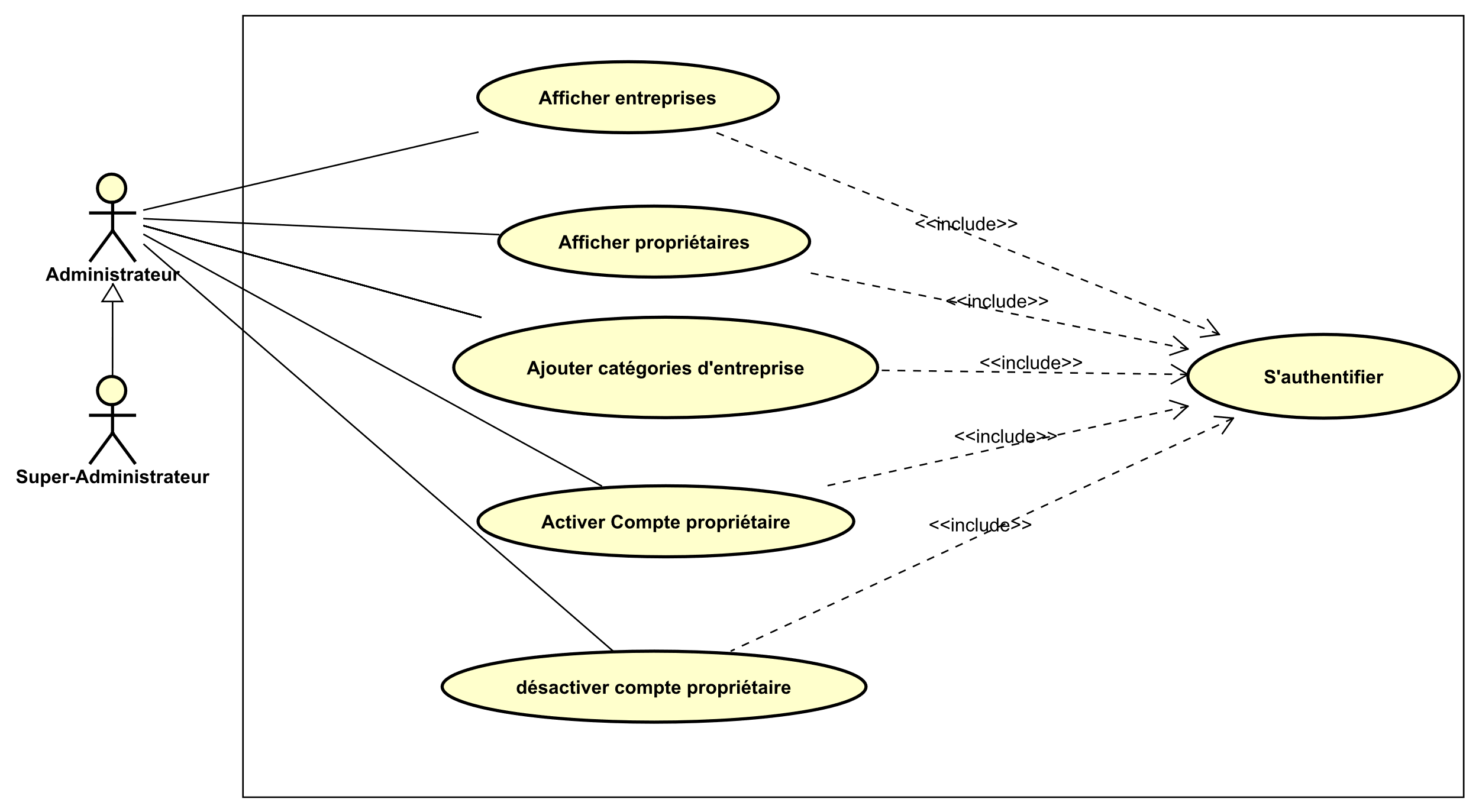
# Sprint 1 : propriétaires et entreprises

|  |  |
| --- | --- |
| **Release 1** | |
| **Sprint 1** | **Sprint 2** |
| Gestion des propriétaires et des entreprises | Configuration et paramétrage de l'application |

* + 1. **Analyse et conception**

**5.3.1.1 Diagramme de cas d’utilisation du Sprint 1**

Dans cette section, nous allons présenter le diagramme de cas d'utilisation global du premier sprint.



*Figure 55: Diagramme de cas d'utilisation du sprint 1 : Gestion des propriétaires et des entreprises*

* + - 1. **Description textuelle du Sprint 1**

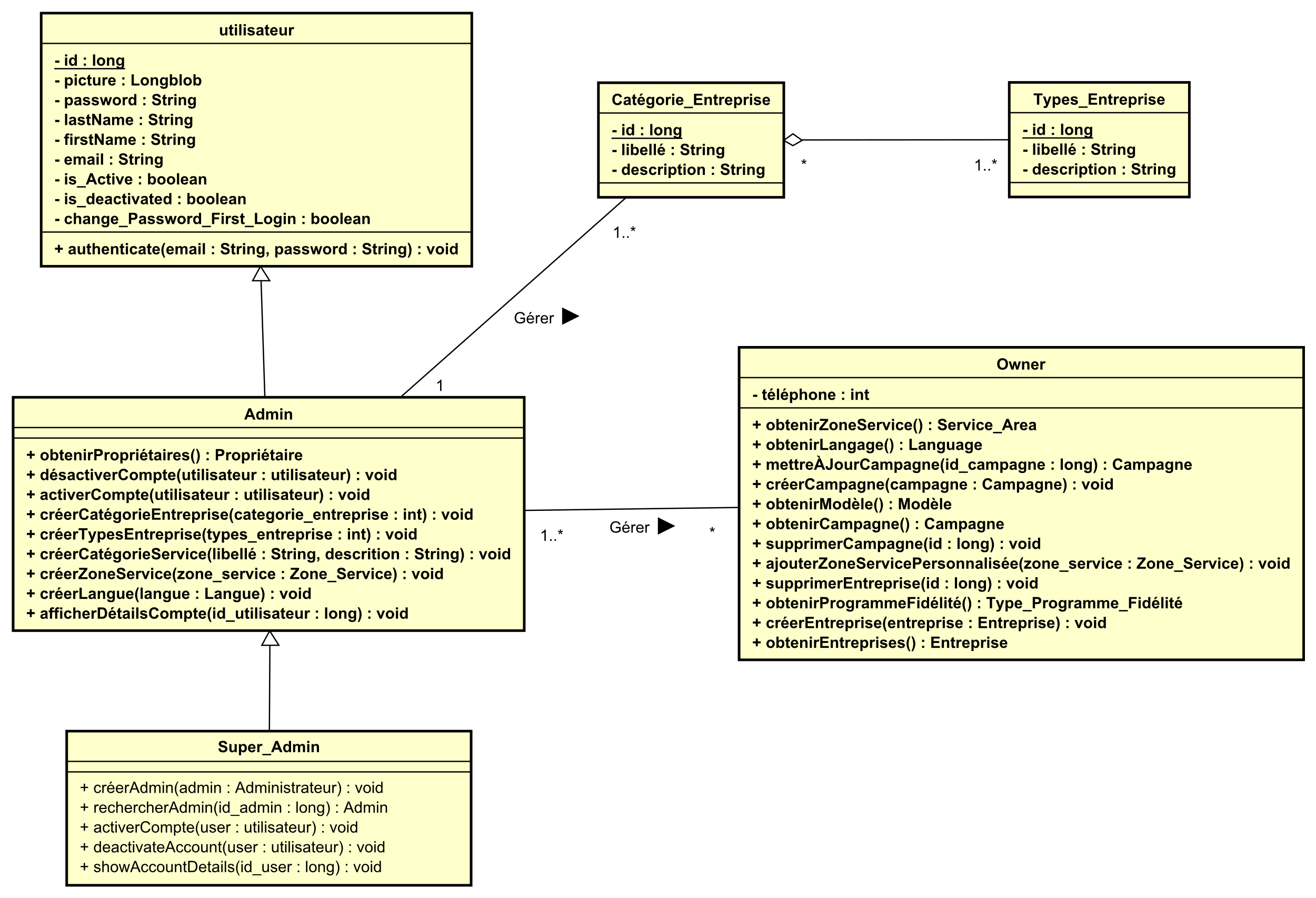
Dans le cadre du sprint 1, nous détaillons les cas d'utilisation suivants :

|  |  |
| --- | --- |
| **Cas d’utilisation** | Gestion des propriétaires et des entreprises |
| **Acteurs** | Super Administrateur, Administrateur |
| **Précondition** | L'utilisateur a réussi à se connecter en tant que super administrateur ou administrateur. |
| **Postcondition** | Les modifications apportées aux comptes des propriétaires et des entreprises sont enregistrées dans la base de données. |
| **Scenario nominal** | * L'administrateur effectue les opérations nécessaires, telles que l’activation ou désactivation d’un compte, la modification ou la suppression de propriétaires ou d'entreprises. * L'administrateur crée des catégories d'entreprises par défaut pour organiser les différentes entreprises inscrites sur la plateforme. * Le système notifie l'administrateur que les modifications ont été effectuées avec succès. |
| **Scenario alternatif** | * L’utilisateur n'est pas connecté en tant que super administrateur ou administrateur, il ne peut pas accéder à l'interface de gestion des propriétaires et des entreprises. |

*Tableau 10: Description Textuelle : Gestion des propriétaires et des entreprises*

* + - 1. **Diagramme des classes du Sprint 1**

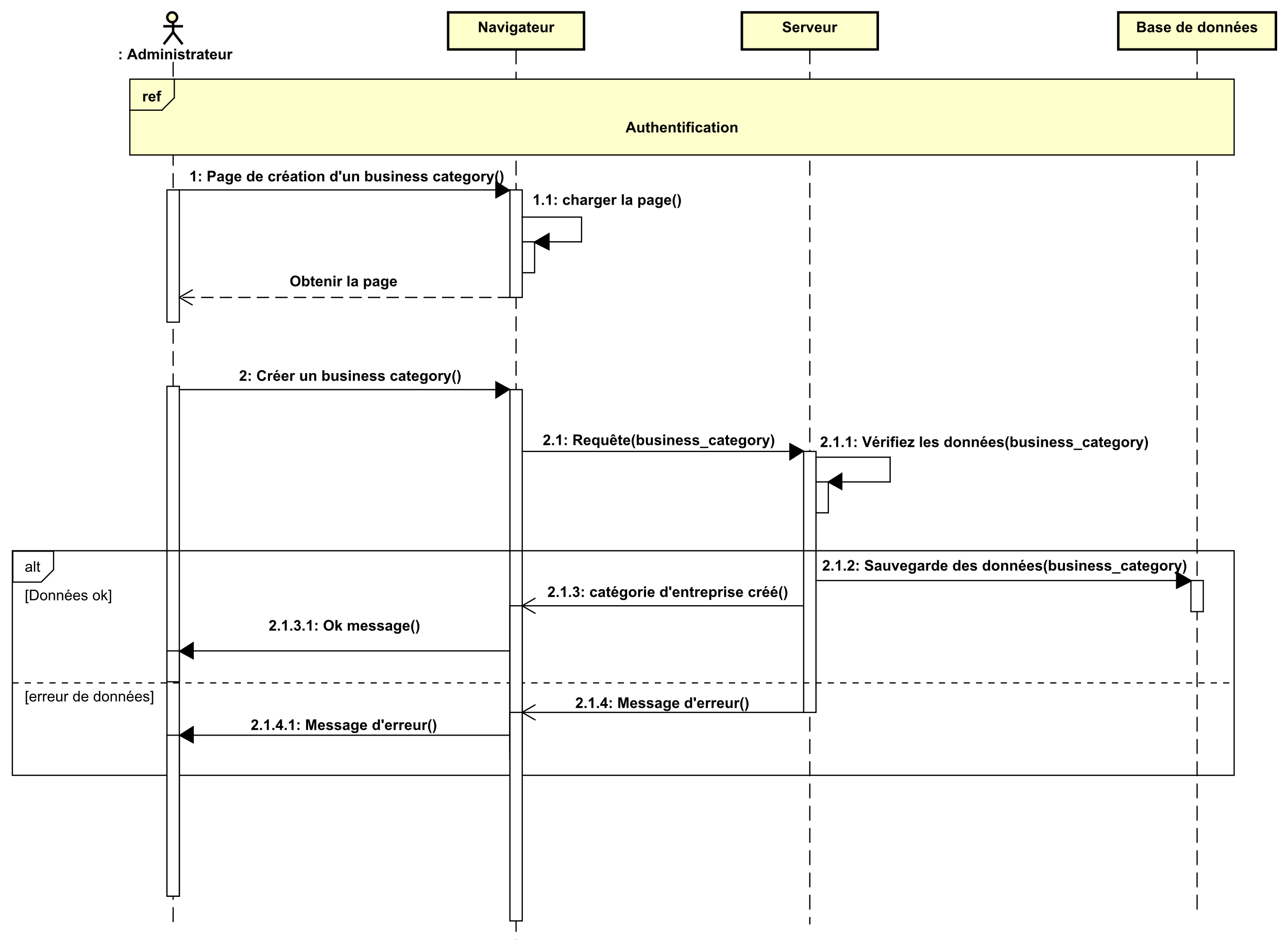
La figure illustre le diagramme des classes du sprint 1.



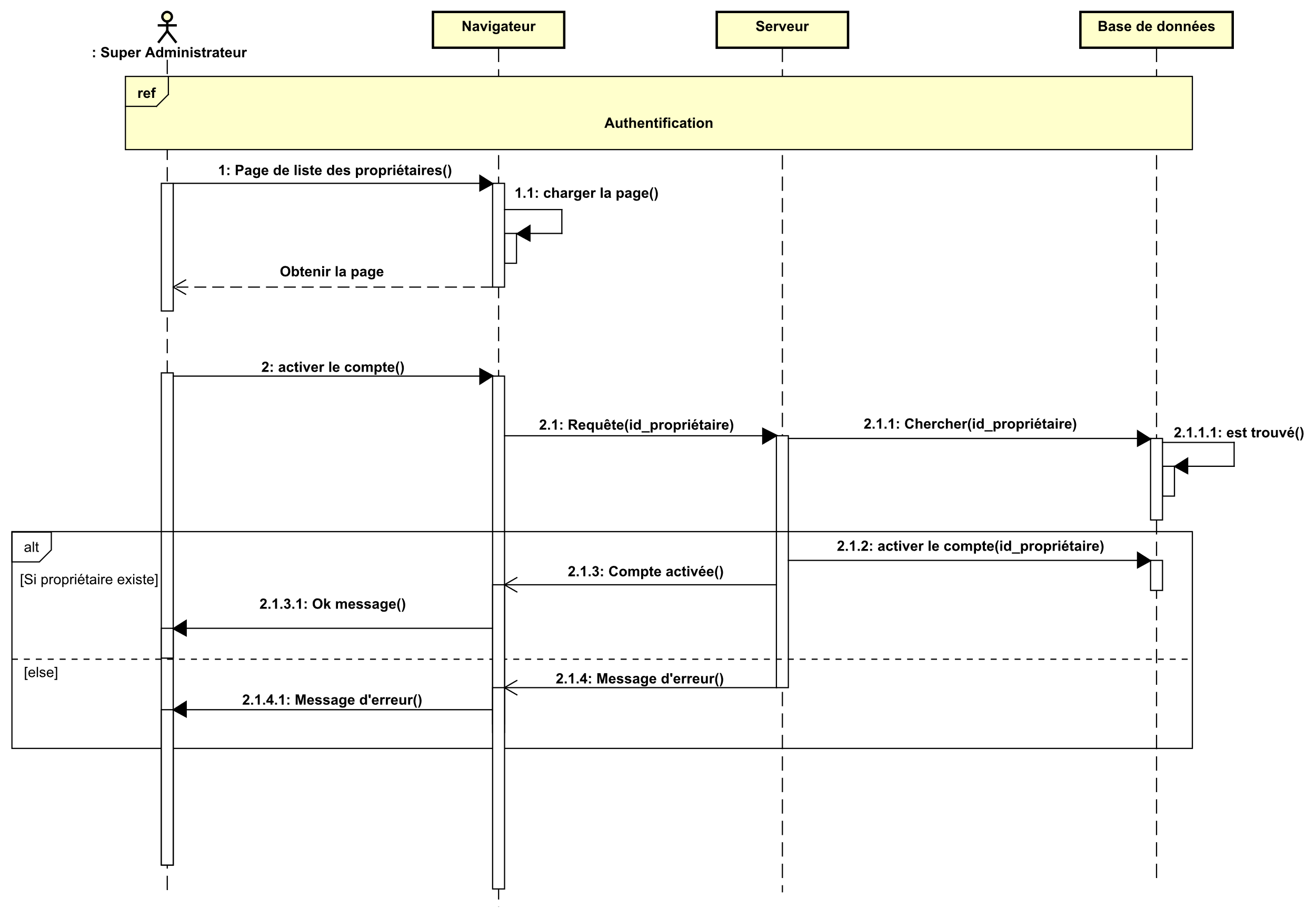
*Figure 56: Diagramme des classes du Sprint 1*

* + - 1. **Diagramme de séquence du Sprint 1**

La figure illustre le diagramme de séquence relative au cas d’utilisation " Créer une catégorie d'entreprise " :

****

*Figure 57: Diagramme de séquence : Créer une catégorie d'entreprise*

La figure illustre le diagramme de séquence relative au cas d’utilisation "Activer Compte " :

*Figure 58: Diagramme de séquence : Activer Compte de propriétaire*

* + - 1. **Réalisation, test et validation**

Dans cette section, nous examinerons quelques interfaces de notre produit final qui sont spécifiques à ce sprint.

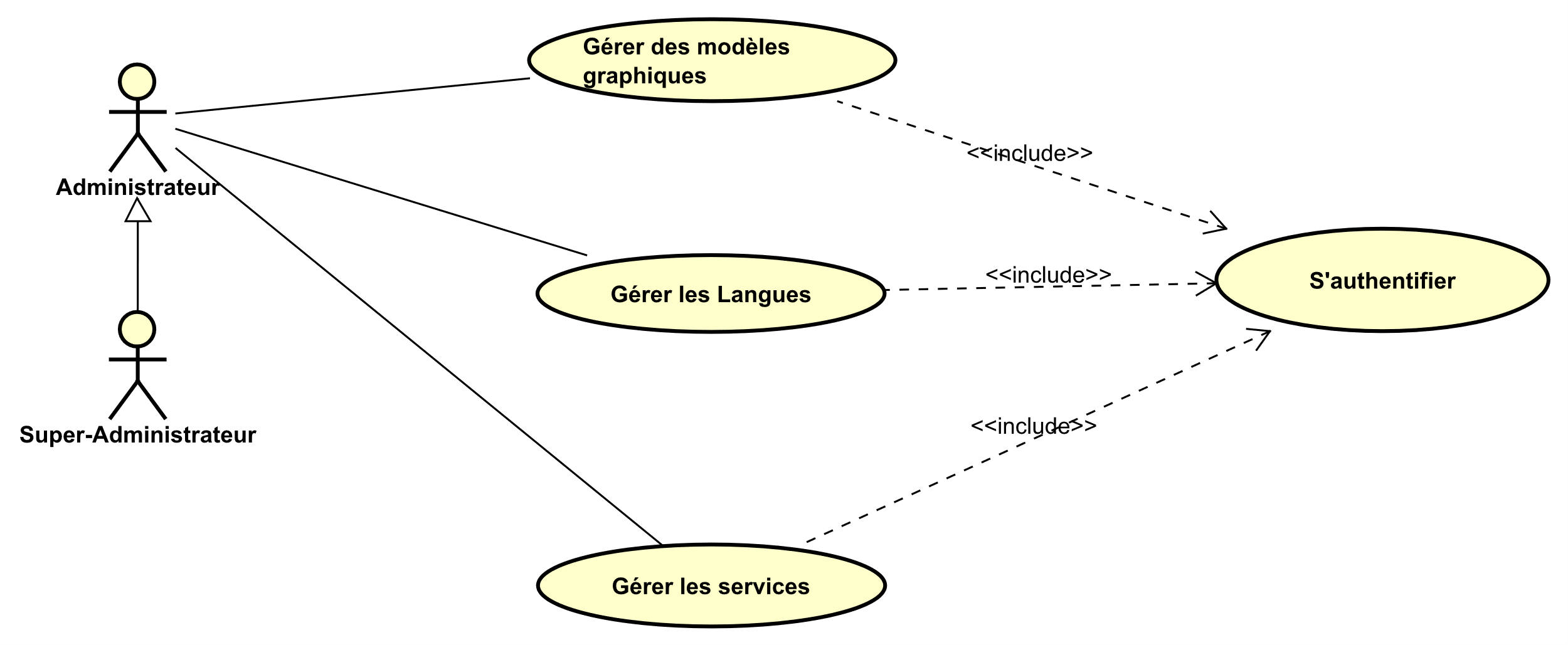
// ici les capture d’écran

# Sprint 2 : paramétrage de l'application

|  |  |
| --- | --- |
| **Release 1** | |
| **Sprint 1** | **Sprint 2** |
| Gestion des propriétaires et des entreprises | Configuration et paramétrage de l'application |

* + 1. **Analyse et conception**
       1. **Diagramme de cas d’utilisation du Sprint 2**

Dans cette section, nous allons présenter le diagramme de cas d'utilisation global du premier sprint.



*Figure 59: Diagramme de cas d'utilisation du sprint 2 : Configuration et paramétrage de l'application*

* + - 1. **Description textuelle du Sprint 2**

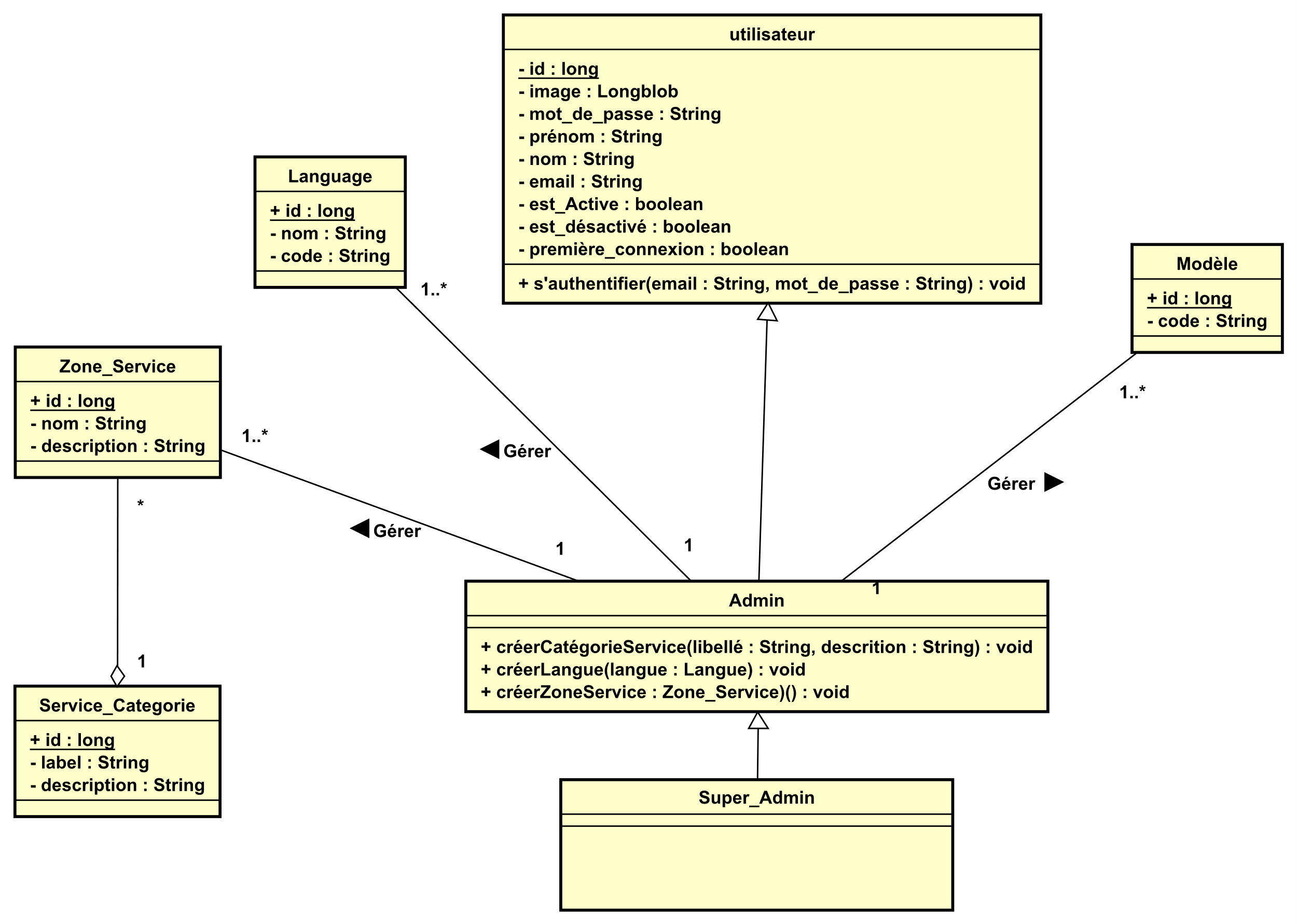
Dans le cadre du sprint 2, nous détaillons les cas d'utilisation suivants :

|  |  |
| --- | --- |
| **Cas d’utilisation** | Configuration et paramétrage de l'application |
| **Acteurs** | Super Administrateur, Administrateur |
| **Précondition** | L'utilisateur est connecté en tant qu'administrateur. |
| **Postcondition** | Les modifications ont été enregistrées avec succès. |
| **Scenario nominal** | * L'administrateur accède à l'interface de gestion. * L'administrateur sélectionne l'option pour gérer les modèles graphiques. * Le système affiche la liste des modèles existants. * L'administrateur choisit de créer, modifier ou supprimer un modèle graphique. * Le système enregistre les modifications et confirme leur succès. * L'administrateur répète les étapes pour gérer les langues et les services. |
| **Scenario alternatif** | * Aucun modèle, langue ou service de zone n'existe dans le système. * Le système rencontre des erreurs lors de l'enregistrement des modifications. * L'administrateur annule une action en cours. |

*Tableau 11: Description Textuelle : Configuration et paramétrage de l'application*

* + - 1. **Diagramme des classes du Sprint 2**

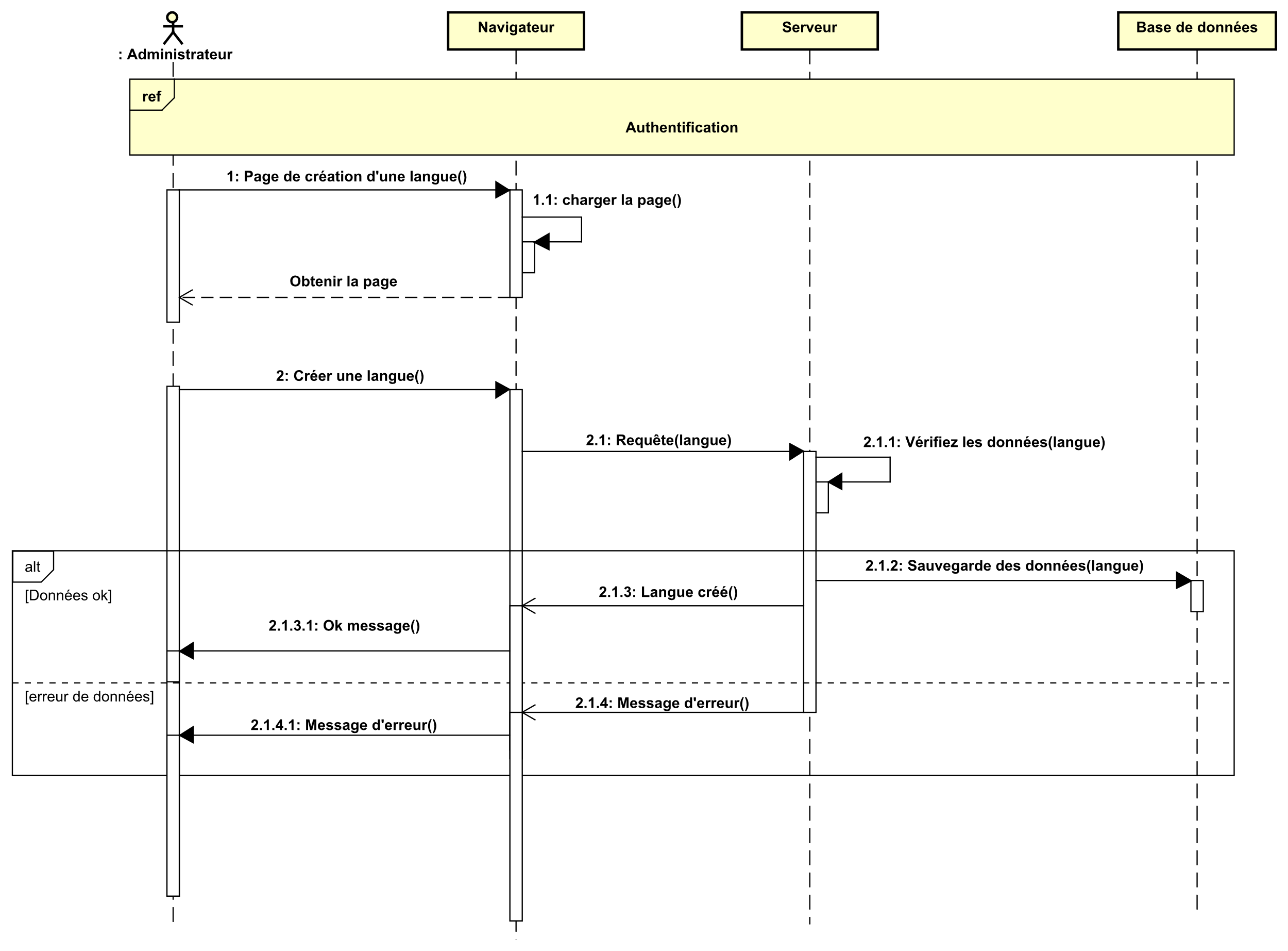
La figure illustre le diagramme des classes du sprint 2.



*Figure 60: Diagramme des classes du Sprint 2*

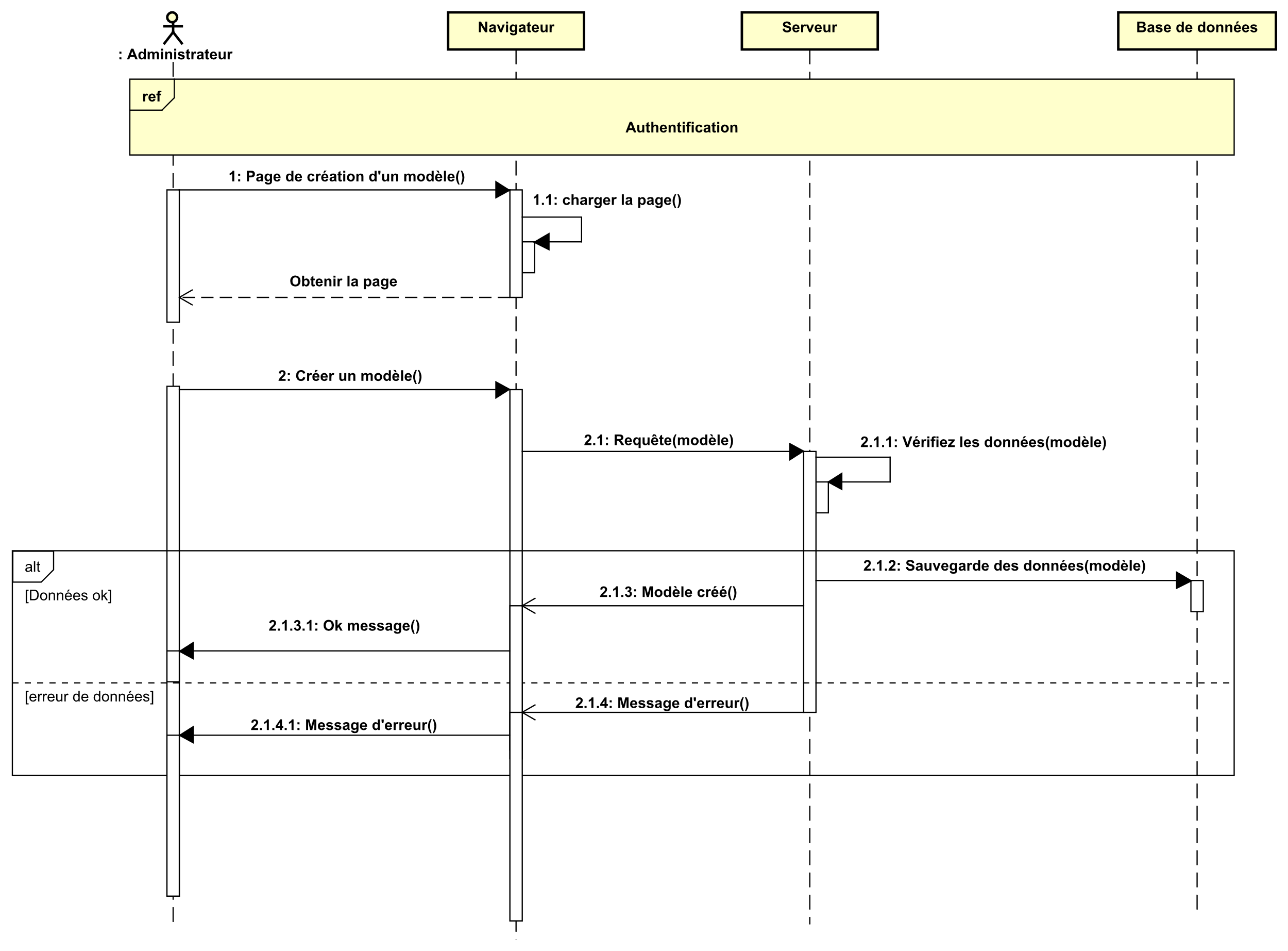
* + - 1. **Diagramme de séquence du Sprint 2**

La figure illustre le diagramme de séquence relative au cas d’utilisation "Ajouter Langue" :

****

*Figure 61: Diagramme de séquence : Ajouter langue*

La figure illustre le diagramme de séquence relative au cas d’utilisation " Ajouter Modèle " :

* + - 1. **Réalisation, test et validation**

*Figure 62: Diagramme de séquence : Ajouter Modèle*

Dans cette section, nous examinerons quelques interfaces de notre produit final qui sont spécifiques à ce sprint.

// ici les capture d’écran

**Conclusion**

Dans cette release, nous avons mis l'accent sur la gestion des propriétaires d'entreprises, offrant ainsi une approche plus personnalisée pour nos utilisateurs. De plus, nous avons enrichi l'expérience utilisateur en permettant le paramétrage de l'application, notamment à travers l'ajout de langues, de modèles graphiques et de services supplémentaires.

**Chapitre 6**

**RELEASE 3**

Dans ce chapitre de la release 3, nous concentrerons sur la gestion avancée des programmes de fidélité. Nous explorerons les fonctionnalités permettant de créer, gérer et personnaliser les programmes de fidélité pour les clients. De plus, l'analyse approfondie des statistiques sur les performances des programmes de fidélité et des enquêtes de satisfaction, dans le but de fournir aux entreprises des insights approfondis pour optimiser leur relation client et améliorer leur expérience globale.

# 6.1 Organisation des Sprints

Pour organiser efficacement les sprints de la release 3, nous avons défini deux objectifs principaux : la gestion des programmes de fidélité, et l'analyse des statistiques. Chaque sprint sera consacré à la réalisation de fonctionnalités spécifiques liées à ces objectifs, permettant ainsi une progression méthodique vers l'achèvement de la release.

|  |  |
| --- | --- |
| Release 3 | |
| Sprint 1 | Sprint 2 |
| Gestion des programmes de fidélité | Analyse et gestion des statistiques |

*Tableau 12: Organisation des sprints de la release 3*

# 6.2 Backlog des sprints pour la release 3

Le backlog des sprints de la release 3 détaille les tâches spécifiques à accomplir pour chaque user story identifiée dans notre backlog produit. Dans cette phase, nous nous concentrons sur deux sprints distincts pour répondre aux objectifs définis. Le sprint 1 est dédié à la gestion des programmes de fidélité, tandis que le sprint 2 se focalise sur la gestion et l’analyse des statistiques.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Backlog des sprints** | | | | |
| **ID** | **Sprint** | **User Story** | **Estimation (En jours)** | **Tâche à réaliser** |
| 3.1 | Gestion des programmes de fidélité | * En tant qu’administrateur, je veux gérer les programmes de fidélité afin de * En tant qu'administrateur, je veux créer des programmes de fidélité par défaut afin d’offrir aux propriétaires d'entreprise des solutions prêtes à l'emploi. | 14 Jours | * Développer les interfaces et la logique backend de l’ajout, de modification et de suppression des programmes de fidélité. * Faire les tests. |
| 3.2 | Gestion et L’analyse des statistiques | * En tant qu’administrateur d'entreprise, je veux consulter les statistiques et les rapports sur la satisfaction afin de comprendre le niveau de satisfaction globale des clients à l'égard des produits ou services de l'entreprise. | * Création des graphiques et des rapports visuels. * Développer les composants nécessaires à l'affichage des statistiques et à leur analyse. * Faire les tests. |

*Tableau 13: Backlog des sprints pour release 3*

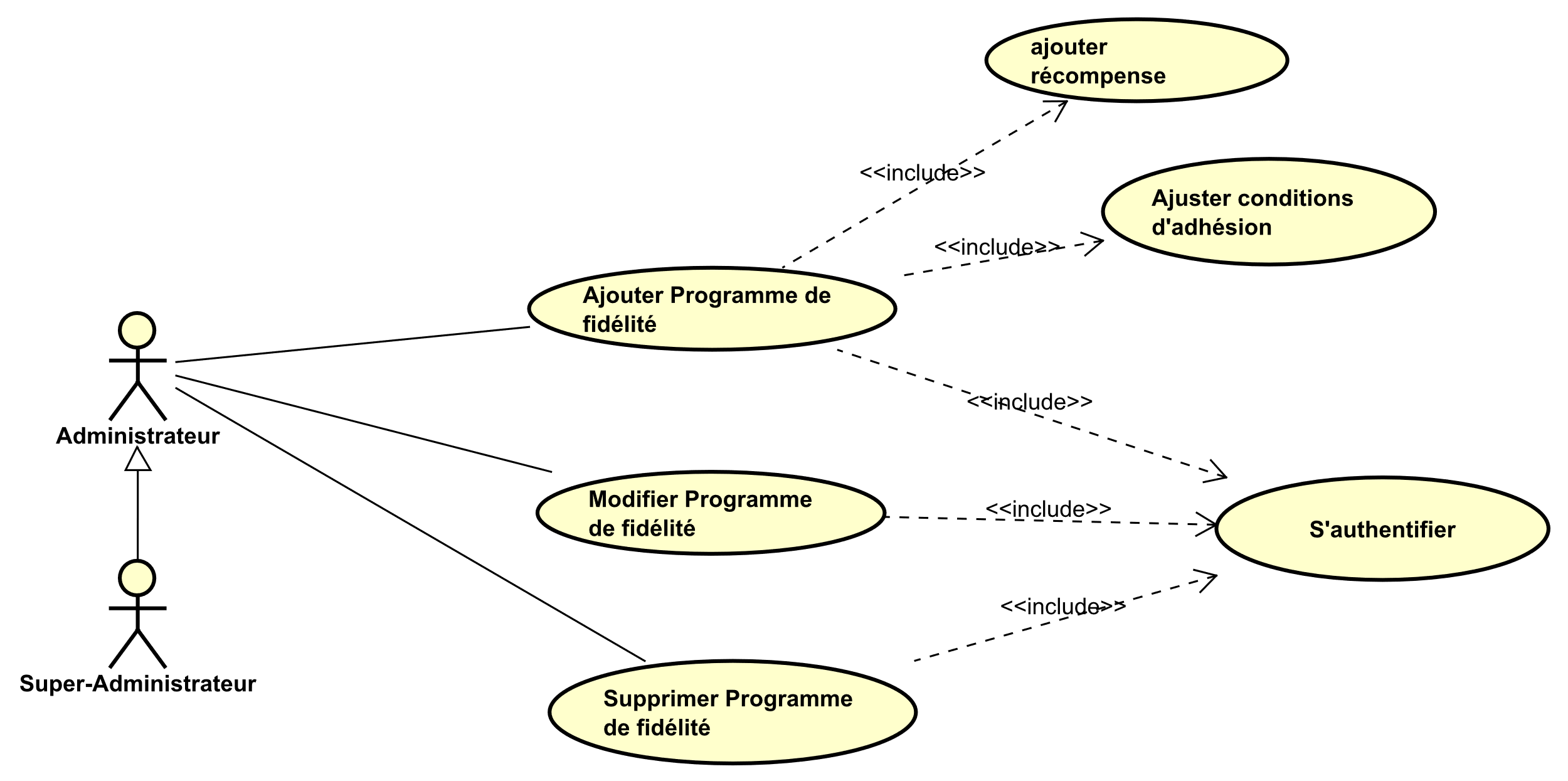
# Sprint 1 : Programmes de fidélité

|  |  |
| --- | --- |
| **Release 1** | |
| **Sprint 1** | **Sprint 2** |
| Gestion des programmes de fidélité | Analyse et gestion des statistiques |

* + 1. **Analyse et conception**

**5.3.1.1 Diagramme de cas d’utilisation du Sprint 1**

Dans cette section, nous allons présenter le diagramme de cas d'utilisation global du premier sprint.



*Figure 63: Diagramme de cas d'utilisation du sprint 1 :* *Gestion des Programmes de fidélité*

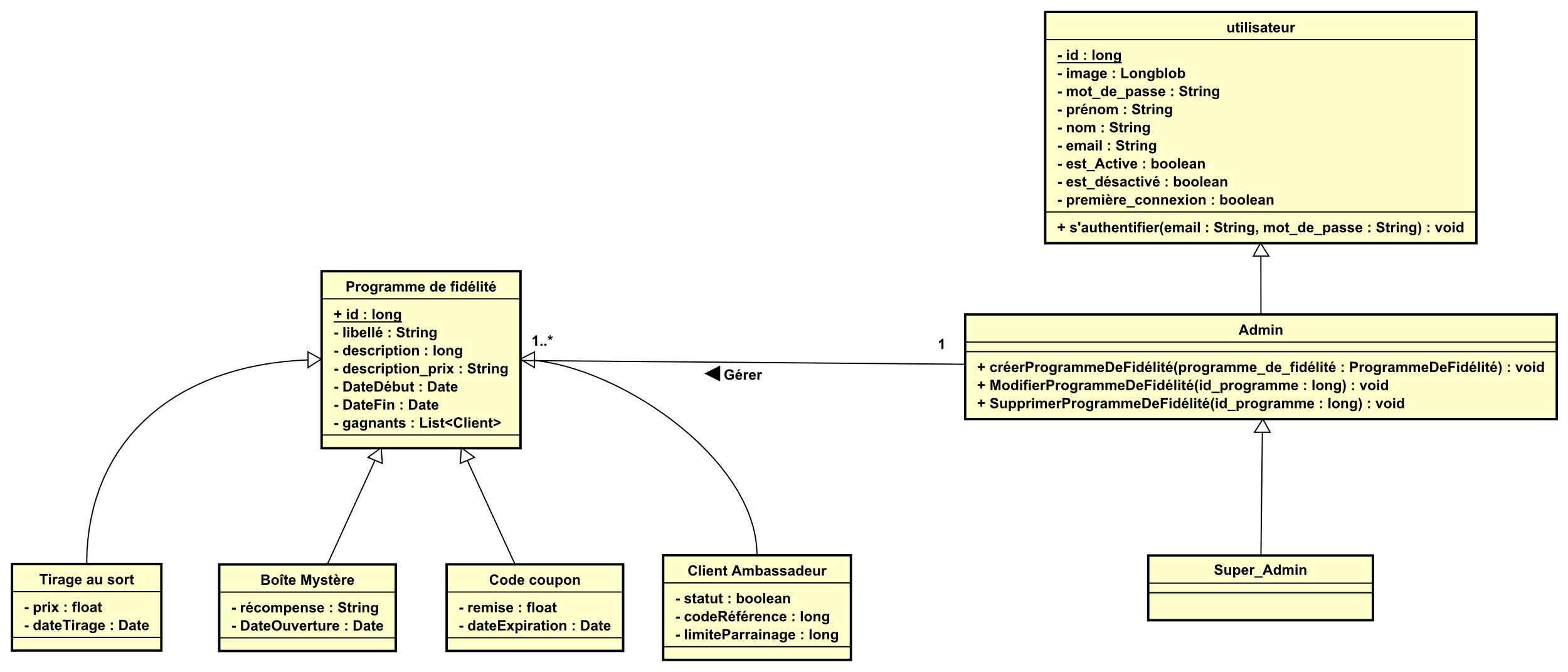
* + - 1. **Description textuelle du Sprint 1**

Dans le cadre du sprint 1, nous détaillons les cas d'utilisation suivants :

|  |  |
| --- | --- |
| **Cas d’utilisation** | Gestion des Programmes de fidélité |
| **Acteurs** | Super Administrateur, Administrateur |
| **Précondition** | L'utilisateur est connecté en tant que super administrateur ou administrateur. |
| **Postcondition** | * Un nouveau programme de fidélité est ajouté avec succès dans le système. * Les modifications apportées aux programmes de fidélité sont enregistrées avec succès dans le système. |
| **Scenario nominal** | * L'administrateur accède à la section de gestion des programmes de fidélité. * L'administrateur choisit l'option pour créer ou modifier un programme de fidélité. * Il saisit les détails du nouveau programme dans le cas de l’ajout. * Il apporte les modifications nécessaires dans le cas de la modification. * Le système envoie un message informant l’administrateur que le programme de fidélité est ajouté ou modifié avec succès. |
| **Scenario alternatif** | * L’utilisateur n'est pas connecté en tant que super administrateur ou administrateur. * Il ne peut pas accéder à l'interface de gestion des programmes de fidélité. |

*Tableau 14: Description Textuelle : Gestion des Programmes de fidélité*

* + - 1. **Diagramme des classes du Sprint 1**

La figure illustre le diagramme des classes du sprint 1.

*Figure 64: Diagramme des classes du Sprint 1*

* + - 1. **Diagramme de séquence du Sprint 1**

****La figure illustre le diagramme de séquence relative au cas d’utilisation " Ajouter programme de fidélité" :

*Figure 65: Diagramme de séquence : Ajouter programme de fidélité*

**Réalisation, test et validation**

Dans cette section, nous examinerons quelques interfaces de notre produit final qui sont spécifiques à ce sprint. Pour chaque cas d'utilisation, nous explorerons tous les scénarios envisageables.

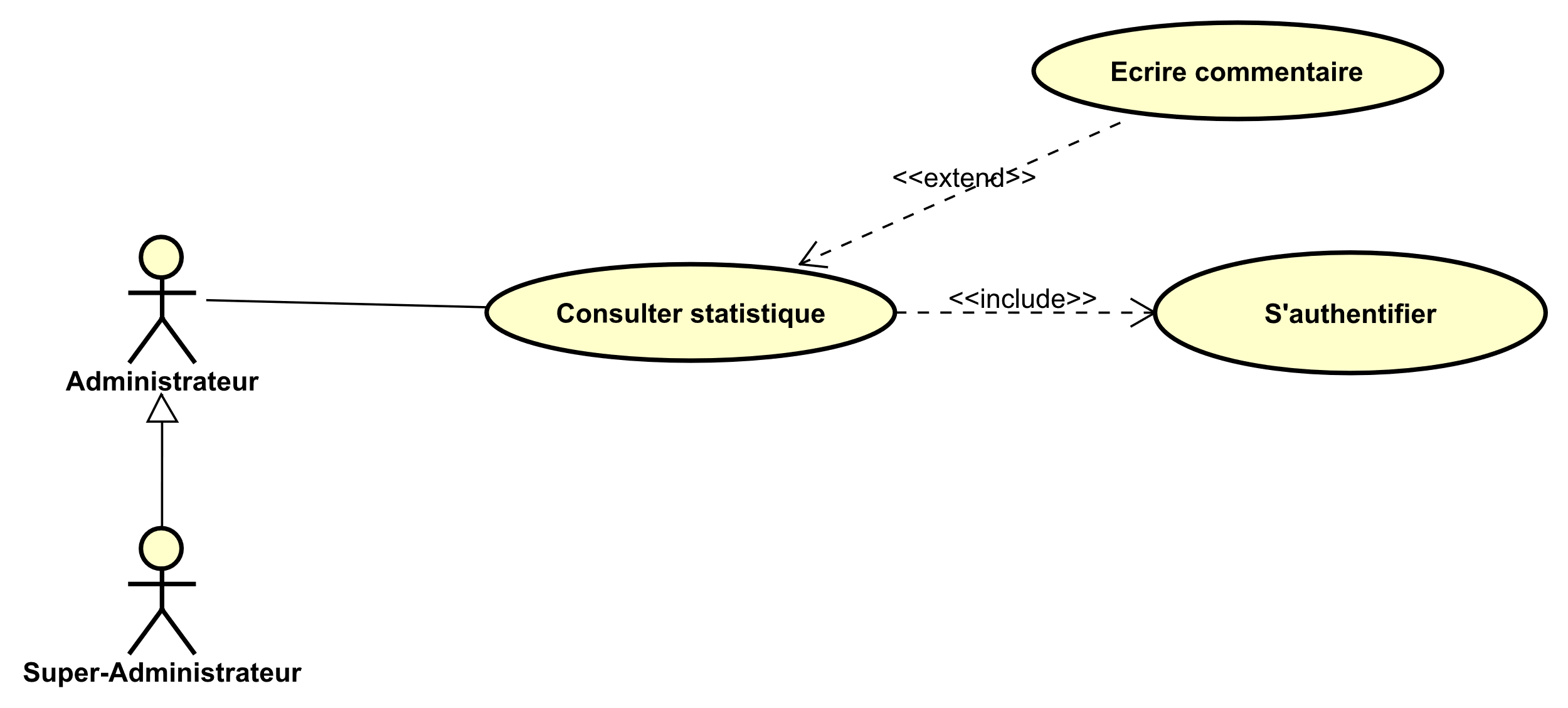
// ici les captures d’écran

# Sprint 2 : Analyse des statistiques

|  |  |
| --- | --- |
| **Release 1** | |
| **Sprint 1** | **Sprint 2** |
| Gestion des programmes de fidélité | Analyse des statistiques |

* + 1. **Analyse et conception**
       1. **Diagramme de cas d’utilisation du Sprint**

Dans cette section, nous allons présenter le diagramme de cas d'utilisation global du deuxième sprint.



*Figure 66: Diagramme de cas d'utilisation du sprint 2 : Analyse des statistiques*

* + - 1. **Description textuelle du Sprint 2**

Dans le cadre du sprint 2, nous détaillons les cas d'utilisation suivants :

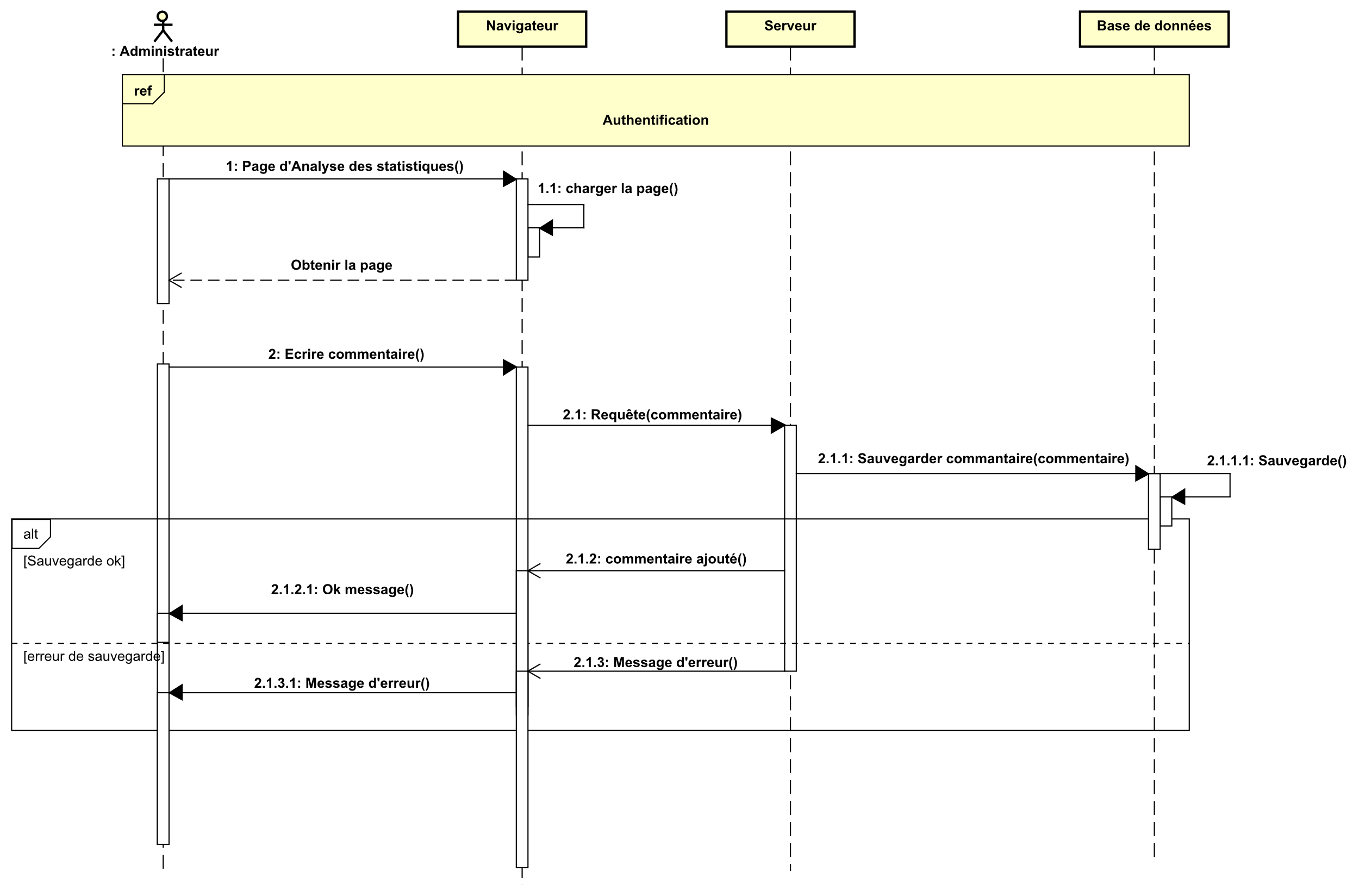
|  |  |
| --- | --- |
| Cas d’utilisation | Analyse des statistiques |
| Acteurs | Super Administrateur, Administrateur |
| Précondition | L'utilisateur est connecté en tant que super administrateur ou administrateur. |
| Postcondition | * L'administrateur a accédé avec succès à la section d'analyse des statistiques de performance. * Le commentaire a été inséré avec succès. |
| Scenario nominal | * L'administrateur accède à la section d'analyse des statistiques de performance. * Il voit le nombre de réponses sur les questionnaires et le NPS pour chaque entreprise. * Le système envoie un message informant l’administrateur que le commentaire a été inséré avec succès. |
| Scenario alternatif | * L’utilisateur n'est pas connecté en tant que super administrateur ou administrateur. * Il ne peut pas accéder à l'interface d'analyse des statistiques. |

*Tableau 15: Description Textuelle : Analyse des statistiques*

*Figure 67: Diagramme des classes du Sprint 2*

* + - 1. **Diagramme de séquence du Sprint 2**

La figure illustre le diagramme de séquence relative au cas d’utilisation " Écrire commentaire" :



*Figure 68: diagramme de séquence : Écrire commentaire*

* + - 1. **Réalisation, test et validation**

Dans cette section, nous examinerons quelques interfaces de notre produit final qui sont spécifiques à ce sprint. Pour chaque cas d'utilisation, nous explorerons tous les scénarios envisageables.

// ici les captures d’écran

**Conclusion**

Dans ce chapitre de la release 3, nous nous sommes concentrés sur la gestion avancée des programmes de fidélité. Nous avons exploré les fonctionnalités qui ont permis de créer, de gérer et de personnaliser ces programmes pour nos propriétaires. De plus, une analyse des statistiques sur les performances des programmes de fidélité et des enquêtes de satisfaction a été menée, dans le but d'offrir aux entreprises des perspectives riches pour optimiser leurs relations clients et améliorer leur expérience globale.

**Conclusion et perspective**

Le présent document est une synthèse complète de notre projet, couvrant différents aspects de notre démarche à travers plusieurs chapitres. Nous avons opté pour la méthode Scrum pour gérer notre projet. Cette approche itérative nous a permis de rester réactifs aux changements, de collaborer efficacement en équipe et de livrer régulièrement des fonctionnalités à tester et à valider.

Notre travail a débuté par la construction minutieuse du cahier des charges, posant ainsi des bases solides pour notre projet. Ensuite, nous avons établi un planning détaillé pour organiser efficacement les différentes phases.

Chaque étape de notre projet a été essentielle. De l'analyse des besoins à la mise en œuvre des fonctionnalités avancées, nous avons progressé grâce à un travail d'équipe méthodique, aboutissant à un produit de qualité répondant aux attentes des utilisateurs.

Jusqu’à présent, on a réalise …

Sur le plan technique, nous avons eu l'opportunité d'appliquer et de développer nos compétences dans un environnement réel, en utilisant des technologies modernes et en relevant des défis concrets.

Sur le plan professionnel, ce projet nous a immergés dans le monde de l'entreprise, nous permettant de comprendre les attentes du marché et de développer des compétences en gestion de projet et en communication.

Enfin, sur le plan académique, ce stage nous a permis de mettre en pratique nos connaissances théoriques et d'enrichir notre parcours universitaire et professionnel. Chaque étape a été une occasion d'apprentissage, de résolution de problèmes et de progression.

Forts des enseignements tirés de cette expérience, nous avons identifié plusieurs pistes d'amélioration et d'innovation pour les développements futurs.

Du point de vue professionnel, nous cherchons à renforcer nos compétences en gestion de projet et en communication, en nous appuyant sur les leçons apprises lors de ce projet. Nous aspirons également à élargir notre réseau professionnel et à explorer de nouvelles opportunités de collaboration avec des entreprises et des partenaires potentiels.

Sur le plan académique, nous envisageons de poursuivre notre formation et notre apprentissage, en exploitant les connaissances acquises lors de ce stage pour enrichir notre parcours universitaire.

**Bibliographie et webographie**

**[**ELYAAKOUBI**]** M. ELYAAKOUBI Mohammed, « Cours 1 » *Génie logiciel,* 2024

**[**ELYOUSFI**]** M. EL YOUSFI, « Cours 1 » *UML,* 2024

**[Spring Boot, VMware tanzu]** Spring Boot Reference Documentation, [En ligne] Date de dernière mise à jour : *18 avril 2024* Disponible sur :

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> , *12 mars 2024*

**[****Spring Boot, VMware tanzu]** Spring Boot, [En ligne] Date de dernière mise à jour : *18 avril 2024* Disponible sur :

<https://spring.io/projects/spring-boot> *10 avril 2024*

**[React]** Spring Boot, [En ligne] Date de dernière mise à jour : *19 avril 2024* Disponible sur :

<https://www.w3schools.com/REACT/DEFAULT.ASP> *16 avril 2024*

**[React, Andrew Clark]** Spring Boot, [En ligne] Date de dernière mise à jour : *10 mars 2024* Disponible sur :

<https://legacy.reactjs.org/docs/getting-started.html>  *14 mars 2024*

**Annexes**

**Annexe A : UML**

**Annexe B : Diagramme de GANTT**

**Annexe C : Diagramme de GANTT**

**Annexe A**

**UML**

* Les années 1970 ont été marquées par l'apparition des premières méthodes d'analyse.
* Dans les années 1980, l'approche systémique a été annoncée, avec l'émergence de méthodes telles que Merise, Axial et IE, qui ont introduit la modélisation des données et des traitements.
* Entre 1990 et 1995, une multitude de méthodes orientées objet ont vu le jour, notamment Booch, Classe-Relation, Fusion, HOOD, OMT, 0OA, OOD, OOM et OOSE. L'année 1995 a été marquée par les premiers consensus autour de méthodes telles qu'OMT (James Rumbaugh), OOD (Grady Booch) et OOSE (Ivar Jacobson).
* Les années 1995 à 1997 ont été dédiées à l'unification et à la normalisation des méthodes, menant à la naissance d'UML (Unified Modeling Language).
* UML est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90.
* **Les diagrammes UML**

UML définit 13 types de diagrammes :

* Diagramme de cas d'utilisation
* Diagramme de classe
* Diagramme d'objet
* Diagramme de séquence
* Diagramme de collaboration
* Diagramme d'états-transitions
* Diagramme d'activité
* Diagramme de composants
* Diagramme de déploiement
* Diagramme de package
* Diagramme de structure composite
* Diagramme de temps
* Diagramme de communication

**Annexe B**

**Diagramme de GANTT**