

Ministère de l'Enseignement Supérieur Et de la Recherche Scientifique

Université de Tunis

École Supérieure des Sciences Économiques et Commerciales de Tunis



Projet de fin d'études en vue de la licence en E-Business

Application RH

Réalisé au sein de la startup Creathech



Réalisé par :

El Ouni Rihab

Jebali Amenallah

Encadré par :

Encadrant Académique
Mr Kooli Anis

Encadrant Professionnel
Mr Mattoussi Helmi

Année Universitaire : 2022-2023

Dédicaces

Remerciements

Table des matières

Introduction générale	1
1 Etude préliminaire du projet	2
1.1 Introduction	2
1.2 Contexte du projet	2
1.3 Présentation de l'organisme d'accueil	2
1.3.1 À Propos	2
1.3.2 Les activités de l'organisme de "Createch"	2
1.4 Etude de l'existant	3
1.4.1 Description de l'existant	3
1.4.2 Critique de l'existant	3
1.4.3 Solution proposée	4
1.5 Méthodologie et langage de conception	4
1.5.1 Méthodes agiles	4
1.5.2 Principes du framework Scrum	5
1.5.3 Langage de modélisation	7
1.6 Style architecturale	8
1.6.1 Modele physique	10
1.6.2 Architecture logique	10
1.6.3 Communication dans le Microservice	11
1.6.4 Rôle du Protocole HTTP	11
1.7 Environnement de travail	12
1.7.1 Langages et framework utilisés	12
1.7.2 Environnement matériel	14
1.7.3 Environnement logiciel	14
1.8 Organisation du travail	17
1.9 Conclusion	17
2 Planification du projet	18
2.1 Introduction	19
2.2 Spécification des besoins	19
2.2.1 Identification des acteurs	19
2.2.2 Besoins fonctionnels	20
2.2.3 Besoins non fonctionnels	21
2.2.4 Diagramme de cas d'utilisation	22
2.3 Pilotage du projet avec Scrum	23
2.3.1 Equipe Et Roles :	23
2.3.2 Contraintes et exigences	23
2.3.3 Risques	23
2.3.4 Backlog du produit	23
2.3.5 Organisation et division du projet :	26
2.4 Diagramme de classes du domaine	27

2.5	Conception de logo :	28
2.6	Conclusion :	29
3	Sprint 1 :Fonctionnalités du responsable RH et de l'employé	30
3.1	Introduction	31
3.2	Sprint Backlog	31
3.3	Analyse et conception	32
3.3.1	Diagramme de cas d'utilisation	33
3.3.2	Diagrammes de séquences détaillés	34
3.3.3	Diagrammes de classes participantes	39
3.4	Réalisation	40
3.4.1	Gérer les années	41
4	Sprint 2 : Authentification , fonctionnalités de l'employé et la gestion de paie	42
4.1	Introduction	43
4.2	Sprint Backlog	43
4.3	Analyse et conception	44
4.3.1	Diagramme de cas d'utilisation	44
4.3.2	Diagrammes de cas d'utilisation raffiné "S'authentifier"	45
4.3.3	Diagrammes de séquence	45
	Webographie	46

Table des figures

1.1	Mode de fonctionnement de la méthodologie Scrum [I1]	6
1.2	Valeurs de la méthodologie Scrum [I2]	7
1.3	Microservices vs monolithique [I3]	9
1.4	Spring logo [I4]	12
1.5	Angular logo [I5]	13
1.6	MySQL logo [I6]	13
2.1	la figure ci-dessus montre les acteurs de notre application	19
2.2	Diagramme de cas d'utilisation global	22
2.3	Structure de decouapge des sprints	26
2.4	Diagramme de classes du domaine	27
2.5	Logo du startup "Createch"	28
2.6	Design et ajustement du logo	28
2.7	Design et ajustement du logo	28
2.8	Logo de microservice RH	29
3.1	Diagramme de cas d'utilisation du Sprint 1	33
3.2	Diagramme de séquence détaillés du cas d'utilisation «Ajouter un employé»	34
3.3	Diagramme de séquence détaillé du cas d'utilisation «Modifier un employé»	36
3.4	Diagramme de séquence détaillé du cas d'utilisation «Modifier un employé»	38
3.5	Diagramme de classes participantes de conception du cas d'utilisation «Gé- rer employé»	40
3.6	Diagramme de classes participantes de conception du cas d'utilisation «Gé- rer entreprise»	40
3.7	L'interface d'ajout d'une entreprise	41
4.1	Diagramme de cas d'utilisation du Sprint 2	44
4.2	Diagramme de cas d'utilisation "S'authentifier"	45
4.3	Diagramme de séquence détaillé du cas d'utilisation «S'authentifier»	45

Liste des tableaux

1.1	Description des machines de développement utilisées	14
2.1	Backlog du produit	24
2.2	Backlog du produit	25
3.1	Backlog du premier sprint	32
3.2	Description textuelle du cas d'utilisation "Ajouter un employé"	35
3.3	Description textuelle du cas d'utilisation "Supprimer un employé"	37
3.4	Description textuelle du cas d'utilisation "Ajouter une entreprise "	39
4.1	Backlog du deuxième sprint	43
4.2	Description textuelle du cas d'utilisation "S'authentifier"	46

Introduction générale

Aujourd'hui, dans le paysage professionnel en constante évolution, la gestion des ressources humaines (RH) est cruciale pour garantir le bon fonctionnement et la croissance des entreprises. Les responsables RH sont confrontés à de multiples tâches administratives et managériales, allant de la gestion des employés à l'administration des salaires, en passant par la planification des congés et des absences. Pour améliorer la productivité et la précision des processus RH, il est nécessaire de trouver une solution efficace et intégrée qui permet d'éviter la répétition et le chronophage de ces tâches.

Dans le cadre de ce projet de fin d'études, la start-up "Createch" s'est donné pour objectif de développer un microservice d'application RH destiné à optimiser la gestion des ressources humaines au sein des entreprises. Contrairement à une application mobile, ce microservice sera conçu pour une utilisation sur ordinateur de bureau, offrant ainsi une solution accessible et conviviale aux entreprises.

Ce rapport est structuré en cinq chapitres, chacun couvrant une phase spécifique du projet :

- Le premier chapitre présente une étude préliminaire du projet, y compris l'organisme d'accueil, le contexte du projet, la méthodologie de travail et l'architecture de l'application.
- Le deuxième chapitre aborde la planification du projet, en détaillant les besoins fonctionnels et non fonctionnels de l'application, ainsi qu'une conception préliminaire.
- Les chapitres trois, quatre et cinq correspondent respectivement aux sprints 1, 2 et 3 du projet, décrivant les fonctionnalités développées pour chaque acteur impliqué dans l'application, y compris les aspects liés à la gestion des ressources humaines.

Nous clôturons le rapport par une conclusion générale.

Chapitre 1

Etude préliminaire du projet

1.1 Introduction

L'étude préliminaire du projet implique de présenter l'organisme d'accueil, d'analyser l'état actuel, de déterminer nos décisions méthodologiques, la méthode d'organisation de notre projet, ainsi que les divers langages, frameworks et architectures que nous utiliserons pour mener à bien notre projet.

1.2 Contexte du projet

Lors de notre formation en Business Computing orientée e-business à l'Université d'Economie et de Commerce de Tunis, nous avons eu l'opportunité de réaliser un projet de fin d'études au sein d'une boîte de développement "Createch". Notre projet est un microservice d'application RH qui vise à simplifier les processus administratifs RH, tels que la gestion des contacts, des congés et de la paie, etc.. En parallèle, une interface web sera développée pour permettre aux acteurs d'accéder facilement aux données essentielles sur la performance du microservice, facilitant ainsi une gestion proactive et informée des ressources humaines.

1.3 Présentation de l'organisme d'accueil

1.3.1 À Propos

L'agence Createch s'est affirmée comme un acteur majeur dans le domaine du numérique, comptant à son actif plusieurs projets couronnés de succès. Nous sommes le terreau fertile où naissent les idées novatrices et les solutions qui propulseront votre projet web ou application vers la réussite. Que vous envisagiez la création d'un site internet, d'une boutique en ligne ou la production de contenu digital, Createch est là pour vous accompagner.

1.3.2 Les activités de l'organisme de "Createch"

Createch propose une offre globale de communication sur internet. De la conception du cahier des charges au référencement du site Internet en passant par la création du site Internet, Createch regroupe toutes les compétences pour faire de votre projet web, une réussite. Nous nous occupons également de votre communication sur les réseaux sociaux et l'optimisation de votre site web pour les appareils mobiles. Createch accorde

une grande importance à la satisfaction et à la réussite de ses clients dans sa stratégie. C'est pourquoi une partie des bénéfices est intégrée chaque année pour la veille technologique est l'amélioration de sa plateforme de suivis clients

1.4 Etude de l'existant

Dans cette section, nous allons décrire l'existant. Ensuite nous présenterons nos critiques ainsi que la solution proposée.

1.4.1 Description de l'existant

Actuellement, la gestion des ressources humaines dans de nombreuses entreprises repose sur des processus manuels et des systèmes disparates. Les tâches clés, telles que la gestion des congés, la paie, et les avantages sociaux, sont souvent réalisées à l'aide de feuilles de calcul, de documents papier, et de logiciels non intégrés. Ce mode de fonctionnement comporte de nombreuses limitations et inefficacités.

La duplication des efforts, les risques élevés d'erreurs humaines et les retards dans le traitement des informations sont souvent des problèmes rencontrés dans les processus manuels. Par exemple, les demandes de congés sont généralement faites par e-mail ou par formulaire papier, nécessitant un suivi manuel et une validation qui peuvent être sources de conflits et de malentendus. De manière similaire, la saisie des fiches de paie requiert souvent des processus manuels ou semi-automatisés, ce qui augmente les risques d'erreurs et de retards.

En outre, les systèmes existants ont souvent du mal à s'adapter aux besoins spécifiques de chaque entreprise en raison du manque de flexibilité et de capacités d'intégration. Par conséquent, les responsables RH passent une grande partie de leur temps à effectuer des tâches administratives répétitives au lieu de se concentrer sur des initiatives stratégiques et sur l'amélioration de la satisfaction des employés.

1.4.2 Critique de l'existant

Plusieurs défis majeurs se posent aux entreprises en ce qui concerne la gestion des ressources humaines :

- Tout d'abord, il est courant de rencontrer des erreurs et des conflits lors de la planification des congés et des absences en raison de processus manuels et non automatisés. Les congés sont souvent approuvés de manière inefficace et sans standardisation, ce qui entraîne des retards et des frustrations.
- Les calculs des primes et de souvent réalisés de manière manuelle, peuvent être sujets à des erreurs et peuvent demander beaucoup de temps.
- Il y a aussi des difficultés à gérer les jours fériés, ce qui peut conduire à des erreurs de planification et de paie.
- En outre, il est difficile de gérer efficacement les avantages liés aux enfants à charge sans automatisation.
- L'établissement des fiches de paie est une autre source couramment utilisée lorsqu'elle est effectuée manuellement, il y a des erreurs et des retards.

- Enfin, il est souvent difficile de prévoir les augmentations de salaire ce qui rend la planification efficace et la satisfaction des employés difficiles.

1.4.3 Solution proposée

Notre application de gestion des ressources humaines offre une solution flexible et configurable, adaptée aux divers besoins des entreprises. Elle couvre toutes les fonctionnalités essentielles, notamment :

- Gestion des congés, des autorisations et des absences : Automatisation des processus de demande et d’approbation, réduction des conflits et des erreurs.
- Gestion des primes et des rémunérations : Calculs automatisés pour minimiser les erreurs et gagner du temps.
- Gestion des jours fériés : Intégration des jours fériés dans le calendrier pour éviter les erreurs de paie et de planification.
- Établissement des fiches de paie : Automatisation du processus pour réduire les erreurs et les retards.
- Prédiction des augmentations de salaire : Outils analytiques pour aider à prévoir et planifier les augmentations de salaire, améliorant la satisfaction et la rétention des employés.

En offrant un accès facile et une gestion centralisée de ces aspects cruciaux de la gestion des ressources humaines, notre application favorise une relation solide entre les employés et les employeurs. Elle assure une transparence accrue, une gestion plus équitable des avantages, et contribue à la satisfaction des employés en leur offrant une meilleure visibilité sur leur rémunération et leur progression professionnelle. En récapitulant, notre application renforce la relation employé-employeur en offrant une plateforme flexible et exhaustive pour la gestion des ressources humaines.

1.5 Méthodologie et langage de conception

La conception est extrêmement importante lors du développement de systèmes informatiques pour répondre aux besoins des clients. Les décisions conceptuelles concernant les approches, les langages de modélisation et les processus de développement sont basées sur des considérations collectives. La méthodologie Scrum est une méthode agile pour laquelle nous avons opté pour mener à bien le projet.

1.5.1 Méthodes agiles

Les méthodes agiles sont des approches de développement logiciel itératives et incrémentales, mises en œuvre de manière très collaborative par des équipes cherchant à produire un logiciel capable de s’adapter aux besoins changeants des utilisateurs. Scott Ambler, expert reconnu dans le domaine des pratiques de développement Agile, décrit ces méthodes comme étant particulièrement adaptées pour répondre à l’évolution constante des besoins des utilisateurs.

Parmi les méthodes agiles les plus fréquemment employées de nos jours, on peut citer :

- Extrême Programming (XP)

- Scrum
- Agile Unified Process (Agile UP ou AUP)

Ces méthodes agiles mettent l'accent sur une approche itérative et incrémentale, une concentration sur les besoins des utilisateurs, une flexibilité pour répondre au changement, une collaboration étroite avec les participants au projet et l'accent mis sur la fourniture régulière de fonctionnalités à valeur ajoutée. Bien que chacune de ces méthodologies agiles ait ses propres pratiques, rôles, rituels et outils, elles partagent toutes les mêmes valeurs et principes fondamentaux de l'agile.

Pour notre projet, nous avons opté pour l'utilisation du cadre Scrum. Le framework agile scrum nous a semblé le plus approprié pour notre travail. Il propose une structure souple qui permet de diviser le projet en plusieurs sprints. Cette méthode itérative nous permettra de gérer notre travail de manière plus efficace en nous focalisant sur des objectifs concrets à atteindre à chaque étape du processus de développement.

1.5.2 Principes du framework Scrum

Scrum est un cadre de travail pour la gestion de projet agile qui est largement utilisé à travers le monde. Il comprend des rôles bien définis, un rythme itératif, des réunions chronométrées et des artefacts tels que le product backlog, le sprint backlog et le graphique d'avancement (Burndown Chart). Ces éléments constituent les piliers du processus de gestion de projet agile offert par Scrum.[1]

Sprint

Tout projet Scrum est organisé autour de «Sprints» de développement pendant lesquels une équipe de développement travaille sur un ensemble de fonctionnalités ou d'objectifs spécifiques.[1]

User story

Les fonctionnalités requises sont présentées sous la forme de "User Stories" dans une liste organisée.[1]

Backlog du produit

Les user stories sont organisées dans le backlog du produit.[1]

Sprint backlog

Le sprint backlog est une liste des éléments du product backlog sélectionnés pour être développés pendant un sprint exacte. C'est une liste de tâches à accomplir pendant le sprint. Les membres de l'équipe de développement se concentrent sur ces tâches pour les livrer à la fin du sprint.[1]

Planning poker

Pendant une réunion nommée "Réunion du Planning Poker", les user stories de chaque sprint sont estimées en points et priorisées.[1]

La mêlée quotidienne

La mêlée quotidienne permet aux membres de l'équipe de se synchroniser régulièrement, de signaler rapidement les problèmes et les défis qu'ils rencontrent et de suivre l'avancement du sprint en temps réel. Cette réunion rapide et efficace est un moyen pour l'équipe de travailler de manière collaborative et d'assurer la réussite du projet.[1]

Revue de sprint

L'objectif de la revue de sprint est d'inspecter l'incrément produit au cours du sprint fini.[1]

Rétrospective

La rétrospective est une réunion organisée à la fin de chaque sprint pour évaluer ce qui a bien fonctionné et ce qui peut être amélioré pour le prochain sprint. L'objectif est d'améliorer continuellement les processus de travail et la collaboration de l'équipe Scrum.[1]



FIGURE 1.1 – Mode de fonctionnement de la méthodologie Scrum [I1]

Rôles du framework Scrum

Il existe trois rôles dans l'organisation d'un projet agile suivant la méthodologie Scrum :

- **Product Owner** : Le Product Owner est celui qui porte la vision du produit à réaliser et travaille en interaction avec les Développeurs. Il apporte ses connaissances pour aider l'équipe de développement à comprendre les besoins et les attentes du client, tout en veillant à ce que le produit final réponde à ces exigences.[1]

- **Scrum Master** : Le Scrum Master est un rôle clé dans le cadre de la méthode Scrum. Il est responsable de s'assurer que l'équipe Scrum suit les principes et les pratiques de Scrum. Le Scrum Master facilite les réunions et les événements Scrum, et aide l'équipe à résoudre les obstacles ou les problèmes qui peuvent les entraver .[1]
- **Scrum Team** : Equipe autogérée et multidisciplinaire constituée de développeurs, testeurs... chargés de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables.[1]



FIGURE 1.2 – Valeurs de la méthodologie Scrum [I2]

1.5.3 Langage de modélisation

Dans le but de visualiser la conception de notre système, nous avons employé le langage de modélisation unifié UML (Unified Modeling Language) pour représenter les exigences du logiciel à développer.

1.6 Style architecturale

L'architecture d'une application réfère à sa structure fondamentale, organisant les composants logiciels en modules distincts pour assurer une meilleure gestion du code et une réutilisation efficace des fonctionnalités. Typiquement organisée en couches telles que la présentation, le logique métier et l'accès aux données, elle facilite la communication entre ces composants via des API ou des messages, favorisant ainsi la scalabilité et la fiabilité de l'application. La sécurité et l'évolutivité sont également des préoccupations majeures, intégrant des mécanismes d'authentification, de gestion des autorisations et de tolérance aux pannes pour garantir le bon fonctionnement de l'application dans des conditions variées.

Modèles d'architecture d'application courants :

Chaque application logicielle est unique, ce qui signifie qu'aucune ne sera conçue exactement de la même manière. Cependant, il existe suffisamment de similitudes entre de nombreuses applications pour que plusieurs motifs ou types d'architectures communs se soient développés.

Les architectures d'application les plus courantes incluent :

1. **Architecture monolithique** : Dans ce modèle, une seule application avec un seul code source gère toutes les fonctions. Cette architecture était couramment utilisée dans de nombreuses anciennes applications d'entreprise, mais elle est de moins en moins préférée en raison de sa lenteur à mettre à jour et de sa difficulté à entretenir. Cependant, elle peut toujours être utile pour de très petites applications maintenues par de petites équipes.
2. **Architecture en couches (N-tiers)** : Les fonctions de l'application sont divisées en couches hiérarchiques, chacune traitant une portion différente de la fonctionnalité. L'architecture client-serveur en deux tiers est la plus courante et la plus simple, tandis que l'architecture en trois tiers est couramment utilisée pour les applications web.
3. **Architecture à microservices** : Chaque fonction d'une application est divisée en un petit service ou composant qui fonctionne de manière indépendante des autres. Les microservices utilisent souvent des interfaces de programmation d'applications (API) pour communiquer entre eux. Cette architecture permet des mises à jour rapides et une évolutivité rapide, mais peut entraîner un cloisonnement de la conception.

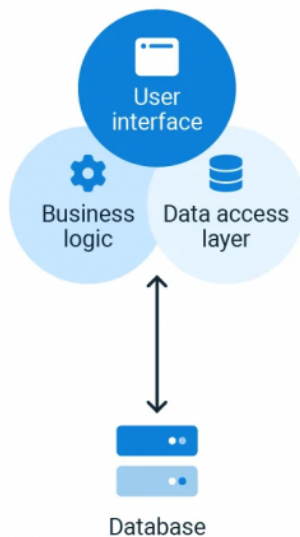
Ces architectures peuvent utiliser diverses méthodes pour atteindre des sous-objectifs, telles que l'architecture sans serveur, les applications monopages ou le cloud computing. Le choix de l'architecture dépendra des besoins spécifiques de chaque application et des contraintes de développement.

Problèmes des architectures monolithiques :

1. **Difficulté de mise à jour** : Les mises à jour peuvent être complexes car toute l'application doit être déployée ensemble, ce qui augmente les risques et ralentit le processus.

2. **Évolutivité limitée** : L'ajout de nouvelles fonctionnalités peut devenir difficile à mesure que l'application grossit, car tout est intimement lié dans un seul codebase.
3. **Maintenance coûteuse** : La maintenance d'une architecture monolithique peut être coûteuse en raison de sa complexité et de sa taille.

Monolithic Architecture



Microservice Architecture

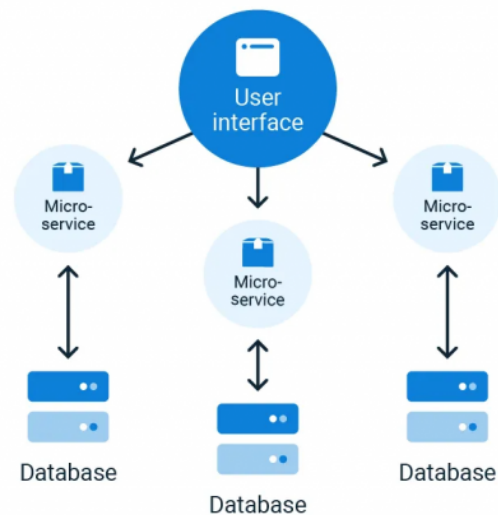


FIGURE 1.3 – Microservices vs monolithique [I3]

Problèmes des architectures N-tiers :

1. **Complexité accrue** : Diviser l'application en différentes couches peut ajouter de la complexité et de la surcharge cognitive pour les développeurs.
2. **Latence** : Chaque interaction entre les différentes couches peut introduire de la latence, ce qui peut affecter les performances de l'application.
3. **Synchronisation** : La gestion de la synchronisation entre les différentes couches peut être difficile, surtout dans les architectures distribuées.

Facteurs et avantages des microservices :

1. **Décomposition Modulaire** : Les microservices décomposent l'application en services indépendants, chacun responsable d'une fonctionnalité spécifique.
2. **Scalabilité granulaire** : Les microservices permettent de faire évoluer chaque service individuellement en fonction des besoins, ce qui améliore la scalabilité.
3. **Déploiement indépendant** : Chaque microservice peut être déployé et mis à jour indépendamment des autres, ce qui facilite les mises à jour continues et réduit les risques.

4. **Meilleure résilience** : En cas de défaillance d'un microservice, le reste de l'application peut continuer à fonctionner, ce qui améliore la résilience de l'ensemble du système.
5. **Technologies adaptées** : Les microservices sont souvent associés à des technologies modernes telles que les conteneurs et les orchestrateurs, ce qui facilite la gestion et l'orchestration des services.

En transitionnant vers les microservices, les organisations cherchent une approche plus flexible et évolutive pour le développement d'applications, répondant ainsi aux défis posés par les architectures monolithiques et N-tiers.

1.6.1 Modèle physique

Pour le modèle physique de notre application de gestion des ressources humaines, nous avons conçu une architecture centralisée accessible via une interface utilisateur conviviale. Une base de données centrale stockera toutes les données relatives aux employés, aux congés, aux autorisations, aux absences, aux primes, aux enfants à charge, aux fiches de paie, aux salaires, ainsi qu'aux historiques et aux prévisions d'augmentations de salaire. Cette base de données sera robuste, sécurisée et évolutive pour gérer efficacement les informations de toutes les entreprises clientes.

1.6.2 Architecture logique

Notre microservice adopte une architecture organisée et modulaire, basée sur le modèle MVC (Modèle-Vue-Contrôleur). Cette approche permet de séparer clairement les différentes responsabilités au sein de l'application, favorisant ainsi sa maintenance et son évolutivité à long terme.

Composants clés de l'architecture :

1. Entités (Entities) :

- Contient les classes qui représentent les entités métier telles que les employés, les congés, les autorisations, etc.
- Chaque entité est associée à une table dans la base de données.

2. DTOs (Data Transfer Objects) :

- Comprend les objets de transfert de données utilisés pour transporter les données entre les différentes couches de l'application, en particulier entre les contrôleurs et les services.
- Ils sont conçus pour transporter uniquement les données nécessaires, améliorant ainsi les performances et la sécurité.

3. Mappers (Mappers) :

- Utilise Mapstruct pour mapper les entités aux DTOs et vice versa.
- Fournit des fonctions de conversion automatique entre les différentes représentations des données.

4. Référentiels (Repositories) :

- Contient les interfaces JpaRepository qui définissent les opérations de base de données (CRUD) pour chaque entité.
- Facilite l'accès aux données stockées dans la base de données.

5. Services :

- Contient la logique métier de l'application, y compris la gestion des employés, des congés, des autorisations, etc.
- Implémente les règles métier et coordonne l'interaction entre les différentes entités et les repositories.

6. Contrôleurs (Controllers) :

- Gère les requêtes HTTP et les réponses associées pour chaque endpoint de l'API REST.
- Transmet les données entre les services et les clients de l'API.

En utilisant cette architecture logique, notre microservice est bien organisé, modulaire et facilement extensible. Chaque couche joue un rôle spécifique dans le traitement des données et des opérations métier, ce qui facilite la maintenance et l'évolution de l'application au fil du temps.

1.6.3 Communication dans le Microservice

1. Communication Interne :

- Les différentes couches de notre microservice RH communiquent entre elles pour traiter les demandes des clients et effectuer les opérations métier nécessaires.
- Par exemple, les contrôleurs communiquent avec les services pour récupérer ou traiter les données, tandis que les services interagissent avec les référentiels pour accéder aux données stockées en base de données.

2. Communication Externe :

- Notre microservice RH expose une API RESTful qui permet aux clients externes, tels que d'autres microservices ou des applications clientes, d'interagir avec lui.
- Les clients envoient des requêtes HTTP aux endpoints de l'API, et le microservice répond avec les données demandées ou les résultats des opérations.

1.6.4 Rôle du Protocole HTTP

1. Protocole de Communication :

- HTTP (Hypertext Transfer Protocol) est le protocole de communication standard utilisé pour les échanges de données sur le Web.
- Il définit la structure des requêtes et des réponses échangées entre les clients et les serveurs.

2. Caractéristiques Clés :

- Stateless : Chaque requête HTTP est traitée de manière indépendante, sans état de session persistant entre les requêtes.
- Verbeux : Les requêtes et les réponses HTTP peuvent contenir des en-têtes et des corps, permettant de transporter des métadonnées et des données.
- Basé sur l'URI : Les ressources sont identifiées par des URI (Uniform Resource Identifiers) dans les requêtes HTTP.

3. Utilisation dans notre Microservice RH :

- Les clients envoient des requêtes HTTP aux endpoints de notre API REST pour effectuer des opérations telles que la récupération, l'ajout, la modification ou la suppression de données.
- Les contrôleurs du microservice reçoivent ces requêtes HTTP, les analysent et déclenchent les actions appropriées dans les services métier.
- Les réponses sont renvoyées aux clients sous forme de réponses HTTP avec des codes d'état appropriés et des données associées, le cas échéant.

En résumé, le protocole HTTP assure la communication entre les clients et notre microservice RH en définissant la structure et les règles des requêtes et des réponses échangées, facilitant ainsi l'interaction et l'intégration avec d'autres systèmes.

1.7 Environnement de travail

Dans cette section, nous détaillons l'environnement de travail à savoir l'environnement matériel, l'environnement logiciel ainsi que les langages et framework utilisés.

1.7.1 Langages et framework utilisés

Les langages et Frameworks utilisés pour l'implémentation de notre solution sont les suivants

Spring Boot

Spring Boot est un framework Java open-source utilisé pour créer des microservices. Il est utilisé pour programmer des applications autonomes de qualité de production basées sur Spring avec un effort minimal. Spring Boot est une extension de convention sur configuration pour la plateforme Java Spring, conçue pour aider à minimiser les préoccupations de configuration lors de la création d'applications basées sur Spring. La plupart des applications peuvent être préconfigurées en utilisant la "vision orientée opinion" de l'équipe Spring sur la meilleure configuration et l'utilisation de la plateforme Spring et des bibliothèques tierces.[2]



FIGURE 1.4 –
Spring logo [I4]

Angular

Angular, également connu sous le nom d'« Angular 2+ » ou « Angular v2 et plus », est un framework open source pour le développement côté client, basé sur TypeScript. Il est co-dirigé par l'équipe du projet Angular chez Google ainsi que par une communauté de contributeurs individuels et d'entreprises. Angular représente une refonte complète d'AngularJS, un framework également développé par la même équipe. Angular est construit sur une architecture de type Modèle-Vue-Contrôleur (MVC), qui permet de séparer clairement les données, la logique de présentation et les actions de l'utilisateur.



FIGURE 1.5 –
Angular logo [I5]

MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL (General Public License) et propriétaire. MySQL est l'un des logiciels de gestion de bases de données les plus largement utilisés au monde, aussi bien par le grand public (notamment dans les applications web) que par les professionnels, rivalisant avec des solutions telles qu'Oracle, PostgreSQL et Microsoft SQL Server.



FIGURE 1.6 –
MySQL logo [I6]

Combinaison Spring Boot Angular

La combinaison de Spring Boot et Angular est une approche populaire pour le développement d'applications web modernes. Voici quelques points clés sur cette combinaison :

☐ Backend avec Spring Boot :

Spring Boot est utilisé pour créer le backend de l'application, qui gère la logique métier, la persistance des données, l'authentification, etc. Spring Boot simplifie le développement en fournissant des fonctionnalités telles que l'auto-configuration, l'injection de dépendances, et une intégration facile avec d'autres frameworks Spring.

☐ Frontend avec Angular :

Angular est utilisé pour construire le frontend de l'application, qui gère l'interface utilisateur, l'interaction utilisateur, et l'affichage des données. Angular offre des fonctionnalités avancées telles que la liaison de données bidirectionnelle, les composants réutilisables, le routage, et la gestion de formulaires.

☐ Communication entre le frontend et le backend :

Spring Boot expose des API RESTful ou utilise GraphQL pour fournir des points de terminaison d'API sécurisés pour les opérations CRUD (Create, Read, Update,

Delete). Angular consomme ces API pour récupérer et manipuler les données, ce qui permet une séparation claire entre le frontend et le backend.

□ Authentification et autorisation :

Spring Security est souvent utilisé avec Spring Boot pour gérer l'authentification et l'autorisation des utilisateurs. Angular peut intégrer des mécanismes d'authentification côté client, tels que les tokens JWT (JSON Web Tokens), pour sécuriser les interactions avec le backend.

□ Déploiement :

L'application peut être déployée en tant qu'archive JAR autonome contenant Spring Boot pour le backend, et les fichiers statiques générés par Angular pour le frontend. Des solutions telles que Docker et Kubernetes peuvent être utilisées pour déployer l'application dans des environnements de production.

En combinant Spring Boot pour le backend et Angular pour le frontend, les développeurs bénéficient d'une approche robuste et modulaire pour la construction d'applications web modernes, offrant à la fois une expérience utilisateur riche et une architecture backend solide et évolutive.

1.7.2 Environnement matériel

Le tableau (Table 1.2) montre les caractéristiques techniques des machines utilisées pour la réalisation de notre application.

Nom de l'ordinateur	ASUS ZenBook	LENOVO LEGION Y520
Propriétaire	Rihab El Ouni	Amenallah Jebali
Processeur	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz	AProcessor : AMD Ryzen 5 1600 Six-Core Processor 3.20 GHz
RAM	8 gb	16 gb 3200 frequency
Système d'exploitation	Windows 11 Famille Unilingue	Windows 10 Famille Unilingue

TABLE 1.1 – Description des machines de développement utilisées

1.7.3 Environnement logiciel

Les logiciels utilisés pour l'implémentation de notre solution sont les suivants :

Git

Git est un système de contrôle de version distribué qui permet de suivre les modifications apportées à n'importe quel ensemble de fichiers informatiques. Il est principalement utilisé pour coordonner le travail entre les programmeurs qui développent collaborativement du code source lors du développement logiciel. Les objectifs de Git comprennent la rapidité, l'intégrité des données et le support des flux de travail distribués et non linéaires.



GitHub

GitHub est en effet une plateforme pour les développeurs qui leur permet de créer, stocker, gérer et partager leur code. Elle utilise le logiciel Git, offrant ainsi un contrôle de version distribué, ainsi que des fonctionnalités de contrôle d'accès, de suivi des bugs, de demandes de fonctionnalités logicielles, de gestion des tâches, d'intégration continue, et de wikis pour chaque projet.[19]



Visual Studio Code (V 1.45.1)

Microsoft a développé un éditeur de code gratuit et open source, compatible avec divers langages de programmation tels que Dart. Il prend également en charge différents environnements de développement tels que Flutter. Nous avons utilisé cet éditeur pour créer à la fois la partie front-end et la partie back-end de notre application en utilisant Flutter et Firebase.[7].



IntelliJ IDEA

IntelliJ IDEA également connu sous les noms d'« IntelliJ », « IDEA » ou « IDJ », est un environnement de développement destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.



Postman

Postman est une application utilisée pour tester des API, créée en 2012 par Abhinav Asthana, Ankit Sobti et Abhijit Kane à Bangalore, en Inde. Initialement, elle était proposée sous forme de module complémentaire pour Google Chrome, puis elle est devenue un client lourd avant de se transformer en client léger. Postman est largement utilisé par plus de 500 000 entreprises dans le monde entier pour tester et déboguer des API.



MySQL Workbench

MySQL Workbench, anciennement connu sous le nom de MySQL Administrator, est un logiciel de gestion et d'administration de bases de données MySQL, créé en 2004. Il offre une interface graphique intuitive permettant aux utilisateurs de créer, modifier ou supprimer des tables, des comptes utilisateurs, ainsi que d'effectuer d'autres opérations liées à la gestion d'une base de données MySQL.



Overleaf

Overleaf est un éditeur en ligne de LaTeX, un langage de composition de documents scientifiques et techniques. Il permet aux utilisateurs de créer, de modifier et de collaborer sur des documents LaTeX en temps réel. Tout au long de notre stage de projet de fin d'étude, nous l'avons utilisé pour travailler ensemble sur un seul document et suivre les modifications apportées par chacune de nous en temps réel.[16]



StarUML

StarUML est un outil de génie logiciel spécialisé dans la modélisation UML, développé par la société coréenne MK-Labs. Il est compatible avec plusieurs plateformes, notamment Windows, Linux et MacOS. La dernière version de StarUML prend en charge l'ensemble des diagrammes définis par UML 2, ainsi que plusieurs diagrammes SysML, les organigrammes, les diagrammes de flux de données et les diagrammes entité-association. Cela en fait un outil polyvalent pour la création de divers types de modèles dans le domaine du génie logiciel.[7]



1.8 Organisation du travail

Pour organiser notre travail durant la réalisation de notre projet, nous avons commencé par réaliser une séance de brainstorming . Nous avons ensuite partagé nos idées avec notre encadrant académique qui nous a aidé à les filtrer et à les développer davantage. Une fois la phase de conception terminée, nous avons entamé la phase de codage. Pour faciliter le partage du code en temps réel, nous avons décidé d'utiliser GitLab (figure 1.17). Chacune de nous a donc pu pousser "push" ses changements dans sa propre branche une fois les tâches accomplies, ce qui nous a permis de suivre l'avancement de chacune et de respecter les délais de réalisation de nos sprints.

1.9 Conclusion

Ce chapitre a été consacré à l'introduction de l'organisme d'accueil, à l'examen des pratiques existantes, et à la description des choix méthodologiques et de l'architecture. Le prochain chapitre se focalisera sur l'organisation du projet et la définition des fonctionnalités de l'application.

Chapitre 2

Planification du projet

2.1 Introduction

L'objectif de ce chapitre est de présenter les différentes parties impliquées dans le projet, les exigences fonctionnelles et non fonctionnelles demandées par le client et le backlog produit. En d'autres termes, il donne un aperçu des éléments clés nécessaires à la compréhension de l'ensemble du projet.

2.2 Spécification des besoins

Dans cette section du chapitre, nous exposons les différents acteurs et les divers besoins fonctionnels et non fonctionnels de notre microservice.

2.2.1 Identification des acteurs

Cette étape consiste à déterminer les acteurs qui vont interagir directement avec le système étudié. Un acteur représente un rôle spécifique joué par une entité externe interagissant avec le système. Cette entité peut être un utilisateur humain, un appareil matériel, ou un autre système. Notre système est conçu pour être utilisé par trois profils d'acteurs :

- **Internaute (visiteur)** : Un utilisateur non authentifié dans l'application est un internaute. Il a la possibilité de consulter les services offerts sur la page d'accueil (landing Page) et de prendre contact avec l'entreprise afin d'obtenir des renseignements sur les services .
- **Responsable des Ressources Humaines (RH)** : Le Responsable des Ressources Humaines est l'acteur principal chargé de la gestion et de l'administration des ressources humaines au sein de l'entreprise. Ses responsabilités incluent la gestion des informations sur les employés, la création et la gestion des postes vacants, ainsi que le suivi des performances et des évaluations des employés.
- **Employé** : Chaque employé peut accéder à son profil personnel, consulter ses informations, soumettre des demandes de congés, des demandes d'autorisation. L'employé interagit principalement avec l'application pour gérer ses propres données et effectuer des demandes auprès du service des ressources humaines.

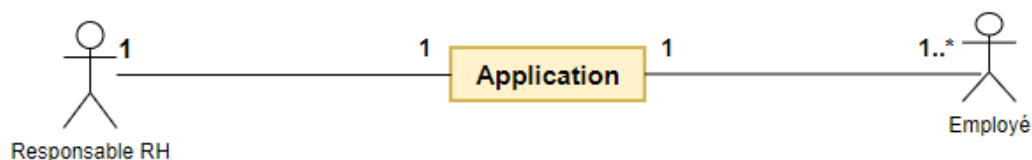


FIGURE 2.1 – la figure ci-dessus montre les acteurs de notre application

2.2.2 Besoins fonctionnels

Il est crucial de spécifier les besoins fonctionnels afin de déterminer les exigences fonctionnelles ainsi que les principaux objectifs et fonctionnalités de l'application. Autrement dit, elle précise ce que le système doit accomplir.

- **S'authentifier** : Toute personne inscrite devra s'authentifier en saisissant son nom d'utilisateur ainsi que son mot de passe afin d'accéder aux fonctionnalités du statut sous laquelle elle est inscrite.
- **Gérer les employés** : Le responsable RH peut gérer les employés à savoir ajouter un employé, supprimer un employé, modifier un employé et consulter des employés.
- **Gérer les entreprises** : Le responsable RH peut gérer les entreprises à savoir : ajouter une entreprise, supprimer une entreprise, modifier une entreprise et consulter la liste des entreprises.
- **Gérer les années** : Le responsable RH peut gérer les années à savoir : ajouter une année , supprimer une année, modifier une année et consulter la liste des années.
- **Gérer les jours fériés** : Le responsable RH peut gérer les jours fériés à savoir : ajouter un jour férié , supprimer un jour férié, modifier un jour férié et consulter la liste des jours fériés.
- **Gérer les congés** : Le responsable RH peut gérer les congés : à savoir : ajouter, supprimer, modifier et consulter la liste des congés.
- **Gérer les absences** : Le responsable RH peut gérer les absences à savoir : ajouter, supprimer, modifier et consulter la liste des absences.
- **Gérer les autorisations** : Le responsable RH peut gérer les autorisations à savoir : ajouter, supprimer, modifier et consulter la liste des autorisations.
- **Gérer les primes** : Le responsable RH peut gérer les primes à savoir : ajouter, supprimer, modifier et consulter la liste des primes.
- **Gérer les types prime** : Le responsable RH peut gérer les types prime à savoir : ajouter, supprimer, modifier et consulter la liste des types prime.
- **Gérer les catégories** : Il est possible pour le responsable RH de gérer les catégories , tels que l'ajout, la suppression, la modification et la consultation de la liste des catégories.
- **Gérer les groupes** : Il est possible pour le responsable RH de gérer les groupes , tels que l'ajout, la suppression, la modification et la consultation de la liste des groupes.
- **Gérer les grades** : Il est possible pour le responsable RH de gérer les grades , tels que l'ajout, la suppression, la modification et la consultation de la liste des grades.
- **Gérer les configurations financières** : Il est possible pour le responsable RH de gérer les configurations financières en ajoutant, supprimant, modifiant et consultant la liste des configurations financières .
- **Gérer les fiches de paie** : Il est possible pour le responsable RH de gérer les fiches de paie en ajoutant, supprimant, modifiant et consultant la liste des fiches de paies.

- **Générer une prediction d'augmentation de salaire** : Il est possible pour le responsable RH de générer et consulter des predictions d'augmentation de salaire pour chaque employé à l'aide à des kpis.
- **Visualiser des données liés aux employés** : Le responsable RH a une vue d'ensemble de quelques informations des employés .
- **Demander un congé** : L'employé a la possibilité de soumettre une demande de congé via l'application. Pour cela, il accède à un formulaire spécifique où il doit préciser les dates de début et de fin du congé.
- **Demander une autorisation** : L'employé a la possibilité de soumettre une demande d'autorisation de 2 heures . Il remplit un formulaire en ligne en précisant la date, l'heure de début et de fin d'autorisation. Cette demande est ensuite envoyée pour approbation au responsable RH.
- **Consulter et/ou télécharger sa fiche de paie** : L'employé peut consulter en ligne ses fiches de paie mensuelles via l'application. Il a également l'option de télécharger ces fiches de paie au format PDF pour les conserver ou les imprimer, facilitant ainsi la gestion de ses documents financiers.
- **Visualiser ses propres données concernant les autorisations, les absences et les congés** : L'employé a une vue d'ensemble de toutes les demandes d'autorisation, d'absence et de congé qu'il a soumises. Il peut vérifier le statut de chacune (approuvée, en attente, ou refusée) et voir quelques graphiques . Cela permet une gestion personnelle efficace et transparente de son temps de travail et de ses absences.
- **Consulter le calendrier des jours fériés** : L'utilisateur de microservice peut accéder au calendrier des jours fériés de l'année en cours. Cela lui permet de planifier ses congés et ses absences en tenant compte des jours non travaillés reconnus par l'entreprise.
- **Consulter et modifier son profil** : L'utilisateur de microservice peut accéder à son profil personnel via l'application. Il peut y visualiser ses coordonnées. Si des informations doivent être mises à jour, comme une adresse ou un numéro de téléphone, l'employé peut les modifier directement dans son profil.
- **Effectuer une récupération de mot de passe** : Si l'utilisateur de microservice oublie son mot de passe, il peut utiliser la fonctionnalité de récupération de mot de passe. En saisissant son adresse e-mail enregistrée, il recevra des instructions pour réinitialiser son mot de passe, garantissant ainsi un accès sécurisé et continu à l'application.
- **Envoyer des messages** : Les utilisateurs de l'application peuvent s'envoyer des messages en surveillant un carrousel affichant les utilisateurs en ligne.

2.2.3 Besoins non fonctionnels

Il est primordial de définir les besoins non fonctionnels afin de définir les caractéristiques et la qualité nécessaires du système. Leur impact n'est pas direct sur les fonctionnalités de l'application, mais peut impacter sa performance et son efficacité globale. Il est important de ne pas les négliger lors de la conception et du développement de l'application. Les principaux besoins non fonctionnels de cette application sont les suivants :

- **Ergonomie** : Pour attirer les utilisateurs potentiels, il est important que le thème, les couleurs et les polices utilisées soient claires, simples et faciles à lire.
- **Convivialité** : Même pour les utilisateurs non experts, l'application est conviviale et facile à utiliser. Sa conception est simple et intuitive pour assurer une manipulation aisée.
- **Rapidité** : Afin d'optimiser le logiciel, il est primordial de garantir des temps de réponse et de traitement des requêtes rapides au sein de l'application.

2.2.4 Diagramme de cas d'utilisation

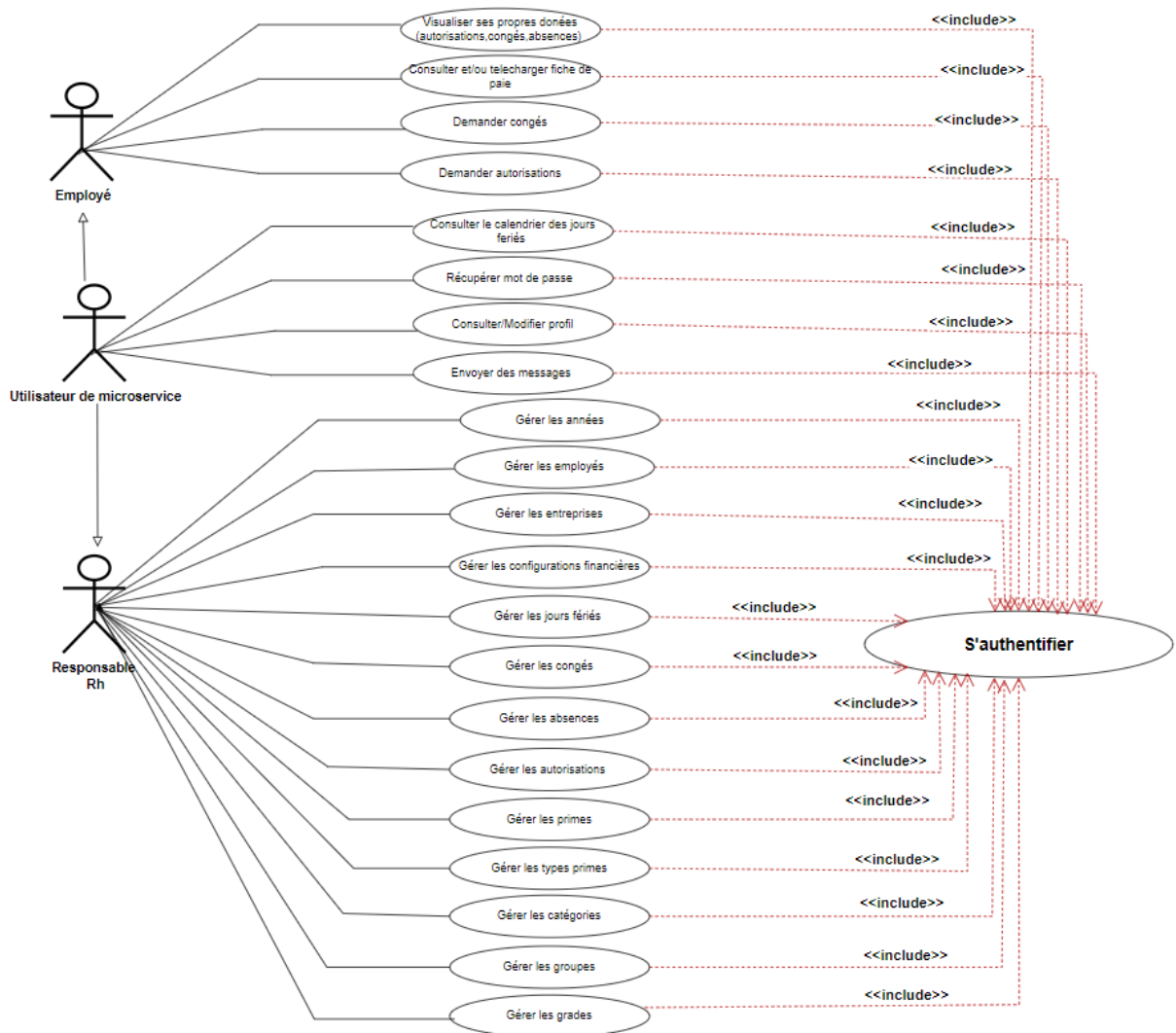


FIGURE 2.2 – Diagramme de cas d'utilisation global

2.3 Pilotage du projet avec Scrum

Cette section présente la gestion de projet à l'aide de Scrum, y compris les équipes de projet et leurs rôles, le backlog produit et la division des projets en sprints.

2.3.1 Equipe Et Roles :

L'équipe en charge du pilotage du projet est constituée de :

- Product Owner : **Helmi Mattoussi**
- Scrum Master : **Helmi Mattoussi**
- Team : **El Ouni Rihab - Jebali Amenallah**

2.3.2 Contraintes et exigences

Le développement et la mise en œuvre réussis de solutions technologiques dépendent de l'identification des véritables besoins des parties prenantes. Certaines fonctionnalités et exigences sont plus importantes que d'autres, et il est important de les aborder en premier pour garantir la continuité et la clarté de votre application. Pour ce faire, il est essentiel d'établir des priorités pour coordonner les tâches à réaliser. Dans ce processus, les besoins sont divisés en trois catégories en fonction de leur importance tout au long du cycle de vie d'une solution technologique : élevée, moyenne et faible.

2.3.3 Risques

La complexité de l'application et les exigences à respecter sont les principaux obstacles qui peuvent freiner notre progression. Il est tout aussi crucial de tenir compte de la représentation des fonctionnalités similaires lors de deux sprints successifs, afin d'éviter toute redondance inutile. Pour limiter les risques, il est préférable de rassembler ces fonctionnalités similaires ensemble, plutôt que de les traiter séparément.

2.3.4 Backlog du produit

Le backlog produit expose la liste des exigences et des fonctionnalités qui doivent être exécutées pour un produit particulier, ainsi que leur priorité et leur niveau de risque. Il sert de document de référence aux équipes de développement, aux parties prenantes et aux clients pour garantir que leur travail est correctement aligné sur les objectifs du projet et les besoins des utilisateurs finaux.

ID	USER STORIES	PRIORITE	Risque
US1	En tant qu'utilisateur de microservice je peux m'authentifier.	Elevée	Faible
US2	En tant que responsable Rh je peux gérer les employés à savoir : ajouter/modifier/supprimer et/ou consulter.	Elevée	Faible
US3	En tant que responsable Rh je peux gérer les entreprises à savoir : ajouter/modifier/supprimer et/ou consulter.	Elevée	Faible
US4	En tant que responsable Rh je peux gérer les années à savoir : ajouter/modifier/supprimer et/ou consulter.	Elevée	Moyen
US5	En tant que responsable Rh je peux gérer les jours fériés à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Elevée
US6	En tant que responsable Rh je peux gérer les congés à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Elevée
US7	En tant que responsable Rh je peux gérer les absences à savoir : ajouter/modifier/supprimer et/ou consulter.	Elevée	Moyen
US8	En tant que responsable Rh je peux gérer les autorisations à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Elevée
US9	En tant que responsable Rh je peux gérer les primes à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Faible
US10	En tant que responsable Rh je peux gérer les types primes à savoir : ajouter/modifier/supprimer et/ou consulter.	Elevée	Faible
US11	En tant que responsable Rh je peux gérer les catégories à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Moyenne
US12	En tant que responsable Rh je peux gérer les groupe à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Faible

TABLE 2.1 – Backlog du produit

ID	USER STORIES	PRIORITE	Risque
US13	En tant que responsable RH je peux gérer les grades à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Faible
US14	En tant que responsable RH je peux gérer les configurations financières à savoir : ajouter/modifier/supprimer et/ou consulter.	Moyenne	Faible
US15	En tant que responsable RH je peux gérer les fiches de paie à savoir : ajouter/modifier/supprimer/télécharger et/ou consulter.	Moyenne	Faible
US15	En tant que responsable RH je peux visualiser quelques données des employés(autorisations/absences,etc..)	Moyenne	Faible
US16	En tant qu'employé je peux consulter et/ou télécharger mes fiches de paie .	Moyenne	Faible
US17	En tant qu'employé je peux demander un congé .	Haute	Moyen
US18	En tant qu'employé je peux demander une autorisation.	Moyenne	Faible
US19	En tant qu'utilisateur de microservice je peux consulter les contacts en ligne et les contacter.	Moyenne	Faible
US20	En tant qu'utilisateur de microservice je peux consulter et/ou modifier mon profil.	Haute	Faible
US21	En tant que responsable RH je peux avoir un e-mail lors de l'ajout d'un nouveau employé.	Haute	Moyen
US22	En tant qu'employé je peux visualiser mes propres données concernant les autorisations, les absences et les congés .	Haute	Moyen
US23	En tant qu'utilisateur de microservices je peux récupérer mon mot de passe .	Haute	Faible
US24	En tant qu'employé je peux avoir un email lors de la création de mon compte.	Haute	Moyen

TABLE 2.2 – Backlog du produit

2.3.5 Organisation et division du projet :

Après la création du Backlog du produit, une réunion est organisée pour prioriser les tâches et les répartir en sprints. Une fois le Backlog bien défini, l'équipe de développement examine les priorités fixées avec le Product Owner et discute des estimations en points d'effort pour chaque fonctionnalité présentée dans l'artéfact

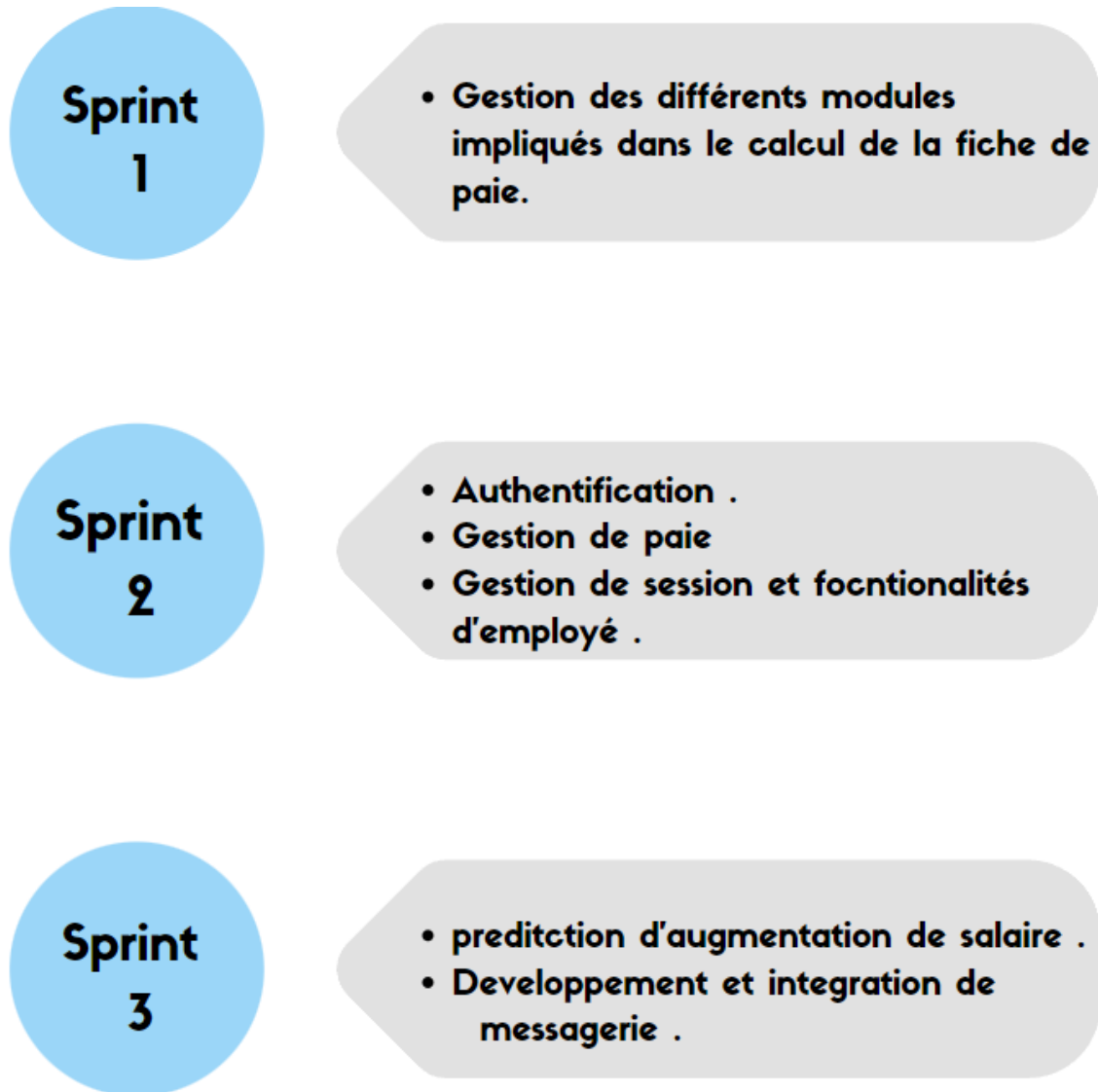


FIGURE 2.3 – Structure de découpage des sprints

- Sprint 1 : -Début : 01/03/2024 ==> -Fin : 30/03/2024
- Sprint 2 : -Début : 01/04/2024 ==> -Fin : 30/04/2024
- Sprint 3 : -Début : 01/05/2024 ==> -Fin : 31/05/2024

2.4 Diagramme de classes du domaine

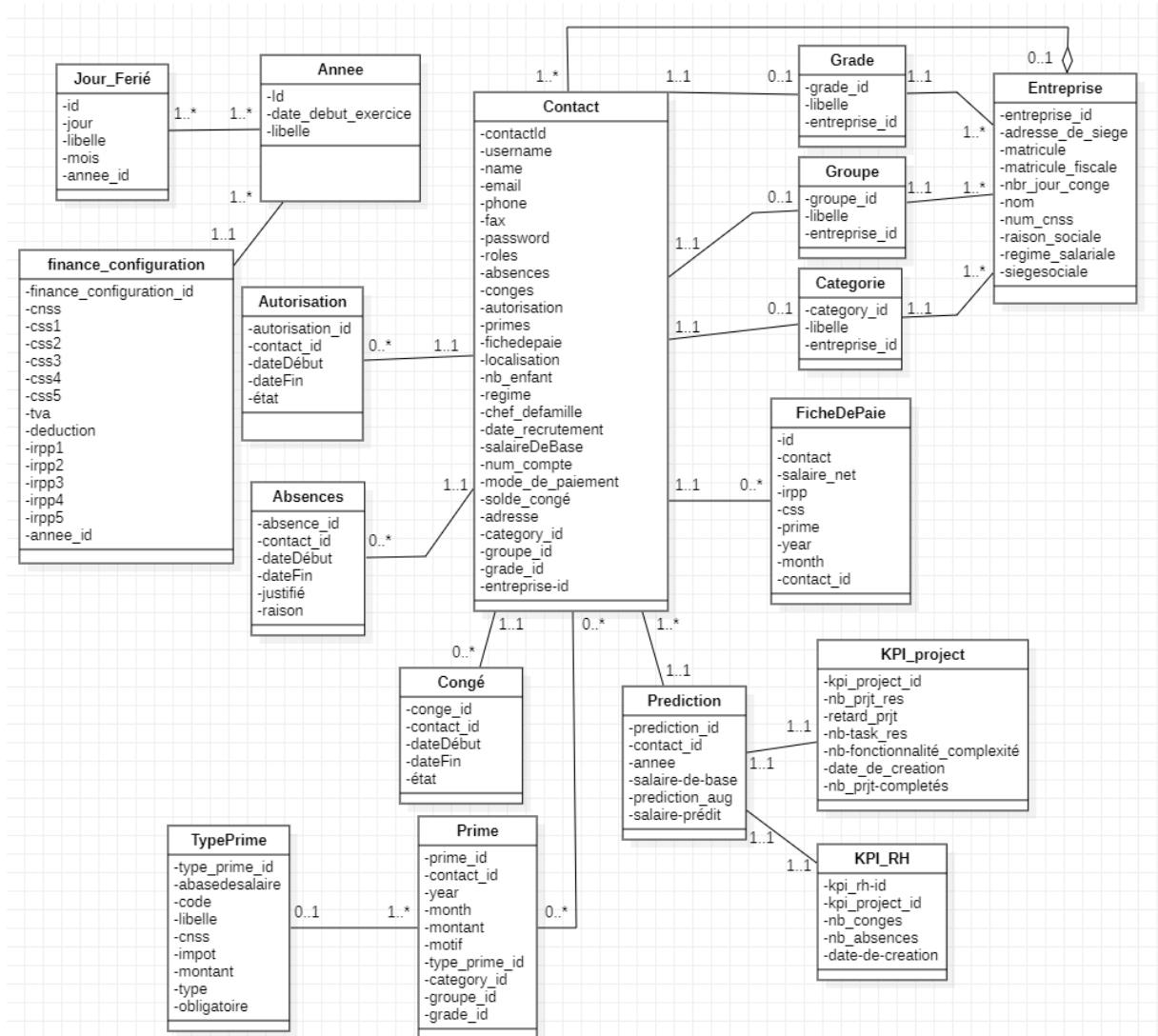


FIGURE 2.4 – Diagramme de classes du domaine

2.5 Conception de logo :

- **Charte graphique :**

Nous avons conceptualisé un logo en utilisant les couleurs définies dans la charte graphique de la startup pour laquelle nous travaillons. Les couleurs principales utilisées sont le bleu (hexa #179AE7) et le blanc (hexa #FFFFFF), avec des accents de noir (hexa #000000). Ce choix de couleurs a été fait pour assurer une cohérence visuelle et renforcer l'identité de la marque.



FIGURE 2.5 – Logo du startup "Createch"

- **Inspiration du logo :**

Le logo est une combinaison d'un symbole d'infini en couleur bleue et de la première lettre du nom de la startup, "C" pour Createch. Le dégradé utilisé dans le logo indique une dimension en 2D, ajoutant une profondeur visuelle.

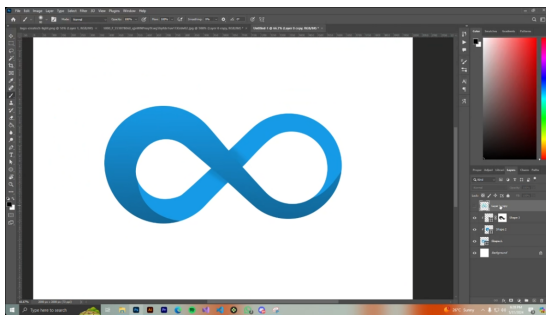


FIGURE 2.6 – Design et ajustement du logo

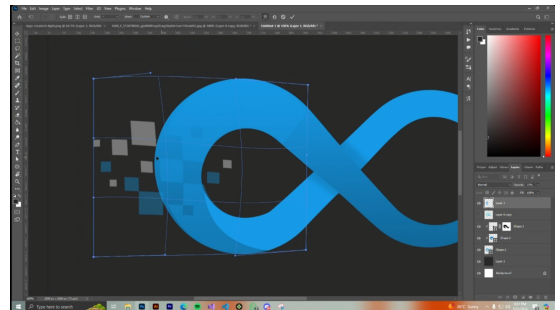


FIGURE 2.7 – Design et ajustement du logo

- **Valeur de logo :**

En haut du logo, il y a deux visages. L'un représente l'employeur et l'autre représente l'employé. Cette dualité symbolise l'impact significatif que l'employé a sur les profits et la réputation de l'employeur. Les deux visages montrent la relation et l'interdépendance entre l'employeur et l'employé, soulignant ainsi l'importance de chaque individu dans la réussite de l'entreprise. Ce logo illustre parfaitement la vision de Createch en mettant en avant l'harmonie et la synergie entre les différents acteurs de l'entreprise.

- Version finale de logo :

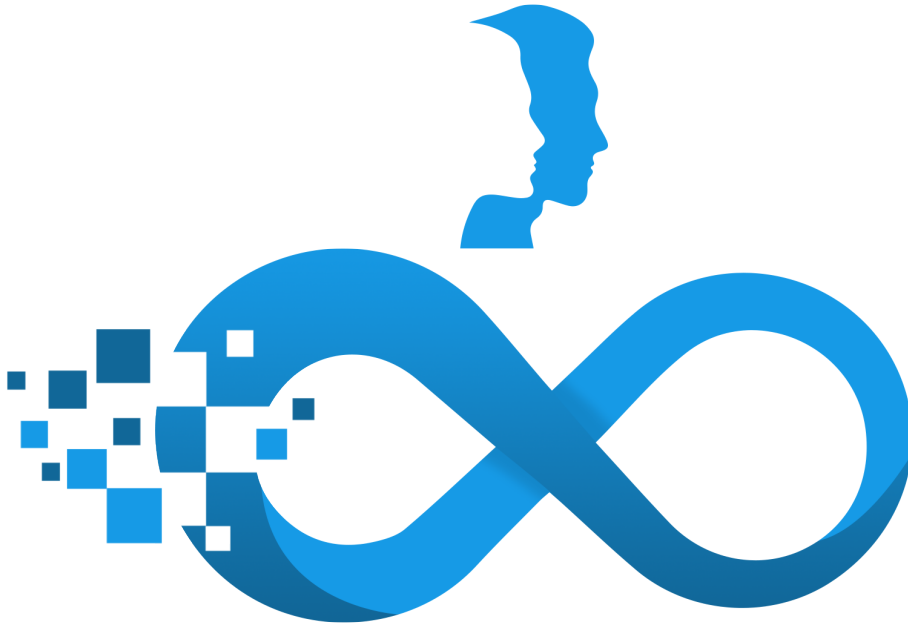


FIGURE 2.8 – Logo de microservice RH

2.6 Conclusion :

Au cours de ce chapitre, nous avons accompli plusieurs tâches indispensables. Tout d'abord, nous avons identifié les acteurs impliqués dans notre application, ainsi que leurs besoins fonctionnels et non fonctionnels. Ensuite, nous avons exposé la première composante de la méthode Scrum : le backlog du produit. Ce backlog recense les exigences fonctionnelles et techniques de notre application. De plus, nous avons défini les différentes tâches au sein de l'équipe du projet et organisé les prochains sprints. Dans le prochain chapitre, nous commencerons notre premier sprint, mettant en pratique toutes les actions que nous avons mises en place jusqu'à présent.

Chapitre 3

Sprint 1 :Fonctionnalités du responsable
RH et de l'employé

3.1 Introduction

Le sprint 1 se focalise sur la mise en place des fonctionnalités de base indispensables au bon fonctionnement du microservice RH. Il englobe principalement les fonctionnalités destinées aux responsables RH. Ce sprint vise à établir une base solide pour les développements futurs, tout en garantissant que le responsable RH peut effectuer les tâches essentielles liées à la gestion des ressources humaines.

3.2 Sprint Backlog

Le sprint backlog est une liste de toutes les tâches sélectionnées à partir du backlog du produit qui doivent être exécutées pendant le sprint en cours. La Table 3.1 illustre le backlog de notre premier sprint

Les ID des tâches du Sprint Backlog correspondent aux identifiants des User Stories du Backlog du produit. Les points d'efforts quant à eux ont été établis grâce au Planning Pocker et estiment l'effort requis pour élaborer chacune des tâches du Sprint Backlog.

ID	Tâches	Points d'effort
US1	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les années.	8
US2	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les entreprises.	
US3	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les configurations financières.	
US4	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les jours fériés.	
US5	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les employés	
US6	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les congés.	
US7	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les absences.	
US8	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les autorisations.	
US9	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les primes.	
US10	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les types primes.	
US11	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les catégories.	
US12	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les grades.	
US13	En tant que responsable Rh je peux ajouter/modifier/supprimer et/ou consulter les groupes.	

TABLE 3.1 – Backlog du premier sprint

3.3 Analyse et conception

Dans le cadre de notre projet informatique, la phase d'analyse revêt une importance capitale pour assurer la réussite du développement. Elle vise à consolider les informations

initiales fournies dans le cahier des charges fonctionnel en les affinant et en les détaillant davantage. Cette démarche implique un examen minutieux du fonctionnement interne du microservice RH à développer, en analysant les besoins spécifiques des utilisateurs finaux. Cette étape est cruciale pour garantir que le produit final répondra de manière optimale aux exigences métier et fonctionnelles.

La phase d'analyse est suivie par celle de la conception, où nous allons élaborer des solutions détaillées pour répondre aux besoins identifiés. Dans cette phase, nous allons également développer les diagrammes correspondant aux fonctionnalités du microservice RH. Ces diagrammes fourniront une vue d'ensemble claire de l'architecture et du fonctionnement du système, ce qui facilitera la compréhension et la communication entre les membres de l'équipe de développement.

3.3.1 Diagramme de cas d'utilisation

La figure 3.1 illustre le diagramme de cas d'utilisation raffiné du sprint 1.

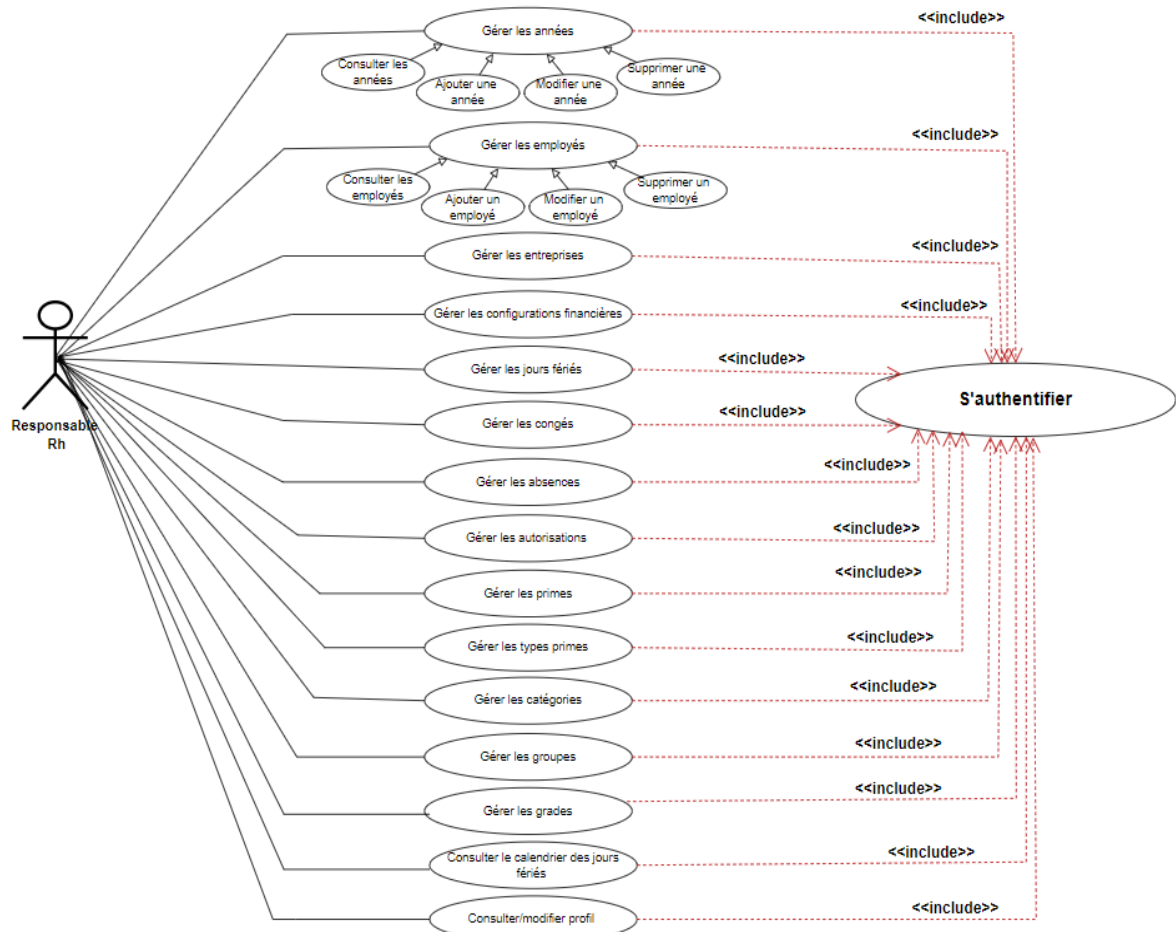


FIGURE 3.1 – Diagramme de cas d'utilisation du Sprint 1

3.3.2 Diagrammes de séquences détaillés

Les diagrammes de séquence détaillés sont des outils de modélisation essentiels qui illustrent de manière précise et séquentielle les interactions entre différents composants d'un système, y compris les microservices RH et diverses parties prenantes comme les employés et les responsables RH.

Diagramme de séquence détaillés du cas d'utilisation «Ajouter un employé»

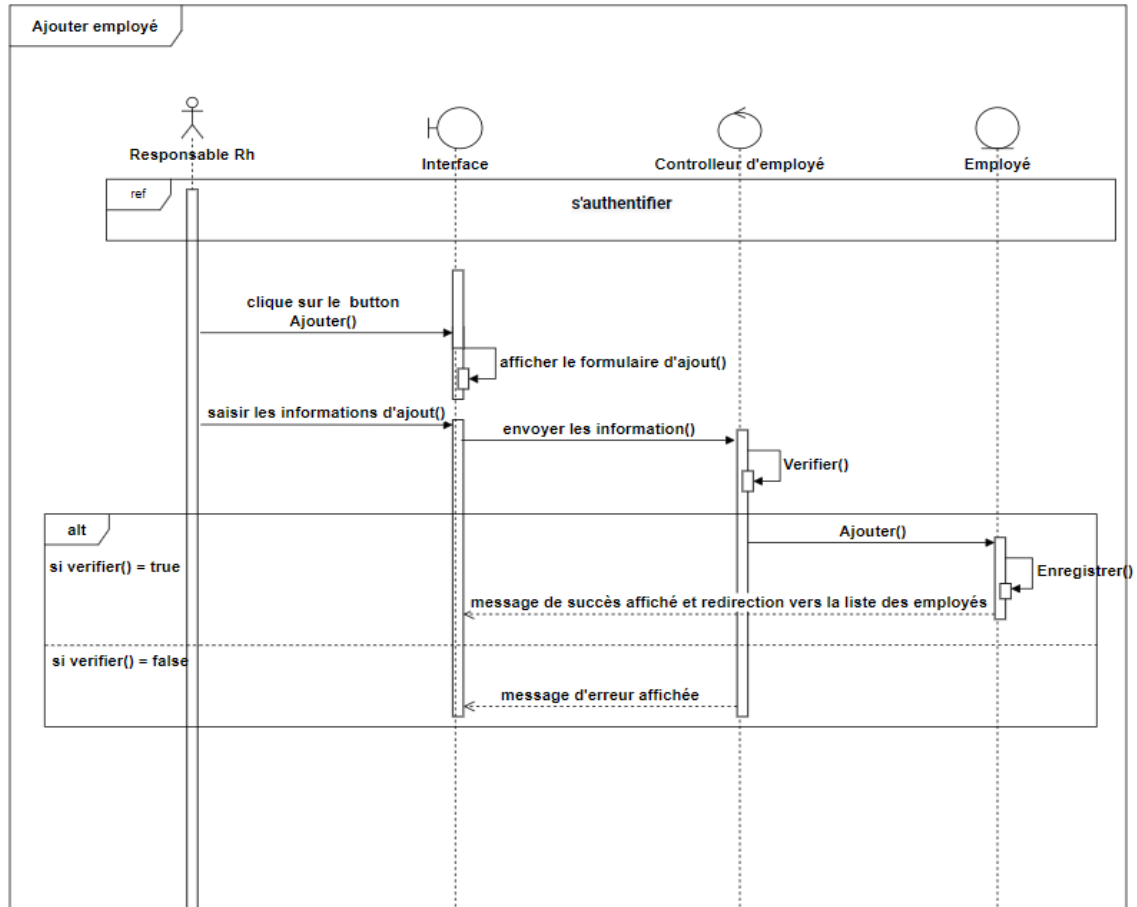


FIGURE 3.2 – Diagramme de séquence détaillés du cas d'utilisation «Ajouter un employé»

- Description textuelle du cas d'utilisation "Ajouter un employé"

Cas d'utilisation	Ajouter un employé
Acteur	Responsable RH
Pré-condition	Responsable RH authentifié.
Post-condition	Le nouvel employé est ajouté à la liste des employés enregistrés. Le responsable RH reçoit une confirmation visuelle de l'ajout réussi de l'employé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le responsable RH accède à la fonctionnalité de gestion des employés depuis le tableau de bord. 2. Le système affiche une liste des employés déjà enregistrés, ainsi qu'une option pour ajouter un nouvel employé. 3. Le responsable RH sélectionne l'option "Ajouter un employé". 4. Le système affiche un formulaire de saisie pour les informations du nouvel employé. 5. Le responsable RH saisit les informations de l'employé, telles que le nom, prénom, adresse, etc. 6. Une fois tous les champs renseignés, le responsable RH valide l'ajout de l'employé. 7. Le système vérifie la validité des données saisies. 8. Si les données sont valides, le système enregistre le nouvel employé dans la base de données. 9. Le système affiche un message de confirmation indiquant que l'employé a été ajouté avec succès. 10. Le responsable RH est redirigé vers la liste des employés où le nouvel employé ajouté est maintenant visible. 11. Le processus d'ajout de l'employé est terminé.
Scénario alternatif	<ol style="list-style-type: none"> 7a. Si les données saisies ne sont pas valides, le système affiche un message d'erreur indiquant les champs incorrects et demande au responsable RH de les corriger. 8a. Le responsable RH corrige les informations incorrectes et soumet à nouveau le formulaire.

TABLE 3.2 – Description textuelle du cas d'utilisation "Ajouter un employé"

Diagramme de séquence détaillé du cas d'utilisation «Modifier un employé»

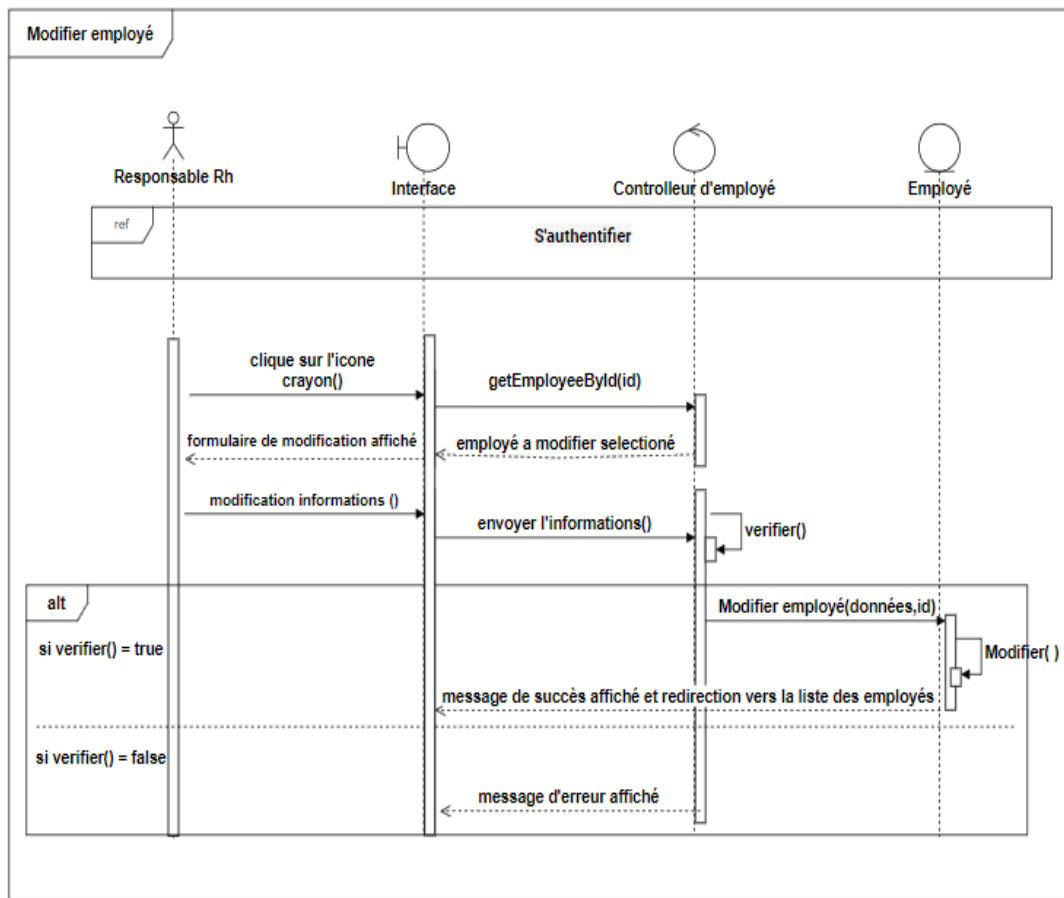


FIGURE 3.3 – Diagramme de séquence détaillé du cas d'utilisation «Modifier un employé»

- Description textuelle du cas d'utilisation «Ajouter une entreprise»

Cas d'utilisation	Modifier un employé
Acteur	Responsable RH
Pré-condition	Responsable RH authentifié.
Post-condition	Les informations de l'employé sont mises à jour dans la base de données. Le responsable RH reçoit une confirmation visuelle de la modification réussie de l'employé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le responsable RH accède à la fonctionnalité de gestion des employés depuis le tableau de bord. 2. Le système affiche une liste des employés déjà enregistrés. 3. Le responsable RH sélectionne l'employé à modifier. 4. Le système affiche un formulaire de saisie pré-rempli avec les informations actuelles de l'employé. 5. Le responsable RH modifie les informations de l'employé, telles que le nom, prénom, adresse, etc. 6. Une fois toutes les modifications effectuées, le responsable RH valide la modification de l'employé. 7. Le système vérifie la validité des données saisies. 8. Si les données sont valides, le système met à jour les informations de l'employé dans la base de données. 9. Le système affiche un message de confirmation indiquant que les informations de l'employé ont été modifiées avec succès. 10. Le responsable RH est redirigé vers la liste des employés où les nouvelles informations de l'employé sont maintenant visibles. 11. Le processus de modification de l'employé est terminé.
Scénario alternatif	<ol style="list-style-type: none"> 7a. Si les données saisies ne sont pas valides, le système affiche un message d'erreur indiquant les champs incorrects et demande au responsable RH de les corriger. 8a. Le responsable RH corrige les informations incorrectes et soumet à nouveau le formulaire.

TABLE 3.3 – Description textuelle du cas d'utilisation “Supprimer un employé”

Diagramme de séquence détaillé du cas d'utilisation «Supprimer un employé»

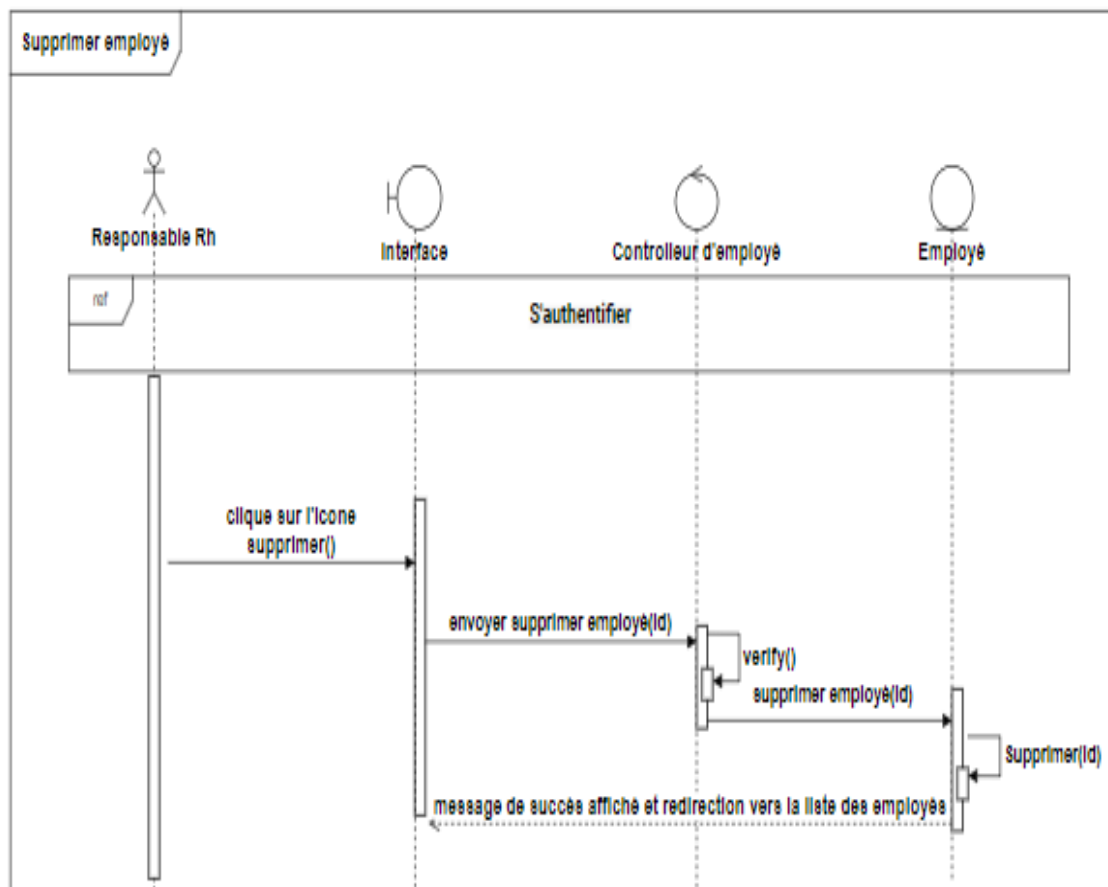


FIGURE 3.4 – Diagramme de séquence détaillé du cas d'utilisation «Modifier un employé»

- Description textuelle du cas d'utilisation «Ajouter une entreprise»

Cas d'utilisation	Ajouter une entreprise
Acteur	Responsable RH
Pré-condition	Responsable RH authentifié.
Post-condition	La nouvelle entreprise est ajoutée à la liste des entreprises enregistrées. Le responsable RH reçoit une confirmation visuelle de l'ajout réussi de l'entreprise.
Scénario nominal	<ol style="list-style-type: none"> 1. Le responsable RH accède à la fonctionnalité de gestion des entreprises depuis le tableau de bord. 2. Le système affiche une liste des entreprises déjà enregistrées, ainsi qu'une option pour ajouter une nouvelle entreprise. 3. Le responsable RH sélectionne l'option "Ajouter une entreprise". 4. Le système affiche un formulaire de saisie pour les informations de la nouvelle entreprise. 5. Le responsable RH saisit les informations de l'entreprise telles que le nom, l'adresse, le secteur d'activité, etc. 6. Une fois tous les champs renseignés, le responsable RH valide l'ajout de l'entreprise. 7. Le système vérifie la validité des données saisies. 8. Si les données sont valides, le système enregistre la nouvelle entreprise dans la base de données. 9. Le système affiche un message de confirmation indiquant que l'entreprise a été ajoutée avec succès. 10. Le responsable RH est redirigé vers la liste des entreprises où la nouvelle entreprise ajoutée est maintenant visible. 11. Le processus d'ajout de l'entreprise est terminé.
Scénario alternatif	<ol style="list-style-type: none"> 7a. Si les données saisies ne sont pas valides, le système affiche un message d'erreur indiquant les champs incorrects et demande au responsable RH de les corriger. 8a. Le responsable RH corrige les informations incorrectes et soumet à nouveau le formulaire.

TABLE 3.4 – Description textuelle du cas d'utilisation "Ajouter une entreprise"

3.3.3 Diagrammes de classes participantes

Un diagramme de classes participantes d'analyse permet de représenter les interactions entre les acteurs et les classes du système, en repérant les responsabilités de chaque acteur et les interactions entre les classes. Les figures 3.5 et 3.6. illustrent les cas d'utilisation «Gérer employé » et «Gérer entreprise »

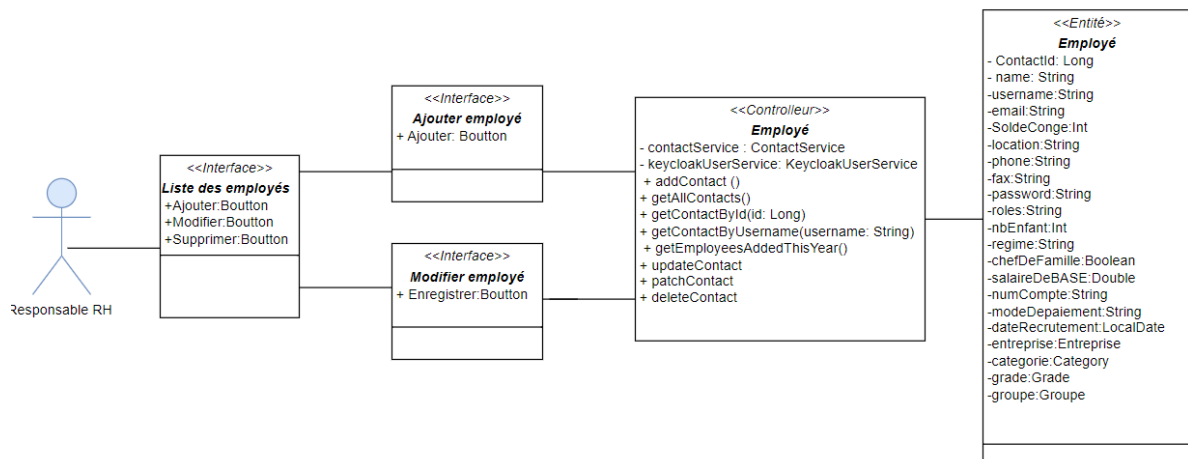


FIGURE 3.5 – Diagramme de classes participantes de conception du cas d'utilisation «Gérer employé»

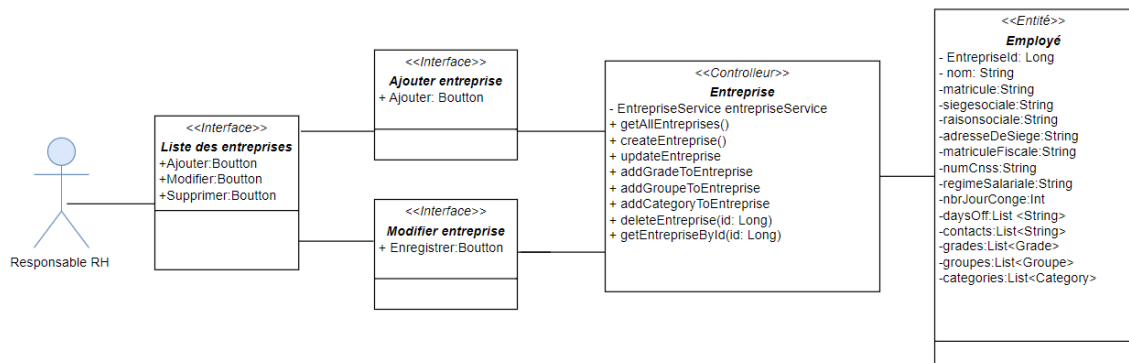


FIGURE 3.6 – Diagramme de classes participantes de conception du cas d'utilisation «Gérer entreprise»

3.4 Réalisation

Dans cette section, nous allons présenter un scénario qui a été élaboré pour les fonctionnalités du premier sprint.

3.4.1 Gérer les années

FIGURE 3.7 – L’interface d’ajout d’une entreprise

Chapitre 4

Sprint 2 : Authentication ,
fonctionnalités de l'employé et la
gestion de paie

4.1 Introduction

Après avoir examiné le premier incrément dans le chapitre précédent, nous allons maintenant aborder le deuxième incrément de notre projet dans ce chapitre.

4.2 Sprint Backlog

Le sprint backlog représente l'ensemble des tâches sélectionnées depuis le backlog du produit qui seront exécutées durant ce sprint. La table 4.1 représente le backlog de notre deuxième sprint.

ID	Tâches	Points d'effort
US1	En tant qu'utilisateur de microservice (Responsable Rh/employé) je peux m'authentifier	
US2	En tant qu'employé je peux consulter et/ou télécharger mes fiches de paie .	
US3	En tant qu'employé je peux demander un congé .	
US4	En tant qu'employé je peux demander une autorisation .	
US5	En tant qu'employé je peux consulter les contacts en ligne et les contacter.	
US6	En tant qu'employé je peux consulter et/ou modifier mon profil.	
US7	En tant qu'employé je peux visualiser mes propres données concernant les autorisations, les absences et les congés .	
US8	En tant qu'employé je peux récupérer mon mot de passe .	
US9	En tant qu'employé je peux avoir un email lors de la création de mon compte.	
US9	En tant qu'employé je peux consulter le calendrier des jours fériés.	
US10	En tant que responsable Rh je peux gérer les fiches de paie .	

TABLE 4.1 – Backlog du deuxième sprint

4.3 Analyse et conception

Cette section met l'accent sur l'analyse et la conception du deuxième sprint.

4.3.1 Diagramme de cas d'utilisation

La figure 4.1 illustre le diagramme de cas d'utilisation de ce sprint.

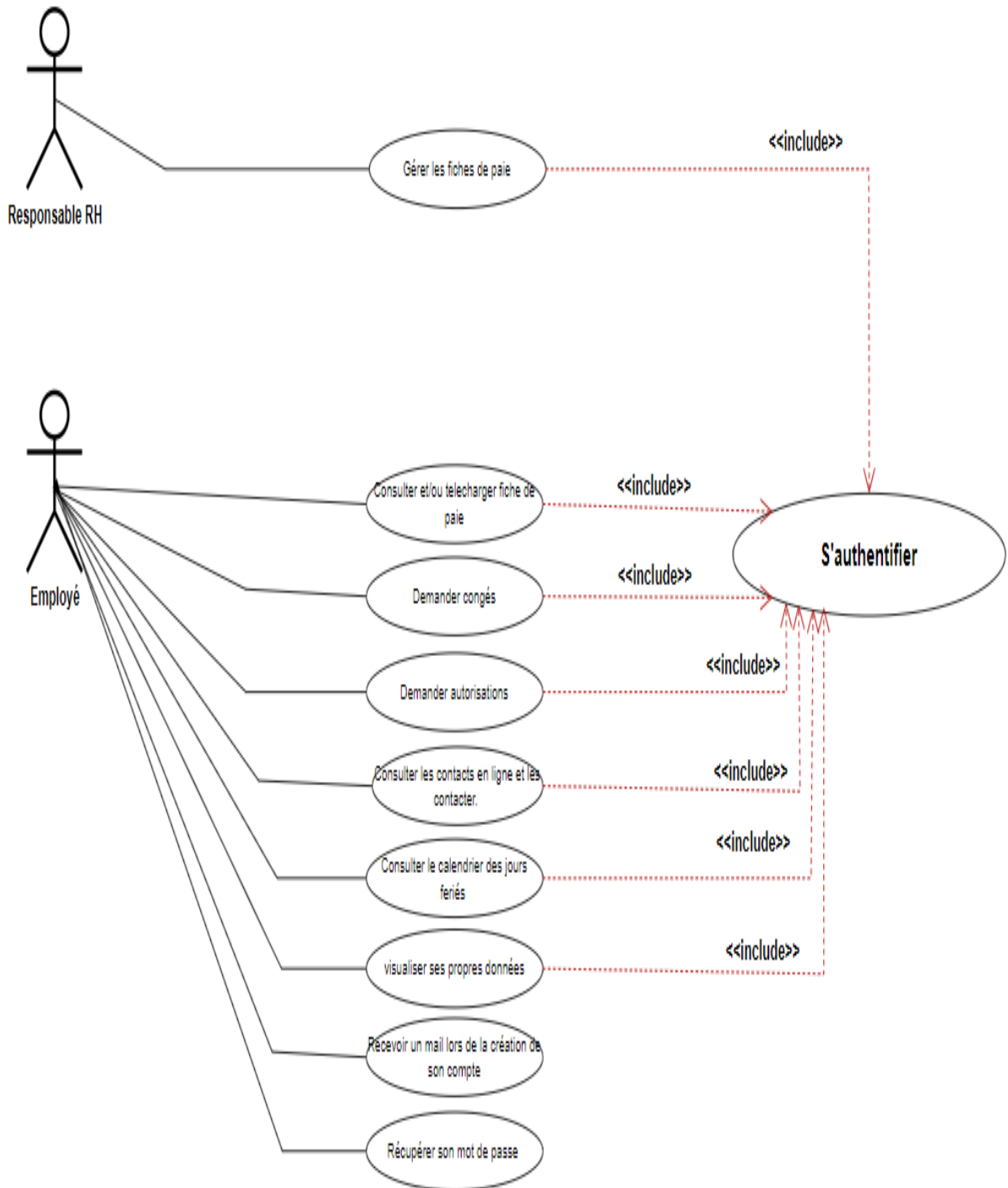


FIGURE 4.1 – Diagramme de cas d'utilisation du Sprint 2

4.3.2 Diagrammes de cas d'utilisation raffiné "S'authentifier"

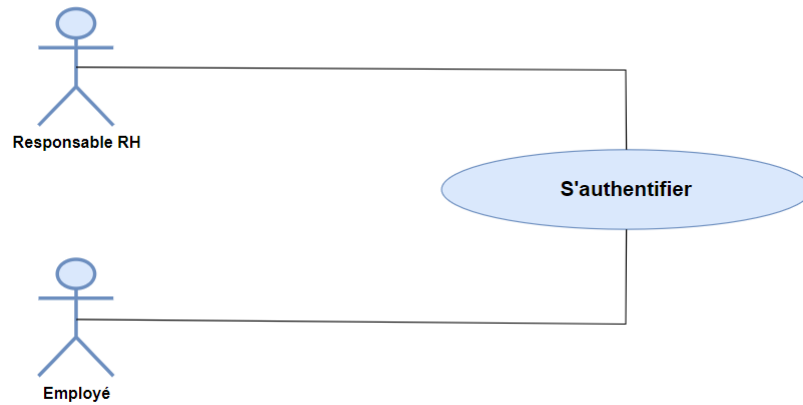


FIGURE 4.2 – Diagramme de cas d'utilisation "S'authentifier"

4.3.3 Diagrammes de séquence

Diagramme de séquence détaillé du cas d'utilisation «S'authentifier»

La figure 4.2 présente le diagramme de séquence détaillé du cas d'utilisation «S'authentifier».

FIGURE 4.3 – Diagramme de séquence détaillé du cas d'utilisation «S'authentifier»

Cas d'utilisation	S'authentifier
Acteur	Responsable RH/Employé
Pré-condition	L'utilisateur de l'application (Responsable RH/Employé) n'a pas de session active en cours.
Post-condition	L'utilisateur de l'application (Responsable RH/Employé) est connecté à l'application
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de connexion. 2. Le système redirige l'utilisateur vers l'écran de connexion Keycloak, où l'utilisateur est invité à entrer son adresse email et son mot de passe. 3. L'utilisateur entre son adresse email et son mot de passe dans les champs appropriés et envoie ses données. 4. Le système vérifie l'adresse email et le mot de passe de l'utilisateur. 5. Si la vérification est réussie, le système envoie les informations de l'utilisateur et un jeton. 6. Le système redirige l'utilisateur vers la page d'accueil de l'application, où il peut accéder aux fonctionnalités disponibles pour les utilisateurs authentifiés.
Scénario alternatif	5a. Si la combinaison d'adresse email et de mot de passe fournie n'est pas correcte, le système affichera un message d'erreur et interrompra le processus de connexion.

TABLE 4.2 – Description textuelle du cas d'utilisation “S'authentifier”

Webographie

[1]<https://agiliste.fr/>

Visité le (28/03/2024). [2]https://en.wikipedia.org/wiki/Spring_Boot

Visité le (30/03/2024).

Sources d'image

[I1] <https://blog.myagilepartner.fr/index.php/2017/11/29/scrum/>

Visité le (28/03/2024). [I2] <https://fr.slideshare.net/slideshow/introduction-scrum-et-aux-51670125>

Visité le (28/03/2024). [I3] <https://www.texnoman.uz/blogs/umumiy-dasturlash>

Visité le (28/03/2024). [I4] <https://github.com/devicons/devicon/issues/435>Spring logo

Visité le (30/03/2024). [I5] <https://digitalogy.co/blog/top-web-development-frameworks/An> logo

Visit  le (30/03/2024). [I6] <https://webbrains.com/MySQL> logo
Visit  le (30/03/2024).