



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université 20 Août 1955 - Skikda  
Facultés des sciences - Département d'informatique

**Mémoire pour l'obtention du diplôme  
de Master en  
Réseaux et Systèmes Distribués**

**Thème**

---

**Développement d'une application web  
pour la gestion du labo LICUS**

---

**Présenté par**

**Selma BOUCHAREB  
Najla SeLOURA**

**Supervisé par**

**D<sup>r</sup> Soufiane BOULEHOUACHE  
M<sup>r</sup> Mohamed Akram SBAGHDI**

Juin 2023

# Résumé

Dans ce mémoire, nous avons présenté le développement d'une application web pour la gestion du laboratoire de recherche LICUS. Cette application web que nous avons nommée LICUS est facile à utiliser et permet de répondre au besoin de chercheurs enseignants et doctorants du LICUS.

Ce travail a été conçu en utilisant le langage de modélisation UML. Pour ce qui est de la réalisation et de l'implémentation, notre choix s'est porté sur le Framework React qui est basé sur les langages HTML, CSS/Tailwind CSS, TypeScript et PostgreSQL pour la base de données.

# Abstract

In this memory, we presented a design and implementation of a web application for the management of the LICUS research laboratory . This web application that we have named LICUS is easy to use and makes it possible to meet the needs of LICUS teaching researchers and doctoral students. This work was designed using the UML modeling language. In terms of design and implementation, our choice fell on the Framwork React which is based on HTML, Css/Tailwind CSS, TypeScript and Postgresql languages for the database.

# Remerciements

Au terme de ce mémoire de fin d'étude, nous tenons à exprimer notre gratitude à Allah, le Tout-Puissant, qui nous a donné la force et la patience nécessaires pour mener à bien ce travail.

Nous remercions également notre encadreur **D<sup>r</sup> Soufiane BOULEHOUACHE** et notre co-encadrant **M<sup>r</sup> Mohamed Akram SBAGHDI** pour leur soutien, leur expertise et leurs conseils précieux tout au long de ce processus.

Nous sommes également reconnaissants envers nos parents et notre famille pour leur soutien inconditionnel et leur encouragement tout au long de notre parcours académique.

Enfin, nous tenons à remercier tous les professeurs et enseignants qui ont pensé à notre formation depuis notre premier cycle d'étude jusqu'à la fin de notre cycle universitaire. Leurs enseignements, leurs conseils et leur mentorat ont été essentiels pour nous aider à atteindre nos objectifs académiques et professionnels.

Nous sommes conscients que ce travail n'aurait pas été possible sans le soutien et l'aide de tous ceux qui nous ont entourés. nous sommes reconnaissants envers chacun d'entre eux et les remercions du fond du coeur.

# Dédicace

Je dédie ce modeste travail a touts personne ayant participé de loin ou de prés a sa réalisation.

A Mes parents Moukhtar, Fatiha

A très chère famille, mes frères Ilyes, Yesser, Saber

A mes soeurs Bouchra, Sabrina

A mon cher petit neveu Layth et sa maman samiha

A mes oncle et tentes.

A ma binôme Selma

A mes amis

A toute la promotion 2022/2023

Najla

# Dédicace

Je tiens à dédier ce modeste travail à tous ceux qui m'ont encouragé durant toute la période de réalisation de ce travail.

En particulier :

A mes chers parents "Ali" et "Cherifa" qui se sacrifient pour me voir réussir.

A mes chers frères et mes chères soeurs.

A ma binôme Najla.

A mes amies.

A tous mes amis de de promotion de 2<sup>ème</sup> année Master RSD en informatique

A tous personne qui occupe une place dans mon coeur

A tous les membres de ma famille et toute personne qui porte le nom BOUCHARB.

Selma

# Table des Matières

Résumé	i
Remerciements	iii
Dédicaces	iv
Dédicaces	v
Table des Matières	vi
Liste des Figures	ix
Liste des Tableaux	x
<b>1 Introduction</b>	<b>1</b>
<b>2 Evolution Et Technologies Web</b>	<b>3</b>
2.1 Préambule . . . . .	3
2.2 L'internet . . . . .	3
2.2.1 Définition . . . . .	3
2.2.2 Les différents protocoles d'internet . . . . .	4
2.2.3 Les services Internet . . . . .	5
2.2.4 Intranet-extranet . . . . .	6
2.3 Le web . . . . .	7
2.3.1 Définition . . . . .	7
2.3.2 Évolution du web . . . . .	7
2.4 Navigateur web . . . . .	9
2.4.1 Principaux navigateurs web du marché . . . . .	9
2.5 Serveur web . . . . .	9

2.6	Les site web . . . . .	10
2.6.1	Les différents types de sites web . . . . .	10
2.7	Les Application web . . . . .	10
2.7.1	Les types des application web . . . . .	11
2.7.2	La différence entre le site Web ou application Web . . . . .	12
2.7.3	Pourquoi choisit le développement application web . . . . .	12
2.7.4	Les avantages et Les inconvénients d'une application web : . . . . .	12
2.8	L'architecture client/serveur . . . . .	15
2.8.1	Types d'architecture client/serveur . . . . .	15
2.8.2	les avantages et les inconvénients de l'architecture . . . . .	17
2.9	Conclusion . . . . .	18
<b>3</b>	<b>Analyse Et Conception</b>	<b>19</b>
3.1	Préambule . . . . .	19
3.2	Spécification des besoins . . . . .	19
3.2.1	les besoins fonctionnels . . . . .	19
3.2.2	les besoins Non fonctionnels . . . . .	20
3.3	Analyse des besoins . . . . .	20
3.3.1	Diagrammes Cas d'utilisation . . . . .	20
3.3.2	Diagramme de cas d'utilisation de chercheur . . . . .	21
3.3.3	Diagramme de cas d'utilisation de Chef d'équipe . . . . .	22
3.3.4	Diagramme de cas d'utilisation Administrateur . . . . .	23
3.3.5	Diagramme de cas d'utilisation Anonyme . . . . .	23
3.3.6	Description textuelle des cas d'utilisation : . . . . .	24
3.3.7	Architecture logicielle . . . . .	29
3.3.8	Diagrammes d'activité . . . . .	29
3.3.9	Diagrammes de classes . . . . .	32
3.3.10	Diagrammes de déploiement . . . . .	33
3.4	Conclusion . . . . .	33
<b>4</b>	<b>Implémentation</b>	<b>34</b>
4.1	Préambule . . . . .	34
4.2	Les technologies utilisé : . . . . .	34
4.2.1	Le frontend . . . . .	34

4.2.2	Le backend . . . . .	37
4.2.3	La base de donnée . . . . .	38
4.3	outils de Développement . . . . .	38
4.3.1	Overleaf . . . . .	38
4.3.2	PlantUML . . . . .	39
4.3.3	Visual Studio code . . . . .	39
4.3.4	Prisma studio . . . . .	39
4.3.5	Docker . . . . .	39
4.3.6	StoryBook . . . . .	40
4.4	Présentation des interfaces . . . . .	40
4.4.1	Page d'accueil . . . . .	40
4.4.2	Liste des travaux . . . . .	41
4.4.3	Formulaire d'enregistrement . . . . .	42
4.4.4	Formulaire d'authentification . . . . .	42
4.4.5	Liste des Modèles Prisma . . . . .	43
4.4.6	Model Prisma - Labo . . . . .	43
4.4.7	Model Prisma - User . . . . .	44
4.4.8	Model Prisma - Team . . . . .	44
4.4.9	Model Prisma - Project . . . . .	45
4.4.10	Model Prisma - Work . . . . .	45
4.5	conclusion . . . . .	46
<b>5</b>	<b>Perspectives</b>	<b>47</b>
5.1	Préambule . . . . .	47
5.2	Conclusion . . . . .	48
<b>6</b>	<b>Conclusion Générale</b>	<b>49</b>
	<b>Bibliographie</b>	<b>50</b>

# Liste des Figures

2.1	une image qui exprime l'internet . . . . .	3
2.2	L'évolution du web . . . . .	8
2.3	Architecture 2-tier . . . . .	15
2.4	Architecture 3-tier . . . . .	16
2.5	Architecture n-tier . . . . .	17
3.1	Diagramme de cas d'utilisaion "chercheur" . . . . .	21
3.2	Diagramme de cas d'utilisation "chef d'équipe" . . . . .	22
3.3	Diagramme de cas d'utilisation "Administrateur" . . . . .	23
3.4	Diagramme de cas d'utilisation "Anonyme" . . . . .	24
3.5	Diagramme d'activité de "l'authentification" . . . . .	30
3.6	Diagramme d'activité "d'ajout" . . . . .	30
3.7	Diagramme d'activité de "Modification" . . . . .	31
3.8	Diagramme d'activité de "suppression" . . . . .	31
3.9	Diagramme de classe . . . . .	32
3.10	Diagramme de déploiement . . . . .	33
4.1	Page d'accueil . . . . .	41
4.2	Liste des travaux . . . . .	41
4.3	Formulaire d'enregistrement . . . . .	42
4.4	Formulaire d'authentification . . . . .	42
4.5	Liste des Modèles Prisma . . . . .	43
4.6	Model Prisma - Labo . . . . .	43
4.7	Model Prisma - User . . . . .	44
4.8	Model Prisma - Team . . . . .	44
4.9	Model Prisma - Project . . . . .	45
4.10	Model Prisma - Work . . . . .	45

# Liste des Tableaux

2.1	les avantages et les inconvénients de l'architecture . . . . .	17
3.1	Description du cas d'utilisation "s'authentifier". . . . .	25
3.2	Description du cas d'utilisation "gérer un membre". . . . .	26
3.3	Description du cas d'utilisation "gérer un équipe". . . . .	27
3.4	Description du cas d'utilisation "gérer un membre". . . . .	28

# Chapitre 1

## Introduction

Un laboratoire de recherche est une entité qui favorise la collaboration entre des chercheurs travaillant sur des problématiques similaires afin de concrétiser un ou plusieurs axes de recherche scientifique et de développement technologique. Ces laboratoires peuvent être établis dans des établissements d'enseignement supérieur, de formation ou d'autres organismes publics. Ils peuvent être exclusifs à un établissement, mixtes ou associés, en fonction de leur création dans le cadre d'une collaboration avec le secteur socio-économique ou d'une coopération scientifique inter-établissements.

L'université de Skikda regroupe un ensemble de laboratoires de recherche affiliés à des facultés relevant de plusieurs domaines de recherche (Informatique, Mathématique, Génie mécanique, Biologie, etc.). Parmi ces laboratoires, il y a le laboratoire d'informatique et de communication de l'université de Skikda LICUS.

Le laboratoire informatique et Communication de l'université de Skikda est une unité de recherche Créé en 2012, il destiné pour d'exécuter des activités de recherche scientifique et de développement technologique, il accueille quatre équipes de recherche et couvre plusieurs axes de l'informatique théoriques et pratiques : intelligence artificiel, modélisation, systèmes intelligents et Autonomes, sécurité .

L'objectif principal de notre travail est la conception et la réalisation d'une application web pour le Laboratoire LICUS rattaché à la département de l'informatique de skikda , Ce système de traitement des activités du LICUS vise a remédier aux insuffisances et aux manques considérables que représente le site web du laboratoire existant. un autre objectif de ce système est d'offrir des fonctionnalités aux différents acteurs du laboratoire , en particulier :

- Permettre aux membres chercheurs du LICUS d'avoir un espace dans l'application et la possibilité d'accéder pour saisir leurs informations, concernant les projets de recherche, publications ou autres cela permet de maintenir à jour la base de données.

- ▶ Permettre à l'administrateur d'avoir toutes les informations nécessaires pour la gestion des activités de recherche du LICUS dans les délais afin d'établir tous les documents nécessaires au bon fonctionnement du laboratoire.
- ▶ Assurer une visibilité de tous les travaux de recherche du LICUS .
- ▶ Établir des statiques concernant : l'évolution des activités de recherche du LICUS, l'évolution des effectifs par grade.
- ▶ Avoir une base de données à jour.

Ce présent travail sera structuré en quatre chapitres :

Le chapitre 1 : Nous évoquons l'évolution et les technologies du web.

Le chapitre 2 : Nous allons faire une conception détaillée de notre application en présentant les cas d'utilisation sous différents diagrammes.

Le chapitre 3 : Sera la partie de l'implémentation et la réalisation de notre application, ou nous présenterons l'environnement de développement et quelques interfaces de celle si.

Le chapitre 4 : Nous allons explorer les perspectives pour notre application .

Enfin, Nous clôturons ce mémoire par une conclusion générale.

# Chapitre 2

## Evolution Et Technologies Web

### 2.1 Préambule

Les applications web jouent un rôle essentiel dans notre vie moderne en nous permettant d'accéder à une multitude de services et d'informations de manière pratique et efficace, quel que soit l'appareil que nous utilisons. Dans ce chapitre, nous présentons l'internet, technologie du web, les applications web et l'architecture client/serveur.

### 2.2 L'internet

#### 2.2.1 Définition

Internet est un réseau informatique qui relie des ordinateurs du monde entier entre eux et qui leur permet d'échanger des informations. Les données sont transmises par l'intermédiaire de lignes téléphoniques, de câbles ou de satellites.

Pour communiquer entre eux, les ordinateurs connectés à internet utilisent un protocole de transmission et de communication constituant un langage commun.

Ce langage s'appelle la Transmission Control protocol / Internet Protocol (TCP-IP), voir Figure 2.1.



FIGURE 2.1 – une image qui exprime l'internet

## 2.2.2 Les différents protocoles d'internet

Un protocole est un ensemble de convention et de règles permettant la gestion de la manière avec laquelle s'effectue la communication entre plusieurs machines.[1]

### 1. les protocoles TCP/IP (Transmission Control Protocole/Internet Protocole)

Pour faire fonctionner un vaste réseau, Internet met en oeuvre des protocoles de communication. Les deux protocoles de base sont TCP (Transmission Control Protocol) et IP (Internet Protocol). Ils sont référencés sous le vocable TCP/IP.

TCP récupère les informations à transmettre, IP les transforme en paquets et les délivre à la couche chargée du transport (composant électronique). Comme ces protocoles sont faciles à mettre en place et permettent d'inter-connecter tous les types de réseaux, ils se sont rapidement développés, et la technologie TCP/IP a permis l'existence de l'internet mais aussi de nombreux inter-réseaux.

#### (a) Le protocole IP

Il est mis en oeuvre au niveau de la couche réseaux, sa tâche principale est de convertir le flux d'information transmis par TCP en paquets de données IP et les envoie vers une destination décrite par une adresse IP attribuer à chaque ordinateur.

- **L'adresse IP** est un nombre binaire codé sur 32bits et attribué à un hôte, elle est utilisée pour toute communication avec cet hôte, elle est toujours représentée dans une notation décimale constituée de quatre nombres, chacun compris entre [0-255] et séparés par un point.  
Les adresses IP sont réparties en fonction du nombre d'octets utilisés pour désigner le réseau.

#### (b) Le protocole TCP

TCP est le protocole de transport le plus important de la suite de TCP/IP, il offre aux programmes d'application un service de transport fiable de flots de données bidirectionnel avec contrôle de flux, Après avoir établi une connexion, les programmes d'application se servent de cette dernière pour s'échanger des données, et TCP garantit que ces données arrivent dans l'ordre et sans duplication.

### 2. HTTP (HyperText transport Protocol)

C'est un protocole léger et rapide, utilisé pour délivrer des fichiers multimédia et HyperText appelés plus généralement ressources, en utilisant internet, il gère la totalité des échanges réalisés entre le client et le serveur .pour chaque demande d'accès à l'URL, une requête http est émise indépendamment par le client.

### **3. DNS (Domaine Name service)**

C'est un ensemble de protocole permettant aux utilisateurs d'un réseau TCP/IP d'accéder aux hôtes à l'aide des noms conviviaux hiérarchisés, le serveur DNS assure la transformation des noms des domaines en adresse IP et inversement, car la structure des adresses IP est complexe à manipuler pour cela on utilise le DNS, sans celui-ci les utilisateurs devraient mémoriser les adresse IP.

## **2.2.3 Les services Internet**

L'internet offre plusieurs services qui sont relatifs aux différents protocoles de communication parmi ces services nous citons : [1]

### **1. Messagerie électronique E-mail (électronique mail)**

C'est un service qui offre la possibilité à ces utilisateurs de communiquer et de recevoir des messages texte, des images, des programmes, des sons, par des messages électroniques.

### **2. Le transfert de fichier : FTP (File Transfer Protocol)**

En se connectant sur Le serveur FTP qui nous donne l'accès à des fichiers, les copies, et les déplacer entre les différents réseaux connectés à l'internet, et le protocole FTP a été mis au point avant l'introduction du web.

### **3. Le service IRC : (Internet Relay Chat)**

"CHAT" est un terme en anglais qui signifie en français" bavarder "ainsi IRC permet le bavardage en directe sur Internet. Il permet l'échange de messages textuels en temps réel.

### **4. Les groupes de new (new group)**

C'est une messagerie électronique qui permet à toute personne de chercher de l'aide, faire la recherche sur un sujet bien défini ou d'envoyer un courrier électronique vers un espace ouvert à tout le monde.

## 5. La visioconférence

La visioconférence permet aux utilisateurs de communiquer en direct sur internet, avec échange de son et d'images, il est utilisé pour ramener l'école, travaille a la maison sans se déplacer.

## 6. Le World Wide Web : en français : Toile d'araignée mondiale

C'est le service le plus connu, le plus récent et maintenant le plus utilisé par le réseau Internet et naviguer entre des documents reliés par des liens hypertexte.

### 2.2.4 Intranet-extranet

Il existe deux types de réseau informatique : [1]

#### 1. Intranet

L'intranet est un réseau informatique utilisé à l'intérieur d'une entreprise ou de toute autre entité organisationnelle utilisant les techniques de communication d'internet (IP, serveurs http). dans les grandes entreprises, l'intranet fait l'objet d'une gouvernance particulière en raison de sa pénétration dans l'ensemble des rouages des organisations.

Un intranet repose généralement sur une architecture à trois niveaux composée : de clients (navigateur internet généralement), d'un ou plusieurs serveurs d'applications (middleware) : un serveur web permettant d'interpréter des scripts PHP, ASP, ou autre, et les traduire en requêtes SQL afin d'interroger une base de données et d'un serveur de bases de données. De cette façon, les machines clientes gèrent l'interface graphique, tandis que les différents serveurs manipulent les données. Le réseau permet de véhiculer les requêtes et les réponses entre clients et serveurs.

#### 2. Extranet

Un extranet est une extension de l'entreprise à des partenaires situés au-delà du réseau et ce de manière sécurisée (authentification par un mot de passe). De cette façon un extranet n'est ni un intranet ni un site web internet, il s'agit d'un système supplémentaire offrant par exemple aux clients, à ses partenaires ou à des filiales un accès privilégié à certaines ressources informatiques de l'entreprise par l'intermédiaire d'une interface web.

## **2.3 Le web**

### **2.3.1 Définition**

Le Web est un système d'information mondial qui permet l'accès à une multitude de ressources et de services à travers Internet. Le Web repose sur des technologies de communication, telles que le protocole HTTP et le langage HTML, qui permettent la création et la publication de pages web sur des serveurs accessibles à tous les utilisateurs connectés à Internet. Le Web permet aux utilisateurs de naviguer entre les pages et les sites web, de rechercher de l'information, de communiquer avec d'autres utilisateurs, de partager des contenus, d'effectuer des transactions commerciales, etc. [2]

### **2.3.2 Évolution du web**

Les différentes phases de cette évolution sont d'écrites comme suit ,voire Figure 2.2. [3]

#### **Web 1.0**

Le web traditionnel également, appelé web 1.0, a vu le jour dans les années 1990. Il s'agissait principalement d'un web statique axé sur la diffusion d'informations, principalement associé aux grandes entreprises. Les sites web étaient généralement orientés vers les produits et ne sollicitaient que peu la participation des utilisateurs, ressemblant souvent à une version en ligne des catalogues papier. Les premiers sites de commerce électronique ont été développés à cette époque. En ce temps là, le coût des programmes et logiciels propriétaires est énorme et l'explosion de la bulle dot.com, en 2000, remet en question cette approche de la toile.

#### **Web 2.0**

Le web 2.0, ou web social, change fondamentalement la donne. Il privilégie la dimension de partage et d'échange d'informations et de contenus (textes, vidéos, images ou autres). Il voit l'émergence des réseaux sociaux, des smartphones et des blogs. Le web se démocratise et se dynamise. C'est l'époque des forums, des wikis, etc. L'avis du consommateur est sollicité en permanence et il prend goût à cette socialisation virtuelle. Toutefois, la prolifération de contenus de qualité inégale engendre une infobésité difficile à contrôler.

## Web 3.0

Le web 3.0, aussi nommé web sémantique, vise à organiser la masse d'informations disponibles en fonction du contexte et des besoins de chaque utilisateur, en tenant compte de sa localisation, de ses préférences, etc. C'est un web qui tente de donner sens aux données. Les algorithmes règnent en maître. C'est aussi un web plus portable qui fait de plus en plus le lien entre monde réel et monde virtuel et répond aux besoins d'utilisateurs mobiles, toujours connectés à travers une multitude de supports et d'applications malines ou ludiques.

## Web 4.0

Le web 4.0, ou web intelligent, est l'évolution logique du web sémantique. Il effraie autant qu'il fascine, puisqu'il vise à immerger l'individu dans un environnement digital de plus en plus prégnant. Basé sur la communication sans fil reliant les personnes et les objets à tout moment et en tout lieu dans le monde physique ou virtuel en temps réel, le web 4.0 pousse à son paroxysme la voie de la personnalisation ouverte par le web 3.0. Mais il pose par la même occasion de nombreuses questions quant à la protection de la vie privée, au contrôle des données, etc .

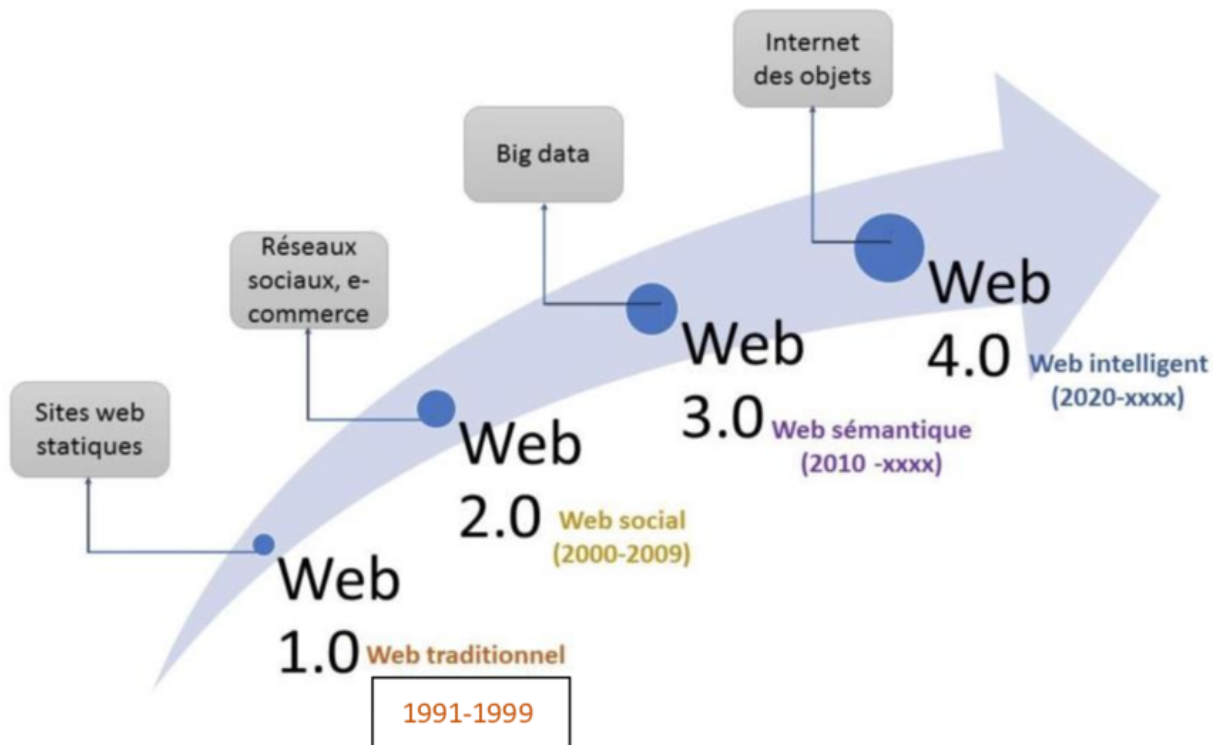


FIGURE 2.2 – L'évolution du web

## 2.4 Navigateur web

Un navigateur web, également appelé (web browser en anglais), est un programme informatique qui permet d'ouvrir et d'afficher une page web sur l'écran de l'utilisateur. Pour cela, il doit être capable de lire et d'interpréter les fichiers html. Les navigateurs accèdent au web grâce au protocole de communication HTTP, en agissant comme un client HTTP qui se connecte à un serveur. [4]

### 2.4.1 Principaux navigateurs web du marché

Les navigateurs web sont utilisés sur tous les appareils qui proposent un accès à Internet et qui offrent la possibilité de consulter des pages web : tablettes, smartphones, ordinateurs personnels. Les plus connus des navigateurs grand public sont : [4]

- ▶ Mozilla Firefox
- ▶ Google Chrome
- ▶ Internet Explorer (Microsoft)
- ▶ Safari (Apple)
- ▶ Opera
- ▶ Android Browser (spécifique aux mobiles Android) Certains d'entre eux proposent une variante mobile spécifique.

## 2.5 Serveur web

Un serveur web est un ordinateur connecté à Internet et sur lequel sont hébergés des sites web, composés de pages HTML (le serveur web, également appelé serveur HTTP, peut également être composé d'un groupe d'ordinateurs). Le logiciel fédérateur, sur un serveur web, est le serveur HTTP (Apache, le plus fréquemment), auquel viennent s'adjoindre un interpréteur de langage dynamique (PHP dans la plupart des cas), un gestionnaire de base de données (tel que postSQL) et d'autres programmes, comme un serveur SMTP.

La fonction d'un serveur web est de répondre aux requêtes des navigateurs Internet (Internet Explorer, Firefox, Chrome, Opéra, Safari, ...) .[5]

## 2.6 Les site web

Un site internet regroupe des pages web et des ressources liées par des liens hypertextes accessibles grâce à une adresse web appelée URL, et enregistrée sous un même nom de domaine. Il comprend des textes, des images et des vidéos, et est hébergé sur un serveur web qui permet d'y accéder via Internet ou un réseau intranet. Une page web est une collection de documents HTML organisée, stockée sur un serveur connecté à Internet, qui est principalement composée de texte et enrichie d'images, de vidéos, d'animations, parfois de sons et de liens hypertextes qui permettent de relier cette page à d'autres pages web.[6]

### 2.6.1 Les différents types de sites web

Il existe différents types de sites web, chaque type correspond à un objectif.

On peut citer :[7]

#### Site web statique

Un site web statique est présenté dans un navigateur web tel qu'il est stocké sur le serveur. Il se compose de pages web à contenu fixe qui sont codées en HTML et fournies sur un serveur web. Il ne subit aucun changement et reste identique, ou "statique", pour tous les visiteurs du site.

#### Site web dynamique

Un site web dynamique nécessite une programmation web et la conception d'une base de données. Un site web dynamique contient des informations et un contenu qui changent, en fonction de facteurs tels que le visiteur du site, l'heure de la journée, le fuseau horaire ou la langue maternelle du pays du visiteur). Le contenu de votre site (texte/images) est stocké sur une base de données ou un système de gestion de contenu. Lorsque l'information est mise à jour ou modifiée dans la base de données, elle change sur le site.

## 2.7 Les Application web

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur favori. La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs. Les technologies utilisées pour développer les applications web sont les mêmes que celles employées dans la création des sites internet. [8]

## 2.7.1 Les types des application web

Il existe cinq (05) types d'application web :[9]

### Application web statique

On désigne par application web statique, les applications web de base qui renferment peu de contenu et offrent peu de possibilités de mouvement. Ces applications web sont souvent conçues en utilisant du CSS et du HTML et peuvent supporter du contenu animé comme les GIFS et les vidéos. Elles se chargent sur le navigateur d'un utilisateur exactement comme elles sont enregistrées sur le serveur web. Ce sont des applications qui manquent de flexibilité et proposent peu ou pas d'interaction. Changer le contenu statique stocké sur les applications web statiques peut être délicat et nécessite d'effectuer de grands changements : télécharger le code HTML, le modifier sur le serveur, etc.

### Application web dynamique

Les applications web dynamiques constituent des outils qui, à l'inverse des applications web statiques, utilisent des bases de données pour charger leurs informations. Elles changent continuellement et peuvent fournir beaucoup d'informations en fonction de l'interaction avec l'utilisateur. Ce style d'application est techniquement plus compliqué que l'application statique et est mis en place pour faire en sorte que les bases de données ou les forums soient régulièrement capables de se mettre à jour ou de changer l'information disponible. Ceci est généralement possible grâce à l'utilisation d'un CMS ou d'un système de gestion de contenu.

### Applications e-commerce

Si vous vous êtes déjà procuré quelque chose en ligne, vous connaissez éventuellement les applications web les plus répandues, car elles font partie du e-commerce. Une application web qui prend la forme d'un magasin est considérée comme du e-commerce. Le processus de développement de cette application est plus compliqué que les deux précédemment mentionnés, car une application de e-commerce doit accepter des paiements électroniques par carte de crédit, PayPal ou autre mode de paiement. Il est donc conseillé de confier votre projet à des entreprises de développeurs web spécialisées comme Globalis qui se chargeront de réaliser convenablement votre projet d'application web.

### Application de portail web

Une application de portail web renvoie simplement à une application dans laquelle différentes sections ou catégories sont accessibles par l'intermédiaire d'une page d'accueil. Ce type de page se compose à la fois des forums, des chats, des courriels, des navigateurs et ainsi de suite.

## **Application web animée**

Un application web animée doivent obligatoirement être compatibles avec la technologie FLASH. Cette dernière peut être complexe pour quiconque n'ayant pas d'expérience avec ce type de technologie. Toutefois, si vous êtes à la recherche d'une application web relativement moderne offrant la possibilité d'ajouter du mouvement, il est préférable de travailler avec la technologie FLASH.

### **2.7.2 La différence entre le site Web ou application Web**

La principale différence entre une application Web et un site Web est qu'un site Web est une collection de pages universellement accessibles et imbriquées qui contiennent un seul nom de domaine. Alors qu'une application web est un programme ou un logiciel qui est accessible à l'aide de n'importe quel navigateur web. Son frontend est typiquement créé en déployant des langages de programmation comme HTML CSS Javascript qui sont renforcés par les principaux navigateurs.

Le backend, quant à lui, utilise des langages plus ingénieux tels que PHP et Python, et parfois JavaScript. Les systèmes de base de données, les configurations de serveur, les cadres de test et d'autres outils de développement sont également employés dans le développement des sites Web et des applications Web.[10]

### **2.7.3 Pourquoi choisit le développement application web**

Choisir le développement d'application web permet principalement de faciliter l'accès et l'usage des services d'une entreprise aux utilisateurs. En effet, l'application web s'active à travers un navigateur web dans le cloud. Les informations quelle contient sont stockées dans un serveur. De ce fait, l'utilisateur peut les avoir en se connectant à un ordinateur, un téléphone ou une tablette. Pour y parvenir, il doit simplement activer sa connexion, ouvrir l'application et insérer son identifiant ainsi que son mot de passe.

### **2.7.4 Les avantages et Les inconvénients d'une application web :**

#### **Les Avantage d'une application web**

Les applications web proposent de nombreux avantages sur différents aspects :[11]

#### **1. L'accessibilité**

Une application web est plus rapidement accessible qu'une application native. L'utilisateur n'a pas besoin de télécharger une application sur son appareil, il peut l'utiliser instantanément. Lorsqu'il souhaite interagir avec l'application, il lui suffit d'une connexion internet et de rentrer l'URL de celle-ci dans un navigateur web, qu'il soit sur son ordinateur, sa tablette ou son smartphone.

## **2. La compatibilité**

L'application web est conçue de manière à être compatible sur tous les systèmes d'exploitation et responsives sur tous les appareils. Étant donné que celle-ci fonctionne grâce à un navigateur, elle ne dépend ni de l'appareil utilisé, ni du système d'exploitation et elle est faite pour s'adapter à tous les formats d'écrans.

## **3. La mobilité**

La mobilité est améliorée grâce aux applications web. Où que vous soyez, votre application est accessible tant que vous possédez une connexion internet. Vous pouvez passer d'un ordinateur à l'autre, de votre ordinateur à votre smartphone, etc... Vos données sont stockées dans l'application, ce qui vous permet de reprendre votre activité à tout moment et de n'importe où !

## **4. Le travail en équipe**

Dans le cadre d'une entreprise, les applications web simplifient le travail d'équipe. Elles permettent de créer des espaces collaboratifs sur lesquels il est très facile de partager le travail avec une équipe entière. Toutes les données sont stockées en ligne et il suffit de partager un lien ou de créer un accès pour accéder aux espaces de collaborations. Plus besoin de travailler en local et d'envoyer les documents par mails à chaque modification !

## **5. Pas de contrainte des magasins d'applications**

Le développement d'une application web permet d'éviter certaines contraintes imposées par les magasins d'applications natives. Votre application étant hébergée en ligne, elle n'est pas soumise aux différentes règles de conformité ou à la vérification des magasins d'application comme Google Play Store ou l'App Store. Cela permet de gagner beaucoup de temps sur le développement.

## **6. Mise à jour et sécurité**

L'application web est toujours à jour. Contrairement aux applications natives qu'il faut mettre à jour manuellement, les applications web ont l'avantage d'être toujours à jour pour tout le monde. La mise à jour se fait côté serveur et non du côté de l'utilisateur, aucune action n'est donc requise par l'utilisateur. Cela permet de renforcer la sécurité. Lorsqu'une faille est détectée, elle est corrigée pour tous les utilisateurs en même temps.

## **7. Le stockage**

Plus de soucis à se faire concernant la limite de stockage de votre ordinateur ou de votre smartphone. Les applications web sont hébergées en ligne et vous n'avez pas de données à stocker directement sur votre appareil. Les applications peuvent aussi permettre d'enregistrer votre travail, vos documents, ou n'importe quels fichiers directement sur le cloud qui va proposer une capacité de stockage quasiment infinie.

### **Les inconvénients d'une application web**

Les applications web proposent de nombreux avantages très attrayants mais elles présentent aussi certains inconvénients que tous devez connaître avant de vous lancer dans le développement : [11]

#### **1. Dépendance à internet**

Une application web ne fonctionne que si vous avez accès à internet. Sans connexion, vous ne pourrez pas vous rendre sur votre navigateur et interagir avec la Web app.

Aujourd'hui, rares sont les situations où aucune connexion internet n'est disponible mais dans ce cas précis, l'application ne vous sera d'aucune utilité.

#### **2. La compatibilité**

Un des avantages de l'application web est sa compatibilité sous différents formats. Mais pour cela, il faut développer une application parfaitement responsive et compatible au niveau technique. C'est une tâche qui peut parfois compliquer le développement et demander beaucoup de temps.

#### **3. Moins de visibilité**

Une application web ne bénéficie pas de la visibilité des magasins d'applications. Celle-ci à la même visibilité qu'un site web classique sur lequel il faut travailler le référencement.

#### **4. Accès restreint à certaines fonctionnalités**

Une application web a un accès restreint aux fonctionnalités de l'appareil contrairement aux applications natives qui ont accès simplifié aux fonctionnalités matérielles comme le GPS, la caméra, le micro, etc.

## 2.8 L'architecture client/serveur

L'architecture client/serveur est un modèle de communication entre plusieurs ordinateurs, dans lequel un ordinateur client envoie des requêtes à un ordinateur serveur pour obtenir des informations ou des services. Le serveur répond aux requêtes en fournissant les données demandées. Ce modèle est utilisé pour la communication dans les réseaux informatiques, notamment pour l'accès à des bases de données et aux applications distribuées. L'architecture client/serveur permet une gestion centralisée des données et une répartition des tâches entre les clients et les serveurs, ce qui facilite la maintenance et l'évolution des systèmes informatiques.[12]

### 2.8.1 Types d'architecture client/serveur

Il y a plusieurs types d'architectures de client/serveur selon les applications et les données qui se trouvent sur le même ordinateur ou non : [13]

#### Présentation de l'architecture à 2 niveaux

L'architecture à deux niveaux (aussi appelée architecture 2-tier , signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service, voir figure 2.3.

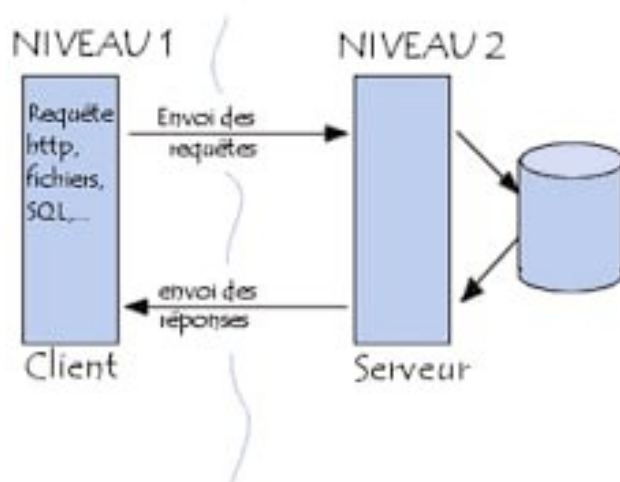


FIGURE 2.3 – Architecture 2-tier

## Présentation de l'architecture à 3 niveaux

Dans l'architecture à 3 niveaux (appelée architecture 3-tier), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre Un client : c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation. Le serveur d'application (appelé également middleware) : chargé de fournir la ressource mais faisant appel à un autre serveur. Le serveur de données : fournissant au serveur d'application les données dont il a besoin, voir figure 2.4.

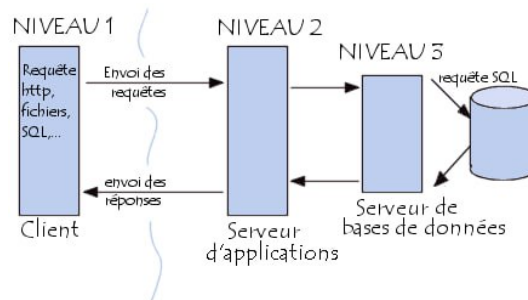


FIGURE 2.4 – Architecture 3-tier

## Comparaison des deux types d'architecture

L'architecture à deux niveaux est donc une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client.

Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont dé-localisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple).

L'architecture à trois niveaux permet :

Une plus grande flexibilité/souplesse .

Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau .

De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

## L'architecture multiniveaux

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) (Figure 3), effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à n niveaux..., voir figure 2.5.

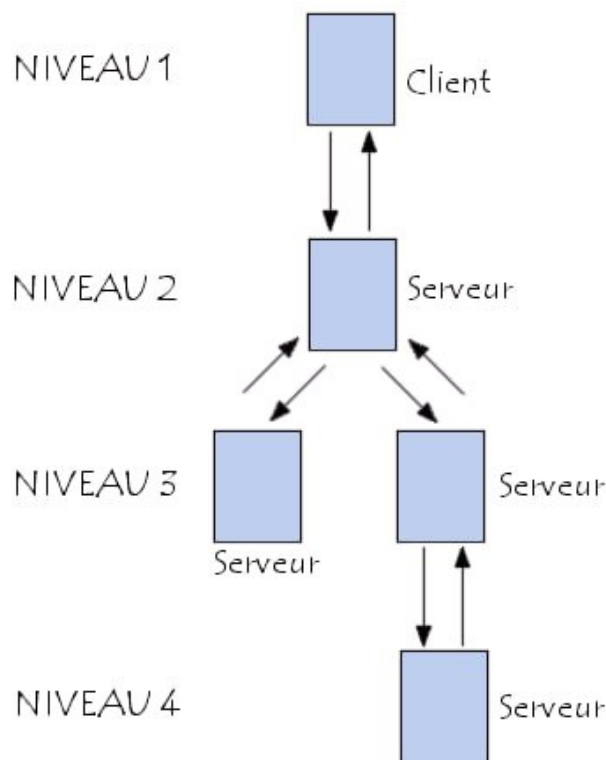


FIGURE 2.5 – Architecture n-tier

### 2.8.2 les avantages et les inconvénients de l'architecture

les avantages	les inconvénients
<p><b>Séparation des responsabilités</b> : l'architecture client/serveur permet de séparer les responsabilités entre le client et le serveur, ce qui facilite la maintenance et le développement des applications.[14]</p> <p><b>Meilleure sécurité</b> : en utilisant une architecture client/serveur, les données sont intégrées sur le serveur plutôt que sur le client, ce qui offre une meilleure sécurité et une meilleure protection contre les attaques malveillantes.[15]</p> <p><b>Gestion des ressources</b> : l'architecture client/serveur permet de mieux gérer les ressources informatiques en les répartissant entre le client et le serveur.[16]</p>	<p><b>Complexité</b> : l'architecture client/serveur peut être plus complexe à mettre en oeuvre que d'autres architectures, ce qui peut nécessiter une expertise technique plus élevée.[17]</p> <p><b>Coût</b> : l'utilisation d'une architecture client/serveur peut être déboursée en raison de l'achat et de la maintenance des serveurs, du matériel et des logiciels nécessaires.[15]</p> <p><b>Dépendance au réseau</b> : l'architecture client/serveur dépend fortement du réseau, ce qui peut entraîner des problèmes de performances ou d'indisponibilité en cas de problèmes de réseau.[18]</p>

TABLE 2.1 – les avantages et les inconvénients de l'architecture

## 2.9 Conclusion

A la fin de ce chapitre, nous avons identifié les différents concepts web qui nous semblent essentiels à la réalisation de notre projet. Nous décrivons ensuite les différents types d'applications web et leurs avantages et inconvénients, ainsi que l'architecture client/serveur.

# Chapitre 3

## Analyse Et Conception

### 3.1 Préambule

Dans le cycle de vie de notre projet, la spécification des exigences est indispensable pour donner une bonne description et analyse du système, ainsi que la conception qui représente une phase primordiale et déterminante pour produire une application de haute qualité.

Dans ce chapitre, nous allons identifier les différentes fonctionnalités que notre application doit offrir, cela en recensant les différents besoins fonctionnels et non fonctionnels, les acteurs qui interagissent avec le système. Ensuite, nous allons présenter les diagrammes de cas d'utilisation pour chaque entité, suivi de leurs diagrammes d'activité et enfin le diagramme de classes.

### 3.2 Spécification des besoins

Un besoin est ce qu'on attend du système faire, ou l'objectif qui doit être accomplis par le système.

Nous distinguons dans ce qui suit les besoins fonctionnels qui présentent les fonctionnalités attendues de notre application, et les besoins non fonctionnels qui permettent d'éviter le d'enveloppement d'une application non satisfaisante.

#### 3.2.1 les besoins fonctionnels

- ▶ S'authentifier pour permettre un accès contrôle et sécurise au système avec des privilèges propres a chaque membre.
- ▶ Gérer les membres du LICUS (ajouter, rechercher, modifier, supprimer un membre).
- ▶ Permettre aux membres de mettre a jour leurs informations personnelles et leurs travaux de recherche.

- ▶ Assurer la gestion des équipes (Ajouter/Modifier/Supprimer/Rechercher).
- ▶ Permettre a un visiteur de consulter les informations publiques du (profils des membres chercheurs, activités de chaque équipe et ses membres).
- ▶ Gérer les activités du LICUS (ajouter, rechercher, modifier, supprimer) par l'administrateur.
- ▶ Établira des statistiques des activités du LICUS.

### 3.2.2 les besoins Non fonctionnels

- ▶ L'application doit garantir un certain niveau de sécurité lors de l'accès à la base de données.
- ▶ le chercheur doit être identifié et authentifier ç travers son login et son mot de passe.
- ▶ Le système doit garantir la sécurité du transfert des données via le réseau de transfert du client à la machine serveur.
- ▶ Le système doit assurer la confidentialité des données personnelles.
- ▶ Offrir aux chercheurs un moyen de spécifier les droits d'accès à leurs documents chargé sur serveur.
- ▶ Limitation du droit de suppression des documents qu'à l'administrateur et le chercheur qui les a publiés.

## 3.3 Analyse des besoins

### 3.3.1 Diagrammes Cas d'utilisation

Le diagramme de cas d'utilisation dans UML est une sorte de diagramme qui utilise des acteurs et des cas d'utilisation pour décrire les fonctionnalités du système.

Les actions, services et fonctions que le système doit exécuter sont ce que vous avez appelé les " cas d'utilisation".

Alors que les entités qui opèrent sous des rôles spécifiques trouvés dans le système sont les "acteurs".

Et le "système" est celui qui est en cours de développement. Une pointe de flèche montre par ailleurs une relation de dépendance entre cas d'utilisation.[19]

Une distinction est faite entre ces deux stéréotypes :[20]

- ▶ **L'association d'inclusion "include"** : Le cas d'utilisation d'où part la ligne de connexion en pointillés inclut un deuxième cas d'utilisation, pointé par la pointe de flèche.
- ▶ **L'association d'extension "extend"** : Le cas d'utilisation d'où part la ligne de connexion en pointillés commence peut, sous certaines conditions, étendre le cas d'utilisation pointé par la tête de flèche. Mais ce n'est pas toujours le cas.

## Identification des acteurs

Il y a quatre différents acteurs qu'on peut trouver dans notre système, on les présente comme suit :

- ▶ **Visiteur** : C'est un utilisateur qui n'est pas connecté qui peut consulter les informations du laboratoire.
- ▶ **Chercheur** : Sont les chercheurs du laboratoire qui sont inscrits comme membres, il gère les travaux et peut gérer son profil .
- ▶ **Chercheur chef d'équipe** : Un membre comme tout autre chercheur, en plus ,Peut gérer son équipe.
- ▶ **Directeur de laboratoire** : Aussi un membre du laboratoire, mais il a une fonction en plus , il peut ajouter ou supprimer une équipe ainsi que l'ajout et la suppression d'un thème de recherche.
- ▶ **Administrateur** : Il dispose de toutes les autorisations sur l'application.

### 3.3.2 Diagramme de cas d'utilisation de chercheur

Le diagramme suivant représente les cas d'utilisateur associé le chercheur du laboratoire, voir Figure 3.1.

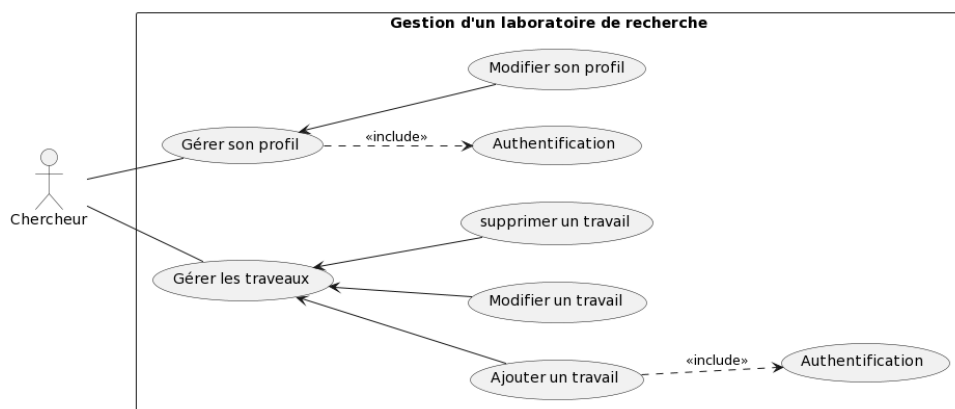


FIGURE 3.1 – Diagramme de cas d'utilisaion "chercheur"

### 3.3.3 Diagramme de cas d'utilisation de Chef d'équipe

Le diagramme suivant représente les différentes tâches réalisées par le chef d'équipe, voir Figure 3.2.



FIGURE 3.2 – Diagramme de cas d'utilisation "chef d'équipe"

### 3.3.4 Diagramme de cas d'utilisation Administrateur

Le diagramme ci-dessous représente les taches spécifiées a un administrateur, voir Figure 3.3 .



FIGURE 3.3 – Diagramme de cas d'utilisation "Administrateur"

### 3.3.5 Diagramme de cas d'utilisation Anonyme

Le diagramme suivante représente les taches spécifiées a un anonyme, voir Figure 3.4.

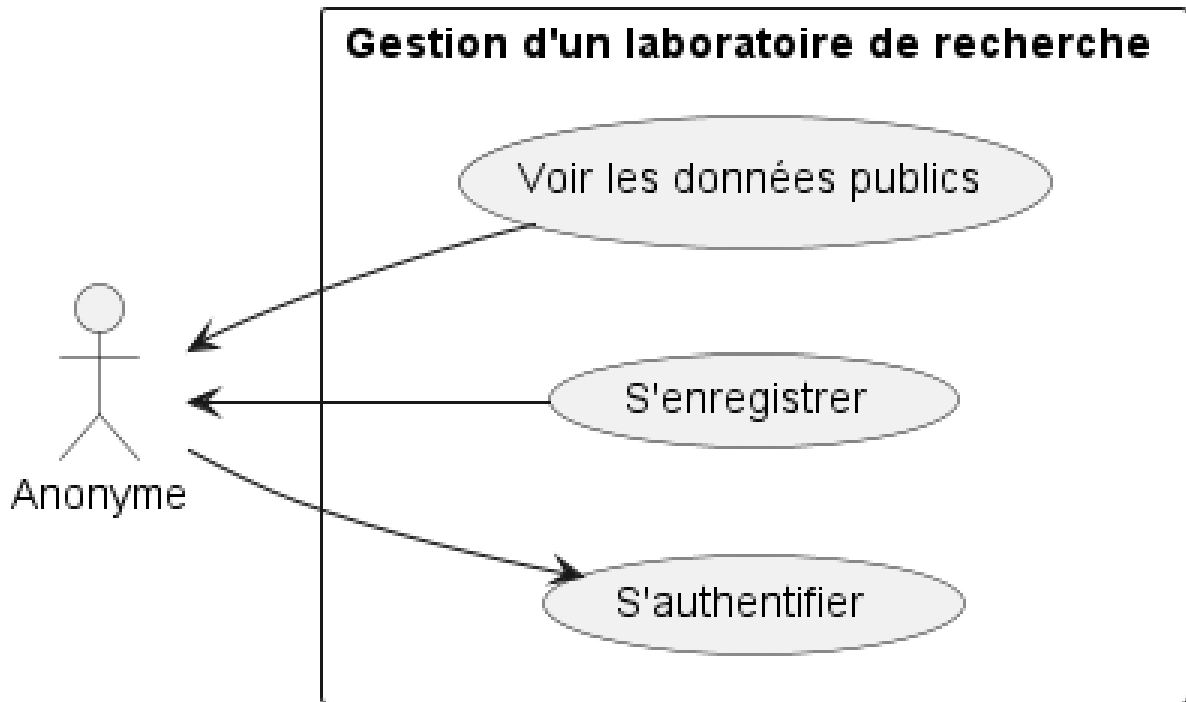


FIGURE 3.4 – Diagramme de cas d'utilisation "Anonyme"

### 3.3.6 Description textuelle des cas d'utilisation :

Chaque cas d'utilisation d'un système doit être défini textuellement, cela consiste à : [21]

**Identifier le cas** :Résumé de son objectif, les acteurs impliqués.

**Scénario nominal** : C'est le scénario "idéal" (tout se passe bien) pour le cas d'utilisation. on décrit un enchaînement d'actions (Acteur Système) qui conduisent au bon déroulement du cas d'utilisation. on parle également de scénario de base, scénario normal.

**Scénario alternatif** : Partant du scénario nominal, on étudie chaque point de l'enchaînement et s'il existe une variante, on propose "une nouvelle façon de dérouler le cas d'utilisation".

**Scénario d'exception** : Même description que pour le scénario alternatif sauf que les post-conditions du cas d'utilisation ne sont pas remplies.

## Cas d'utilisation "s'authentifier"

Cas d'utilisation	s'authentifier
Acteur	chercheur, Chef d'équipe, Administrateur
Objectif	Permet d'identifier l'utilisateur et de savoir son rôle dans l'application
Pré-condition	L'utilisateur doit être déjà inscrit
Scénario nominal	<ol style="list-style-type: none"><li>1. L'utilisateur demande de se connecter.</li><li>2. Le système affiche la fenêtre d'authentification .</li><li>3. L'utilisateur saisit son nom d'utilisateur et son mot de passe.</li><li>4. Le système approuvé si les informations d'identification sont correctes.</li><li>5. Le système donne accès a l'interface d'accueil correspondant a chaque utilisateur.</li></ol>
Scénario alternatif	Si les informations d'identification sont incorrectes, le système affiche un message d'erreur et retourne à l'étape 2.

TABLE 3.1 – Description du cas d'utilisation "s'authentifier".

## Cas d'utilisation "gérer les membres "

Cas d'utilisation	gérer un membre
Acteur	Chef d'équipe,Administrateur
Objectif	Pouvoir d'ajouter,modifier,supprimer des membres du laboratoire
Pré-condition	Authentification
Scénario nominal	<p><b>cas 01</b> :ajouter un membre</p> <ol style="list-style-type: none"> <li>1. L'utilisateur s'authentifie.</li> <li>2. L'utilisateur demande l'interface d'ajout d'un membre.</li> <li>3. Le système affiche la fenêtre d'ajout d'un membre.</li> <li>4. L'utilisateur saisit les données nécessaires et valide.</li> <li>5. Le système vérifie la validité des données saisies.</li> <li>6. Le système enregistre les donnés .</li> </ol> <p><b>cas 02</b> :modifier le profil d'un membre</p> <ol style="list-style-type: none"> <li>1. L'utilisateur s'authentifie.</li> <li>2. L'utilisateur demande l'interface de modification d'un membre.</li> <li>3. Le système affiche la fenêtre de modification d'un membre.</li> <li>4. L'utilisateur saisit les données a modifier et valide.</li> <li>5. Le système vérifie la validité des données saisies.</li> <li>6. Le système enregistre les données.</li> </ol> <p><b>cas 03</b> :supprimer un membre</p> <ol style="list-style-type: none"> <li>1. L'utilisateur choisit le membre a supprimer.</li> <li>2. Le système demande une confirmation.</li> <li>3. L'utilisateur confirme ou annule la suppression.</li> <li>4. Le système supprime le membre de la base de donnée.</li> </ol>
Scénario alternatif	<p><b>cas 01</b>:Si le nom d'utilisateur ou le mot de passe introduit sont incorrects,le système affiche un message d'erreur pour le champ erroné sur l'interface d'authentification et reprend au point (2) du scénario nominal.</p> <p><b>cas 02</b>:Si les données saisies sont invalides, le système affiche un message d'erreur et reprend au point (3) du scénario nominal.</p> <p><b>cas 03</b>:si membre inexistant ,le système affiche un message d'erreur.</p>

TABLE 3.2 – Description du cas d'utilisation "gérer un membre".

## Cas d'utilisation "gérer les équipes".

Cas d'utilisation	gérer un équipe
Acteur	Chef d'équipe,Administrateur
Objectif	Pouvoir d'ajouter,modifier,supprimer des equipes de recherche
Pré-condition	Authentification
Scénario nominal	<p><b>cas 01</b> :ajouter un équipe</p> <ol style="list-style-type: none"> <li>1. L'utilisateur s'authentifie.</li> <li>2. L'utilisateur demande l'interface d'ajout d'un équipe.</li> <li>3. Le système affiche la fenêtre d'ajout dun membre.</li> <li>4. L'utilisateur saisit les données nécessaires et valide.</li> <li>5. Le système vérifie la validité des données saisies.</li> <li>6. Le système enregistre les donnés .</li> </ol> <p><b>cas 02</b> :modifier un équipe</p> <ol style="list-style-type: none"> <li>1. L'utilisateur s'authentifie.</li> <li>2. L'utilisateur demande l'interface de modification d'un équipes.</li> <li>3. Le système affiche la fenêtre de modification d'un equipe.</li> <li>4. L'utilisateur saisit les données a modifier et valide.</li> <li>5. Le système vérifie la validité des données saisies.</li> <li>6. Le système enregistre les données.</li> </ol> <p><b>cas 03</b> :supprimer un équipe</p> <ol style="list-style-type: none"> <li>1. L'utilisateur choisit l'équipe a supprimer.</li> <li>2. Le système demande une confirmation.</li> <li>3. L'utilisateur confirme ou annule la suppression.</li> <li>4. Le système supprime le équipe de la base de donnée.</li> </ol>
Scénario alternatif	<p><b>cas 01</b>:Si les données introduit sont incorrects,le système affiche un message d'erreur pour le champ erroné sur l'interface d'authentification et reprend au point (2) du scénario nominal.</p> <p><b>cas 02</b>:Si les données saisies sont invalides, le système affiche un message d'erreur et reprend au point (3) du scénario nominal.</p> <p><b>cas 03</b>:Si équipe inexistant , le système affiche un message d'erreur.</p>

TABLE 3.3 – Description du cas d'utilisation "gérer un équipe".

## Cas d'utilisation "gérer les projet de recherche".

Cas d'utilisation	gérer un projet de recherche
Acteur	Chef d'équipe,Administrateur
Objectif	Pouvoir d'ajouter,modifier,supprimer des projet de recherche
Pré-condition	Authentification
Scénario nominal	<p><b>cas 01</b> :ajouter un projet de recherche</p> <ol style="list-style-type: none"> <li>1. L'utilisateur s'authentifie.</li> <li>2. L'utilisateur demande l'interface d'ajout d'un projet de recherche.</li> <li>3. Le système affiche la fenêtre d'ajout dun projet de recherche.</li> <li>4. L'utilisateur saisit les données nécessaires et valide.</li> <li>5. Le système vérifie la validité des données saisies.</li> <li>6. Le système enregistre les donnés .</li> </ol> <p><b>cas 02</b> :modifier le projet de recherche</p> <ol style="list-style-type: none"> <li>1. L'utilisateur s'authentifie.</li> <li>2. L'utilisateur demande l'interface de modification d'un projet de recherche.</li> <li>3. Le système affiche la fenêtre de modification d'un projet de recherche.</li> <li>4. L'utilisateur saisit les données a modifier et valide.</li> <li>5. Le système vérifie la validité des données saisies.</li> <li>6. Le système enregistre les données.</li> </ol> <p><b>cas 03</b> :supprimer un projet de recherche</p> <ol style="list-style-type: none"> <li>1. L'utilisateur choisit le projet a supprimer.</li> <li>2. Le systeme demande une confirmation.</li> <li>3. L'utilisateur confirme ou annule la suppression.</li> <li>4. Le système supprime le projet de la base de donnée.</li> </ol>
Scénario alternatif	<p><b>cas 01</b>:Si les donnees saisies sont incorrects,le système affiche un message d'erreur pour le champ erroné sur l'interface d'authentification et reprend au point (2) du scénario nominal.</p> <p><b>cas 02</b>:Si les données saisies sont invalides, le système affiche un message d'erreur et reprend au point (3) du scénario nominal.</p> <p><b>cas 03</b>:si le projet inexistant ,le système affiche un message d'erreur.</p>

TABLE 3.4 – Description du cas d'utilisation "gérer un membre".

### 3.3.7 Architecture logicielle

Une architecture est un modèle générique et conceptuel qui se rapporte à un sujet et qui représente la fonctionnalité, la structure, le positionnement, l'interrelation des différents types d'éléments (hardware, logiciels, infrastructure) qui la composent :

#### **La couche présentation (frontend)**

Cette couche est fonctionnelle sur le navigateur de l'utilisateur et est créée avec les technologies suivantes : Typescript avec JSX, React, TailwindCSS, ReactRouter et MobX.

Cette couche est responsable de la présentation de l'interface utilisateur de l'application. Elle communique avec la couche de service pour récupérer les données et les afficher à l'utilisateur.

#### **La couche de service**

Cette couche est responsable de la logique métier de l'application. Elle est exécutée sur le serveur et est créée avec Typescript sans JSX, nodejs, next js et Prisma (ORM).

Cette couche communique avec la couche de présentation pour récupérer les données et effectuer des opérations sur ces données. Elle communique également avec la couche de persistance pour accéder aux données étendues dans la base de données.

#### **La couche de persistance**

Cette couche est responsable de l'accès aux données prolongées dans la base de données. Elle est créée avec Postgresql et communiquée avec la couche de service pour récupérer et stocker les données.

### 3.3.8 Diagrammes d'activité

Ce type de diagramme présente une présentation visuelle d'une série d'actions ou d'un flux de contrôle dans un certain système qui est identique à un organigramme ou un diagramme de flux de données.

Le diagramme d'activité a généralement un début ou l'état initial et une fin qui est l'état final .[19]

#### **Diagramme d'activité de l'authentification**

Le diagramme d'activité d'authentification nous permet de voir les comportements internes du système, lors du démarrage de l'application par l'utilisateur, le système lui affiche le formulaire d'authentification, après que le mot de passe soit saisi le système vérifie sa validité et affiche la page d'accueil sinon il affiche un message d'erreur, voir Figure 3.5.

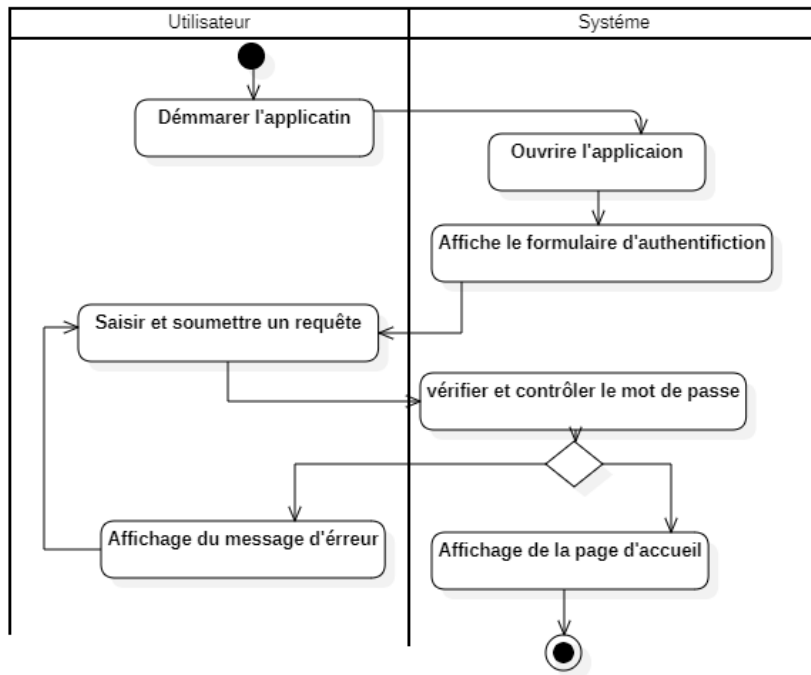


FIGURE 3.5 – Diagramme d'activité de "l'authentification"

### Diagramme d'activité d'ajout

Le diagramme d'activité d'ajout ,voire Figure 3.6.

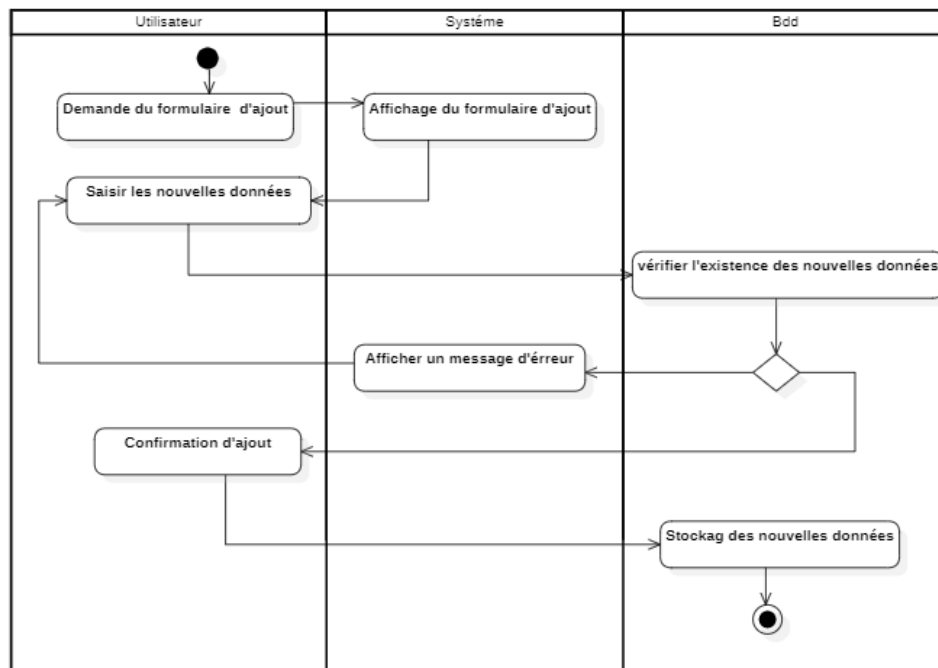


FIGURE 3.6 – Diagramme d'activité "d'ajout"

## Diagramme d'activité de modification

Le diagramme d'activité de modification,voire Figure 3.7 .

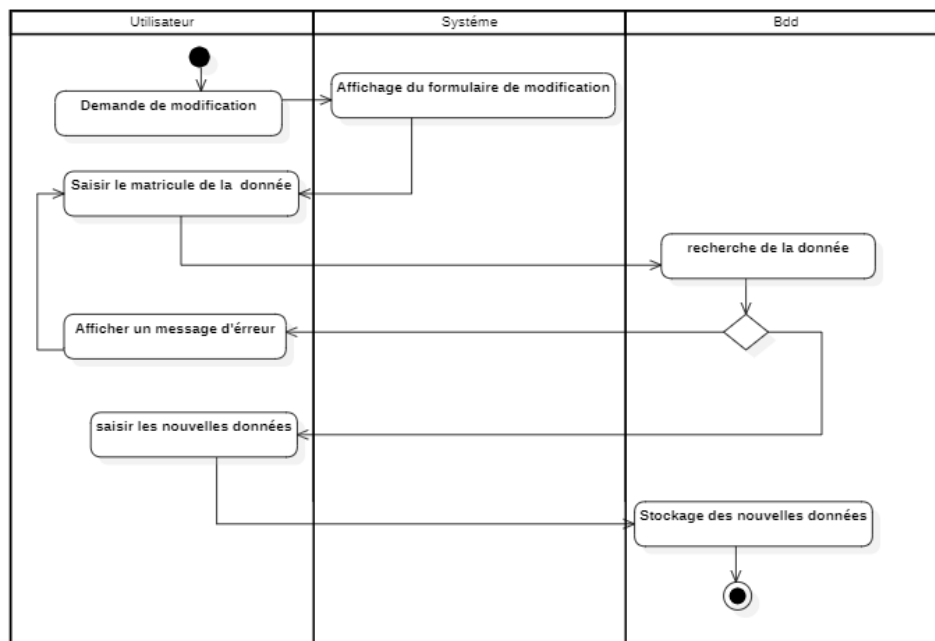


FIGURE 3.7 – Diagramme d'activité de "Modification"

## Diagramme d'activité de suppression

Le diagramme d'activité de suppression,voire Figure 3.8.

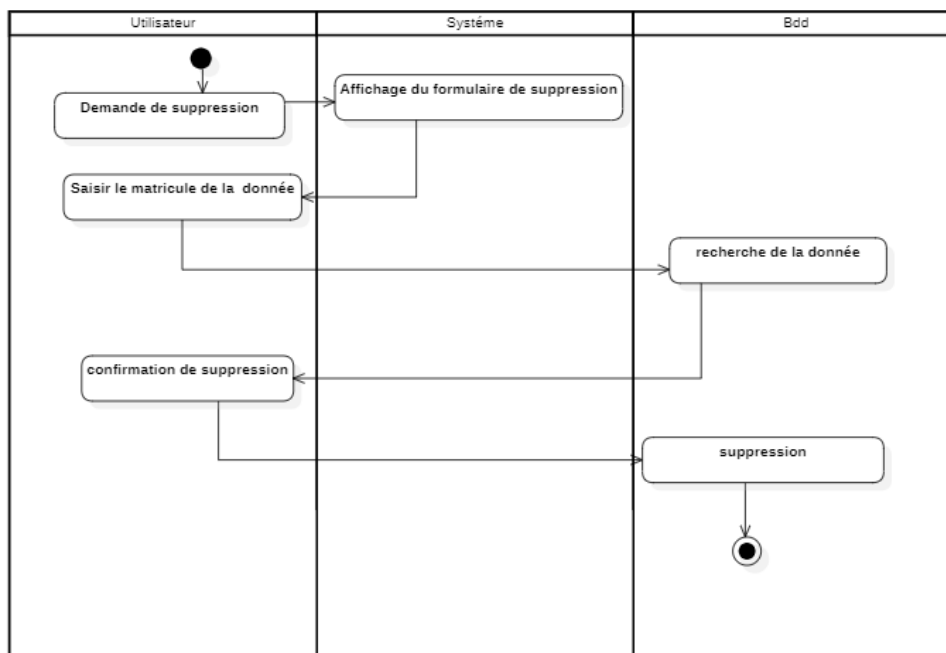


FIGURE 3.8 – Diagramme d'activité de "suppression"

### 3.3.9 Diagrammes de classes

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles.

Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction, voire la figure 3.9. [22]

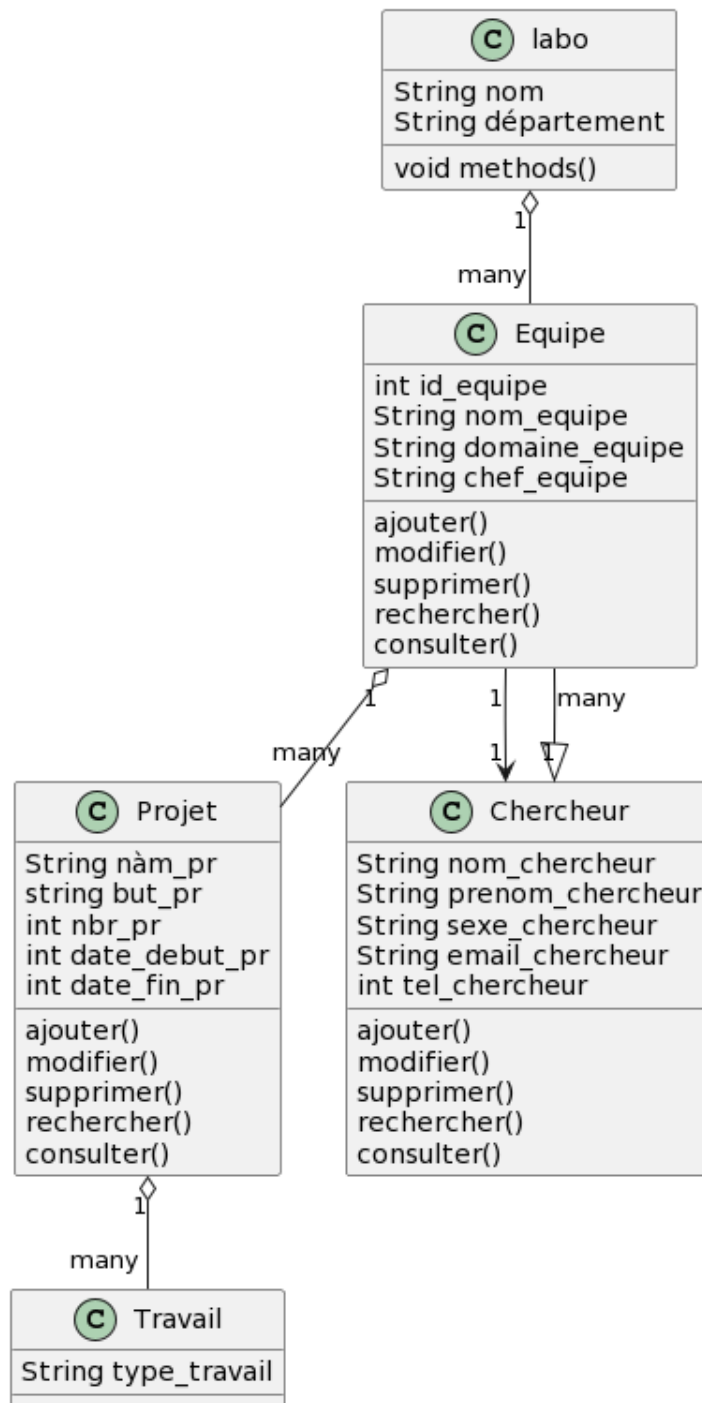


FIGURE 3.9 – Diagramme de classe

### 3.3.10 Diagrammes de déploiement

Les diagrammes de déploiement indiquent la manière dont les logiciels sont déployés sur les composants matériels d'un système.

Ces visuels sont particulièrement utiles pour les ingénieurs système et ils illustrent généralement les performances, l'évolutivité, la maintenabilité et la portabilité.

Lorsque les composants matériels sont représentés les uns par rapport aux autres, il est plus facile de suivre l'ensemble de votre infrastructure informatique et de s'assurer que tous les éléments sont pris en compte lors d'un déploiement, voir figure 3.10. [23]

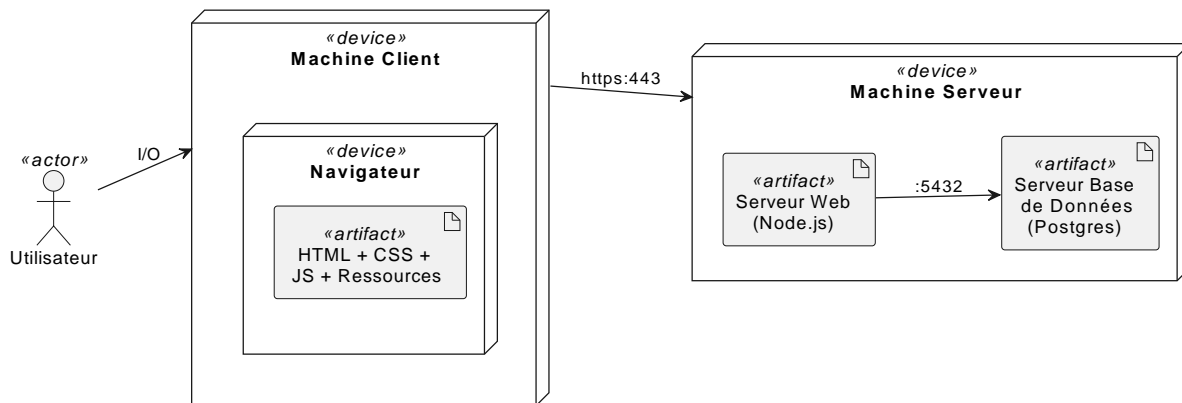


FIGURE 3.10 – Diagramme de déploiement

## 3.4 Conclusion

Dans ce chapitre, nous avons examiné les besoins fonctionnels et non fonctionnels de notre application, ainsi que les différents types de diagrammes UML qui peuvent être utilisés pour modéliser ces besoins.

Les diagrammes de cas d'utilisation sont utiles pour représenter les différentes actions que les utilisateurs peuvent effectuer dans le système et les diagrammes de classe permettent de détailler les différentes classes et relation entre celles-ci ainsi les diagrammes d'activité permettent de modéliser les processus métier et les diagrammes de déploiement permettent de décrire l'environnement technique dans lequel le système sera déployé, ensemble ces outils de modélisation peuvent aider à développer un système qui répond aux besoins de l'entreprise et des utilisateurs tout en étant efficace et bien structuré.

# Chapitre 4

## Implémentation

### 4.1 Préambule

Dans ce chapitre, on présente la réalisation d'application web, tout d'abord, on commence par les différents technologies utilisé ainsi que les outils de développement et quelque exemple d'interface graphique de notre système.

### 4.2 Les technologies utilisé :

#### 4.2.1 Le frontend

Le frontend est la partie visuelle de l'application Web qui est affichée dans les navigateurs.

#### Html

HTML signifie Hyper Text Markup Language est le langage de balisage standard pour la création de pages Web, html décrit la structure d'une page Web, se compose d'une série d'éléments indiquant au navigateur comment afficher le contenu, Les éléments HTML étiquettent des éléments de contenu tels que "ceci est un titre", "ceci est un paragraphe", "ceci est un lien", etc. [24]

#### Css

CSS signifie feuilles de style en cascade CSS décrit comment les éléments HTML doivent être affichés à l'écran, sur papier ou sur d'autres supports CSS économise beaucoup de travail. il peut contrôler la mise en page de plusieurs pages Web à la fois Les feuilles de style externes sont stockées dans des fichiers CSS. [24]

## typescript

TypeScript est un sur-ensemble syntaxique de JavaScript qui ajoute un typage statique.

Cela signifie essentiellement que TypeScript ajoute une syntaxe au-dessus de JavaScript, permettant aux développeurs d'ajouter des types.

### Pourquoi devrais-je utiliser TypeScript ?

JavaScript est un langage faiblement typé. il peut être difficile de comprendre quels types de données sont transmis en JavaScript.

En JavaScript, les paramètres de fonction et les variables n'ont aucune information ! Les développeurs doivent donc consulter la documentation ou deviner en fonction de l'implémentation.

TypeScript permet de spécifier les types de données transmises dans le code et a la capacité de signaler des erreurs lorsque les types ne correspondent pas. par exemple, TypeScript signalera une erreur lors du passage d'une chaîne dans une fonction qui attend un nombre. JavaScript ne le fera pas. [24]

## Tailwind CSS

Tailwind CSS est un framework CSS utilitaire qui vous aide à concevoir une interface utilisateur réactive. il prend en charge la thématisation, l'animation, le rembourrage et les marges prédéfinis, le flex, les grilles, etc. [25]

Tailwind CSS a été développé dans le but d'accélérer le processus de prototypage de pages Web pour répondre aux exigences (personnalisées), Tailwind CSS essaie d'adopter une autre approche dans le développement Web, qui a été dominé par Bootstrap, de sorte qu'il n'y a plus le terme "vue d'un million de personnes" en raison des similitudes entre un site Web et un autre (car les deux utilisent le thème Bootstrap par défaut , personnalisation minimale).

Un exemple est que les paramètres de rembourrage sont déclarés pour contenir uniquement la propriété padding, pourquoi est- ce parce que les utilisateurs peuvent librement ajuster l'apparence du site Web sans être bloqués par les règles établies par le framework qui définissent plus d'une propriété css dans 1 classe Points importants du CSS Tailwind : [26]

1. Tailwind CSS fournit des tonnes de classes CSS prêtes à l'emploi pour créer des interfaces utilisateur selon les besoins.
2. Minimiser l'écriture de CSS personnalisés.
3. Réactif car chaque utilitaire Tailwind a des types réactifs qui facilitent la création d'interfaces réactives sans avoir besoin en utilisant CSS personnalisé.
4. Compatible avec les composants bien que nous ne puissions faire que beaucoup de choses avec en utilisant des classes d'utilité, ainsi qu'en étendant un projet, il sera utile pour coder des modèles communs en abstractions de haut niveau.

## React

React est une bibliothèque JavaScript pour créer des interfaces utilisateur. React est utilisé pour créer des applications d'une seule page. React nous permet de créer des composants d'interface utilisateur réutilisables..[27]

### Historique de React.JS

La version actuelle de React.JS est la V18.0.0 (avril 2022).

La version initiale au public (V0.3.0) était en juillet 2013. React.JS a été utilisé pour la première fois en 2011 pour la fonction Newsfeed de Facebook.

L'ingénieur logiciel de Facebook, Jordan Walke, l'a créé.

La version actuelle de create-react-app est la v5.0.1 (avril 2022). create-react-app inclut des outils intégrés tels que webpack, Babel et ESLint. [24]

### Quand utiliser React ?

React est une solution universelle et offre des possibilités de développer tout types d'application. En particulier : [28]

1. Les application dynamiques React est un excellent choix pour développer des applications dynamiques notamment car il permet de refléter rapidement les changements de données dans la vue.
2. Les SPA (Single Page Application) Étant donné que React peut afficher les modifications de contenu sans recharger la page actuelle, il s'agit d'une excellente solution pour les applications dites SPA.
3. Applications mobiles React Native permet la création d'applications mobiles dites hybrides avec un rendu natif.

### React Router :

React router est la bibliothèque la plus populaire et la plus utilisée pour le routage dans les applications React, au fur et à mesure que votre application nécessite plusieurs vues et itinéraires, il est idéal de choisir un bon routeur pour gérer la transition entre les vues, les redirections, obtenir facilement les paramètres d'URL, etc.

Le routage est de la plus haute importance dans presque toutes les architectures d'application. Plus votre application est grande, plus votre fonctionnalité de routage devient complexe, des scénarios de routage simples à ceux profondément imbriqués.[29]

## Mobx (gestion réactive des états de l'application)

MobX est une bibliothèque de gestion d'état simple, évolutive et puissante. Tout comme React, qui utilise un DOM virtuel pour rendre les éléments de l'interface utilisateur dans nos navigateurs, réduisant le nombre de mutations DOM, MobX fait la même chose mais dans notre état d'application.

MobX a été créé en utilisant TFRP (application transparente programmation réactive) . Nous pouvons le comprendre comme étant une bibliothèque réactive : cela rend l'état de notre application cohérent et sans bogue en utilisant un graphe d'état de dépendance réactif qui ne sera mis à jour que lorsque cela sera nécessaire.

MobX traite l'état de votre application comme une feuille de calcul, ce qui est vrai. Un autre avantage de MobX est qu'il s'agit d'une bibliothèque indépendante du framework, ce qui signifie que vous pouvez également l'utiliser dans d'autres environnements JS, ou même dans d'autres langages par exemple, Flutter.[30]

### 4.2.2 Le backend

Le backend est le côté serveur d'un site Web. Il conserve et organise les données et garantit que tout le côté client du site Web fonctionne correctement.

#### Node.js

Node.js est un environnement de serveur open source et gratuit Node.js fonctionne sur différentes plates-formes (Windows, Linux, Unix, Mac OS X, etc.) Node.js utilise JavaScript sur le serveur .[24]

#### Que peut faire Node.js ?

- ▶ Node.js peut générer du contenu de page dynamique
- ▶ Node.js peut créer, ouvrir, lire, écrire, supprimer et fermer des fichiers sur le serveur
- ▶ Node.js peut collecter des données de formulaire
- ▶ Node.js peut ajouter, supprimer, modifier des données dans votre base de données

#### Next.js

Next.js est un framework JavaScript open-source pour créer des applications web avec React. Il fournit des fonctionnalités pour faciliter le développement d'applications web statiques et dynamiques, comme la prise en charge de l'exécution côté serveur pour les composants React, la gestion automatique des routes, la pré-rendu, la prise en charge des médias

statiques, etc. Il permet aux développeurs de créer des applications web performantes et évolutives en utilisant les avantages de React et des outils modernes de développement web. Il est souvent utilisé pour créer des applications web d'entreprise, des sites web blogs, des plateformes e-commerce, etc.[31]

### 4.2.3 La base de donnée

#### Prisma

Prisma est un ORM moderne qui prend en charge plusieurs bases de données SQL, MongoDB et CockroachDB (à l'aide d'un package externe). Il offre une gamme de fonctionnalités qui facilitent et rendent plus productif le travail avec les bases de données, notamment un outil CLI, des schémas, un générateur de requêtes de type sécurisé, des migrations et une interface graphique pour inspecter votre base de données et votre schéma.

Avec Prisma, vous définissez votre schéma de base de données à l'aide du langage de définition de schéma (SDL) de Prisma, et Prisma génère une API de création de requêtes de type sécurisé basée sur ce schéma. Cela signifie que vous pouvez écrire des opérations de base de données d'une manière plus sécurisée et conviviale pour les développeurs. Cela inclut la saisie semi-automatique, la coloration syntaxique et la vérification des erreurs.[32]

#### Postgresql

Postgresql est un système de base de données open source, très stable, qui prend en charge différentes fonctions de SQL, comme les clés étrangères, les sous-requêtes, les déclencheurs, ainsi que différents types et fonctions définis par l'utilisateur. Il augmente encore le langage SQL en proposant plusieurs fonctionnalités qui permettent de mettre à l'échelle et de réserver méticuleusement les charges de travail de données. Il est principalement utilisé pour stocker les données de nombreuses applications mobiles, web, géospatiales et analytiques.[33]

## 4.3 outils de Développement

### 4.3.1 Overleaf



Overleaf est une plate-forme en ligne gratuite permettant d'éditer du texte en LATEX sans aucun téléchargement d'application. En outre, elle offre la possibilité de rédiger des documents de manière collaborative, de proposer ses documents directement à différents éditeurs (IEEE Journal, Springer, etc.) ou plateformes d'archives ouvertes (arXiv, engrxiv, etc.) pour une éventuelle publication, cette plateforme est très compatible avec différents supports tels que tablettes et smartphones.[34]

### 4.3.2 PlantUML



PlantUML est un outil de modélisation Open Source qui permet aux utilisateurs de créer des diagrammes à partir d'un langage de texte brut. Ce langage utilise une syntaxe simple qui inclut des constructions courantes de l'UML, mais suit également des conventions spécifiques à chaque type de diagramme. Grâce à PlantUML, les utilisateurs peuvent créer des diagrammes rapidement et facilement, sans avoir besoin de connaissances approfondies en programmation ou en graphisme.[35]

### 4.3.3 Visual Studio code



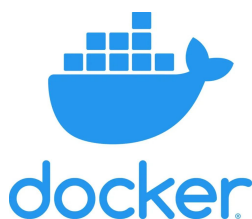
Visual Studio Code (VSCode) est un éditeur de code source et un environnement de développement intégré (IDE) de Microsoft. Il est open-source et cross-platform, c'est-à-dire qu'il fonctionne sur Windows, Linux et Mac. Il a été conçu pour les développeurs web, mais il prend en charge de nombreux autres langages de programmation tels que C#, C++ ,Python, Java, etc. Il offre de nombreuses fonctionnalités comme la coloration syntaxique, l'auto-complétion, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion de versions, l'intégration avec Git, et beaucoup d'autres. Il est également extensible à l'aide d'une grande variété d'extensions développées par la communauté, permettant aux développeurs de personnaliser l'éditeur selon leurs besoins.[36]

### 4.3.4 Prisma studio



Prisma Studio est une interface graphique de base de données qui permet aux développeurs de gérer et de visualiser les données capables dans leurs bases de données Prisma. Selon le site officiel de Prisma, Prisma Studio est "une interface utilisateur graphique pour interagir avec vos bases de données Prisma. Elle permet de visualiser et de gérer les données transférées dans vos bases de données Prisma, de gérer les schémas de base de données et de générer des requêtes SQL". Prisma Studio facilite la gestion et l'exploration des données pour les développeurs travaillant avec Prisma[37].


### 4.3.5 Docker



Docker est une plateforme de conteneurisation qui permet aux développeurs de créer, de réussir et d'exécuter des applications dans des conteneurs logiciels isolés. Selon le site officiel de Docker, "un conteneur est une unité standard de logiciel qui regroupe le code et toutes ses dépendances, ce qui permet à l'application de s'exécuter rapidement et de manière fiable dans n'importe quel

environnement". Docker permet aux développeurs de créer des images de conteneurs qui peuvent être distribuées facilement et exécutées sur n'importe quel système d'exploitation ou infrastructure cloud. Docker est largement utilisé dans le développement d'applications modernes pour sa facilité de déploiement et sa portabilité.[38]

### 4.3.6 StoryBook

 **Storybook** est un outil open-source pour le développement d'interfaces utilisateur (UI) isolées, réutilisables et testables. Selon le site officiel de Storybook, "Storybook est un environnement de développement pour les composants UI, il vous permet de parcourir un ensemble de composants, de les développer et de les tester en isolation de votre application". Storybook permet aux développeurs de concevoir, de développer et de tester des composants UI en dehors de leur application principale, ce qui facilite la réutilisation et la maintenance des composants. Storybook prend en charge de nombreux frameworks front-end populaires, tels que React, Vue.js, Angular, etc.[39]

## 4.4 Présentation des interfaces

Les figures ci-dessous représentent les différentes sections de l'interface vitrine de notre application :

### 4.4.1 Page d'accueil

il s'agit de la première page qui apparaît à l'utilisateur, elle englobe les informations les plus importantes de notre application et joue le rôle de portail de cette dernière.



## Laboratoire d'Informatique et de Communication de l'Université 20 Août 1955-Skikda

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Enim rem accusamus tempora voluptas consequatur? Dicta, quis. Labore debitis optio, magnam dolorum accusamus veritatis blanditiis suscipit quos assumenda facilis adipisci amet?

Iusto a eveniet vero beatae, fugiat dicta provident repellendus, animi impedit eius inventore natus ipsa aspernatur molestiae maiores? Cumque aspernatur nobis aliquid dignissimos iure maxime. Sapiente adipisci harum nam enim! Ea, nostrum porro quibusdam commodi maiores itaque voluptatibus! Sed amet ducimus, deserunt culpa praesentium, perferendis veniam facilis nobis sint dolore nisi? Saepe quasi eos quaerat aperiam velit. Vero, totam dolore.

FIGURE 4.1 – Page d'accueil

### 4.4.2 Liste des travaux



#### Works

Name	Description	Project	Author	Comment	Date	Files
Work 1	Description of Work 1	Project B 1	teamMemberPermB teamMemberPermB	Work comment	Tue Jul 04 2023	<a href="#">LOGO.PNG</a>

FIGURE 4.2 – Liste des travaux

### 4.4.3 Formulaire d'enregistrement

The screenshot shows the 'Sign Up Form' interface. At the top left is the LICUS logo. At the top right are 'SIGN UP' and 'LOG IN' buttons. The form itself is a white box with the following fields: Email, Password, First Name, Last Name, University, and Grade. Each field has a corresponding input box. Below the Grade field is a 'SIGN UP' button. To the left of the form, there is a table with two columns: 'Name' and 'Description'. The first row contains 'Work 1' and 'Description Work 1'. To the right of the form, there is a 'Files' section with a 'LOGO.PNG' button.

FIGURE 4.3 – Formulaire d'enregistrement

### 4.4.4 Formulaire d'authentification

The screenshot shows the 'Log In Form' interface. At the top left is the LICUS logo. At the top right are 'SIGN UP' and 'LOG IN' buttons. The form is a white box with the following fields: Email and Password. Each field has a corresponding input box. Below the Password field is a 'LOG IN' button. To the left of the form, there is a table with two columns: 'Name' and 'Description'. The first row contains 'Work 1' and 'Description Work 1'. To the right of the form, there is a 'Files' section with a 'LOGO.PNG' button.

FIGURE 4.4 – Formulaire d'authentification

## 4.4.5 Liste des Modèles Prisma

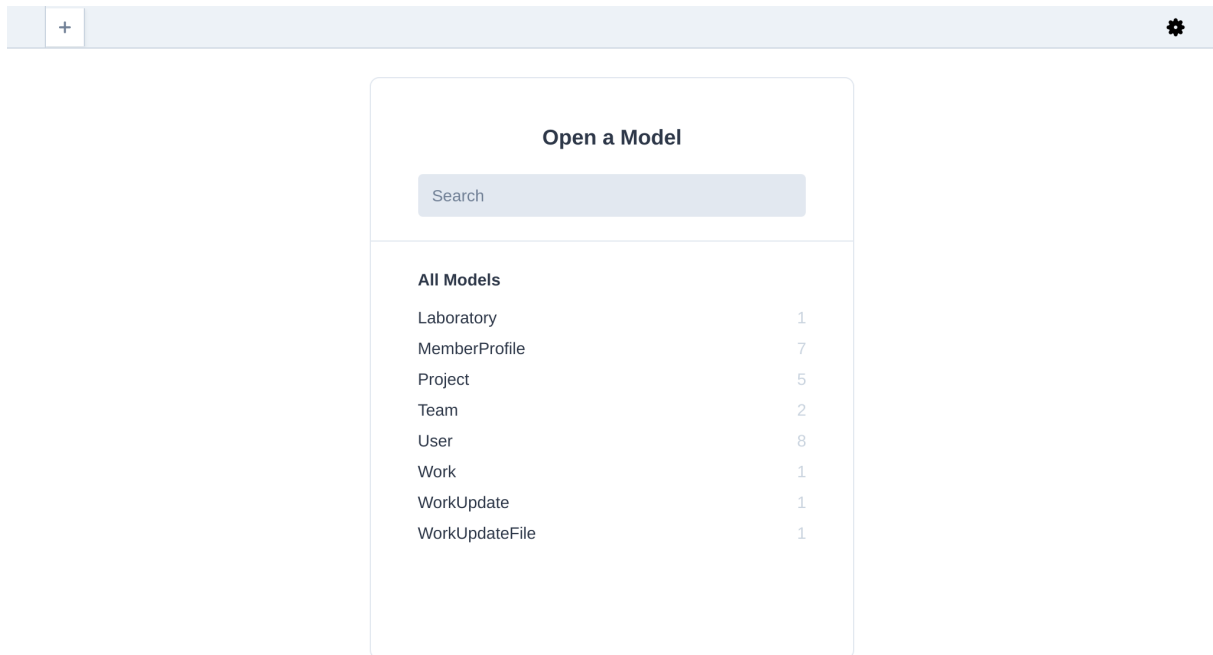


FIGURE 4.5 – Liste des Modèles Prisma

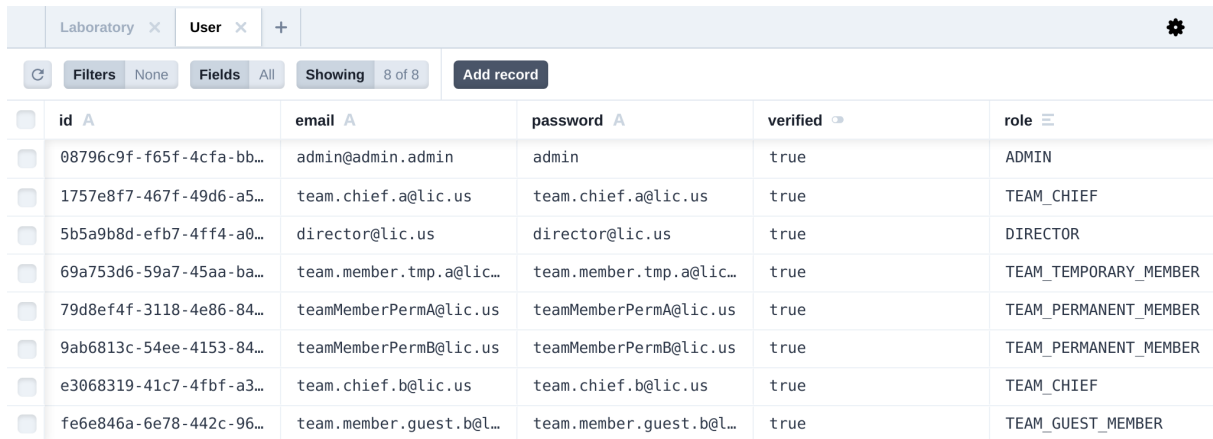
## 4.4.6 Model Prisma - Labo

The screenshot shows the Prisma interface for the "Laboratory" model. The table has the following columns: id, shortName, longName, description, and logo. There is one record displayed.

id	shortName	longName	description	logo
8595fe7a-7d04-4d1a-98...	LICUS	<b>L</b>aboratoire d'...	<p>Lorem, ipsum dolor...	iVBORw0KGgoAAAANSUHEU...

FIGURE 4.6 – Model Prisma - Labo

## 4.4.7 Model Prisma - User

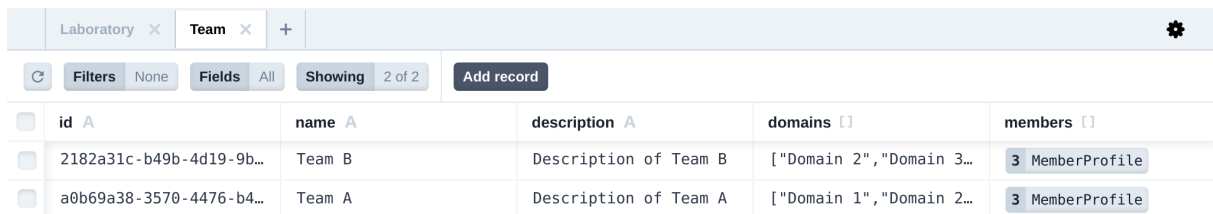


The screenshot shows a web interface for managing users. At the top, there are tabs for 'Laboratory' and 'User', with a plus sign and a settings gear icon. Below the tabs, there are buttons for 'Filters' (set to 'None'), 'Fields' (set to 'All'), 'Showing 8 of 8', and 'Add record'. The main content is a table with the following columns: 'id', 'email', 'password', 'verified', and 'role'. The table contains 8 rows of user data.

id	email	password	verified	role
08796c9f-f65f-4cfa-bb...	admin@admin.admin	admin	true	ADMIN
1757e8f7-467f-49d6-a5...	team.chief.a@lic.us	team.chief.a@lic.us	true	TEAM_CHIEF
5b5a9b8d-efb7-4ff4-a0...	director@lic.us	director@lic.us	true	DIRECTOR
69a753d6-59a7-45aa-ba...	team.member.tmp.a@lic...	team.member.tmp.a@lic...	true	TEAM_TEMPORARY_MEMBER
79d8ef4f-3118-4e86-84...	teamMemberPermA@lic.us	teamMemberPermA@lic.us	true	TEAM_PERMANENT_MEMBER
9ab6813c-54ee-4153-84...	teamMemberPermB@lic.us	teamMemberPermB@lic.us	true	TEAM_PERMANENT_MEMBER
e3068319-41c7-4fbf-a3...	team.chief.b@lic.us	team.chief.b@lic.us	true	TEAM_CHIEF
fe6e846a-6e78-442c-96...	team.member.guest.b@l...	team.member.guest.b@l...	true	TEAM_GUEST_MEMBER

FIGURE 4.7 – Model Prisma - User

## 4.4.8 Model Prisma - Team

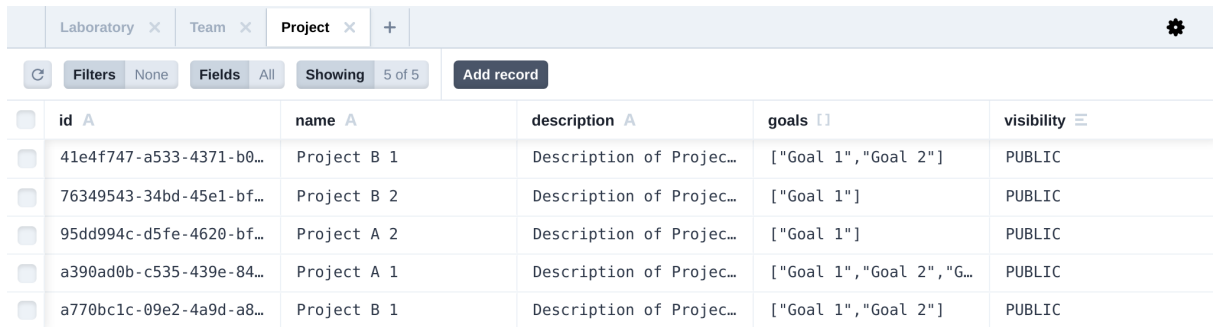


The screenshot shows a web interface for managing teams. At the top, there are tabs for 'Laboratory' and 'Team', with a plus sign and a settings gear icon. Below the tabs, there are buttons for 'Filters' (set to 'None'), 'Fields' (set to 'All'), 'Showing 2 of 2', and 'Add record'. The main content is a table with the following columns: 'id', 'name', 'description', 'domains', and 'members'. The table contains 2 rows of team data.

id	name	description	domains	members
2182a31c-b49b-4d19-9b...	Team B	Description of Team B	["Domain 2", "Domain 3...]	3 MemberProfile
a0b69a38-3570-4476-b4...	Team A	Description of Team A	["Domain 1", "Domain 2...]	3 MemberProfile

FIGURE 4.8 – Model Prisma - Team

## 4.4.9 Model Prisma - Project

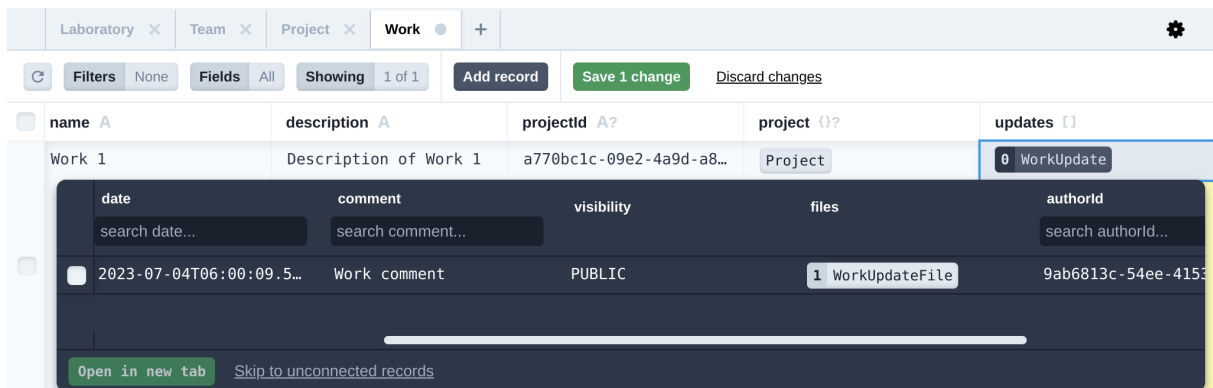


The screenshot shows the Prisma Project model interface. At the top, there are tabs for 'Laboratory', 'Team', and 'Project'. Below the tabs, there are buttons for 'Filters', 'Fields', and 'Showing 5 of 5'. A table with 5 columns is displayed: 'id', 'name', 'description', 'goals', and 'visibility'. The table contains 5 rows of project data.

id	name	description	goals	visibility
41e4f747-a533-4371-b0...	Project B 1	Description of Projec...	["Goal 1", "Goal 2"]	PUBLIC
76349543-34bd-45e1-bf...	Project B 2	Description of Projec...	["Goal 1"]	PUBLIC
95dd994c-d5fe-4620-bf...	Project A 2	Description of Projec...	["Goal 1"]	PUBLIC
a390ad0b-c535-439e-84...	Project A 1	Description of Projec...	["Goal 1", "Goal 2", "G...	PUBLIC
a770bc1c-09e2-4a9d-a8...	Project B 1	Description of Projec...	["Goal 1", "Goal 2"]	PUBLIC

FIGURE 4.9 – Model Prisma - Project

## 4.4.10 Model Prisma - Work



The screenshot shows the Prisma Work model interface. At the top, there are tabs for 'Laboratory', 'Team', 'Project', and 'Work'. Below the tabs, there are buttons for 'Filters', 'Fields', and 'Showing 1 of 1'. A table with 5 columns is displayed: 'name', 'description', 'projectId', 'project', and 'updates'. A modal overlay is shown over the table, displaying a list of work records with columns for 'date', 'comment', 'visibility', 'files', and 'authorId'.

name	description	projectId	project	updates
Work 1	Description of Work 1	a770bc1c-09e2-4a9d-a8...	Project	0 WorkUpdate

date	comment	visibility	files	authorId
2023-07-04T06:00:09.5...	Work comment	PUBLIC	1 WorkUpdateFile	9ab6813c-54ee-4153...

FIGURE 4.10 – Model Prisma - Work

## 4.5 conclusion

Ce dernier chapitre nous avons décrit en premier tous les technologies et les outils de développements que nous avons utilisés pendant la réalisation de notre projet. Ensuite nous avons présenté les captures les plus importants de l'application.

# Chapitre 5

## Perspectives

### 5.1 Préambule

L'application web développée pour le laboratoire de recherche informatique et communication de l'université de Skikda LICUS présente un grand potentiel d'amélioration et de développement dans divers domaines.

Dans ce chapitre, Nous allons examiner les différentes perspectives qui pourraient être améliorées afin d'optimiser l'expérience utilisateur et l'efficacité de l'application.

#### **PWA (Progressive Web App)**

pourrait améliorer l'expérience utilisateur, ce qui pourrait une utilisation plus fluide et rapide de l'application, même hors ligne. Les utilisateurs peuvent installer l'application sur leur écran d'accueil pour y accéder facilement et rapidement.

#### **SEO**

l'application pourrait bénéficier d'une amélioration du référencement naturel (SEO) pour améliorer sa visibilité et son accessibilité sur les moteurs de recherche. Les techniques de SEO permettraient de mieux positionner l'application dans les résultats de recherche, ce qui attirerait d'avantage d'utilisateurs.

#### **Corrections les bugs**

Des corrections régulières des bugs devraient également être dégradées pour assurer la stabilité et la fiabilité de l'application. Cela permettra de maintenir la qualité de l'application et de garantir une expérience utilisateur satisfaisante.

## **Améliorations des performances**

L'amélioration des performances est également un domaine clé pour l'application. Les temps de chargement doivent être réduits autant que possible pour offrir une utilisation plus rapide et plus fluide, ce qui améliorerait l'expérience utilisateur. L'optimisation des performances devrait également permettre de mieux gérer les ressources système et d'améliorer la sécurité de l'application.

## **Amélioration de l'IHM (UI/UX)**

L'amélioration de l'interface utilisateur (UI/UX) est également importante, car elle peut rendre l'application plus attrayante et plus facile à utiliser. L'interface utilisateur doit être intuitive et facile à comprendre, afin que les utilisateurs puissent tirer le meilleur parti de l'application

## **Ajouts de fonctionnalité**

En termes de fonctionnalités, l'ajout d'un éditeur de texte pour la collaboration entre les chercheurs, utilisant la technologie CRDT, serait une amélioration majeure pour l'application. Cela impliquerait aux chercheurs de travailler de manière plus efficace et de mettre en oeuvre plus facilement.

## **Amélioration de l'outil de modélisation pour une meilleure expérience utilisateur**

L'application pourrait bénéficier d'une amélioration de l'outil de modélisation PlantUML ou de l'adoption d'un autre outil plus flexible. Cela pourrait mieux modéliser les systèmes et offrir une meilleure expérience utilisateur.

## **5.2 Conclusion**

En somme, l'application web développée pour le laboratoire de recherche informatique et communication de l'université de Skikda LICUS a un grand potentiel d'amélioration et de développement dans divers domaines.

En adoptant la technologie de Progressive Web Apps (PWA), en diminuant le référencement naturel (SEO), en corrigeant régulièrement les bugs, en diminuant les performances et l'interface utilisateur (UI/UX), en ajoutant un éditeur de texte pour la collaboration entre les chercheurs et en faciliter l'outil de modélisation, l'application pourrait offrir une meilleure expérience utilisateur, une meilleure collaboration et une meilleure productivité.

# Chapitre 6

## Conclusion Générale

A travers ce modeste travail, on a tenté d'expliquer notre thème de projet de fin d'études du point de vue théorique et pratique, un thème qui consiste à la conception et la réalisation d'une application web dynamique pour le laboratoire **LICUS (Laboratoire d'informatique et de communication de l'université de Skikda)**.

Nous avons présenté les concepts fondamentaux d'internet et du web, ainsi que l'architecture client/serveur.

Ensuite, nous avons abordé l'analyse des besoins pour notre applications web, en se concentrant sur la spécification des besoins fonctionnels et non fonctionnels, nous avons également présenté les diagrammes de cas d'utilisation, diagramme de classes, d'activité et de déploiement pour aider à comprendre les exigences et les fonctionnalités du système.

Nous avons réalisé notre application web, et traité les phases nécessaires à la réalisation, où nous avons appris à manipuler le Framework React, et approfondir nos connaissances sur le langage HTML, CSS/Tailwind CSS, TypeScript, PostgreSQL et Node.js .

Enfin, nous proposons également quelques perspectives autour PWA, SEO, Correction des bugs, Amélioration des performances, Amélioration de l'IHM (UI/UX), Ajouts de fonctionnalités, Amélioration de l'outil de modélisation pour une meilleure expérience utilisateur.

La réalisation de ce projet nous a été extrêmement rentable, car elle nous a permis de prendre conscience de la complexité et de l'importance d'une étude préalable approfondie, ainsi que de mettre en pratique les connaissances théoriques que nous avons acquises. En outre, ce projet nous a permis d'approfondir nos connaissances pratiques et théoriques en matière de conception et de programmation.

# Bibliographie

- [1] Aoudia THANINA et Bachir SIHAM. “Conception et réalisation d’une application web avec J2EE pour la gestion de scolarité cas:” 2 intepartners””. Thèse de doct. Université Mouloud Mammeri, 2014.
- [2] Jérémie CLÉMENT. *Web et Marketing : Du marketing traditionnel au webmarketing*. Pearson France, 2015.
- [3] *Du web 1.0 au web 4.0*. URL : <https://c-marketing.eu/du-web-1-0-au-web-4-0/>.
- [4] *navigateur*. URL : <https://www.1min30.com/dictionnaire-du-web/navigateur-web-browser>.
- [5] *Serveur web*. URL : <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimedia-informatique/s-glossaire-informatique-et-multimedia/491-serveur-web-definition.html>.
- [6] *Site web*. URL : <https://www.takttilcommunication.com/glossaire/projets-et-sites-internet/site-web.html>.
- [7] *Quelle est la différence entre un site web statique et un site web dynamique?* URL : <https://ibracilinks.com/blog/quelle-est-la-difference-entre-un-site-web-statique-et-un-site-web-dynamique>.
- [8] Narimane BOUBLAL et Asma OULALOU. “Conception et réalisation d’une application web pour le suivi de la recherche scientifique (production scientifique)”. In : ().
- [9] *Quels sont les différents types d’applications web disponibles sur le marché?* URL : <https://le-consultant-digital.com/site-web/quels-sont-les-differents-types-dapplications-web-disponibles-sur-le-marche%E2%80%89>.
- [10] *Application web vs site web : Quelles différences?* URL : <https://mobiskill.fr/blog/conseils-emploi-tech/application-web-vs-site-web-quelles-differences/>.
- [11] *Quels sont les avantages et les inconvénients d’une application web?* URL : <https://agence-scroll.com/blog/avantages-et-inconvenients-dune-application-web>.
- [12] Faten KARRAY. “Conception et développement d’un système d’information géographique pour la gestion des ressources en eau en Tunisie”. Thèse de doct. Tunisie : Université de Carthage, 2016.
- [13] *Architecture client/serveur à 3 niveaux*. URL : <https://web.maths.unsw.edu.au/~lafaye/CCM/initiation/3tier.htm>.

- [14] Anne-Lise COURBIS et Frédéric ADAM. *Architecture des systèmes d'information*. Dunod, 2014.
- [15] Guy PUJOLLE. *Réseaux informatiques - Notions fondamentales*. Eyrolles, 2018.
- [16] Guillaume DERVAL. *Introduction aux réseaux*. Dunod, 2015.
- [17] Dimitri KONSTANTAS. *Architecture des réseaux*. Presses Polytechniques et Universitaires Romandes, 2016.
- [18] Olivier HERSENT, David GURLE et Jean-Pierre PETIT. *Réseaux et télécoms*. Dunod, 2016.
- [19] *Diagramme de cas d'utilisation*. URL : [https://gitmind.com/fr/types-diagrammes-uml.html#\\_8](https://gitmind.com/fr/types-diagrammes-uml.html#_8).
- [20] *Le diagramme Use-Case*. URL : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/diagramme-de-cas-de-utilisation/>.
- [21] *UML Description textuelle des cas d'utilisation*. URL : <https://www.clicours.com/uml-description-textuelle-des-cas-dutilisation/>.
- [22] *Définition des diagrammes de classes UML 1.5*. URL : [https://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition\\_des\\_diagrammes\\_de\\_classes\\_UML\\_1.5](https://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_de_classes_UML_1.5).
- [23] *diagrammes de déploiement*. URL : <https://www.lucidchart.com/blog/fr/types-de-diagrammes-UML>.
- [24] URL : <https://www.w3schools.com/>.
- [25] Sourabh SHARMA. *tailwindcss*. URL : [https://www.google.dz/books/edition/Modern\\_API\\_Development\\_with\\_Spring\\_and\\_S/3wYxEAAAQBAJ?hl=ar&gbpv=0](https://www.google.dz/books/edition/Modern_API_Development_with_Spring_and_S/3wYxEAAAQBAJ?hl=ar&gbpv=0).
- [26] Ariq Rafi KUSUMAH RONI ANDARSYAH. *tailwindcss*. URL : [https://www.google.dz/books/edition/5\\_Tahap\\_Membuat\\_Dashboard\\_Admin\\_Untuk\\_Ke/B0u5EAAAQBAJ?hl=ar&gbpv=0](https://www.google.dz/books/edition/5_Tahap_Membuat_Dashboard_Admin_Untuk_Ke/B0u5EAAAQBAJ?hl=ar&gbpv=0).
- [27] CORY-GACKENHEIMER. *introduction to react*. URL : <https://www.amazon.com/Introduction-React-Cory-Gackenhimer/dp/1484212460>.
- [28] *Les technos front-end*. URL : <https://www.ennov.io/les-technos-front-end/>.
- [29] *React Router 4*. URL : <https://leprogrammeurmarocain.com/react-router-4-introduction-pratique/>.
- [30] *Le guide de MobX*. URL : <https://blog.arcoptimizer.com/le-guide-de-mobx>.
- [31] *Next.js : Définition et avantages*. URL : <https://bility.fr/definition-nextjs/>.
- [32] *Prisma TypeScript ORM*. URL : <https://www.thisdot.co/blog/prisma-typescript-orm-an-introduction-and-exploration/>.
- [33] *Quest-ce que PostgreSQL ?* URL : <https://kinsta.com/fr/base-de-connaissances/qu-est-postgresql/>.
- [34] *Latex*. URL : <https://paris-sorbonne.libguides.com/c.php?g=497641&p=4637541>.
- [35] *PlantUML*. URL : <https://www.openriskmanual.org/wiki/PlantUML>.

- [36] *Définition Visual Studio Code*. URL : <https://bility.fr/definition-visual-studio-code/>.
- [37] *Prisma Studio*. en. 2021. URL : <https://www.prisma.io/studio> (visité le 12/10/2021).
- [38] *Docker*. en. 2021. URL : <https://www.docker.com/resources/what-container> (visité le 13/10/2021).
- [39] *Storybook*. en. 2021. URL : <https://storybook.js.org/docs/react/get-started/introduction> (visité le 13/10/2021).