Chapitre 1

Analyse et Spécifications besoins

1.1 Introduction

Cette partie consiste en une étape analytique dans laquelle nous allons recenser et factoriser les besoins des utilisateurs de l'application. Ceci est fortement lié à l'étude préalable menée au Cours du premier chapitre. Pour ce faire cette phase doit répondre aux questions suivantes : Quels sont les besoins fonctionnels de l'application ? Quelles sont les contraintes qui doivent être prises en considération ?

1.2 Les acteurs du système

C'est une entité externe qui agit sur le système (opérateur, autre système, ...).Il peut consulter Ou modifier l'état du système. En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. Le principal acteur de système :

- **L'administrateur :** entité externe principale. Son rôle est qui a le droit de gérer un projet (créer, modifier, supprimer). Aussi son rôle et de gérer l'affectation des taches aux membres correspondants.
- **Le membre :** entité externe secondaire, il peut se connecter pour consulter la tâche en cours que l'administrateur lui a effectué et la marquer comme terminée ou non.

1.3 Spécifications besoins

1.3.1 Besoins fonctionnels

Le besoin primordiale de notre application et de permettre à l'administrateur de « Cherchini » de gérer les projets et ceci consiste à :

Gérer les projets

- La création d'un projet.
- La modification d'un projet.
- La suppression d'un projet.
- Consulter les projets et les tâche et les détails correspondants.
- Affecter les membres correspondants à chaque projet.

Gérer les tâche

- La création d'une tâche.
- La modification d'une tâche.
- La suppression d'une tâche.
- Consulter les tâche et les détails correspondants.
- Affecter le membre correspondant à chaque tâche.

Gérer les membres

- La création d'un membre.
- La modification d'un membre.
- La suppression d'un membre.
- Consulter les membres et leurs détails correspondants.
- Affecter les membres correspondants à chaque projet.

Gérer les clients

- La création d'un .
- La modification d'un client.
- La suppression d'un client.
- Consulter les clients et leurs détails correspondants.

Suivre le déroulement des projets

- Suivre le travail des équipes en consultant le diagramme Gantt pour chaque projet.
- Consulter les rapports des coûts et les durées selon les projets et les clients .
- Consulter la carte géographique des géolocalisations des .

1.3.2 Besoins non fonctionnels

Les besoins non fonctionnels spécifient les propriétés du système afin de garantir la Cohérence, la confidentialité et l'intégrité des données. Le système doit être fiable : la validité de l'application. Réutilisabilité : aptitude de site à être utilisé en tout ou en partie dans de nouvelles applications.

La performance d'exécution

Le temps d'exécution du système doit être minimal pour ne pas gêner l'utilisateur. Ce temps dépend de la complexité du code implémenté, du serveur d'application utilisé, du débit de la ligne de connexion et de la conception de la base de données.

La sécurité

Le système doit respecter un niveau de sécurité élevé afin de garantir la confidentialité de l'accès des membres

L'ergonomie

L'interface de cette application doit être ergonome, conviviale et voire même apte à aider l'utilisateur à mieux gérer son espace de travail.

1.4 Diagramme de cas d'utilisation générale

Le diagramme de cas d'utilisation permet de décrire l'interaction entre l'acteur et le système. Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Un acteur et un cas d'utilisation sont mis en relation par une association

représentée par une ligne. Le but principal du diagramme du cas d'utilisation est de définir le système du point de vue des utilisateurs et de définir les limites précises du système en utilisant une notation très simple

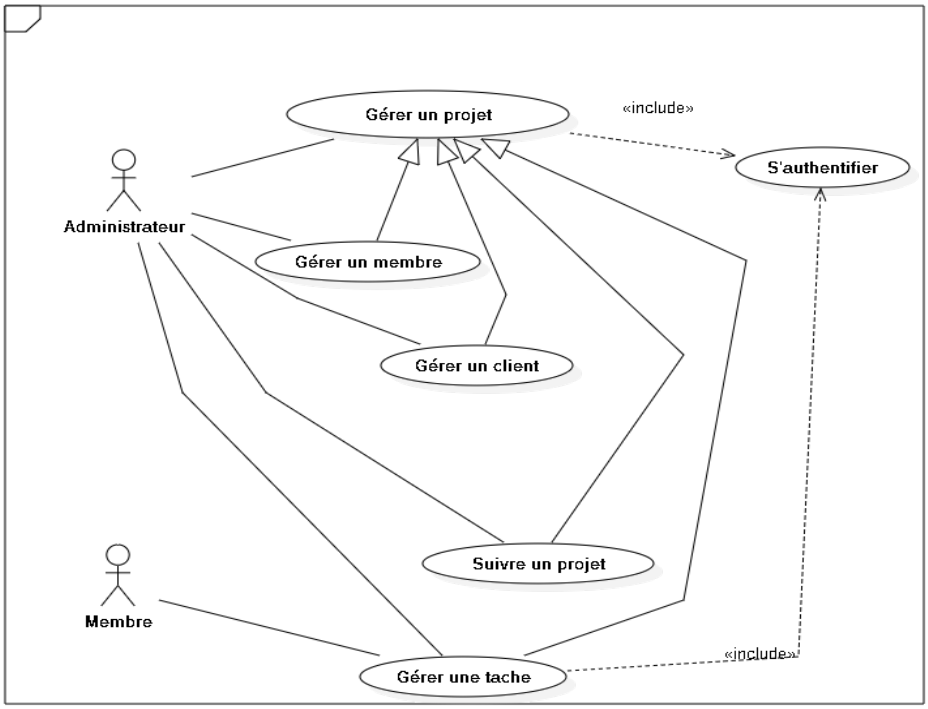

FIGURE 1.1 – Diagramme de cas d'utilisation générale.

1.5 Backlog de planning

Aprés avoir définit les acteurs des systémes et les différentes interactions nous pouvons maintenant définir notre Product Backlog puis nous précisions la planification des sprints.

1.5.1 Les fonctionnalités du Backlog

Le Backlog est un artéfact très important dans SCRUM. C'est l'ensemble des caractéristiques fonctionnelles ou techniques qui constituent le produit souhaité. Nous allons les décrire en détails dans le tableau qui suit :

Fonctionnalité	Acteur	Description
Gérer un projet	Administrateur	L'administrateur peut gérer un projet et ses tâches correspondantes d'affectation
Mettre à jour un Projet	Administrateur	L'administrateur peut changer les détails du projet ainsi que l'affectation des membres au projet
Créer ,Modifier , Supprimer un membre	Administrateur	L'administrateur peut manipuler les données des membres
Créer ,Modifier , Supprimer un client	Administrateur	L'administrateur peut manipuler les données des clients
Consulter les rapports	Administrateur	L'administrateur peut accéder aux rapports
Consulter les coordonnées des clients sur la carte géographique	Administrateur	L'administrateur peut accéder aux coordonnées géographiques des clients
Changer l'état et la progression approximative de ses tâches	Membre	Le membre peut changer ses taches courantes selon l'avancement.

TABLE 1.1 – Product Backlog

1.5.2 Planification des sprints

Release 1	Gestion des membres	14 jours
(L'application de gestion de projets)	Gestion des clients	5 jours
	Gestion des projets	20 jours
	Authentification	5 jours
	Interfaces administrateur	2 jours
	Interfaces membre	1 jour
Release 2	Affichage des rapports	15 jours
(La partie informatique décisionnelle)		

TABLE 1.2 – Planification des sprints