

**République Tunisienne**

**Ministère de l’Enseignement Supérieur et**

**de la Recherche Scientifique**

**Rapport de Projet de Fin d’Études**

Pour l’obtention d’une Master Professionnel Parcours :

Développement Logiciel Et nouvelles technologies

*Intitulé :*

**Conception et développement d’un système d’une Application de Gestion des services Administratives**

*Réalisé par :*

Wissem Mhiri

*Au sein de*

ArabSoft



*Encadré par*

*:*

**Mr. Hamouda Bahri (ArabSoft)**

**Mme. Imen Lassoued (ISI)**

**Année Universitaire 2024-2025**

[Introduction générale](#_bookmark1) 1

1. Chapitre [Etude Préalable](#_bookmark2) 2
2. [Introduction](#_bookmark3)
3. [Contexte](#_bookmark4)
   * 1. [Présentation de l’organisme d’accueil](#_bookmark5) 2
     2. [Problématique](#_bookmark7) 3
   1. [Étude de l’existant](#_bookmark8) 3
      1. [Description de l’existant](#_bookmark9) 3
      2. [Critique de l’existant](#_bookmark10) 4
   2. [Expression des besoins](#_bookmark12) 5
      1. [Identification des acteurs](#_bookmark13) 5
      2. [Identification des besoins](#_bookmark14) 5
   3. [Méthodologie de travail](#_bookmark15) 7
      1. [Choix de méthodologie](#_bookmark16) 7
      2. [Principes de la méthodologie SCRUM](#_bookmark17) 7
      3. [Équipe et rôles](#_bookmark18) 8
   4. [Méthodologie de conception](#_bookmark20) 8
   5. [Planification du projet](#_bookmark21) 9
   6. [Conclusion](#_bookmark23) 9
4. Chapitre [Mise en place du projet](#_bookmark24) 10
   1. [Introduction](#_bookmark25) 10
   2. [Environnement et choix techniques](#_bookmark26) 10
      1. [Environnement de développement](#_bookmark27) 10
   3. [Architecture du système](#_bookmark29) 12
      1. [Architecture logique](#_bookmark30) 12
      2. [Architecture physique](#_bookmark32) 14
   4. [Backlog de produit](#_bookmark34) 15
   5. [Création de product backlog](#_bookmark35) 15
   6. [Planification des sprints](#_bookmark36) 19
   7. [Conclusion](#_bookmark37) 19

[Sprint 1](#_bookmark39) 20

* 1. [Introduction](#_bookmark40) 20
  2. [Objectifs du Sprint 1](#_bookmark41) 20
  3. [Backlog du sprint 1](#_bookmark42) 20
  4. [Diagramme d’activité](#_bookmark45) 21
     1. [Diagramme d’activité “ Authentification”](#_bookmark46) 21
  5. [Conception des interfaces : le maquettage](#_bookmark48) 22
     1. [Scénarios et maquette d’user story : Authentification](#_bookmark49) 22
     2. [Scénarios et maquette d’user story : Ajout d’un utilisateur](#_bookmark51) 23
  6. [Conception du Sprint](#_bookmark52) 23
     1. [Diagramme de package](#_bookmark53) 24
     2. [Diagramme de classe de conception](#_bookmark55) 24
     3. [Diagramme de séquence objet de u.s “Authentification”](#_bookmark58) 26
     4. [Diagramme séquence objet "Vérification d’autorités"](#_bookmark60) 27
     5. [Diagramme séquence objet "Ajout utilisateur"](#_bookmark62) 27
  7. [Réalisation du Sprint 1](#_bookmark64) 29
     1. [Interfaces](#_bookmark65) 29
     2. [Tests](#_bookmark68) 30
  8. [Burndown Chart](#_bookmark72) 31
  9. [Conclusion](#_bookmark74) 31

[Sprint 2](#_bookmark75) 32

* 1. [Introduction](#_bookmark76) 32
  2. [Objectifs du Sprint 2](#_bookmark77) 32
  3. [Backlog du Sprint 2](#_bookmark78) 32
  4. [Diagramme d’activité](#_bookmark80) 33
     1. [Diagramme d’activité “ Modifier article”](#_bookmark81) 33
  5. [Conception des interfaces : le maquettage](#_bookmark84) 34
     1. [Scénarios et maquette d’user story : Ajouter article](#_bookmark85) 34
  6. [Conception du Sprint](#_bookmark87) 34
     1. [Diagramme de package](#_bookmark88) 35
     2. [Diagramme de classe de conception](#_bookmark90) 35
     3. [Diagramme de séquence objet de u.s “Consulter mode de paiement”](#_bookmark92) 37
     4. [Diagramme séquence objet "Supprimer échantillon"](#_bookmark94) 37
  7. [Réalisation du Sprint 2](#_bookmark96) 38
     1. [Interfaces](#_bookmark97) 38
     2. [Tests](#_bookmark99) 39
  8. [Burndown Chart](#_bookmark103) 40
  9. [Conclusion](#_bookmark105) 40

|  |  |
| --- | --- |
| [**Sprint 3**](#_bookmark107) | **41** |
| [5.1 Introduction](#_bookmark108) . . . . . . . . . . . . . . . . | . 41 |
| [5.2 Objectifs du Sprint 3](#_bookmark109) . . . . . . . . . . . | . 41 |
| [5.3 Backlog du Sprint 3](#_bookmark110) . . . . . . . . . . . . | . 41 |
| [5.4 Diagramme d’activité](#_bookmark112) . . . . . . . . . . | . 42 |

[5.4.1 Diagramme d’activité “Mouvement de stock”](#_bookmark113) 42

* 1. [Conception des interfaces : le maquettage](#_bookmark116) 43
     1. [Scénarios et maquette d’user story : “Passer commande”](#_bookmark117) 43
     2. [Scénarios et maquette d’user story : “Génerer acquit à caution”](#_bookmark119) 44
  2. [Conception du Sprint](#_bookmark121) 45
     1. [Diagramme de package](#_bookmark122) 45
     2. [Diagramme de classe de conception](#_bookmark123) 45
     3. [Diagramme de séquence objet de u.s “Passer Commande”](#_bookmark126) 47
     4. [Diagramme de séquence objet de u.s “Génerer Acquit à caution”](#_bookmark128) 48
  3. [Réalisation du Sprint 3](#_bookmark130) 49

[Interfaces](#_bookmark131)

**Introduction Générale**

**Chapitre I : Étude de projet**

**1.1 Introduction**

L’objectif de ce chapitre est de fournir un aperçu global du projet en le plaçant dans son contexte approprié. Pour cela, nous commencerons par présenter l’organisation d’accueil en décrivant son domaine d’activité. Ensuite, nous procéderons à une brève description du projet et de l’étude de la situation actuelle. Enfin, nous exposerons le problème à résoudre ainsi que les objectifs spécifiques que nous cherchons à atteindre.

**1.2 Contexte**

Dans le cadre de l’obtention d’un mastère professionnel en science informatique avec spécialisation en génie logiciel et système d’information, nous avons entrepris un projet de fin d’études intitulé : " Portail web de gestion des services administratifs “. Ce projet a été mené en collaboration avec la société ArabSoft.

1.2.1 Présentation de l’organisme d’accueil

ARAB SOFT a été créé en 1985 par Mohamed TRIKI et s’est spécialisé dans l’étude, le développement et la distribution de solutions de gestion pour le secteur public. Ce qui a permis à l’entreprise de se démarquer, c’est la proposition de solutions bilingues en français et en arabe, contrairement aux autres solutions proposées uniquement en français ou en anglais. Pendant plus de 30 ans, ARAB SOFT est devenue le principal éditeur tunisien de solutions de gestion grâce à son esprit d’entrepreneur et d’innovation, ce qui a contribué à établir une relation durable avec ses clients.[2] Les produits standards d’ArabSoft

: — AMIN : Gestion Comptable et Financière

— AJIR : Gestion des Ressources Humaines

— MARKAZIA : Centrale d’information et des risques

— ARZAK : Gestion des achats, stocks et des immobilisations

— XLIA : Gestion Hôtelière Intégrée

— HAYET : Gestionnaire de réseaux et Fournisseurs d’énergie

— JIBAYA : Gestion des dossiers des contribuables, impôts, déclarations Organigramme L’organigramme de la Figure 1 décrit la structure de la société. Nous avons été accueillis durant la période de ce projet au sein du service développement.

Figure 1: Organigramme d’ArabSoft

**Cadre général du travail :**

Vue la richesse et la diversité des ressources dans l’administration des entreprises, l’évolution des méthodes de traitement de l’information est devenue un besoin fondamental et indispensable. Mettre en place une plateforme web dédiée aura pour objectif d’accélérer et d'améliorer la précision dans le traitement des informations tout en garantissant un haut niveau de sécurité et de stabilité, afin de faciliter les tâches administratives et les rendre plus efficaces.

Pour réussir à déployer une solution adéquate pour la gestion des employés, il est crucial de définir en premier lieu le rôle de ce service au sein de l’établissement. L’administration d’ARABSOFT est composée de plusieurs services et départements, auxquels sont rattachés des employés chargés de tâches spécifiques, suivis et gérés par le service du personnel.

La plateforme à développer pour la gestion des services administratifs devra alléger la charge de travail des responsables du service du personnel, tout en facilitant l’échange d’informations entre les différents employés. De plus, cette stratégie visera à automatiser et accélérer l’accomplissement des tâches au sein du service.

**Problématique :**

Dans un contexte où les administrations des entreprises deviennent de plus en plus complexes, la gestion des services administratifs, y compris le suivi des employés, la répartition des tâches et la communication interne, représente un défi majeur. Actuellement, ces tâches sont souvent effectuées de manière manuelle ou à l'aide de systèmes disparates, ce qui peut entraîner des inefficacités, des erreurs humaines, et un manque de coordination entre les différents départements. La multiplication des informations à traiter, l'augmentation du

Volume des données, et les besoins croissants en sécurité et en accessibilité ajoutent une couche de complexité supplémentaire.

Comment alors mettre en place une solution intégrée et sécurisée qui permettra de centraliser, d'automatiser, et de faciliter la gestion des services administratifs tout en améliorant la collaboration entre les employés et les différents services de l'entreprise ?

**1.3 Étude de l’existant**

L’étude de l’existant permet de comprendre en profondeur le domaine d’étude et de faire un bilan critique de celui-ci. Elle permet également d’identifier les lacunes et de proposer des solutions pour les résoudre. En d’autres termes, l’étude de l’existant est une étape importante pour comprendre les enjeux et les défis du domaine en question.

**Description de l'existant**

Dans l’état actuel, la gestion des services administratifs repose sur des processus manuels et décentralisés, ce qui entraîne des inefficacités, un manque de transparence, et des risques accrus pour la sécurité des données. Ces limitations mettent en évidence la nécessité d'une solution plus intégrée, automatisée, et sécurisée, qui puisse centraliser toutes les opérations administratives sur une plateforme unique.

**Critique de l’existant**

Organisation des Services Administratifs : Les services administratifs sont gérés de manière traditionnelle, divisés en plusieurs départements utilisant des outils et méthodes variés. Les processus comme la gestion des employés et le traitement des demandes sont réalisés manuellement ou avec des logiciels non intégrés.

Gestion des Informations Employés : Les données des employés sont stockées localement dans des fichiers Excel ou des bases de données non centralisées. Les mises à jour sont effectuées manuellement, ce qui peut causer des retards et des erreurs.

Traitement des Demandes : Les demandes des employés, telles que les congés ou permissions, sont soumises par email ou papier, et traitées sans suivi automatisé, rendant le processus lent et inefficace.

Communication Interne : La communication se fait principalement par email ou réunions physiques, sans plateforme centralisée, ce qui peut entraîner des pertes d'information et un manque de coordination.

Sécurité et Gestion des Accès : Le contrôle d'accès est basique, avec des mots de passe simples et sans gestion avancée des rôles, exposant les données sensibles à des risques de sécurité.

**Solution proposée**

Pour répondre à cette problématique, la mise en place d'un portail web de gestion des services administratifs est envisagée. Ce portail aura pour rôle de centraliser toutes les opérations administratives dans une plateforme unique et accessible en ligne. Il proposera les fonctionnalités suivantes :

1. Première partie - Administration :
   * Cette section sera accessible par l’administrateur du système. Elle permettra de consulter la page d’accueil, de gérer les comptes des employés et leurs rôles, ainsi que de traiter les différentes demandes soumises par les utilisateurs. L'administrateur disposera d'un accès complet pour assurer une gestion centralisée et sécurisée des données.
2. Deuxième partie - Chefs :
   * Cette section sera destinée aux chefs d’équipe ou de service. Elle leur permettra de consulter la liste des employés sous leur supervision, facilitant ainsi la gestion des équipes et la coordination des tâches. Les chefs pourront également avoir accès à d'autres fonctionnalités comme l'évaluation des performances ou l'approbation des demandes, selon les besoins de l'organisation.
3. Troisième partie - Employés :
   * Cette section sera accessible par les employés, leur offrant la possibilité de soumettre des demandes (congés, matériel, etc.) et de mettre à jour leurs informations personnelles. Cette approche vise à réduire la charge administrative et à encourager une gestion plus autonome de leurs données.

L’accès à ces différentes parties sera contrôlé par des droits d’accès spécifiques, adaptés en fonction du rôle de l’utilisateur, afin d’assurer la sécurité et la confidentialité des informations.

## Méthodologie de travail

### Choix de méthodologie

[[18]](#_bookmark205) Le développement logiciel nécessite une approche basée sur un langage de modélisation. Selon la complexité du projet, différents modèles peuvent être utilisés, tels que Waterfall, Y- model ou méthodes Agiles (Scrum et Extreme Programming). Pour augmenter nos chances de succès, il est important de choisir une approche itérative et incrémentale comme Scrum, qui fait gagner du temps, améliore la qualité du développement et permet de réagir rapidement aux demandes de changement. Notre projet adopte cette approche afin de garantir la suffisance du résultat final et la satisfaction des utilisateurs.

### Principes de la méthodologie SCRUM

Scrum est un cadre de gestion de projet utilisé pour le développement de logiciels et autres domaines. Les équipes de dix membres ou moins divisent leur travail en objectifs pour des itérations appelées sprints, qui durent généralement deux semaines. Les progrès sont évalués lors de daily scrums, des réunions quotidiennes courtes. À la fin du sprint, il y a une revue pour démontrer le travail et une rétrospective pour permettre à l’équipe de [s’améliorer.[3]](#_bookmark208)

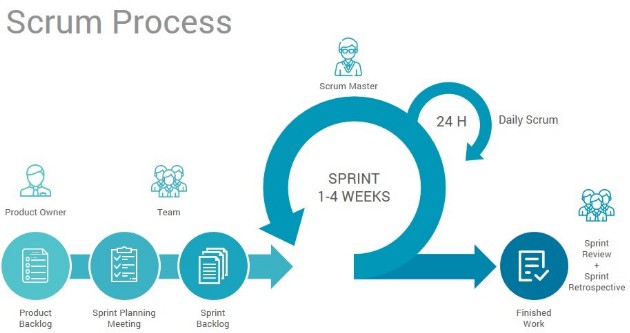


Figure 1.3 – le cycle de vie d’un projet en utilisant la méthode Scrum

### Équipe et rôles

Dans la méthode Scrum, on peut identifier trois rôles principaux impliqués dans la gestion de projet :

* + - * **Le Scrum Master** est un membre de l’équipe dont l’objectif est d’optimiser la capacité de production de l’équipe. Pour atteindre cet objectif, le Scrum Master aide l’équipe à travailler de manière autonome tout en cherchant continuellement à s’améliorer.
      * Le rôle principal de la **Scrum Team** (l’équipe Scrum) est de se gérer en toute autono- mie et de prendre en charge le développement du produit. Contrairement à une équipe hiérarchisée, toutes les décisions sont prises ensemble, sans aucune distinction de rôle ou de grade. L’équipe Scrum regroupe tous les acteurs nécessaires au bon déroulement d’un projet, tels que l’architecte, le concepteur et le développeur, afin d’optimiser la producti- vité et la qualité du produit final.

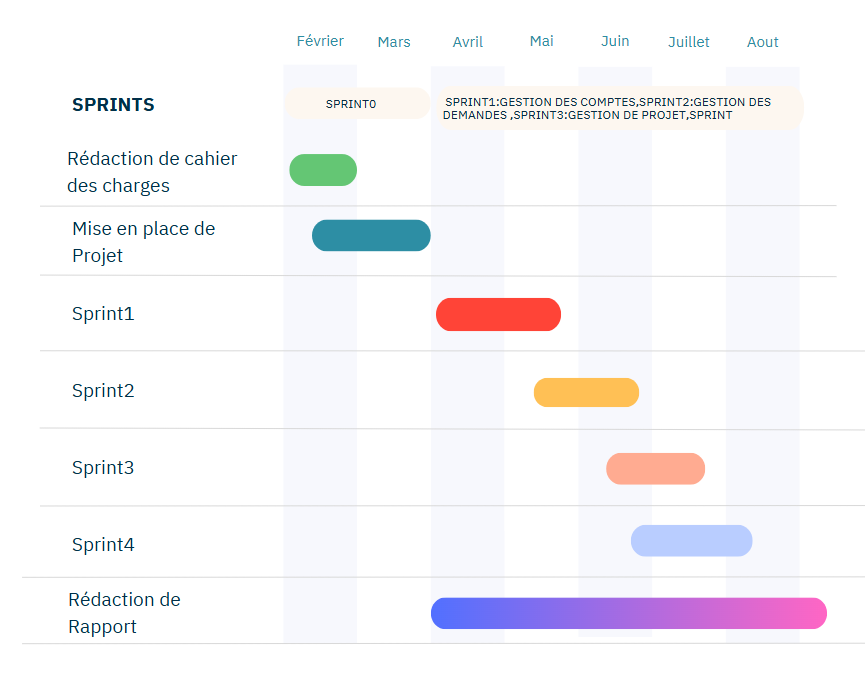
Notre équipe est formée comme suit :

* + - * Mr. Hammouda Bahri : Scrum Master
      * Wissem Mhiri : Développeur

## Méthodologie de conception

Le langage de modélisation unifié (UML) est utilisé pour la conception de systèmes infor- matiques. Il se compose de diagrammes qui permettent la visualisation et la description de la structure et du comportement des objets système, Les diagrammes de classes représentent les classes et les relations entre elles, les diagrammes de séquence décrivent comment les ob- jets interagissent les uns avec les autres dans un scénario donné, les diagrammes d’activités

modélisent les processus métier et les packages d’organigrammes représentent la structure et



**Diagramme de GANTT**

**Conclusion**

Dans ce premier chapitre, nous avons présenté l'organisme d'accueil ainsi que le cadre général du travail, en soulignant les objectifs et les attentes liés à ce projet. Nous avons ensuite identifié la problématique centrale, en mettant en évidence les défis actuels de la gestion des

services administratifs. Une étude de l'existant a été réalisée pour évaluer les méthodes et outils actuellement utilisés, révélant des lacunes significatives en termes de centralisation, d'automatisation, et de sécurité. En réponse à ces constats, une solution a été proposée sous la forme d'un portail web de gestion des services administratifs, conçu pour améliorer l'efficacité, la coordination, et la sécurité au sein de l'organisation.

**Chapitre II : Mise en place du Projet**

* 1. **Introduction**

Dans ce chapitre, nous exposons l’environnement matériel et logiciel que nous avons utilisé. Nous mentionnons également nos choix techniques concernant les langages de programmation et les outils utilisés. Enfin, nous incluons une version du backlog de produit avec le plan des sprints.

* 1. **Environnement et choix techniques**

Dans cette partie on va présenter l’environnement matériel (les deux ordinateurs) et les différents logiciels utilisés lors du développement de notre application tout au long de période de projet.

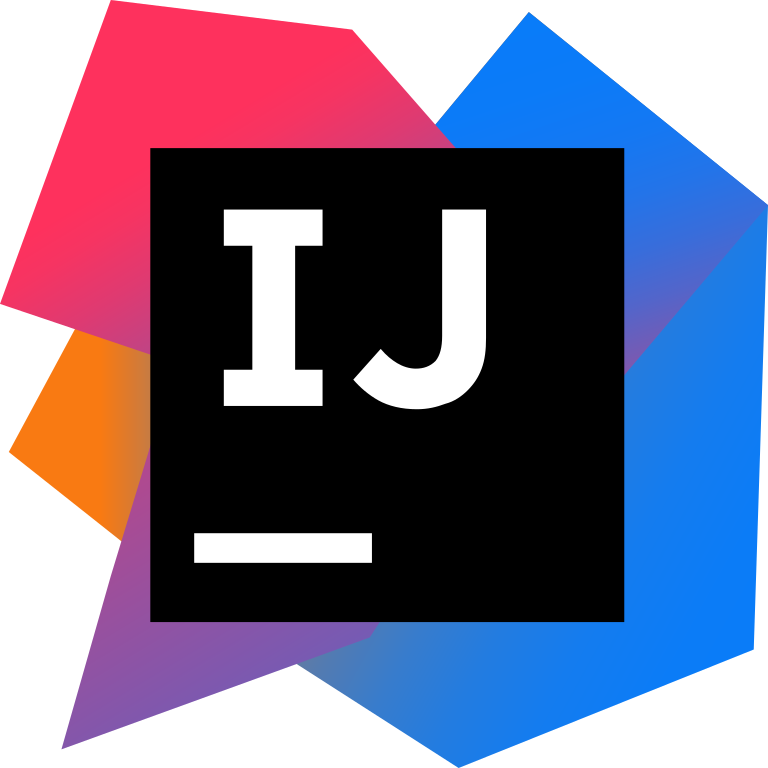
* + 1. **Environnement de développement**

**Environnement matériel**

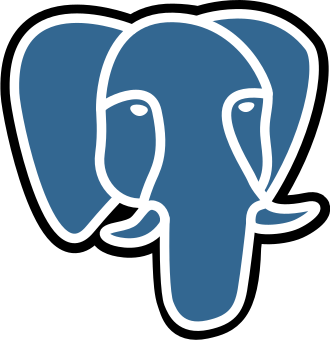
|  |
| --- |
| **Ordinateur** |
| Processeur : Intel Core i5 11dth gen  Mémoire vive : 8 GB Disque dur : 2 TB  Système d’exploitation : Windows 11 Education |

Table 2.1 – Table d’environnement matériel

**Envirenement :**

**Intellij Ultimate Edition :** IntelliJ IDEA est un IDE intelligent et tenant compte du contexte qui permet de travailler sur toutes sortes d’applications en Java et dans d’autres langages de la JVM tels que Kotlin, Scala et Groovy. La version ultimate nous a permet aussi de travailler sur les application et les frameworks en Javascript et Typ[escript..[4]](#_bookmark209)

**Base de Données :**

**PostgreSQL :** est un système de gestion de base de données relationnelle et objet (SGBDRO). C’est un outil libre disponible selon les termes d’une licence de type BSD. Ce système est comparable à d’autres systèmes de gestion de base de données, qu’ils soient libres (comme MariaDB et Firebird), ou propriétaires (comme Oracle, MySQL, Sybase, DB2, Informix et Microsoft SQL Serve[r)..[5]](#_bookmark210)

**Test :**

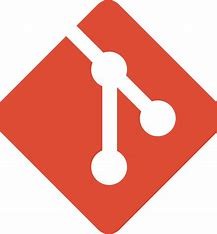
**Postman :** est un client API utilisé pour développer, tester, partager et documen- ter des API. Il est utilisé pour les tests de backend où nous entrons l’URL du point final, il envoie la requête au serveur et reçoit la réponse du serveur. [[6]](#_bookmark211)

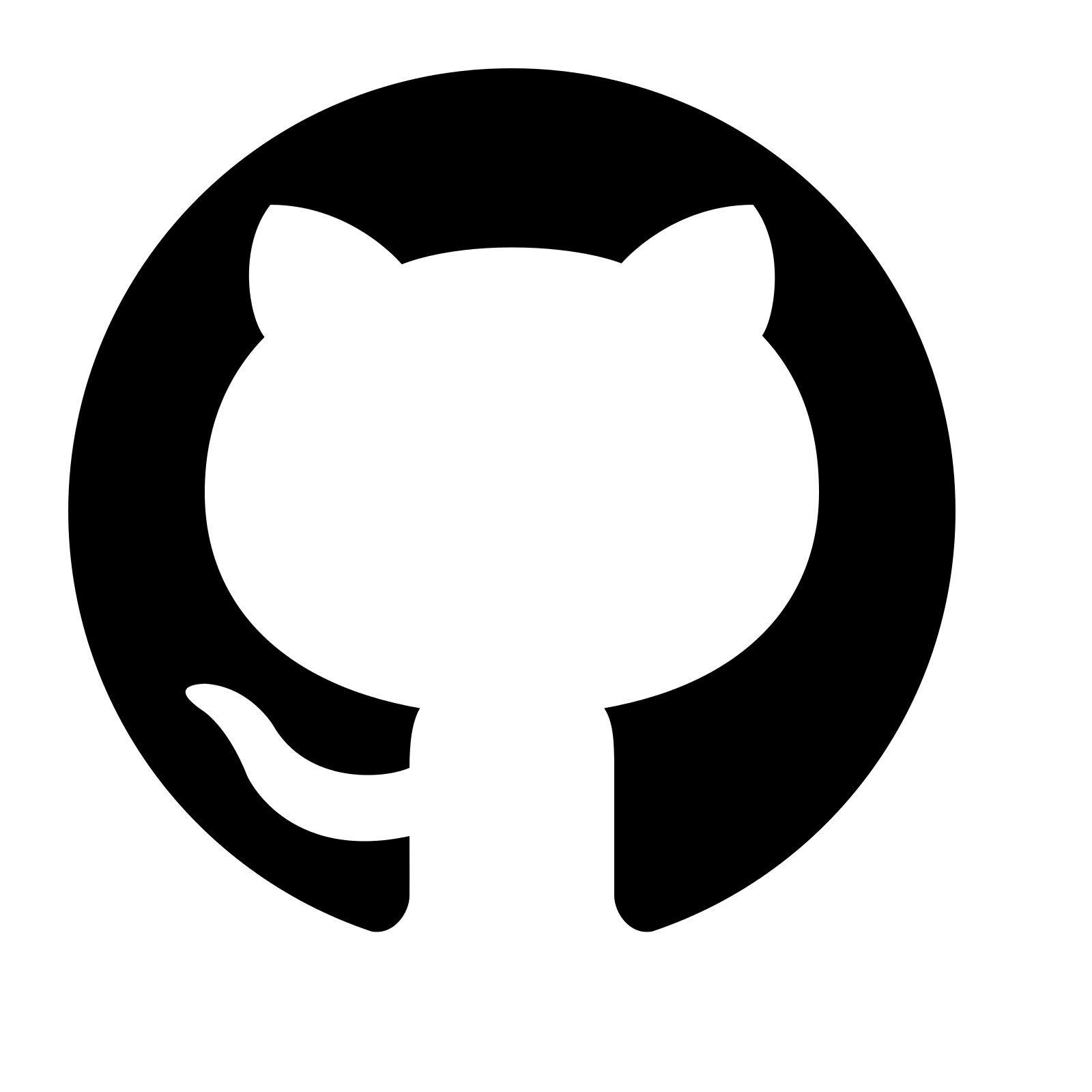
**Tests unitaires :**

**Junit5 :** JUnit est un framework de test unitaire pour le langage de programmation Java, créé par Kent Beck et Erich [Gamma.[7]](#_bookmark212)

**Mockito :** Mockito est un framework de test qui permet de bouchonner les classes autres que celle à tester et de définir des comportements sur leurs méthodes afin de répondre aux cas de [test[8]](#_bookmark213)

**Les outils de collaboration et de planification :**

**Git :** Git est un logiciel de gestion de version qui permet de conserver un historique des modifications effectuées sur un projet afin de pouvoir rapidement identifier les chan- gements effectués et de revenir à une ancienne version en cas de problème. [[9]](#_bookmark214)

**Github :** Github est une plateforme collaborative qui permet aux équipes tech de stocker et de modifier du code pour tous leurs projets (logiciels, programmes informatiques, développement de sites web, applications mobiles, [etc.)[10]](#_bookmark197)

**Jira :** Jira est un produit de suivi des problèmes propriétaire développé par Atlassian qui permet de suivre les bugs , de planifier et gérer les projets [agiles.[11]](#_bookmark198)

**Les outils de prototypage et de conception :**

**Figma :** Figma est une plateforme collaborative pour éditer des graphiques vectoriels et faire du prototypage. Elle permet de concevoir des designs système pour faciliter la création de sites web et d’applications mobiles. [[12]](#_bookmark199)

**Draw.io :** Diagrams.net (anciennement draw.io) est un logiciel de dessin graphique multi- plateforme gratuit et open source développé en HTML5 et JavaScript . Son interface peut être utilisée pour créer des diagrammes tels que des organigrammes , des structures filaires , des diagrammes UML , des organigrammes et des diagrammes de [réseau[13]](#_bookmark200)

**Frameworks :**

**Frontend :**

**Angular :** est un Framework open source écrit en JavaScript qui permet la création d’applications Web et plus particulièrement de ce qu’on appelle des « Single Page Ap- plications » : des applications web accessibles via une page web unique qui permet de fluidifier l’expérience utilisateur et d’éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVVM et permet donc de séparer

les données, le visuel et les actions pour une meilleure gestion des resp[onsabilités.[14]](#_bookmark201)

**Bootstrap :** Bootstrap est une collection d’outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d’applications web. C’est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que

des extensions JavaScript en option. C’est l’un des projets les plus populaires sur la plateforme de gestion de développement [GitHub[15]](#_bookmark202)

**Backend :**

**Springboot :** Le Spring Framework est très largement utilisé dans la communauté Java. Il permet d’accélérer le développement d’applications d’entreprise (notamment le développement d’applications Web et d’API Web). Mais on trouve des applications basées sur le Spring Framework dans bien d’autres [domaines..[16]](#_bookmark203)

* 1. **Architecture du système**
     1. **Architecture logique**

Dans cette section, nous allons examiner l’architecture logique que nous avons mise en place pour notre plateforme. Nous avons opté pour une architecture en couches, également connue sous le nom d’architecture multi-tiers, qui consiste à concevoir le système comme une superposition de couches, chacune responsable d’une tâche spécifique. Cette approche sépare les différents composants de

l’application en couches distinctes, ce qui facilite la maintenance et l’évolutivité de l’application. Le schéma suivant illustre cette architecture :

Figure 2.5 – Schéma de l’architecture logique de l’application

Partie frontend

L’architecture de projet fait référence à la structure et à l’organisation générale d’une appli- cation Angular. Cela comprend la façon dont l’application est divisée en modules, composants, services et autres éléments, ainsi que la manière dont ces éléments interagissent les uns avec les autres.

-**La couche de présentation** d’un projet Angular gère l’affichage de l’interface graphique pour l’utilisateur. Elle est composée de différents éléments tels que les composants, directives, pipes, templates et styles, et est chargée de gérer l’état de l’interface utilisateur et l’interaction avec l’utilisateur.

-**La couche service** regroupe les fonctionnalités communes et la logique métier de l’ap- plication. Les services sont des classes injectables qui permettent de séparer la logique métier de la logique de présentation. Ils facilitent la réutilisation du code, améliorent la maintena- bilité,manipulation de données. Les services sont utilisés pour encapsuler la logique métier complexe et fournir une interface simplifiée aux composants, tout en permettant de centraliser et de partager des données entre les différents composants de l’application.

-**La couche de données** est responsable de l’accès aux données, qu’elles soient stockées localement ou à distance. Cette couche peut inclure des classes de modèle représentant les don- nées de l’application, des services permettant de récupérer et de manipuler les données, ainsi que des intercepteurs HTTP pour la gestion des erreurs et la configuration des requêtes. Son rôle est crucial pour la gestion des données dans l’application.

-**Couche de sécurité** fournit des fonctionnalités de sécurité pour l’application, telles que l’authentification, l’autorisation et la gestion des rôles. Elle peut contenir des services et des

intercepteurs HTTP pour intercepter les requêtes et les réponses, ainsi que des gardes de route pour protéger les routes de l’application.

Partie Backend

Le couplage faible est un principe de conception logicielle qui vise à minimiser la dépendance entre les différentes parties d’un système. En d’autres termes, il s’agit de concevoir les différents modules ou composants d’un système de manière à ce qu’ils soient le moins interdépendants possible. Cela permet de faciliter la maintenance, l’évolution et le test des différents compo- sants du système, et de réduire les risques d’effets de bord ou de répercussions inattendues lors de la modification ou de l’ajout de fonctionnalités. Pour garantir que notre application a un couplage faible, nous adoptons cette architecture multicouches développée avec Spring Boot :

-**La couche web** : contient principalement le contrôleur qui est responsable de la gestion du flux de contrôle de l’application et de l’interaction avec la couche de service pour récupérer les données nécessaires à la réponse.

-**La couche Service** : La couche Service nous aide à former une encapsulation claire et stricte du code qui a mis en œuvre des tâches métier, des calculs et des processus.

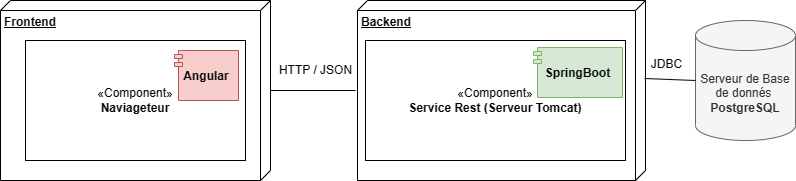
-**La couche accès au données (dao)** : Les classes de cette catégorie définissent les don- nées applicatives échangées entre le système et l’utilisateur.

* **Entities** : sont des classes qui représentent les objets métier dans une application Spring- Boot.

Repository est une interface qui permet d’effectuer des opérations de persistance de données sur une entité définie dans la couche des entités (entities).

* **La couche sécurité** : est fournie par Spring Security, qui permet de protéger l’application contre les attaques courantes. La méthode JWT est utilisée en conjonction avec Spring Security pour authentifier les utilisateurs dans une application web. Les tokens JWT contiennent des informations sur l’utilisateur et sont utilisés pour accéder aux ressources
* protégées..

### Architecture physique



## Backlog de produit

Le backlog est une liste de tâches à effectuer en priorité, qui définit les caractéristiques d’un produit. C’est l’un des éléments essentiels de la méthodologie Scrum. C’est le principal outil dont dispose le chef de produit pour recueillir les besoins des parties concernées et les transformer en une liste de fonctionnalités qui seront développées par l’équipe de développemen[t.[17]](#_bookmark204)

## Création de product backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id feature | Feature | Id user story | User story | Priorité | story points |
| 1 | Gestion de Utilisateurs | 1 | En tant qu'admin je veux m'authentifier | très élevée | 40 |
| 2 | En tant qu'admin je veux ajouter un utilisateur selon son rôle | élevée | 30 |
| 3 | En tant qu'admin je veux modifier un utilisateur ( y compris activer ou désactiver un utilisateur ) | moyenne | 20 |
| 4 | En tant qu'admin je veux consulter la liste des utilisateurs et leurs détails | élevée | 10 |
| 5 | En tant que utilisateur (ou client) je veux reçevoir un email d’activation lors de la création de mon compte par l’admin | très élevée | 5 |
| 6 | En tant que utilisateur (ou client) je veux m'authentifier selon mon rôle | très élevée | 40 |
| 7 | En tant que utilisateur (ou client) je veux modifier mon profile | faible | 10 |
| 8 | En tant que utilisateur (ou client) je veux changer mon mot de passe | faible | 10 |
| 9 | En tant que admin je veux exporter en excel la liste des utilisateurs et des clients | faible | 5 |
| 10 | En tant qu'admin je veux me déconnecter | faible | 20 |
| 2 | Gestion des Demandes1 | 11 | En tant qu'emplyé je veux ajouter un document | trés élevée | 40 |
| 12 | En tant qu'admin je veux consulter la liste des articles et leurs details | moyenne | 10 |
| 13 | En tant qu'admin je veux supprimer un article | moyenne | 20 |
| 14 | En tant qu' admin je veux modifier( y compris activer ou désactiver) un article | élevée | 30 |
| 15 | En tant qu'admin je veux charger une image à un article | élevée | 30 |
| 16 | En tant qu'admin je veux ajouter un échantillon relative à un article | élevée | 40 |
| 17 | En tant qu'admin je veux consulter la liste des échantillons et leurs details | élevée | 10 |
| 18 | En tant qu'admin je veux supprimer un échantillon | moyenne | 20 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 19 | En tant qu' admin je veux modifier ( y compris activer ou désactiver) un échantillon | moyenne | 30 |
| 3 | Gestion de Demandes 2 | 20 | En tant qu' admin je veux ajouter un paramètre | élevée | 40 |
| 21 | En tant qu' admin je veux supprimer un paramètre | moyenne | 20 |
| 22 | En tant qu' admin je veux modifier un paramètre | moyenne | 30 |
| 23 | En tant qu' admin je veux consulter un paramètre | élevée | 10 |
| 4 | Gestion de Projet 1 | 24 | En tant que client je veux consulter les articles depuis ma page d’acceuil | très élevée | 20 |
| 25 | En tant que client je veux passer une commande | très élevée | 40 |
| 26 | En tant que responsable de vente je veux aussi passer une commande pour un client | très élevée | 40 |
| 27 | En tant que client je veux suivre mes commandes en cours | moyenne | 20 |
| 28 | En tant que client je veux récupérer des mail qui décrit les états de mes commandes | faible | 5 |
| 5 | Gestion de Projet2 | 29 | En tant que responsable de dépot je veux ajouter un article au stock | élevée | 10 |
| 30 | En tant que responsable de dépot je veux retirer un article de stock | élevée | 10 |
| 31 | En tant que responsable de dépot je veux consulter l’historique de mouvemen de stock | moyenne | 40 |
| 6 | Gestion des commandes | 32 | En tant que responsable de vente je veux valider la commande de client | très élevée | 20 |
| 33 | En tant que responsable de vente je veux refuser la commande de client | moyenne | 10 |
| 34 | En tant que responsable de vente je veux consulter les détails de la commande de client | élevée | 20 |
| 35 | En tant que responsable de vente je veux générer les acquits à caution | très élevée | 40 |
| 36 | En tant que responsable de vente je veux télécharger les acquits à caution en PDF | faible | 5 |

###### Table 2.5 Backlog de produit

**Introduction**

L’étape de spécification des besoins est conçue pour la détermination des différentes fonctionnalités attendues du système. En effet, dans cette étape nous présentons tout d’abord, les acteurs concernés de notre système. Puis, nous entamons l’étude des besoins fonctionnels et non fonctionnels. Ces besoins seront exprimés sous la forme de diagrammes de cas

d’utilisation qui permettent de détailler les scénarios possibles que peuvent réaliser les différents acteurs.

1. **Identification des acteurs**

Tout système interactif, doit assurer et faciliter l’interaction avec ses utilisateurs (utilisateur humain ou non). Un acteur représente le rôle d’une entité externe exploitant le système à travers ses différentes interfaces. Nous spécifions dans le cadre de ce projet les acteurs du système, qui se divisent en trois catégories comme suit :

Administrateur : peut consulter la plateforme, affecter des rôles, créer un projet et créer des actualités et valider finalement les demandes.

Chef du projet : peut s'authentifier, consulter les listes d'employés et leurs coordonnées, consulter les actualités et valider partiellement les demandes de congé et d'autorisation de sortie, puis les envoyer à l'Administrateur pour finaliser la validation.

Employé : peut s'authentifier, mettre à jour sa fiche de renseignements personnels, faire des demandes de congé, de formation, de prêt avance et d'autorisation de sortie, et consulter les actualités.

1. **Identification des besoins**

Dans cette section, nous allons exposer les besoins fonctionnels et non fonctionnels de notre application.

1. **Besoins fonctionnels**

Les besoins fonctionnels sont ceux qui doivent répondre aux exigences du futur système en termes de fonctionnalités. Ils permettent de générer les cas d’utilisation. Les besoins recensés sont comme suit :

**La gestion des utilisateurs** : L’administrateur assure la gestion des sessions de connexion et gère les comptes des utilisateurs.

**Gestion des Projet :** Cette tâche sera effectuée par l'administrateur du système. L'administrateur aura la capacité d'ajouter de nouveaux projets, de supprimer des projets

existants, de modifier les informations relatives à un projet (comme les dates de début et de fin, les responsables, les équipes assignées, etc.), et d'effectuer d'autres opérations liées à la gestion des projets. Cette fonctionnalité permettra de maintenir à jour les données de projet et de faciliter la coordination entre les différents départements.

**Gestion des Demandes :**

L'administrateur est responsable de la gestion de tous les types de demandes, qu'il s'agisse de demandes de congés, d'autorisations de sortie, de formations, ou de documents administratifs. L'administrateur a la capacité de consulter toutes les demandes, de suivre leur état, et de décider de les accepter ou de les refuser.

En outre, le chef de projet dispose également de certaines responsabilités dans la gestion des demandes, mais de manière partielle. Il peut, par exemple, approuver ou refuser les demandes qui relèvent de sa compétence directe, avant qu'elles ne soient validées ou suivies par l'administrateur.

Cette double gestion permet de répartir les responsabilités tout en assurant un suivi rigoureux et une validation adéquate des demandes au sein de l'organisation.

**Gestion des Tâches :**

Le chef de projet est responsable de la gestion des tâches. Il a la capacité de créer de nouvelles tâches, de modifier les informations associées à une tâche existante, d'affecter les tâches aux employés appropriés, et de supprimer des tâches si nécessaire. Cette gestion permet au chef de projet de superviser efficacement le travail de son équipe, d'assurer une bonne répartition des responsabilités, et de suivre l'avancement des projets en cours.

1. **Besoins non fonctionnels**

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles le système est soumis pour sa réalisation et son bon fonctionnement.

* **Fiabilité**

Notre application doit fonctionner de façon cohérente sans erreurs.

* **Gain de temps**

Notre application doit optimiser les traitements pour avoir un temps de réponse minimal.

L’étude approfondie des spéciations permet de dégager plusieurs cas d’utilisation.

**Lisible**

L’application doit être simple et facile à utiliser.

**Modélisation des besoins :**

Dans cette section, nous allons identifier les acteurs ainsi que leurs cas d’utilisation :

**I.3.1. Identification des acteurs et des cas d’utilisation**

**L’administrateur :**

Son rôle est la gestion des sessions de connexion, l’application lui permettre de :

- S’authentifier

- De créer les paramètres d’accès des utilisateurs.

- la gestion des utilisateurs

-la gestion des demandes (demande congé, demande formation, demande autorisation sortie et demande document)

-la gestion des projets

- affectation des rôles : Employé ou chef

**Chef du projet :**

L’application lui permettre de faire :

-S’authentifier

-Consulter les listes des employées et ses coordonnés

-la gestions des taches

-Valider partiellement la demande de congé puis l’envoyer a l’administrateur

-Valider partiellement la demande d’autorisation sortie puis l’envoyer a Administrateur

-Tous les fonctionnalités d’Employé

**Employé**

L’Employé utilise l’application pour :

-S’authentifier

-mettre à jour sa fiche de renseignements personnels :

• Informations personnelles

• Informations sociales

• Informations professionnelles

-demande congé

-demande autorisation de sortie

-demande document

-demande congé

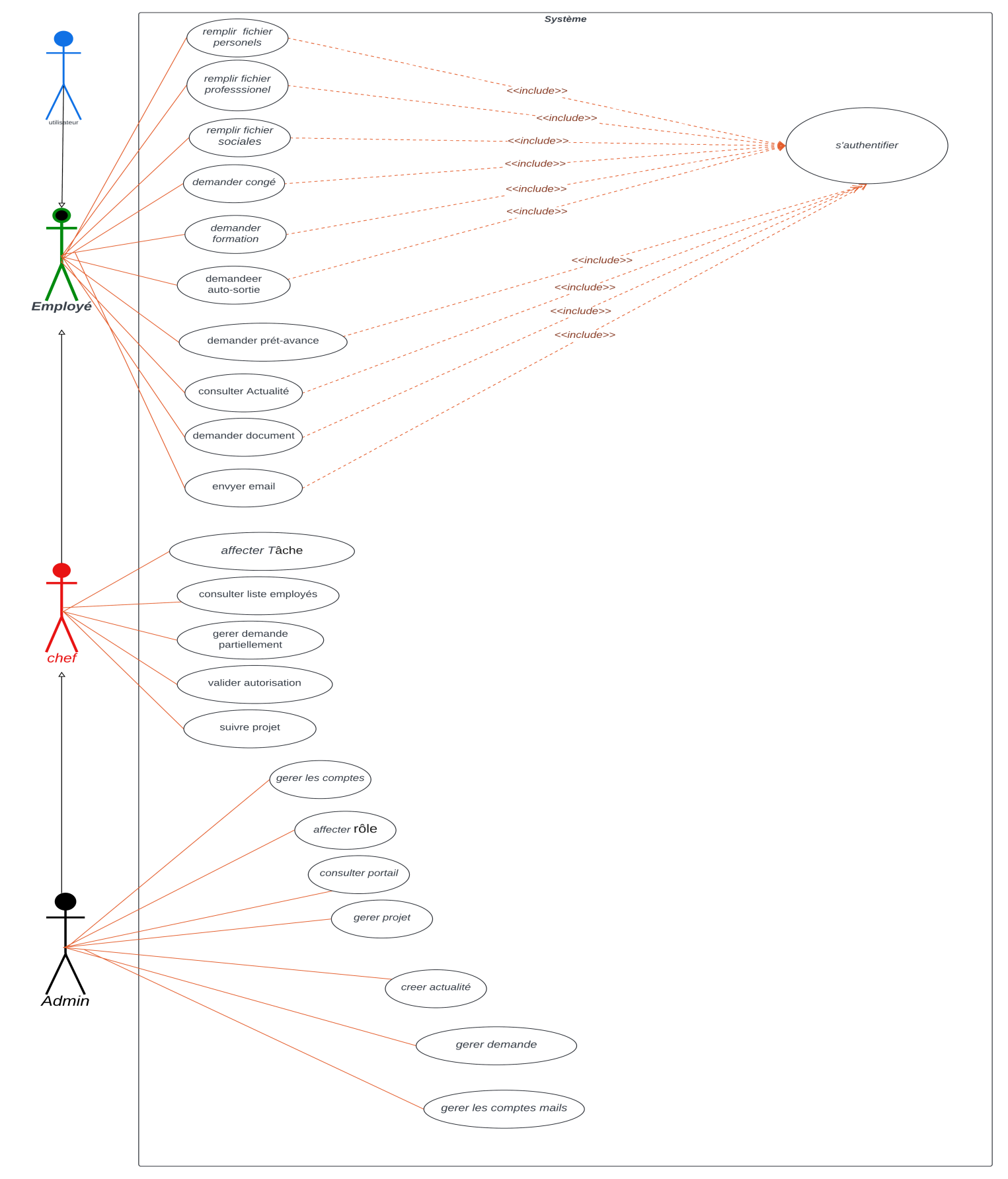
-suivre l’état de sa demande

-consulter les tâches qui leur ont été assignées par le chef de projet

-Suivre l'évolution de leurs tâches

1. **Diagramme de cas d’utilisation**

Un cas d’utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondants d’un système. Nous représentons dans la figure 1 ci-dessous, toutes les cas utilisations de base afin d’avoir une vue globale du fonctionnement de notre application, ainsi que les éventuelles relations qui peuvent avoir lieu.



**Diagramme de cas d’utilisation globale**

**CHAPITRE III: SPRINT 1**

## Introduction

Dans ce chapitre, nous allons détailler le travail réalisé lors du premier sprint. Tout d’abord, nous présentons le backlog du sprint et nous enchaînons avec la phase de conception des user stories identifiés. Enfin, nous présentons quelques interfaces homme-machine réalisées.

## Objectifs du Sprint 1

Dans ce chapitre, nous allons détailler le travail réalisé lors du premier sprint qui porte sur la gestion des comptes administrateurs , C , Employés et authentification de ces utilisateurs .

## Backlog du sprint 1

Le tableau ci-dessous présente le backlog de notre premier sprint : Authentification et gestion des comptes :

|  |  |  |
| --- | --- | --- |
| **ID** | **User Story** | **Story Point** |
| 1 | En tant qu’admin je veux m’authentifier | 40 |
| 2 | En tant qu’admin je veux ajouter un utilisateur selon son rôle | 30 |
| 3 | En tant qu’admin je veux modifier un utilisateur (y compris  activer ou désactiver un utilisateur) | 20 |
| 4 | En tant qu’admin je veux consulter la liste des utilisateurs et  leurs détails | 10 |
| 5 | En tant que utilisateur (ou client) je veux recevoir un email  d’activation lors de la création de mon compte par l’admin | 5 |
| 6 | En tant que utilisateur (ou client) je veux m’authentifier selon  mon rôle | 40 |
| 7 | En tant que utilisateur (ou client) je veux modifier mon profil | 10 |
| 8 | En tant que utilisateur (ou client) je veux changer mon mot  de passe | 10 |
| 9 | En tant que admin je veux exporter en excel la liste des uti-  lisateurs et des clients | 5 |
| 10 | En tant qu’admin je veux me déconnecter | 20 |

Table 3.2 – Backlog du Sprint1

Pour chaque user story mentionnée précédemment, nous entreprendrons les tâches sui- vantes :

20

* Analyse des user stories
* Prototypage et conception
* Implémentation du backend
* Implémentation du frontend
* Tests

.

|  |
| --- |
| **Acteurs**  Administrateur  **But**  Accéder à l’application  **Préconditions**  L’administrateur est authentifié  Le profil de l’administrateur est récupéré |

**Enchaînement :**

Tableau 1 : Enchainement du cas d’utilisation « Ajouter un nouvel utilisateur »

|  |  |
| --- | --- |
| Action Acteur | Action système |
| 1.L’acteur demande l’accès au système |  |
|  | 2-Le système affiche le formulaire d’authentification |
| 3-L’acteur saisit son identifiant et son mot de passe |  |
|  | 4-Le système vérifie les champs (champs obligatoire) |
|  | 5-Le système vérifie l’existence de l’utilisateur |
|  | 6-le système affiche la page d’accueil. |

**Enchainement d’exception :**

A1 : Login ou mot de passe non valide

1-Le système affiche un message d’erreur

2-Le scénario reprend de point 2

A2 : Mot de passe oubliée

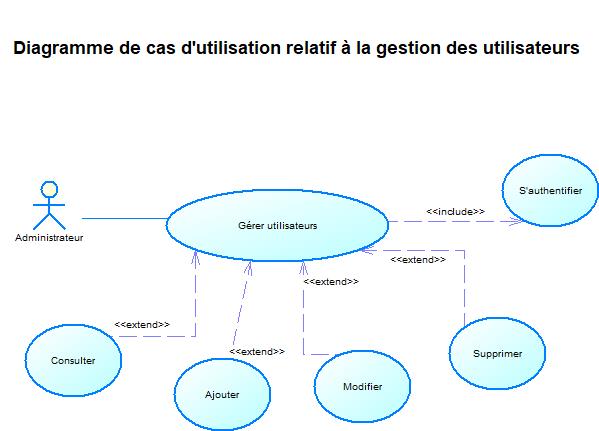
1-introduire (courriel)

2-Envoyer password au courriel

3- Le scénario reprend de point 3

**3.1.1 Diagramme de cas d’utilisation relatif à la gestion des utilisateurs**

Figure 2 : Diagramme du cas d’utilisation détaillé relatif à la gestion des utilisateurs



Dans ce qui suit, nous détaillons les différents cas d’utilisation mentionnés dans la figure 2.

* **Ajouter un nouvel utilisateur**

\*

|  |
| --- |
| **Acteurs**  Administrateur  **But**  Ajouter un nouvel utilisateur  **Préconditions**  L’administrateur est authentifié  Le profil de l’administrateur est récupéré |

**Enchaînement :**

Tableau 1 : Enchainement du cas d’utilisation « Ajouter un nouvel utilisateur »

|  |  |
| --- | --- |
| Action Acteur | Action système |
|  | 1. Le système affiche à l’administrateur  un formulaire à remplir contenant les  informations d’un utilisateur. |
| 2. L’administrateur saisit les informations de  l’utilisateur et valide son choix. |  |
|  | 3. Le système crée l’utilisateur et  synchronise avec l’Active directory. |
|  | 4. Le système affiche la liste des utilisateurs. |

**Enchainement d’exception :**

E1 : Détection de champs vides.

Le système affiche un message d’erreur.

L’enchainement E1 démarre au point 2 du scénario nominal.

E2 : L’administrateur choisit d’annuler l’opération.

L’enchainement E2 démarre au point 2 du scénario nominal.

Le scénario nominal reprend au point 1.

**Post condition :**

L’ajout d’un nouvel utilisateur est effectué.

* **Modifier les informations d’un utilisateur**

|  |
| --- |
| **Acteurs**  Administrateur  **But**  Modifier les informations d’un utilisateur  **Préconditions**  L’administrateur est authentifié  Le profil de l’administrateur est récupéré |

**Enchaînement :**

Tableau 2 : Enchainement du cas d’utilisation « Modifier les informations d’un utilisateur »

|  |  |
| --- | --- |
| Action Acteur | Action système |
|  | 1. Le système affiche la liste des utilisateurs. |
| 2. L’administrateur sélectionne un utilisateur. |  |
|  | 3. Le système affiche le formulaire contenant  les informations de l’utilisateur. |
| 4. L’administrateur modifie les informations  et valide son choix. |  |
|  | 5. Le système enregistre les données et  affiche la nouvelle liste. |

**Enchainement d’exception :**

E1 : L’acteur choisit d’annuler l’opération.

Le système affiche un message d’erreur.

L’enchainement E1 démarre au point 4 du scénario nominal.

Le scénario nominal reprend au point 2.

**Post condition :**

La modification des informations de l’utilisateur est effectuée.

* **Supprimer un utilisateur**

|  |
| --- |
| **Acteurs**  Administrateur  **But**  Supprimer  **Préconditions**  L’administrateur est authentifié  Le profil de l’administrateur est récupérer |

**Enchaînement :**

Tableau 3: Enchainement du cas d’utilisation «Supprimer un utilisateur»

|  |  |
| --- | --- |
| Action Acteur | Action système |
|  | 1. Le système affiche la liste des utilisateurs. |
| 2. L’administrateur sélectionne l’utilisateur à supprimer. |  |
|  | 3. Le système demande une confirmation de  Suppression. |
| 4. L’administrateur confirme. |  |
|  | 5. Le système enregistre les données et  affiche la nouvelle liste. |

**Enchainement d’exception :**

E1 : L’acteur choisit d’annuler l’opération.

Le système affiche un message d’erreur.

L’enchainement E1 démarre au point 4 du scénario nominal.

Le scénario nominal reprend au point 2.

**Post condition :**

La suppression est effectuée.

|  |
| --- |
| En tant que Employé je veux mettre a jour mes informations |
| En tant que Employé je veux Remplir une demande |
| En tant que Employé je veux cliquer pour envoyer le demande au chef |
| En tant que Chef je veux met mon decision de ce demande : je veux accepter partiellement et envoyer a l’admin ou refuser la demande |
| En tant que Admin je veux accepter la demande envoyé par le chef ou refuser finalement |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Les acteurs principaux |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |