

# Introduction Générale

Dans un contexte professionnel où la collaboration et la gestion efficace des projets sont devenues essentielles, il est crucial de disposer d'une plateforme centralisée pour le partage de projets et de documents. La prolifération des outils et des canaux de communication peut souvent entraîner une dispersion de l'information et une diminution de l'efficacité. Pour remédier à ce problème, notre équipe a développé une plateforme innovante nommée "HiveShare", conçue pour faciliter la collaboration et la gestion des projets au sein des équipes.

HiveShare centralise toutes les informations relatives aux projets, améliorant ainsi la communication entre les membres de l'équipe et optimisant la gestion des ressources. En intégrant les divers aspects de la gestion de projet en un seul outil, HiveShare offre une solution complète et intégrée, simplifiant le suivi et la coordination des tâches. Cette plateforme inclut des fonctionnalités de gestion des utilisateurs, de création et de suivi de projets, et de consultation de statistiques, le tout dans une interface utilisateur intuitive et sécurisée.

Ce rapport détaille le processus de conception et de développement de HiveShare, en mettant en lumière les défis rencontrés, les solutions apportées et les résultats obtenus. Il s'inscrit dans le cadre de notre projet PDL à l'université Sésame, démontrant notre capacité à appliquer les connaissances acquises et à contribuer efficacement à des projets innovants en milieu professionnel.

# Chapitre 1

## Cadre général de projet

### I. Introduction

L'étude de projet est une démarche stratégique visant à organiser le bon déroulement d'un projet et à assurer la conduite de toutes les phases qui le constituent. Dans ce chapitre, nous nous intéressons à la présentation de l'université Sésame, où nous avons réalisé notre projet PDL. Ensuite, nous présenterons notre projet, suivi de la méthodologie adoptée pour sa conception et son développement.

### II. Présentation de l'organisme d'accueil

L'École Supérieure des Sciences Appliquées et de Management – Université SESAME – est un établissement privé de formation supérieure agréé par l'État tunisien (ministère de l'Enseignement supérieur), situé dans le technopôle Elgazala. L'Université SESAME combine une école d'ingénieurs et une école de management. Forte de son expérience en formation continue, elle a débuté en 2011 la formation d'ingénieurs en informatique. En 2012-2013, elle a lancé un cycle préparatoire intégré pour les études d'ingénieurs, des licences appliquées en informatique et des formations en management (licences appliquées et masters professionnels).



Figure 1 : Logo Université Sésame

### III. Présentation de projet

Notre plateforme “HiveShare” de partage de projets et de documents permet aux équipes de collaborer efficacement. Elle facilite l'échange de fichiers et l'organisation des projets en un seul endroit centralisé. Les membres de l'équipe peuvent accéder aux documents, suivre l'avancement des projets et travailler ensemble de manière synchronisée. La gestion des projets devient simple et fluide, assurant ainsi une meilleure coordination et productivité.

#### **Méthodologies de travail**

Les méthodes « Agiles » sont des ensembles de pratiques pour le développement informatique (conception de logiciels) qui peuvent s'appliquer à divers types de projets.

Leur objectif principal est la satisfaction réelle du client, au-delà des termes d'un contrat de développement. Elles reposent sur un cycle de développement itératif, incrémental et adaptatif.

## **Méthodologie SCRUM**

Pour éviter le développement des choses qui ne seront pas utilisées, on a choisi l'approche Scrum. Cette méthodologie suit les principes de la méthodologie Agile, c'est-à-dire l'implication et la participation active du client tout au long du projet.

### **2.1) Équipe SCRUM**

Scrum définit un modèle d'équipe optimisant la flexibilité, la créativité et la productivité. Nous définissons, notre équipe de travail ainsi que le rôle de chaque équipier selon les rôles définis par Scrum

Le Product Owner : M.Mohamed Ali Hamzaoui

Le Scrum Master : M. Mohamed Ali Hamzaoui

L'équipe : Riahi Ahlem , Belhadj mohamed Emna , Blouza Aziz

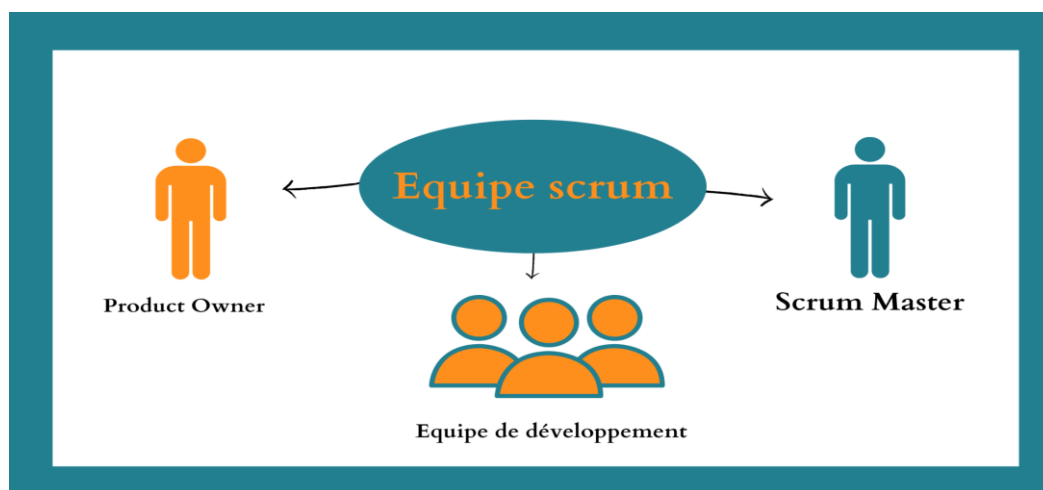


Figure 2 : L'équipe de SCRUM

### **2.2) Cycle de vie d'un projet SCRUM**

La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps.

— Planification du Sprint : au cours de cette réunion, l'équipe de développement sélectionne les éléments prioritaires du « Product Backlog »(liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du sprint (en accord avec le « Product Owner »).

— Revue de Sprint : au cours de cette réunion qui a lieu à la fin du sprint, l'équipe de développement présente les fonctionnalités réalisées au cours du sprint et recueille les feedbacks du « Product Owner »et des utilisateurs finaux. C'est également le moment

d'anticiper le périmètre des prochains sprints et d'ajuster au besoin la planification de release (nombre de sprints restants).

— Rétrospective de Sprint : la rétrospective qui a généralement lieu après la revue de sprint est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail) à la lueur du vécu sur le sprint écoulé (principe d'amélioration continue).

— Mêlée quotidienne : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout (elle est aussi appelée « stand up meeting ») en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici à la prochaine mêlée ? Quels obstacles me retardent ? »

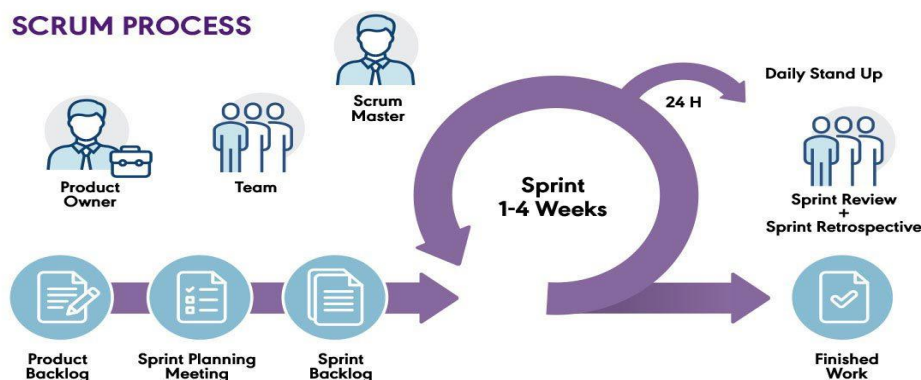


Figure 3 : Cycle de vie de la méthode SCRUM

#### iv. Langage de modélisation UML

Après avoir choisi la méthodologie, nous avons besoin d'un langage de modélisation pour concevoir notre système. Nous avons opté pour le Unified Modeling Language (UML) en raison de ses points forts, notamment sa standardisation et la variété de diagrammes qu'il propose. UML est idéal pour schématiser des systèmes complexes de manière graphique et textuelle, offrant une vue simplifiée et normalisée du projet.

#### v. Conclusion

À la fin de ce premier chapitre, nous avons présenté notre projet ainsi que la méthodologie de conception à adopter pour le développement de notre application : la méthode Agile Scrum associée au formalisme UML.

# Chapitre 2

## Planification et architecture

### I. Introduction

Dans le chapitre précédent, nous avons choisi d'adopter la méthodologie Scrum pour la conception de notre futur système. Dans ce chapitre, nous allons définir les besoins de notre projet, concevoir la conception solution à développer, élaborer le backlog produit, et décrire l'environnement matériel et logiciel utilisé, ainsi que les outils et langages de programmation choisis. Nous présenterons également l'architecture du travail.

### II. Recueil des besoins

Cette partie se portera sur trois volets : l'identification des acteurs, leurs besoins fonctionnels d'une part et non fonctionnels d'autre part.

#### **Identification des acteurs**

Les acteurs qui interagissent avec notre système sont :

**L'administrateur** : Il est le principal responsable de l'administration de l'ensemble du site web. Ses responsabilités incluent :

- Ajouter, modifier, et supprimer des projets
- Consulter et voir les détails des projets
- Ajouter des todolist à tous les projets
- Gérer les membres de l'équipe et les clients
- Consulter les statistiques
- S'authentifier et se déconnecter
- Glisser et déposer les todolist
- Ajouter des documents

**Le membre de l'équipe** : il peut :

- Consulter, modifier, supprimer et voir les détails de tous les projets
- Ajouter des todolists à tous les projets
- Glisser et déposer les todolist
- Ajouter des documents

## Besoins fonctionnels

On va vous présenter les besoins fonctionnels par acteur :

### a) Besoins fonctionnels de l'acteur administrateur

- **S'authentifier** : Avant de réaliser toute action, l'administrateur doit se connecter avec un login et un mot de passe.
- **Gérer les membres de l'équipe** : L'administrateur peut ajouter, modifier et supprimer des membres de l'équipe ainsi que leur donner accès.
- **Gérer les clients** : L'administrateur peut ajouter, supprimer, modifier et consulter la liste des clients.
- **Gérer les projets** : L'administrateur peut ajouter, consulter, supprimer, modifier et voir les détails des projets.
- **Télécharger les documents** : L'administrateur peut télécharger les documents existants dans un projet.
- **Ajouter des documents** : L'administrateur peut ajouter des documents aux projets.
- **Ajouter des Todolist** : L'administrateur peut ajouter des todolists pour chaque projet.
- **Glisser et déposer des Todolist** : L'administrateur peut glisser et déposer des todolist pour chaque projet.
- **Consulter les statistiques** : L'administrateur peut consulter les statistiques.

### b) Besoins fonctionnels de l'acteur membre de l'équipe

- **S'authentifier** : Avant de réaliser toute action, le membre doit se connecter avec un login et un mot de passe.
- **Gérer les projets** : Le membre peut ajouter, consulter, supprimer, modifier et voir les détails des projets.
- **Ajouter des Todolists** : Le membre peut ajouter des todolists pour chaque projet.
- **Télécharger les documents** : Le membre peut télécharger les documents existants dans un projet.
- **Ajouter des documents** : Le membre peut ajouter des documents aux projets.

## Besoins non fonctionnels

Dans cette partie on va présenter les besoins de notre site web en termes de performances.

<b>Simplicité</b>	La plateforme doit être claire et facile à être utilisée.
<b>Sécurité</b>	Afin de sécuriser l'application, on doit bien gérer les profiles et les droits d'accès. D'autre part, il faut sécuriser l'accès aux données. D'où la nécessité d'un stockage fiable des données.
<b>Ergonomie</b>	L'application doit présenter une interface simple et conviviale pour que le collaborateur ait une manipulation aisée.

Tableau 1 : Les besoins non fonctionnel

### III. Diagramme de cas d'utilisation global

Nous présentons le diagramme de cas d'utilisation globale :

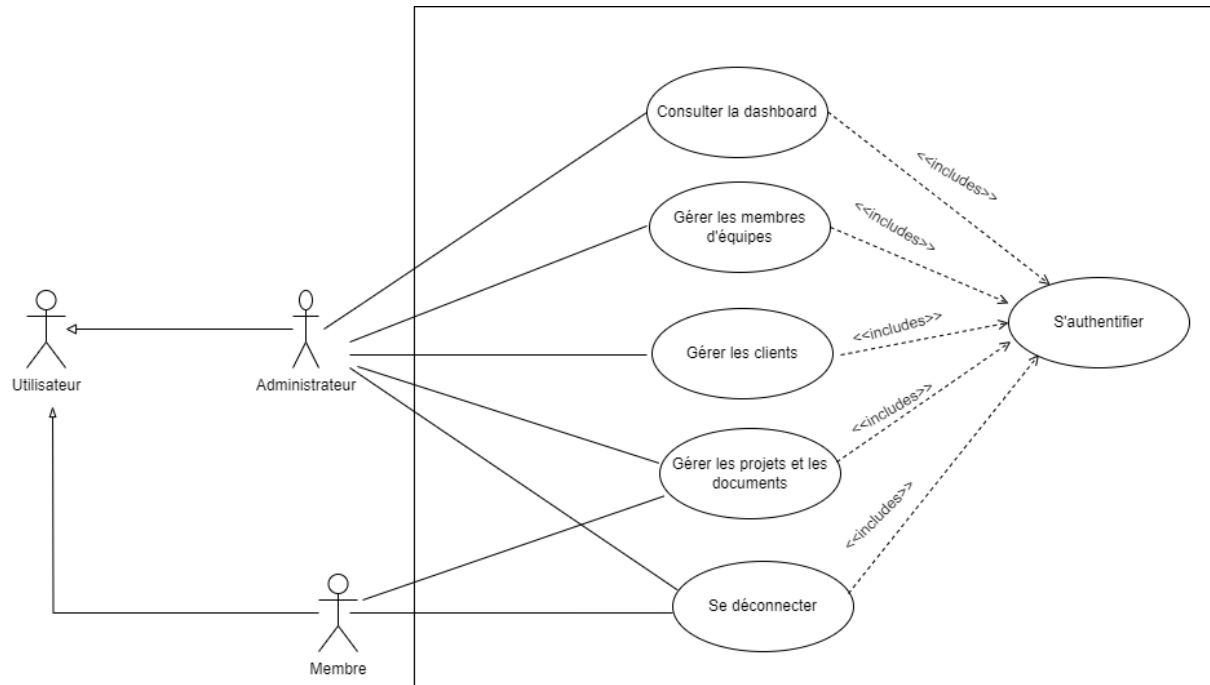


Figure 4 : Diagramme de cas d'utilisation global

### IV. Diagramme de classe

Afin de bien comprendre notre plateforme et les relations entre ses modules, nous allons commencer par le diagramme de classe de HiveShare, accompagné d'une description de la solution présentée dans la Figure 13.

La plateforme HiveShare se compose de quatre classes : Utilisateur, Projet, Client et Board.

- Un ou plusieurs clients peuvent appartenir à un ou plusieurs projets.
- Un membre de l'équipe ou un administrateur peut ajouter des listes de tâches (todolists) dans chaque projet.
- Un administrateur ou un ou plusieurs membres peuvent gérer un ou plusieurs projets.

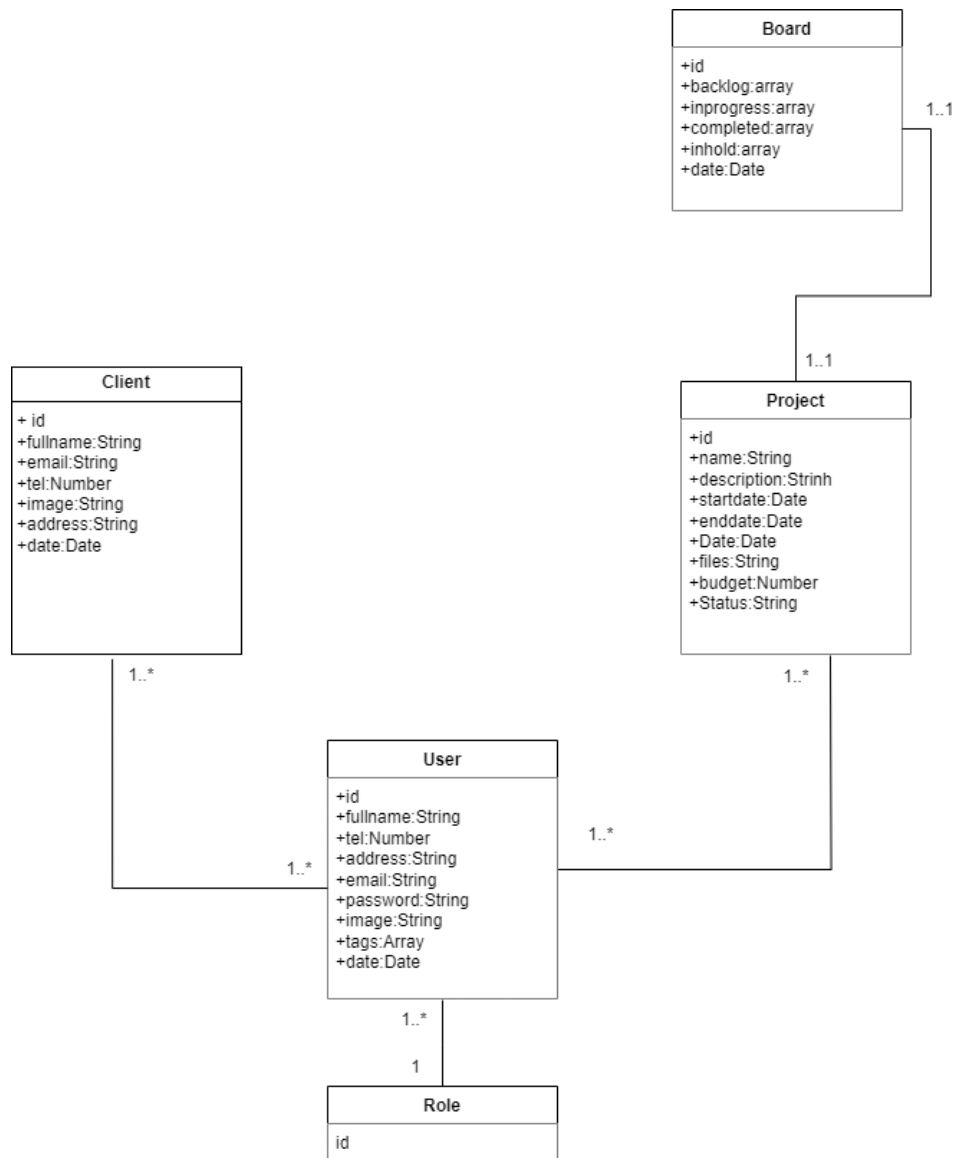


Figure 5 : Diagramme de classe

## v. Backlog du produit

Le backlog du produit présente une liste qui comprend les exigences initiales des clients appelées « user stories ». User story est défini par une priorité et la méthode MoSCoW est un outil pratique et simple pour les organiser et fixer

- M-Must Have : Les fonctionnalités indispensables.
- S-Should Have : Les fonctionnalités importantes mais non-essentiellles.
- C-Could Have : Les fonctionnalités à fournir si possible.
- W-Won't Have : Les fonctionnalités à ne pas fournir pour le moment.

En tant que	ID	User story	Priorité
-------------	----	------------	----------



Administrateur	1	En tant qu'administrateur, je dois m'authentifier pour avoir l'accès à la plateforme.	M
	2	En tant qu'administrateur, je peux ajouter un client.	M
	3	En tant qu'administrateur, je peux consulter la liste des clients.	S
	4	En tant qu'administrateur, je peux modifier un client.	M
	5	En tant qu'administrateur, je peux consulter la liste des projets.	S
	6	En tant qu'administrateur, je peux supprimer un client.	M
	7	En tant qu'administrateur, je peux ajouter un projet.	M
	8	En tant qu'administrateur, je peux modifier un projet.	M
	9	En tant qu'administrateur, je peux supprimer un projet.	M
	10	En tant qu'administrateur, je peux consulter la liste des projets.	S
	11	En tant qu'administrateur, je peux consulter les détails de chaque projet.	M
	12	En tant qu'administrateur, je peux ajouter dans le Todo list.	S
	13	En tant qu'administrateur, je peux gliser et déposer dans le Todo list.	S
	14	En tant qu'administrateur, je peux ajouter un membre d'équipe.	M

	15	En tant qu'administrateur, je peux modifier un membre d'équipe.	M
	16	En tant qu'administrateur, je peux supprimer un membre d'équipe.	M
	17	En tant qu'administrateur, je peux me déconnecter.	M
	18	En tant qu'administrateur, je peux ajouter des documents	M
	19	En tant qu'administrateur, je peux consulter les statistiques via le dashboard.	S
Membre de l'équipe	20	En tant que mebmre , je dois m'authentifier pour avoir l'accès à la plateforme.	M
	21	En tant que membre, je peux consulter un projet.	M
	22	En tant que membre, je peux ajouter un projet.	M
	23	En tant que membre, je peux modifier un projet.	M
	24	En tant que membre, je peux supprimer un projet.	M
	25	En tant que membre, je peux consulter les détails de chaque projet.	M
	26	En tant que membre, je peux ajouter des le Todo list.	S
	27	En tant que membre, je peux gliser et déposer dans le Todo list.	S
	28	En tant que membre, je peux ajouter des documents	M

	29	En tant que membre, je peux me déconnecter	S
--	----	--	---

Tableau 2 : Backlog “HiveShare”

Dans notre application, nous avons 4 sprints :

- Sprint 1 : Authentification et gestion de rôle
- Sprint 2 : Gestion clients, projets et membres
- Sprint 3 : Statistique
- Sprint 4 : Déploiement

## VI. Environnement du travail

Cette partie contiendra la configuration matérielle, ainsi que l’environnement logiciel utilisé pour le développement et la mise en place de notre site web.

### Environnement matériel



Pour la réalisation de ce projet, nous avons disposé de deux postes de travaux caractérisés comme suit :

Poste	ASUS Vivobook	LENOVO
<b>Processeur</b>	<b>13th Gen Intel(R) Core(TM) i7-13700H 2.40 GHz</b>	Intel(R) Core(TM) i5-4288U CPU @ 2.60GHz 2.21 GHz
<b>RAM</b>	<b>8,00 Go</b>	8,00 Go
<b>Disque Dur</b>	<b>1T</b>	1T
<b>Système d’exploitation</b>	Windows10 Professionnel 64-bits	Windows10 Professionnel 64-bits

Tableau 3 : Environnement matériel

### Environnement logiciel

Dans ce qui suit, nous avons énuméré les différentes technologies et les logiciels utilisées tout au long de ce projet pour l’étude et la mise en place de notre plateforme.

Logo	Description
	<b>CANVA</b> est utilisé pour la génération de logo de notre plateforme.
	<b>Figma</b> est un outil d'interface utilisateur en ligne gratuit pour la création, la collaboration, le prototypage et le transfert.

	Il est utilisé pour la conception de sites Web, d'applications, de pages de renvoi, etc.
	<b>Draw.io</b> est utilisé pour la conception et à la modélisation des nos diagrammes de classe, de séquence et cas d'utilisation.
	<b>Visual Studio Code</b> est un éditeur de code extensible développé par Microsoft. C'est un éditeur de code multiplateforme, open source et gratuit, supportant une dizaine de langages.
	<b>Postman</b> sert à tester et appeler nos web APIs, nous avons l'utilisé car il offre beaucoup de fonctionnalités avancées et propose une interface graphique claire qui permet la prise en main rapide de l'outil.
	<b>MongoDB Compass</b> est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma de base de données orienté.
	<b>Node JS</b> est un environnement de développement JavaScript coté serveur. C'était notre choix pour développer la partie backend.
	<b>Angular</b> est un Framework de développement coté client qui se base sur TypeScript. C'était notre choix pour développer la partie frontend.

Tableau 4 : Environnement logiciel

## vii. Architecture utilisée

Cette partie contiendra l'architecture technique et applicative utilisée pour la réalisation de notre projet

### Architecture technique

Pour décrire d'une manière symbolique et schématique les différents éléments du système, on a recours à une architecture 3 tiers qui garantit la stabilité et l'efficacité de notre plateforme. L'architecture de notre plateforme est partagée entre :

Un client léger sous forme de navigateur web qui va permettre de formuler des requêtes aux serveurs web. Serveur Web a pour rôle de gérer la communication entre le client et le serveur de base de données. Le serveur de base de données fournit les données au serveur web.

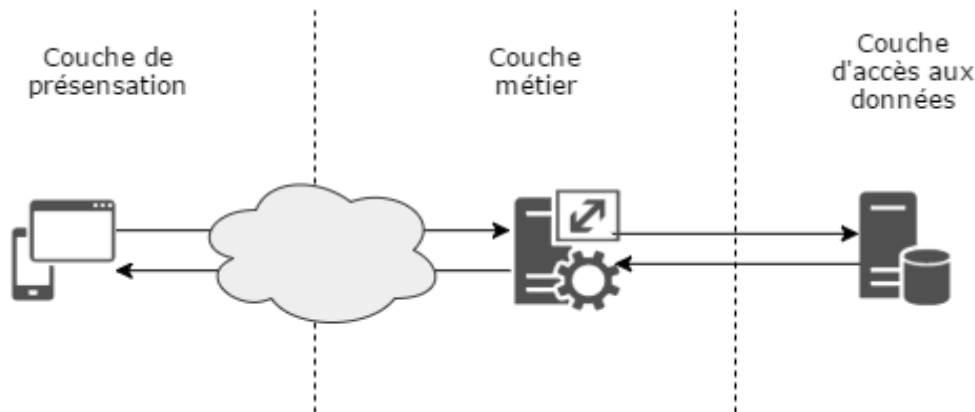


Figure 6 : Architecture technique

### Architecture applicative

L'architecture MVC c'est le meilleur choix pour que le code soit bien organisé, il nous permet de bien distinguer les fichiers et leurs rôles. Cette architecture sépare le code en trois parties :

- **Modèle** : c'est la partie en relation avec la base de données, elle récupère les données nécessaires les organise pour qu'il soit traitées après par le contrôleur
- **Vue** : Cette partie ne fait presque aucun traitement, c'est la partie responsable de l'affichage
- **Contrôleur** : c'est la partie intermédiaire entre le model et la vue. Principalement c'est la partie responsable aux traitements. Il analyse les données récupérées du modèle, prend les décisions et les envoie à la partie vue pour l'affichage.

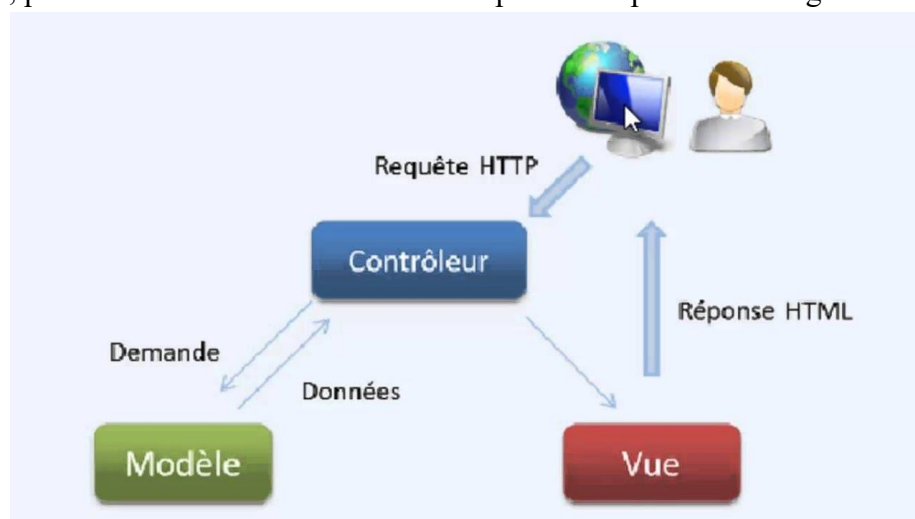


Figure 7 : Architecture applicative

## VIII. Conclusion

Dans ce chapitre, nous avons préparé notre plan de travail en capturant les besoins fonctionnels et non fonctionnels de notre application. Nous avons également identifié l'équipe de travail, établi le backlog du produit et planifié les sprints, puis nous avons arrêté nos choix techniques.

# Chapitre 3

## Réalisation de la solution

### I. Introduction

Ce chapitre se concentre sur la mise en œuvre pratique des quatre sprints précédemment abordés. Il détaille les résultats obtenus et les étapes franchies dans chacun des sprints, mettant en lumière les réalisations concrètes tout au long du processus.

### II. Sprint 1 : Authentification et gestion de rôle

Après avoir étudié les différentes tâches du sprint 1, nous avons établi une maquette pour l'interface d'authentification.

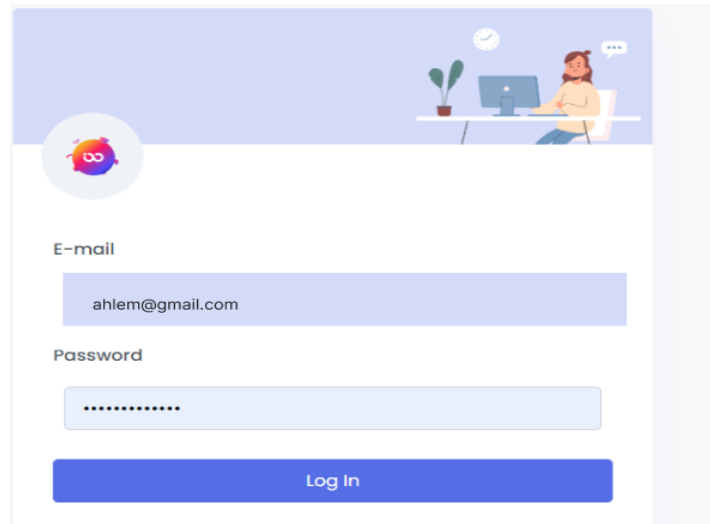


Figure 8 : maquette authentification

### Description textuelle du cas d'utilisation « Authentification »

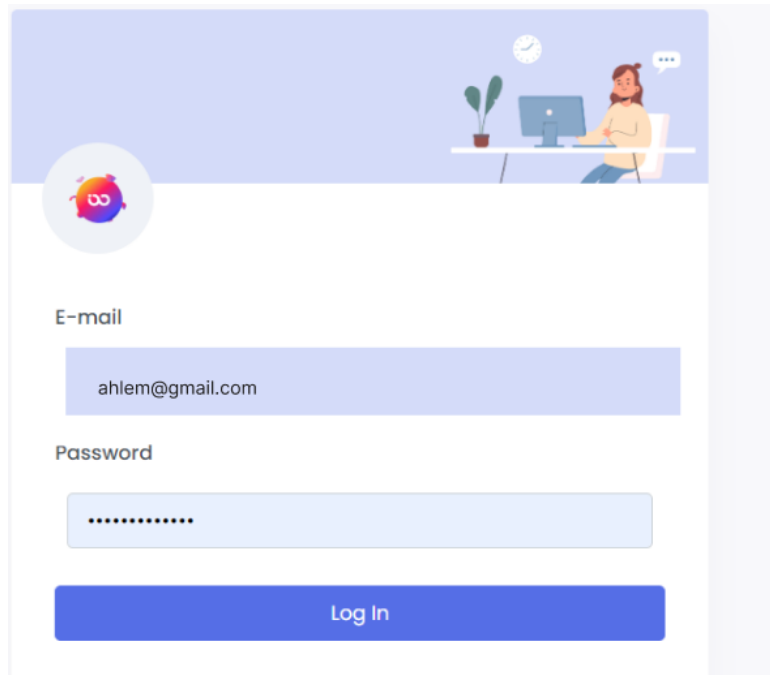
Le tableau ci-après illustre la description textuelle du cas d'utilisation « Se connecter à l'application »

Titre	Authentification
Acteur	Utilisateur de système (administrateur ou membre de l'équipe)
Pré-condition	L'utilisateur existe dans la base de données.
Scénario nominal	1) L'utilisateur demande l'accès à l'application via le navigateur Web.
	2) Il doit taper son email et son mot de passe puis il doit valider le formulaire.
	3) Le système vérifie les champs reçus. En cas de succès, il affecte le rôle nécessaire
	4) Le système redirige l'utilisateur vers l'interface principal
Scénario d'exception	L'utilisateur saisit des informations erronées. Il reçoit, dans ce cas, un message d'erreur.
Post-condition	L'utilisateur est connecté et il peut accéder à l'application.

Tableau 5 : Description textuelle du cas d'utilisation « Authentification »

### Interface d'authentification

Ci-dessous se trouve l'interface de connexion, où l'administrateur ou le membre de l'équipe remplit les champs correspondants à son identifiant et à son mot de passe.



The login interface features a header with a circular logo on the left and an illustration of a person working at a desk on the right. Below the header, there are two input fields: 'E-mail' containing 'ahlem@gmail.com' and 'Password' with masked characters. A blue 'Log In' button is positioned at the bottom of the form.

Figure 9 : Interface d'authentification

### III. Sprint 2 : Gestion de projet, client et membres

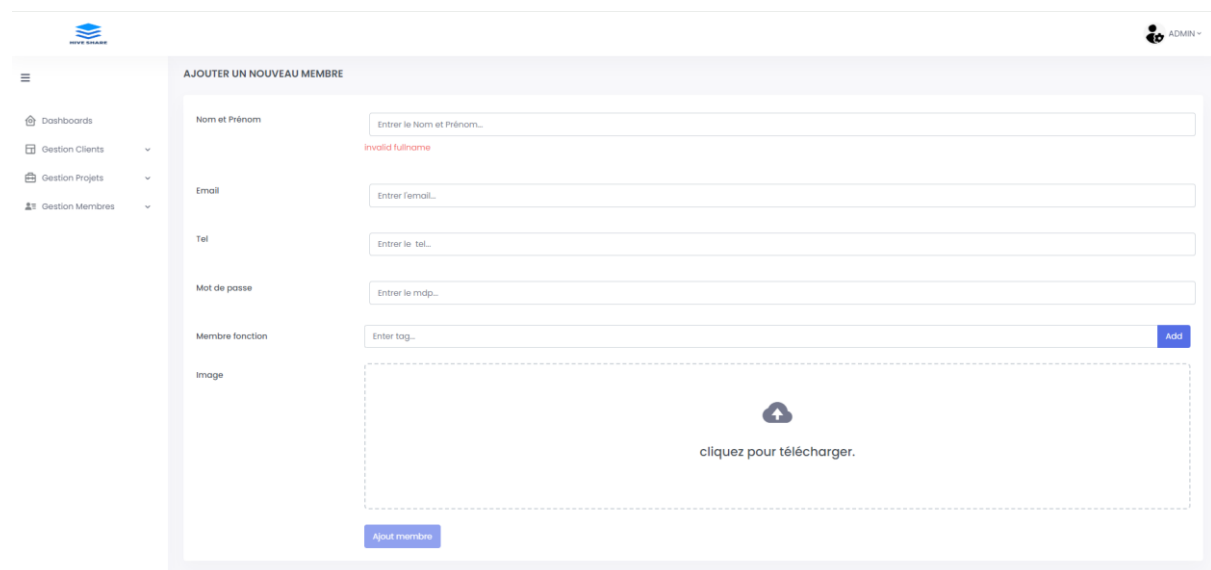
Dans ce sprint, nous allons présenter les opérations de base de notre plateforme, qui incluent la gestion des clients, des membres et des projets.

#### Gestion des membres

Après authentification l'administrateur a la possibilité de gérer les membres.

##### a) Ajout membre

Comme présenté dans la Figure 10 l'administrateur peut ajouter un membre qui travaille sur un ou plusieurs projets



The 'AJOUTER UN NOUVEAU MEMBRE' form is located within a dashboard sidebar. It includes the following fields: 'Nom et Prénom' (with a red 'invalid fullname' error), 'Email', 'Tel', 'Mot de passe', 'Membre fonction' (with an 'Add' button), and 'Image' (with a download icon and text 'cliquez pour télécharger'). A blue 'Ajout membre' button is at the bottom left of the form area.

Figure 10 : Interface ajout membre



### b) Diagramme de séquence <<Ajout membre >>

Le diagramme de la Figure 11 décrit le scénario de la fonctionnalité ajouter membre, le seul acteur responsable est l'administrateur qui peut ajouter un membre.

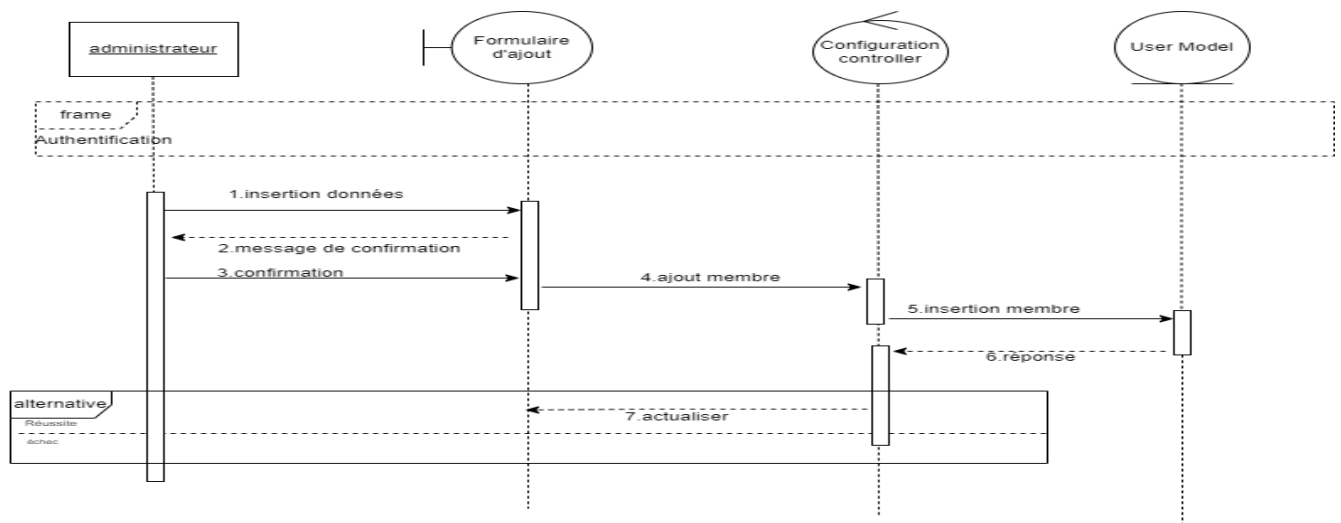


Figure 11 : Diagramme d'activité s'ajout d'un membre

### c) Modifier membre

Comme présenté dans la Figure 12 l'administrateur peut modifier les données un membre.

The form is titled 'MODIFIER LE MEMBRE'. It contains several input fields: 'Nom et Prénom' (with value 'arichi20'), 'Email' (with value 'testuser4@gmail.com'), 'Tel' (with value '12345678'), 'Mot de passe' (with placeholder 'Entrer mdp..'), and 'Membre fonction' (with placeholder 'Enter tag..'). Below the 'Membre fonction' field are two buttons labeled 'web' and 'design', and an 'Add' button. There is also an 'Image' section with a dashed box containing a cloud upload icon and the text 'cliquez pour télécharger.'. At the bottom left is a 'Modifier membre' button.

Figure 12 : Interface de modification d'un membre

### d) Supprimer membre

Comme présenté dans la Figure 13 l'administrateur peut supprimer un membre.

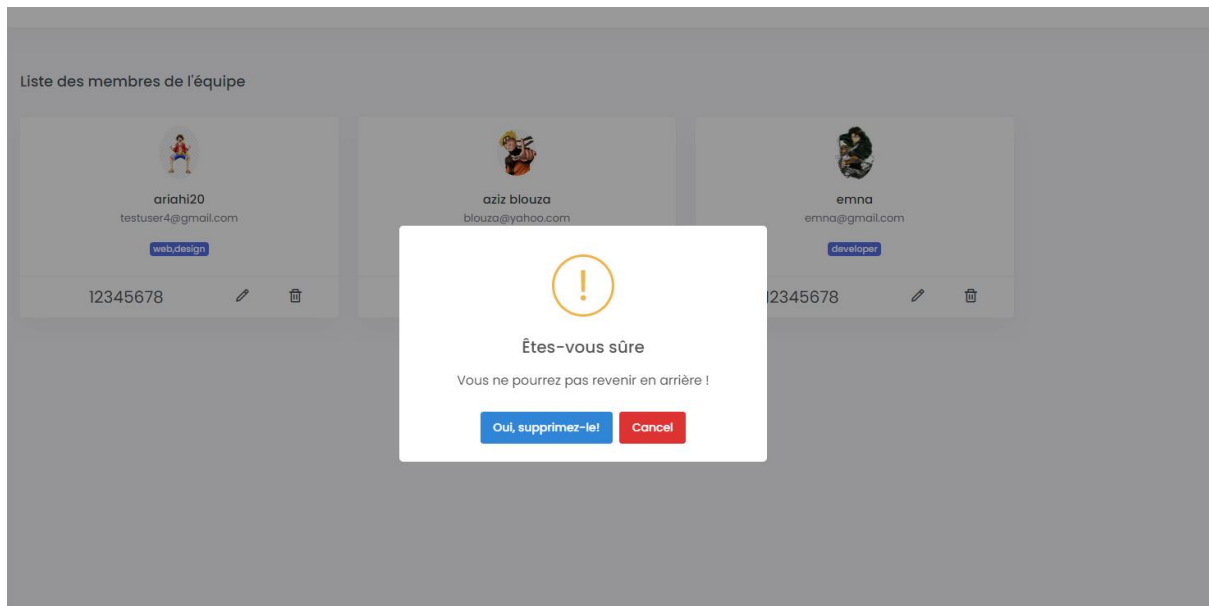


Figure13 : Interface de suppression d'un membre

### e) Consulter membre

Comme présenté dans la Figure 14 l'administrateur peut consulter la liste des membres

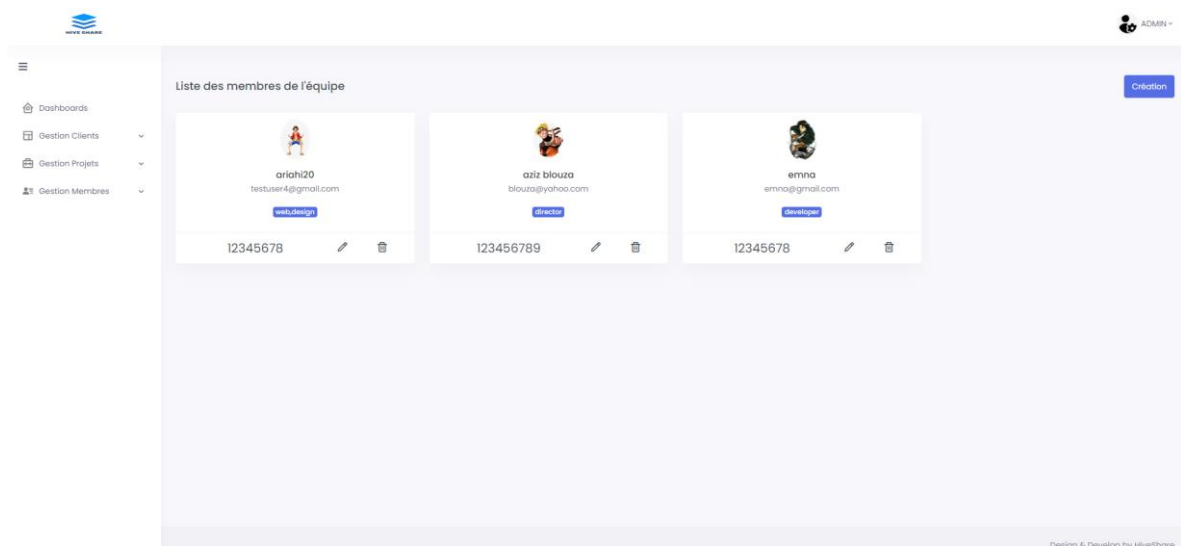


Figure 14 : Interface de la liste des membres

## Gestion des Clients

Après authentification l'administrateur a la possibilité de gérer les clients.

Ci-dessous on va présenter les fonctionnalités mentionnées précédemment.

### a) Ajout client

Comme présenté dans la Figure 15 l'administrateur peut ajouter un client.

CRÉER UN NOUVEAU CLIENT


Nom et Prénom

email

Tel

Adresse

Image 

  
Cliquez pour télécharger.

Design & Develop by HiveShare

Figure15 : Interface d'ajout d'un client

## b) Modifier client

Comme présenté dans la Figure 16 l'administrateur peut modifier les données d'un client.

MODIFIER CLIENT


Nom et Prénom

email

Tel

Adresse

Image 

  
Cliquez pour télécharger.

Design & Develop by HiveShare

Figure 16 : Interface de modification d'un client

### c) Supprimer client

Comme présenté dans la Figure 17 l'administrateur peut supprimer un client .

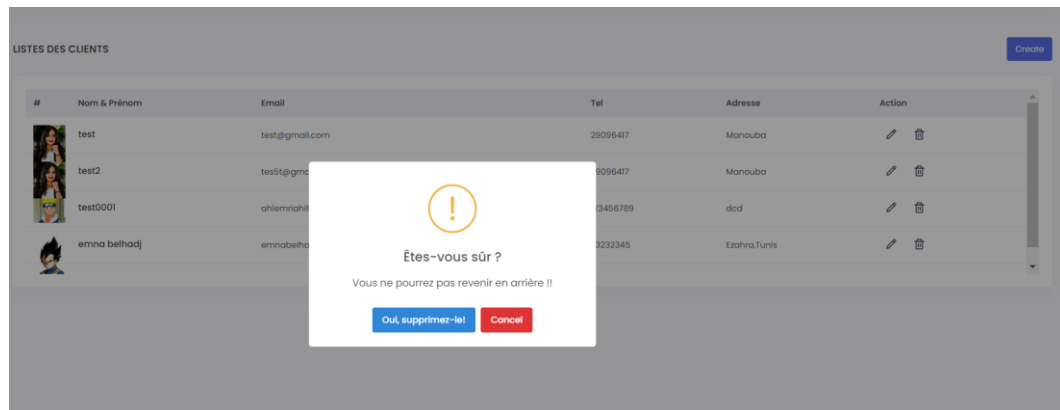


Figure 17 : Interface de suppression d'un client

### d) Consulter client

Comme présenté dans la Figure 18 l'administrateur peut consulter la liste des clients.

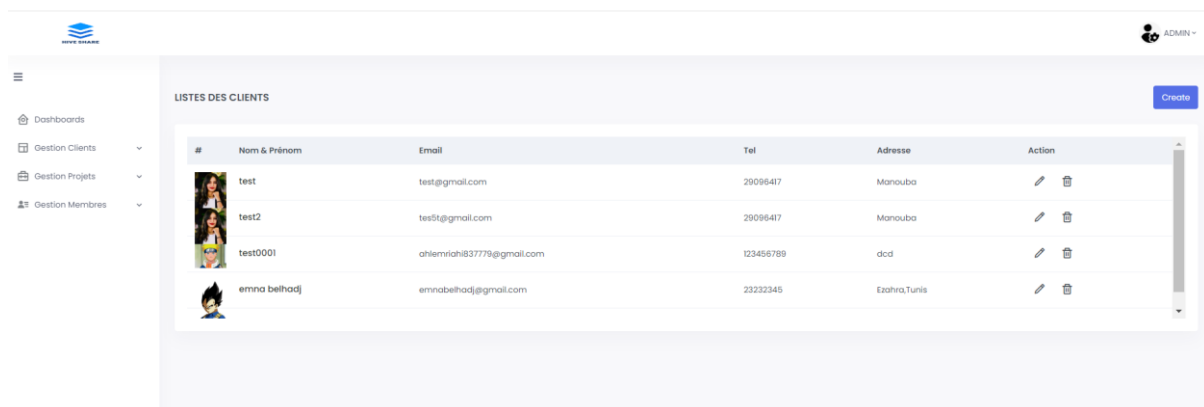


Figure 18 : Interface de la liste des clients

## 1) Gestion des projets et documents

Après authentification l'administrateur ou un membre d'équipe a la possibilité de gérer les projets ainsi les documents

Ci-dessous on va présenter les fonctionnalités mentionnées précédemment.

### a) Ajout du projet

Comme présenté dans la Figure 19 l'administrateur ou le membre peut ajouter un projet

Figure 19 : Interface de modification client

## b) Diagramme de séquence <<ajout projet>>

Le diagramme de la Figure 20 décrit le scénario de la fonctionnalité ajouter membre, l'administrateur ou le membre peut ajouter un projet.

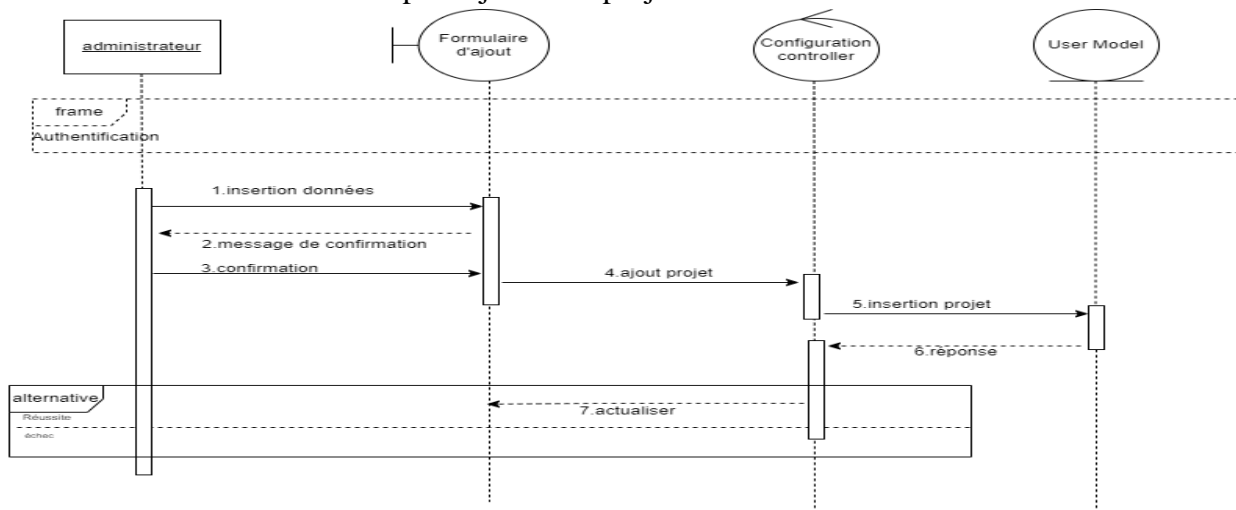


Figure 20 : diagramme d'activité d'ajout d'un projet

### c) Modifier projet et document

Comme présenté dans la Figure 21 l'administrateur ou le membre peut modifier les données d'un projet ou ajouter des documents.

Modifier le Projet

Nom de projet: angular

Description: dsasvdsd

Statut: terminé

Client: test000

Membres de l'équipe: J'ai réussi, Changer

Date de projet: 2024-06-09T00:00:00.000Z - 2024-07-07T00:00:00.000Z

Budget: 50

Pièces jointes: Cliquez pour télécharger.

Modifier Projet

Figure 21 : Interface de mise à jour d'un projet

### d) Supprimer projet

Comme présenté dans la Figure 22 l'administrateur ou le membre peut supprimer un projet.

#	Projet	Date début	Date fin	Statut	Équipes	Action
	angular dsasvdsd...	2024-06-09T00:00:00.000Z	2024-07-07T00:00:00.000Z	complet		
	Projetest dsasv...	2024-06-16T00:00:00.000Z	2024-06-08T00:00:00.000Z	en cours		
	dfabgl f...	2024-06-15T00:00:00.000Z		complet		
	angular qical...	2024-06-20T00:00:00.000Z		en cours		

Êtes-vous sûr ?

Vous ne pourrez pas revenir en arrière !!

Oui, supprimer-le! Cancel

Figure 22 : Interface de suppression d'un projet

### e) Consulter liste des projets

Comme présenté dans la Figure 23 l'administrateur ou le membre peut consulter la liste des projets .

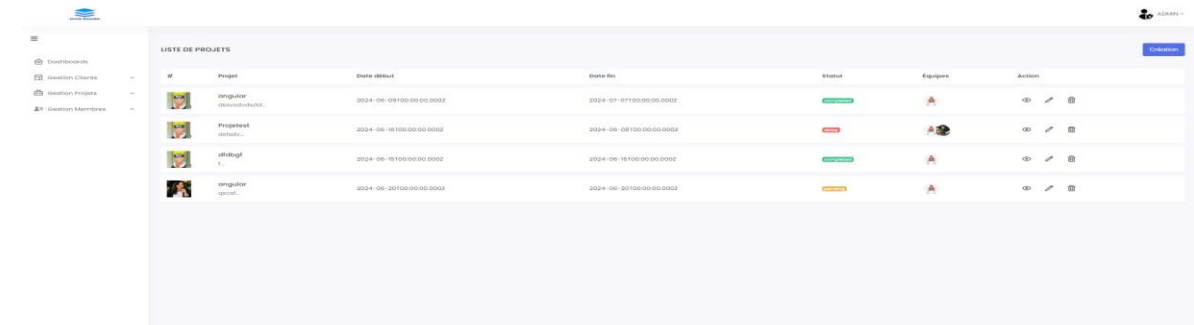


Figure 23 : Interface de liste des projets

#### f) Consulter les détails d'un projet

Comme présenté dans la Figure 24 l'administrateur ou le membre peut consulter les détails d'un projet .

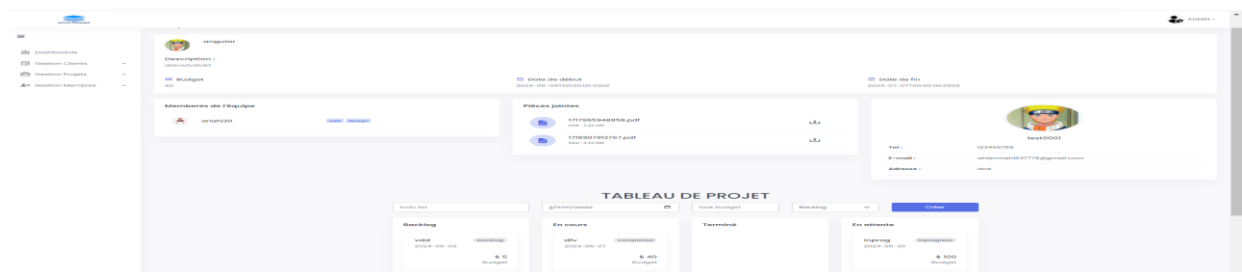


Figure 24 : Interface des détails d'un projet

#### g) Ajouter des tâches(todolist)

Comme présenté dans la Figure 25, Via l'interface détails du projet l'administrateur ou le membre peut ajouter des tâches (todolist) appartient à ce projet.

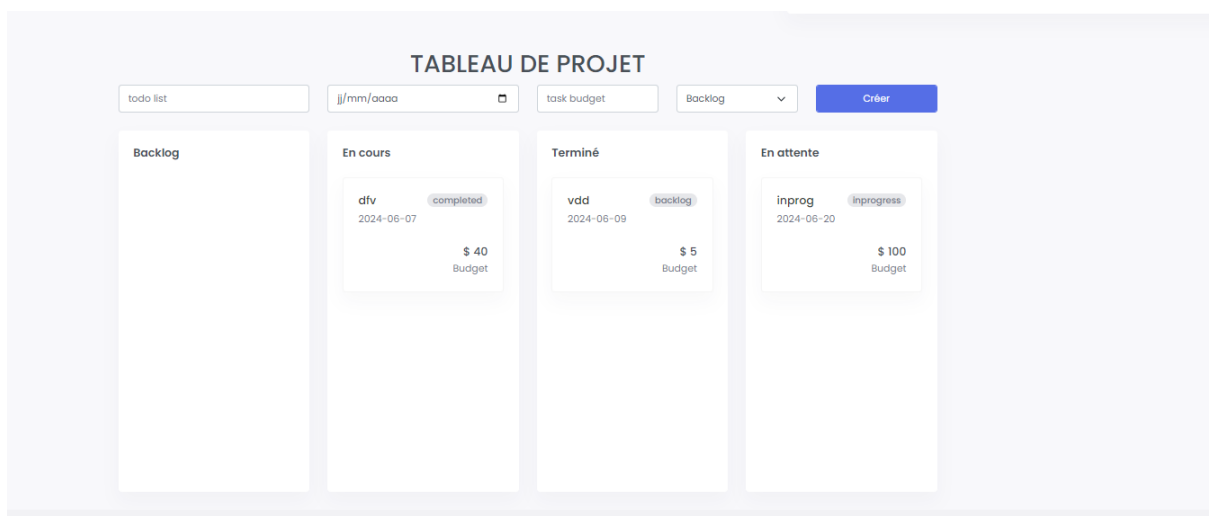


Figure 25 : Interface d'ajout d'une tâche (todolist)

#### h) Glisser et déposer les tâches

Comme présenté dans la Figure 26, Via l'interface détails du projet l'administrateur ou le membre peut glisser et déposer les tâches(todo list).

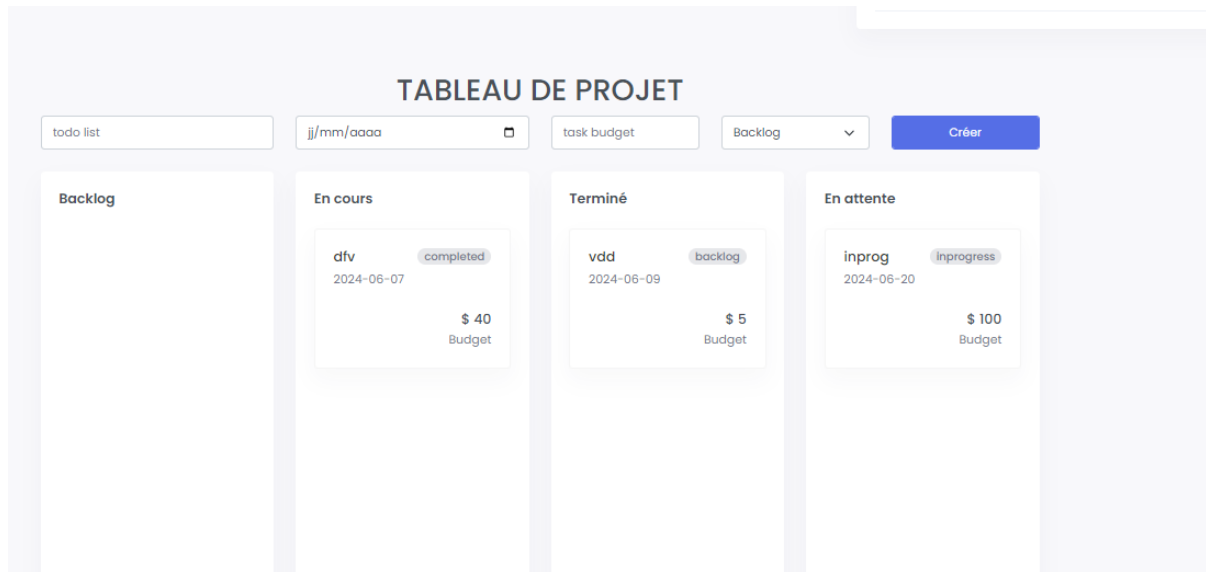


Figure 26 : Interface de glisser-déposer d'une liste de tâches (todolist)

#### IV. Sprint 3 : Statistique

Ce sprint se concentre sur les opérations supplémentaires de notre plateforme, notamment la mise en œuvre du tableau de bord (dashboard). Le dashboard fournit une vue d'ensemble des statistiques clés, incluant le nombre de clients, le nombre de projets et le nombre de membres d'équipe. Ces informations sont présentées de manière graphique, comme illustré dans l'image ci-dessous.

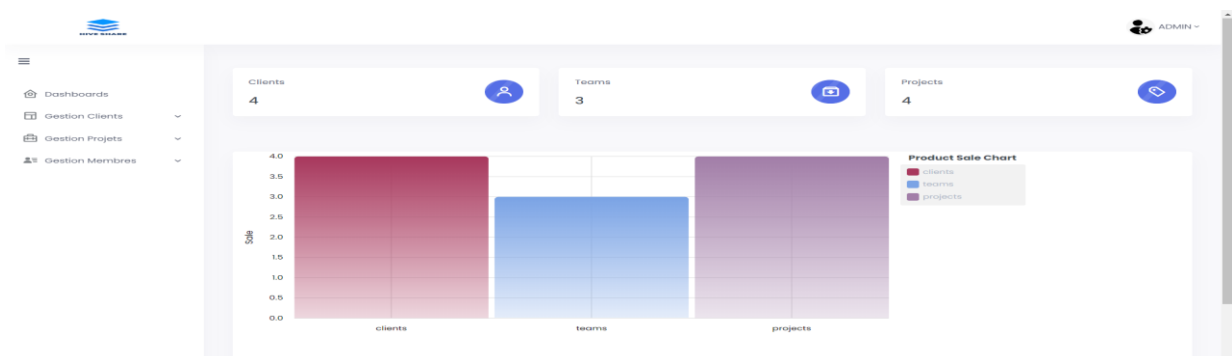


Figure 27 : Interface de Statistique



## Chapitre 4

### Déploiement de la plateforme

#### I. Introduction

Ce chapitre est dédié au processus de déploiement de notre plateforme, mettant en lumière l'utilisation de Netlify et la solution adoptée pour déployer le site. Nous examinerons en détail les étapes de configuration, les choix technologiques, ainsi que les considérations pratiques pour assurer un déploiement efficace et fiable de notre application web.

#### II. Présentation et Configuration de site Netlify

##### **C'est quoi netlify**

Netlify est une plateforme de déploiement et d'hébergement web conçue pour les applications statiques et les sites web modernes. Elle offre des fonctionnalités telles que le déploiement continu (continuous deployment), l'hébergement global sur CDN (Content Delivery Network), la gestion des domaines, ainsi que des outils intégrés pour la gestion des formulaires, le routage, et d'autres fonctionnalités de développement web. Netlify facilite le

déploiement et la gestion de sites web statiques, offrant une infrastructure robuste et scalable pour les développeurs et les équipes de projet.



Figure 28 : Logo de Netlify

## Configuration

- Après s'être connecté à Netlify, nous procédons au déploiement manuel de notre projet en glissant-déposant le dossier de notre projet.

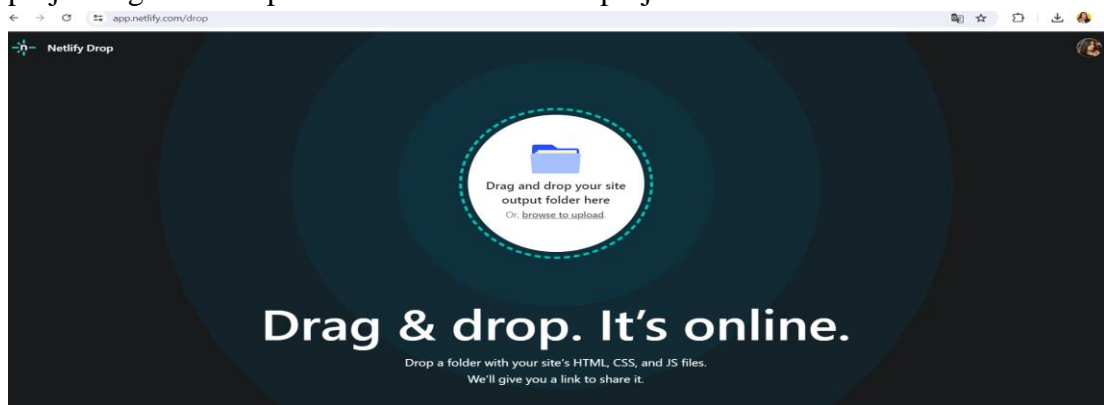


Figure 29 : Interface de glisser-déposer pour le projet

- Ensuite, via l'onglet "Détails du site", nous pouvons modifier le nom de notre site. Enfin, nous exécutons notre plateforme sur le site généré, accessible à l'adresse <https://hiveshare.netlify.app/>.

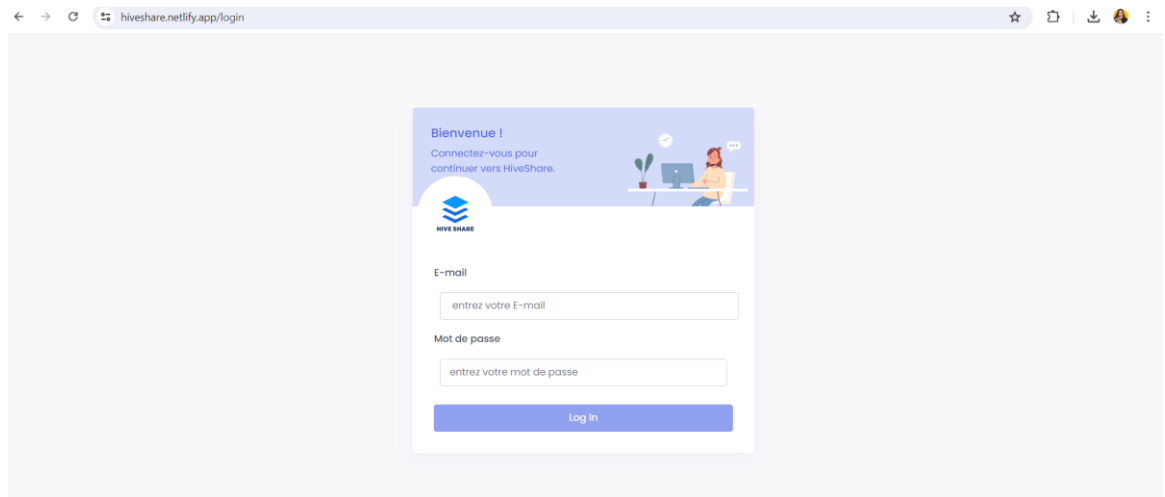


Figure 30 : Interface de plateforme héberger

### III. Conclusion

Dans ce chapitre, nous avons consacré du temps à réaliser le déploiement de notre site.

## Conclusion générale

Le travail présenté dans ce rapport a été réalisé dans le cadre du projet de PDL (Processus de Développement Logiciel).

L'objectif de ce rapport est d'introduire la plateforme que nous avons développée au cours de notre étude. Malgré les défis rencontrés, nous avons maintenu notre détermination et nos efforts tout au long du processus.

Nous avons surmonté de nombreux changements et obstacles techniques durant le développement de cette plateforme, enrichissant ainsi notre expérience grâce à des défis stimulants. Nous sommes parvenus à atteindre tous les objectifs que nous avions fixés.

Cependant, ce projet reste perfectible et ouvert à des améliorations continues. Nous envisageons d'introduire des mesures et des initiatives actives pour soutenir son évolution et son développement futur.

