 Université de Jendouba

Institut Supérieur d’Informatique de Kef

**Mémoire de fin d’études**

**Présenté en vue de l’obtention du diplôme de Licence en Sciences De l’Informatique**

**Spécialité : Génie Logiciel et Système d’Information**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Elaboration d’une application de gestion des congés et des absences des personnels travaillant dans une entreprise

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Réalisé au sein de : **ARAB SOFT**

Réalisée par : Marwa Ayari

Souhayeb Hmissi

**Chapitre 1 : Présentation de l’organisme d’accueil et étude préalable**

**Introduction**

Afin de mettre en évidence le contexte dans lequel s’inscrit notre projet, ce chapitre présente deux points :

Dans un premier temps, nous présentons l’organisme d’accueil dans lequel notre stage a été effectué.

Dans un second lieu, nous abordons le contexte et les objectifs du projet.

Enfin, nous conclurons ce chapitre par la planification du projet et la spécification des besoins.

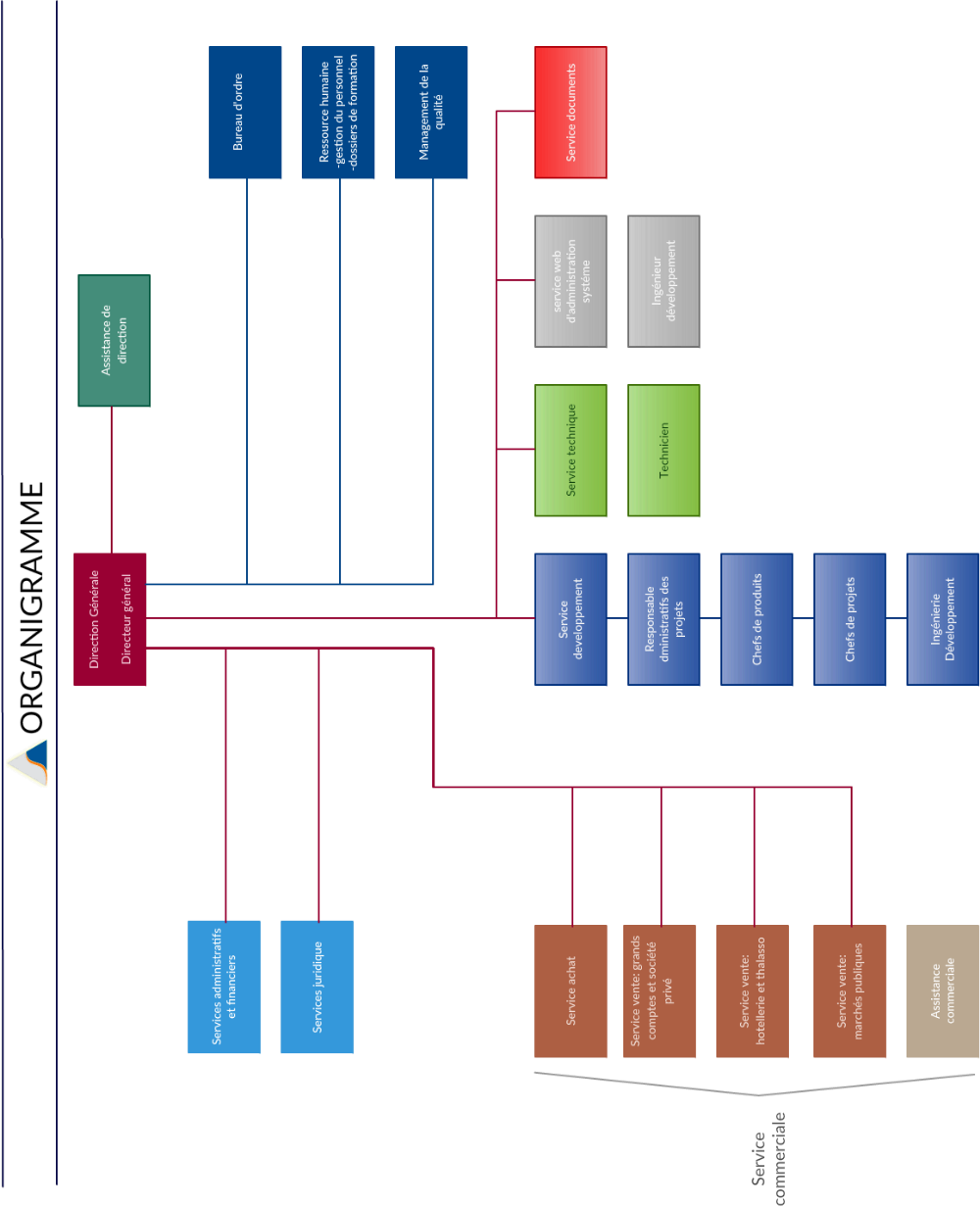
1. **Présentation générale**
   1. **Présentation de l’organisme**

ARABSOFT est une entreprise spécialisée dans l’étude, le développement et la distribution de solution de gestion notamment pour le secteur public.



*Figure1 : Logo de la société ARABSOFT*

Créé en 1985 par Mohamed Triki, au fil des années, l’entreprise s’est engagée dans un processus continu visant à renforcer sa présence à l’international, à répondre aux défis métiers spécifiques de ses clients, à proposer des solutions innovantes en mode SAAS et à consolider ses relations avec sa clientèle.

****

* 1. **Organigramme de l’organisme**

*Figure2 : Organigramme de la société*

1. **Etude préalable**
   1. **Etude de l’existant**

L’étude de l’existant constitue le cœur de la phase d’analyse d’un projet, qui a pour objectif d’aborder et de présenter une bonne spécification des besoins. Nous tiendrons de prévoir et d’analyser de manière précise les fonctionnalités et les résultats attendus de notre projet. Pour cela, nous commencerons par une étude des outils et des solutions existantes, par la suite, nous dégagerons les anomalies et mettre en valeur l’apport de notre solution.

* 1. **Description de l’existant**

Actuellement, la gestion des congés et des absences à ARAB SOFT est principalement gérée de manière manuelle et fragmentée, reposant sur des processus administratifs et des outils de communication traditionnels. Voici un aperçu de l'existant :

* Gestion manuelle des données du personnel
* Demandes de congés et d’absences par formulaire papier ou email
* Validation des congés par les gestionnaires
* Suivi des congés et des absences à l’aide de feuille de calcul
  1. **Critique de l’existant**
  2. **Solutions proposées**

1. **Méthodologie de gestion de projet**

Une méthodologie de gestion de projet se compose d’un ensemble de principes, techniques et procédures que les professionnels d’une discipline particulière utilisent pour gérer leurs projets.

* 1. **Méthodologie agile**

La méthode Agile se caractérise par sa focalisation sur des principes clés tels que la collaboration étroite entre les équipes de développement et les parties prenantes métier, la livraison régulière de fonctionnalités utilisables, une communication ouverte et directe, ainsi que l'acceptation des changements de besoins du client tout au long du projet. Elle favorise également la créativité, l'excellence technique et la motivation des développeurs, ce qui permet de créer des logiciels de haute qualité répondant aux besoins du client de manière efficace et adaptative.

1. **Choix de la m´méthodologie : SCRUM**

J’ai décidé de suivre le processus de d´enveloppement agile SCRUM vu qu’il présente la m´méthodologie la plus appropriée `à ma démarche projet. Ce processus possède plusieurs avantages tels que [KQK12] :

• Une gestion de projet plus transparente :

Il offre un espace de travail collaboratif.

• Une meilleure communication : La m´méthode Scrum met l’accent sur la communication au sein de l’´équipe, la collaboration et la solidarité • Une meilleure qualité du produit final : Des présentations régulières permettent de recueillir rapidement des commentaires et des retours d’information des parties prenantes.

**Conclusion**

Dans ce premier chapitre, j’ai donné un large aperçu de mon projet. J’ai couvert la société hôte, l’étude de l’existant, la m´méthodologie adoptée et le langage de modélisation que je vais utiliser. A ce stade, je peux d´désormais passer au chapitre suivant, qui contiendra Analyse et Spécification des besoins .

**Chapitre 2 : « SPRINT 0 » Analyse et spécification des besoins**

**Introduction**

Comme indiqué dans le chapitre précédent, nous avons choisi d’adopter la méthodologie Scrum pour la conception de notre plateforme. Dans ce chapitre « Analyse et spécification des besoins » nous mettrons en œuvre la première étape de la méthodologie Scrum, au cours de laquelle nous allons identifier les rôles des utilisateurs, et nous allons recenser les besoins fonctionnels et non fonctionnels pour les mettre dans un backlog de produit et puis planifier les sprints afin de réaliser la partie conception globale de l’application.

1. **Analyse des besoins** 
   1. **Besoins fonctionnelles**

Cette partie vise à compter les besoins des utilisateurs de notre application afin de bien comprendre les fonctionnalités de chaque acteur.

**Identification des acteurs**

Les acteurs sont des entités externes qui interagissent avec le système, comme une personne humaine ou un robot. Une même personne (ou robot, ...) peut jouer le rôle de plusieurs acteurs pour un système, c'est pourquoi les acteurs devraient être décrits par leur rôle, ce rôle décrit les besoins et les capacités de l'acteur.

Dans cette partie, je présenterai les acteurs de l'application qui sont :

• Personnel : Représente un acteur important car il consomme les services fournis par notre application.

• Administrateur : L’acteur principal dans l’application qui gère tous les fonctionnalités de la plateforme.

• Chef Hiérarchique : le chef hiérarchique joue un rôle essentiel, en veillant à ce que les politiques et les procédures soient respectées, que les opérations se déroulent de manière efficace et que les membres de l'équipe reçoivent le soutien nécessaire.

**Les fonctionnalités communs**

• Gérer congés:

♣ Un Administrateur peut ajouter, consulter, modifier et planifier les congés.

♣ Un Administrateur peut appliquer les règles de gestion sur les différents types et demandes de congés.

♣ Un Chef Hiérarchique peut consulter, modifier et planifier des congés.

♣ Un Chef Hiérarchique peut appliquer les règles de gestion sur les différents types et demandes de congés.

♣ Un Personnel peut consulter, ajouter, modifier et supprimer des congés.

• Gérer absences:

♣ Un Administrateur peut consulter, ajouter et modifier des absences.

♣ Un Chef hiérarchique peut consulter des absences.

♣ Un Personnel peut consulter des absences.

• Gérer messageries :

♣ Un Administrateur peut consulter, ajouter et supprimer des messages.

♣ Un Personnel peut consulter, ajouter et supprimer des messages.

♣ Un Chef hiérarchique peut consulter, ajouter et supprimer des messages.

• Gérer notifications :

♣ Un Administrateur peut consulter et supprimer des notifications.

♣ Un Chef hiérarchique peut consulter et supprimer des notifications.

♣ Un Personnel peut consulter et supprimer des notifications.

• S’authentifier:

♣ Un Administrateur peut authentifier.

♣ Un Chef hiérarchique peut authentifier.

♣ Un Personnel peut authentifier.

♣ Dans le cas où un administrateur a perdu son mot de passe il peut créer un nouveau mot de passe à travers le lien « Forgot password »

**2.1.1. Les fonctionnalités de l’administrateur**

• Gérer personnels :

♣ Un Administrateur peut ajouter, consulter, modifier, supprimer et sanctionner un personnel. • Gérer Dashboard :

♣ Un Administrateur peut consulter, modifier, le Dashboard.

**2.3 Besoins non fonctionnels**

Après avoir détaillé les besoins fonctionnels de notre application dans la partie précédente, il nous reste maintenant à examiner les besoins non fonctionnels qu'elle propose. Donc nous essayons de satisfaire au maximum possible toutes les besoins non fonctionnelles suivants à travers les choix méthodologiques et technologique ainsi que les approche et les outils utilisé pour réaliser ce projet.

**2.3.1 Performance**

Notre logiciel doit être avant tout performant, c’est-à-dire, à travers ses fonctionnalités, il doit répondre à toutes les exigences des usagers d’une manière optimale. En utilisant un Framework qui assure la technique SPA dans la partie frontend pour favoriser un affichage rapide des composants.

**2.3.2 Convivialité**

Les interfaces utilisateur doivent être simples et ergonomiques, puisque notre application va être utilisée par des personnes de niveaux académiques différents et des utilisateurs non connaisseurs aux outils informatiques.

**2.3.3 EXTENSIBILITÉ**

L’application devra être extensible, c’est-à-dire qu’il pourra y avoir une possibilité d’ajouter ou de modifier de nouvelles fonctionnalités. Pour garantir ce besoin on va utiliser la technique d’injection des dépendances, pour éviter la dépendance du code de l’application.

**2.3.4 SÉCURITÉ**

Il faudra aussi noter que l’application devra être sécurisée, car les informations ne devront pas être accessibles à tout le monde, d’où l’utilisation de JWT pour chaque acteur.

**2.3.5 MAINTENABILITÉ**

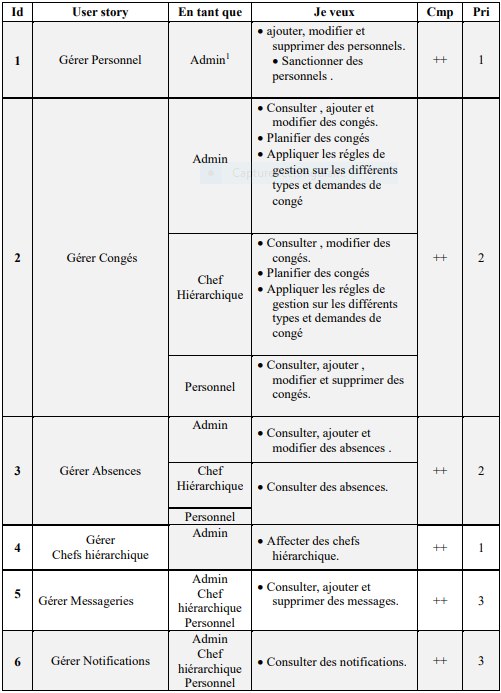
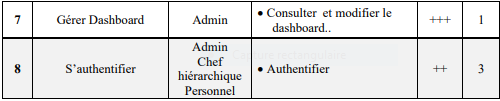
La maintenabilité de notre application il doit être assuré automatiquement lors du développement avec une combinaison costaux « Spring-Boot et Angular » et avec l’aspect de l’orienté objet ainsi le test et le contrôle de la complexité des codes développés.

STRUCTURE DE DECOUPAGE DE PROJET

**3.Equipe scrum :**

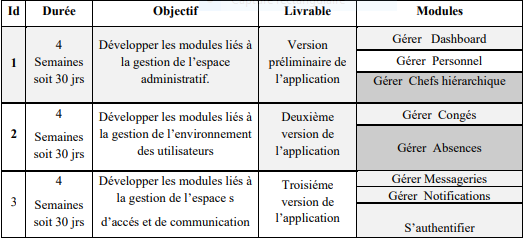
**3.2 Backlog produit**

Le backlog produit est un document qui contient les exigences initiales dressées puis hiérarchisées avec le client en début de projet. Néanmoins il va évoluer tout au long de la durée du projet, en fonction des divers besoins du client. ¬ Priorité (Pri) : L’échelle de la priorité s’étale de 1 pour dire moins prioritaire à 2 pour dire plus prioritaire. ¬ Complexité (Cmp) : Échelle de complexité : + moins compliqué, ++ compliqué, +++ très compliqué. Critères de complexité : la complexité de chaque module dépend de nombre des APIs à implémenter, et des nombres d’heures de travail nécessaire pour compléter les tâches de ce module



**3.3 Planification des sprints**

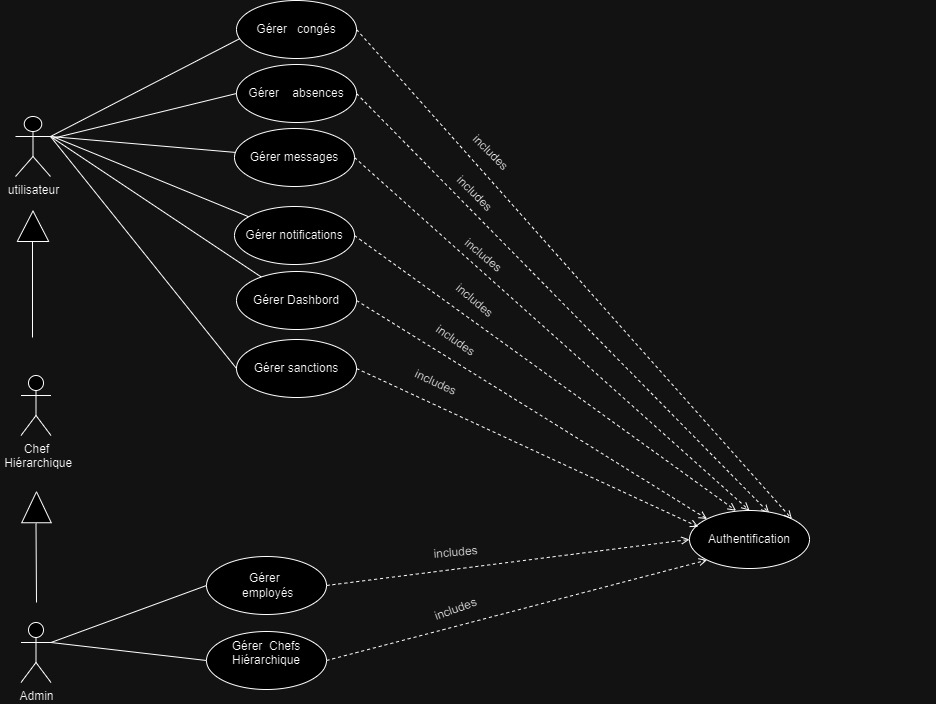
La planification des sprints c’est le livrable d’une réunion de l’équipe Scrum. Dans notre cas les modules sont reparties de tel sort tous les sprints seront équilibrés au niveau de la complexité et triés par ordre d’importance.

****

**4 Conception global**

**4.1 Diagramme Cas D’utilisation Globale**

Dans cette section nous présentons les besoins de notre système de manière formelle. C'est à- dire en utilisant le diagramme des cas d’utilisation globale. Les fonctionnalités globales offertes par l’application sont représentées dans le diagramme ci- dessous. Tous les cas d’utilisation nécessitent une authentification comme précondition.



**4.1 Diagramme De Classe Globale**

Le diagramme de classes de conception représente la structure statique du code, à travers des attributs et des relations entre les classes. Il s'agit d'un diagramme principal, qui est la vue de plus haut niveau avec toutes les classes de l'application.

**Conclusion**

Dans ce chapitre, nous avons approfondi le niveau de vision du projet et nous avons permis de bien comprendre les tâches à réaliser. Entre autres, on a pu prévoir les problèmes à rencontrer et essayer de chercher des solutions pour les surmonter. Nous avons essayé de mettre en valeur les besoins fonctionnels et techniques du projet et les différents scénarios d'utilisation de l’application. Les chapitres suivants vont mieux clarifier le bon fonctionnement de notre projet à travers le raffinement de la conception des modules de chaque sprint.