

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement supérieur et de la Recherche Scientifique  
Université ABDERRAHMANE MIRA - BEJAIA -



Faculté science exacte  
Département Informatique

Mémoire pour l'obtention du diplôme de master en informatique  
Option : Génie logiciel

# Application de e-commerce B2C de vente promotionnelle

---

Élaboré par :

- BOUHADDI ABDELLAH
- KHAMIS IDIR

Encadré par :

- ABDERRAHMANE SIDER

Devant le jury :

- ZINEB TAHAKOURT NÉE YOUNSI
- KHALED BEDJOU

2021-2022

# Table des matières

---

---

Table des matières

---

---

Liste des Tableaux

---

---

Liste des figures

---

---

Introduction Générale

---

---

---

---

## Chapitre I introduction et généralités sur le développement web

---

---

<b>I.1</b>	<b>Introduction</b> .....	<b>3</b>
<b>I.2</b>	<b>Développement web</b> .....	<b>3</b>
	I.2.1 Evolutions majeures des application web .....	3
	I.2.1.1 <i>Progressive Web App</i> .....	5
	I.2.1.2 <i>Différence entre les PWA et les applications natives</i> .....	6
	I.2.1.3 <i>Migration vers PWA</i> .....	6
<b>I.3</b>	<b>Framework</b> .....	<b>7</b>
	I.3.1 Définition .....	7
	I.3.2 Différents types de framework .....	7
	I.3.2.1 <i>Framework frontend</i> .....	8
	I.3.2.2 <i>Framework backend</i> .....	8
<b>I.4</b>	<b>Méthode de conception</b> .....	<b>9</b>
	I.4.1 Méthodes de conception Agile .....	9
	I.4.2 Processus Unifié .....	9
	I.4.2.1 <i>Caractéristiques du processus unifié</i> .....	9
	I.4.2.2 <i>Différents étapes du processus UP</i> .....	10
<b>I.5</b>	<b>Sécurité des applications web et mobiles</b> .....	<b>11</b>
<b>I.6</b>	<b>Conclusion</b> .....	<b>12</b>

**Chapitre II Analyse et Spécification des besoins**

---

<b>II.1</b>	<b>Introduction</b> .....	<b>14</b>
<b>II.2</b>	<b>Problématique</b> .....	<b>14</b>
<b>II.3</b>	<b>Objectif du projet</b> .....	<b>15</b>
<b>II.4</b>	<b>Spécification des besoins</b> .....	<b>16</b>
II.4.1	Analyse de l'existant .....	16
II.4.1.1	<i>Identification des acteurs</i> .....	16
II.4.1.2	<i>Description des rôles de chaque acteur</i> : .....	16
II.4.1.3	<i>Spécification des besoins fonctionnels</i> .....	16
II.4.1.4	<i>Spécification des besoins no fonctionnels</i> .....	17
II.4.1.5	<i>Identiffication des cas d'utilisation</i> .....	17
II.4.1.6	<i>Description textuelle des cas d'utilisations</i> .....	18
<b>II.5</b>	<b>Conclusion</b> .....	<b>24</b>

**Chapitre III Conception de l'application**

---

<b>III.1</b>	<b>Introduction</b> .....	<b>26</b>
<b>III.2</b>	<b>Diagramme de cas d'utilisation</b> .....	<b>26</b>
III.2.1	Diagramme de cas d'utilisation associe à l'Utilisateur .....	26
III.2.2	Diagramme de cas d'utilisation associe à l'Admin et le modérateur .....	27
III.2.3	Diagramme de cas d'utilisation associe à l'Client et le visiteur .....	28
<b>III.3</b>	<b>Diagramme de séquence</b> .....	<b>28</b>
III.3.1	Étude de cas d'utilisation .....	28
III.3.1.1	<i>Diagramme de séquence «s'authentifie»</i> .....	29
III.3.1.2	<i>Diagramme de séquence «gérer un produit»</i> .....	30
III.3.1.3	<i>Diagramme de séquence« gérer un compte»</i> .....	31
III.3.1.4	<i>Diagramme de séquence «gérer un commentaire»</i> .....	32
III.3.1.5	<i>Diagramme de séquence «gérer les signalisation»</i> .....	33
III.3.1.6	<i>Diagramme de séquence «Rechercher un produit »</i> .....	34
<b>III.4</b>	<b>Conclusion</b> .....	<b>34</b>

**Chapitre IV Réalisation**

---

<b>IV.1</b>	<b>Introduction</b> .....	<b>36</b>
-------------	---------------------------	-----------

<b>IV.2</b>	<b>Environnement et outils de programmation :</b> .....	<b>36</b>
IV.2.1	Html .....	36
IV.2.2	CSS .....	36
IV.2.3	JavaScripts cote client .....	36
IV.2.4	JavaScript (ES6) .....	37
IV.2.5	Node JS .....	37
	IV.2.5.1 <i>Npm</i> .....	38
	IV.2.5.2 <i>Node-sass</i> .....	38
	IV.2.5.3 <i>Js-cookie</i> .....	39
	IV.2.5.4 <i>Multer</i> .....	39
IV.2.6	Redux .....	40
IV.2.7	Axios .....	40
IV.2.8	React .....	41
IV.2.9	Github .....	41
<b>IV.3</b>	<b>Serveur d'application</b> .....	<b>41</b>
IV.3.1	ExpressJS .....	41
IV.3.2	MongoDB .....	42
IV.3.3	Mongoddb compass .....	42
IV.3.4	JSON .....	42
IV.3.5	Postman .....	43
<b>IV.4</b>	<b>Sécurité d'application</b> .....	<b>43</b>
IV.4.1	Cors .....	43
IV.4.2	Bcrypt .....	44
IV.4.3	Jsonwebtoken .....	45
<b>IV.5</b>	<b>Architecture de l'Application</b> .....	<b>46</b>
IV.5.1	Architecture de redux .....	46
<b>IV.6</b>	<b>Principales captures d'interfaces</b> .....	<b>47</b>
IV.6.1	Interface Inscription : .....	47
IV.6.2	Interface d'authentification : .....	47
IV.6.3	Interface pofile commerçant .....	48
IV.6.4	Interface Profile d'utilisateur : .....	49
IV.6.5	Interface page d'accueil d'utilisateur : .....	50
IV.6.6	Interface age d'accueil commerçant .....	51
IV.6.7	Interface gestion des comptes : .....	52
IV.6.8	Captures d'interfaces accueil commerçant sur smartphone .....	53
IV.6.9	Interface profil commerçant sur smartphone .....	54

	IV.6.10 Interface accueil utilisateur sur smartphone .....	55
<b>IV.7</b>	<b>Lighthouse test .....</b>	<b>56</b>
<b>IV.8</b>	<b>Conclusion .....</b>	<b>56</b>

---

---

Conclusion Générale

---

---

# Liste des tableaux

II.1	Différents acteurs et leur descriptions . . . . .	16
II.2	Liste des cas d'utilisations . . . . .	17
II.3	Cas d'utilisations s'inscrire . . . . .	18
II.4	Cas d'utilisations s'authentifier . . . . .	19
II.5	Cas d'utilisations modifie une publication . . . . .	19
II.6	Cas d'utilisations supprimer une publication . . . . .	20
II.7	Cas d'utilisations ajouter une publication . . . . .	20
II.8	Cas d'utilisations signaler une publication . . . . .	21
II.9	Cas d'utilisations ajouter un commentaire . . . . .	21
II.10	Cas d'utilisations modifier un commentaire . . . . .	21
II.11	Cas d'utilisations Archiver un commentaire . . . . .	22
II.12	Cas d'utilisations Modifier un compte . . . . .	22
II.13	Cas d'utilisations Bloquer un compte . . . . .	23
II.14	Cas d'utilisations Réactiver un compte . . . . .	23
II.15	Cas d'utilisations Gestion des signalisations . . . . .	24

# Table des figures

I.1	Applications web des années 2000	3
I.2	Applications web des années 2010	4
I.3	Applications web des années 2015	5
I.4	Applications web des années 2015	6
I.5	Framework frontend	8
I.6	Framework backend	9
I.7	Les étapes du processus UP	10
II.1	Le gaspillage alimentaire mondiale	14
II.2	Diagramme de séquence gérer un j aime	15
III.1	Diagramme de cas d'utilisation Utilisateur	26
III.2	Diagramme de cas d'utilisation admin et modérateur	27
III.3	Diagramme de cas d'utilisation client et visiteur	28
III.4	Diagramme de séquence s'authentifie	29
III.5	Diagramme de séquence gérer un produit	30
III.6	Diagramme de séquence gérer un compte	31
III.7	Diagramme de séquence gérer un commentaire	32
III.8	Diagramme de séquence gérer les signalisation	33
III.9	Diagramme de séquence Rechercher un produit	34
IV.1	Logo de HTML	36
IV.2	Logo de css	36
IV.3	Logo de javaScript	37
IV.4	Logo de Node JS	37
IV.5	Logo de Npm	38
IV.6	Logo de node-sass	38
IV.7	Logo de js-cookie	39
IV.8	Logo de multer	39
IV.9	Logo de Redux	40
IV.10	Logo de axios	40
IV.11	Logo de react	41
IV.12	Logo de Github	41
IV.13	Logo de expressJS	42
IV.14	Logo de mongoDB	42
IV.15	Logo de JSON	43
IV.16	Logo de postman	43
IV.17	Logo de Cors	44
IV.18	Logo de bcrypt	44
IV.19	Logo de Jsonwebtoken	45
IV.20	Architecture de l'Application	46

IV.21	Architecture de redux . . . . .	46
IV.22	Captures d'interfaces d'nscription . . . . .	47
IV.23	Captures d'interfaces d'authentification . . . . .	47
IV.24	Captures d'interfaces profile de commerçant . . . . .	48
IV.25	Captures d'interfaces profile d'utilisateur . . . . .	49
IV.26	Captures d'interfaces page d'accueil d'utilisation . . . . .	50
IV.27	Captures d'interfaces page d'acceuil commerçant . . . . .	51
IV.28	Captures d'interfaces gestion des comptes . . . . .	52
IV.29	Captures d'interfaces acceuil commerçant sur smartphone . . . . .	53
IV.30	Captures d'interfaces profil commerçant sur smartphone . . . . .	54
IV.31	Captures d'interfaces acceuil utilisateur sur smartphone . . . . .	55
IV.32	Captures d'interfaces acceuil utilisateur sur smartphone . . . . .	56



## *Dédicaces*

Je dédie ce travail :  
À nos chers parents, pour leur persévérance,  
et pour avoir suscité notre vocation  
et permis d'achever nos études en tant que nous sommes actuellement,  
À nos sœurs, et nos frères  
pour leur soutien moral  
et leur intérêt envers notre travail,  
À tous nos amis et collègues,  
À tous ceux qui nous ont aidés,  
À tous ceux qui nous sont chers,  
À tous ceux qui nous avons omis

## *Remerciements*

Quelques lignes ne pourront jamais exprimer la reconnaissance que nous éprouvons envers tous ceux qui, de près ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leurs amitiés à l'aboutissement de ce travail.

Nous remercions le bon Dieu de nous avoir donné le courage d'aller au bout de nos objectifs.

Nos vifs remerciements accompagnés de toute notre gratitude vont tout d'abord à notre promoteur Dr .abderrahmane SIDER, pour nous avoir guidé pour réaliser ce projet.

Egalement aux :

Président de jury de nous avoir honoré de sa présence, et aux jurys d'avoir accepté d'examiner notre travail.

Notre reconnaissance va également à tous nos enseignants

# Introduction Général

Tout au long de ce travail nous avons décrit les applications web et les PWA avec leur différents avantages. L'application que nous avons réalisée a été conçue et modéliser de manière à ce qu'elle soit accessible pour tous, ergonomique, conviviale, simple d'utilisation sont autant de qualités consacré à faciliter la navigation des différents utilisateurs.

Pour la réalisation de l'application, nous avons eu recours à différents langages et outils de développement et de programmation telle que React js et Redux pour la mise en place des différents traitements et fonctionnalités de l'application ainsi qu'HTML, CSS, JavaScript pour la réalisation de son aspect graphique, sans oublier MongoDB comme système de gestion de base de données.

Ce projet en plus de nous avoir permis d'assimiler certaines connaissances sur le e-Commerce et la programmation et nous a permis de suivre une méthode de conception pour la réalisation d'un projet en partant d'une idée pour arriver à un résultat final.

Pour mener à bien ma mission, j'ai mis en œuvre toutes mes connaissances acquises durant notre cursus universitaire.

# Chapitre I

introduction et généralités sur le  
développement web

## I.1 Introduction

Pour aborder le travail qui va suivre, il est nécessaire de définir les concepts utiles à la compréhension de ce projet. Nous allons présenter un aperçu sur globale sur les application Web leur evolution ainsi que l'environnement de travail adequat pour notre projet .

## I.2 Développement web

Le développement Web va de la création de pages en texte brut à des applications Web complexes, des applications de réseaux sociaux et des applications commerciales électroniques. Les applications web ne sont pas conçues uniquement pour les Smartphone et tablette, mais pour tout navigateur internet ordinateur ou mobile.[7]

### I.2.1 Evolutions majeures des application web

Le développement d'applications web a connu trois évolutions majeures.[7]

#### Première période

La première période se situe depuis la naissance du web jusqu'au milieu des années 2000. Il s'agissait de développer des applications classiques présentant des documents et des formulaires. Les technologies clientes utilisées se limitaient souvent au langage HTML et pour la fin de la période à l'utilisation de feuilles de style avec CSS. Voir figure ci-dessous



FIGURE I.1 – Applications web des années 2000

#### Deuxième période

La deuxième période se situe entre 2005 et 2010. Cette période se distingue par l'arrivée des applications web riches (RIA : Rich Internet Application). La couche logique a été grandement déportée vers le serveur (grâce à la popularisation du langage côté serveur PHP). C'est aussi pendant cette période que le développement d'AJAX est apparu avec le style d'architecture REST (services web). De cette période, la technologie la plus utilisée a été la bibliothèque JQuery. Cette bibliothèque a facilité la manipulation de la partie de la structure HTML afin d'obtenir des applications qui réagissaient sans avoir à rafraichir l'intégralité d'une page complète. Malheureusement, c'est aussi à cette période que le nombre de navigateurs a explosé. Le développement était donc un peu chaotique, car il fallait supporter toutes les subtilités des

navigateurs qui ne se conformaient pas forcément aux standards existants. Des initiatives via des bibliothèques Chapitre I Généralité sur les technologies web et Etude préliminaire comme GWT sont apparues pour tenter de résoudre ce problème, mais cela restait quand même du bricolage. Voir figure ci-dessous

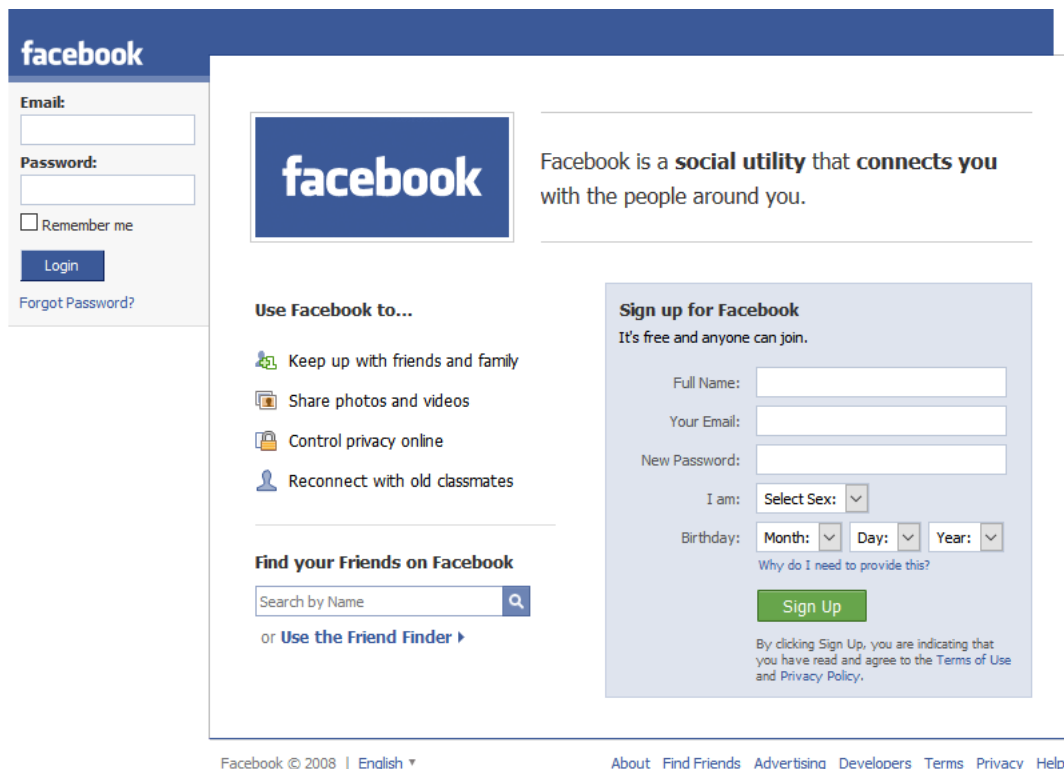


FIGURE I.2 – Applications web des années 2010

### Troisième période

La troisième période se situe après 2010 jusqu'à 2015. Nous sommes passés du terme applications web riches à applications web monopage (plus connu, en anglais sous le nom de SPA pour Single-Page application). Il s'agit désormais d'une application web où tout est accessible sur une page web unique afin de rendre l'expérience utilisateur la plus fluide possible. Pour obtenir ce résultat soit tout le contenu est chargé en avance soit à la demande, mais en tâche de fond. Il y a eu aussi un déport important de la logique vers le client afin de libérer des ressources côté serveur. Les effets de bord à ce transfert de code côté client a naturellement obligé les développeurs à structurer leur code et de facto à utiliser des outils adaptés (Grunt, Gulp, Yeoman, Webpack) pour réaliser des opérations sur cette masse de code (compilation, qualité de code, ...). C'est ainsi que des Framework web JavaScript sont apparus. Voir figure ci-dessous

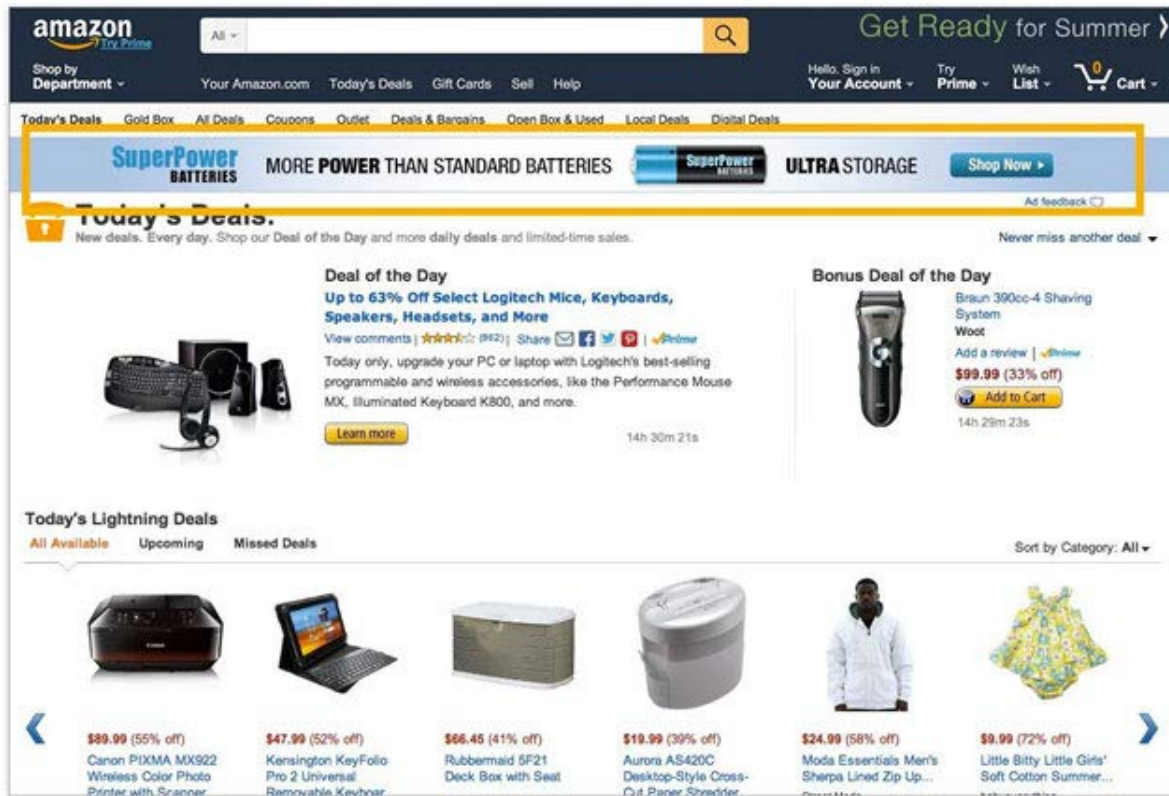


FIGURE I.3 – Applications web des années 2015

En 2015, le designer Français Berriman et l'ingénieur Google Chrome Alex Russell ont inventé le terme "applications Web progressives" pour décrire les applications tirant parti des nouvelles fonctionnalités prises en charge par les navigateurs modernes. Y compris les techniciens de service et les manifestes d'applications Web, qui permettent aux utilisateurs de mettre à jour les applications Web. Google a ensuite déployé des efforts considérables pour promouvoir le développement de PWA pour Android.

En 2019, les PWA étaient prises en charge par les versions de bureau de la plupart des navigateurs, y compris Microsoft Edge (sur Windows) et Google Chrome (sur Windows, macOS, Chrome OS et Linux). Voir figure ci-dessous

### I.2.1.1 Progressive Web App

Les PWA sont des applications Web développées à l'aide d'un certain nombre de technologies spécifiques et de modèles standard pour leur permettre de tirer parti des fonctionnalités des applications Web et natives. Par exemple, les applications Web sont plus détectables que les applications natives ; il est beaucoup plus facile et rapide de visiter un site Web que d'installer une application, et on peut également partager des applications Web en envoyant un lien. D'autre part, les applications natives sont mieux intégrées au système d'exploitation et offrent donc une expérience plus transparente aux utilisateurs. On peut installer une application native pour qu'elle fonctionne hors ligne, et les utilisateurs adorent appuyer sur des icônes pour accéder facilement à leurs applications préférées, plutôt que d'y accéder à l'aide d'un navigateur. Les PWA nous donnent la possibilité de créer des applications Web qui peuvent profiter de ces mêmes avantages. Ce n'est pas un tout nouveau concept - de telles idées ont été revisitées à plusieurs reprises sur la plate-forme Web avec diverses approches dans le passé. L'amélioration progressive et la conception réactive nous permettent déjà de créer des sites Web adaptés aux

mobiles. Les PWA, cependant, fournissent tout cela et plus encore sans perdre aucune des fonctionnalités existantes qui font la grandeur du Web.[8] Voir figure ci-dessous

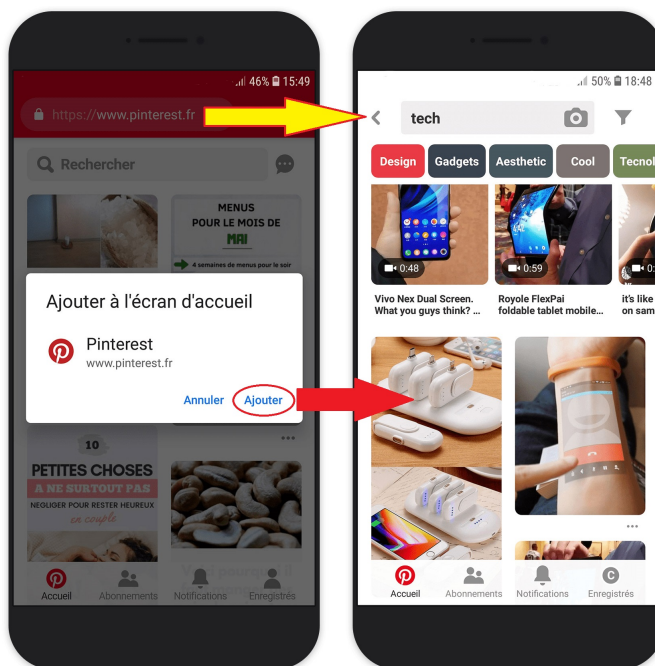


FIGURE I.4 – Applications web des années 2015

### I.2.1.2 Différence entre les PWA et les applications natives

Le développement d'une application Web progressive et d'une application native est en fait assez différent. Du point de vue du développeur, les applications natives sont codées pour être installées sur les smartphones, tablettes et autres appareils intelligents en fonction du système d'exploitation et du matériel associé à l'appareil.

Les PWA, d'autre part, sont essentiellement des sites Web conçus comme des applications qui peuvent être exécutées dans un navigateur de site Web ou installées directement sur l'appareil et accessibles comme une application native. Une PWA utilise CSS, Javascript et HTML tandis que les applications natives sont développées avec le langage de programmation pour les plateformes individuelles, y compris Java qui est le langage le plus courant pour Android et Swift ou Objective-C pour iOS. Cela rend le développement d'une PWA beaucoup plus rapide et plus facile, car on a besoin de créer qu'une seule version qui peut être affichée de manière transparente sur le navigateur Web de presque tous les appareils mobiles.[22]

### I.2.1.3 Migration vers PWA

Comme nous l'avons indiqué ci-dessus, les PWA ne sont pas créées avec une seule technologie. Ils représentent une nouvelle philosophie pour la création d'applications Web, impliquant des modèles, des API et d'autres fonctionnalités spécifiques. Il n'est pas si évident de savoir si une application Web est une PWA ou non à première vue. Une application peut être considérée comme une PWA lorsqu'elle répond à certaines exigences qu'on trouve sur le test fourni gratuitement par LightHouse (voir chapitre 4.7) ou implémente un ensemble de fonctionnalités données : fonctionne hors ligne, est installable, est facile à synchroniser etc.

De plus, il existe des outils pour mesurer le degré de complétude (en pourcentage) d'une application Web, comme Lighthouse. En mettant en œuvre divers avantages technologiques, nous pouvons rendre une application plus progressive, aboutissant ainsi à un score Lighthouse plus élevé. Mais ce n'est qu'un indicateur approximatif.

Il y a quelques principes clés qu'une application Web doit essayer d'observer pour être identifiée comme une PWA et on peut les considérer aussi comme étant des avantages des PWA.[8]

Ça devrait être :

- **Découvrable** : de sorte que le contenu peut être trouvé via les moteurs de recherche.
- **Installable** : il peut donc être disponible sur l'écran d'accueil ou le lanceur d'applications de l'appareil.
- **Liab**le : vous pouvez donc le partager en envoyant une URL.
- **Indépendant du réseau** : il fonctionne donc hors ligne ou avec une mauvaise connexion réseau.
- **Progressivement amélioré** : il est donc toujours utilisable à un niveau de base sur les anciens navigateurs, mais entièrement fonctionnel sur les derniers.
- **Réactivable** : il est donc capable d'envoyer des notifications chaque fois qu'un nouveau contenu est disponible.
- **Conçu de manière réactive** : il est donc utilisable sur n'importe quel appareil doté d'un écran et d'un navigateur : téléphones portables, tablettes, ordinateurs portables, téléviseurs etc.
- **Sécurisé** : de sorte que les connexions entre l'utilisateur, l'application et votre serveur sont sécurisées contre tout tiers essayant d'accéder à des données sensibles.
- **Responsive** : Quand on parle de responsive, il s'agit avant tout de « Responsive design ». Cela consiste à rendre un site web accessible et adaptable à tous les devices : tablettes, smartphones, etc.

## I.3 Framework

### I.3.1 Définition

Un framework (**infrastructure logicielle** en français ) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. C'est le framework, encore appelé structure logicielle, canevas ou socle d'applications en français, qui établit les fondations d'un logiciel ou son squelette applicatif. Tous les développeurs qui l'utilisent peuvent l'enrichir pour en améliorer l'utilisation. L'objectif du framework est de simplifier et d'uniformiser le travail des développeurs. Il fonctionne comme un cadre ou un patron, mais son maniement suppose d'avoir déjà un profil expérimenté. En général, une infrastructure logicielle est associée spécifiquement à un langage de script ou de programmation. Par exemple, Hibernate est un framework pour JavaScript et Django pour Python.[1 ]

### I.3.2 Différents types de framework

Il existe aujourd'hui plusieurs catégories de framework utilisés couramment par les développeurs [1 ] :

- Les framework d'infrastructure système permettent le développement des systèmes d'exploitation (OS) et des interfaces graphiques.



- Les framework d'intégration intergicielle fédèrent des applications hétérogènes
- Et les framework d'entreprises sont spécifiques aux applications utilisées par les entreprises.
- Enfin, les framework de gestion de contenu ont pour mission de créer, collecter, classer, stocker et publier des informations numérisées.

#### I.3.2.1 Framework frontend

Ces frameworks servent à définir ce que va voir le visiteur lorsque qu'il va arriver sur le site. Ils utilisent des langages clients tels que le langage HTML, CSS ou Javascript. En exemple de framework frontend on peut retrouver Vue.JS, React ou bien AngularJS.[2 ]



FIGURE I.5 – Framework frontend

#### I.3.2.2 Framework backend

Ces frameworks correspondent à l'interface administration du site. C'est la partie immergée de l'iceberg sans quoi le site ne peut fonctionner. Ils utilisent eux des langages serveurs comme PHP, Java, Python ou Ruby et comme exemple on peut citer Symfony, Laravel, Django ou Zend ,ect. [2 ]



FIGURE I.6 – Framework backend

## I.4 Méthode de conception

### I.4.1 Méthodes de conception Agile

Une méthodologie agile prévoit la fixation d'objectifs à court terme utilisés dans la gestion de projets. Alors que la méthode traditionnelle prévoit la planification totale du projet avant même la phase développement. Le projet est ainsi fragmenté en plusieurs sous-parties que les équipes de développement qui en a la charge se doivent d'atteindre progressivement en ajustant si nécessaire les objectifs pour répondre le plus possible aux attentes du client.. Les méthodes agiles permettent de renforcer les relations entre les membres d'une équipe, mais aussi entre l'équipe et le client.[4]

Agile a quatre valeurs principales :[4]

- a. Les individus et interactions sont plus importants que les processus et les outils.
- b. Le logiciel de travail est plus important qu'une documentation complète.
- c. La collaboration du client est plus importante que la négociation du contrat.
- d. Répondre au changement est plus important que de suivre un plan.

les méthodes Agiles disponibles sont nombreuses .Parmi les méthodes Agiles les plus populaires en usage aujour c'est le Processus Unifié.C'est une version simplifiée du Rational Unified Process, ou RUP. Il s'agit d'une méthode de développement d'applications métier utilisant les techniques agiles du TDD (Test Driven Development ou développement piloté par les tests).

### I.4.2 Processus Unifié

Un processus de développement logiciel contient une séquence d'étapes afin de produire un système logiciel de qualité ou évoluer un système existant, tout en répondant aux besoins et aux exigences des utilisateurs et en prenant en considération le temps et les coûts prévisibles. Dans notre projet nous avons opté pour le processus unifié qui est un processus de développement itératif adapté à tout type de projet. Il couvre toutes les activités de la conception du projet jusqu'à la livraison du produit [5].

#### I.4.2.1 Caractéristiques du processus unifié

Les principales caractéristiques du processus unifié sont :[5]

- **UP est piloté par les cas d'utilisation** : Le but de créer un logiciel est de satisfaire les utilisateurs et ceci en étudiant leurs besoins. UP permet la spécification et l'analyse des besoins fonctionnels des utilisateurs, ces besoins sont illustrés par des cas d'utilisation qui représentent toutes les fonctionnalités du logiciel.
- **UP est centré sur l'architecture** : UP donne une vue sur l'architecture du système à réaliser, cette architecture émerge des besoins fonctionnels exprimés par les utilisateurs et des besoins non fonctionnels (ergonomie, performance ...).
- **UP est itératif et incrémental** : UP est un ensemble d'itérations où un ensemble d'opérations est répété et à chaque fin d'itération une version exécutable du logiciel est produite ce qui permet l'élimination des problèmes imprévus, la prise en compte des besoins des utilisateurs et l'accélération du rythme de développement. Incrémental signifie que à chaque itération une version améliorée du système sera créée.

#### I.4.2.2 Différents étapes du processus UP

UP répète un ensemble d'opérations en 4 étapes :[5]

- 1) **Analyses des besoins** : Cette phase donne une vue globale sur le produit fini, elle permet d'identifier les acteurs, de déterminer les besoins fonctionnels et non fonctionnels du produit.
- 2) **Élaboration** : L'élaboration permet de concevoir une architecture du système en réalisant un modèle initial de cas d'utilisation pour couvrir les besoins fonctionnels et pour planifier la phase de construction, et en identifiant les risques, les coûts, le calendrier et le budget.
- 3) **Construction** : Dans cette phase il faut implémenter tous les cas d'utilisation et fournir une version exécutable du système.
- 4) **Transition** : Dans cette phase les utilisateurs essayent le produit pour détecter les anomalies à corriger et pour vérifier si les services offerts par le système répondent à leurs exigences [5].

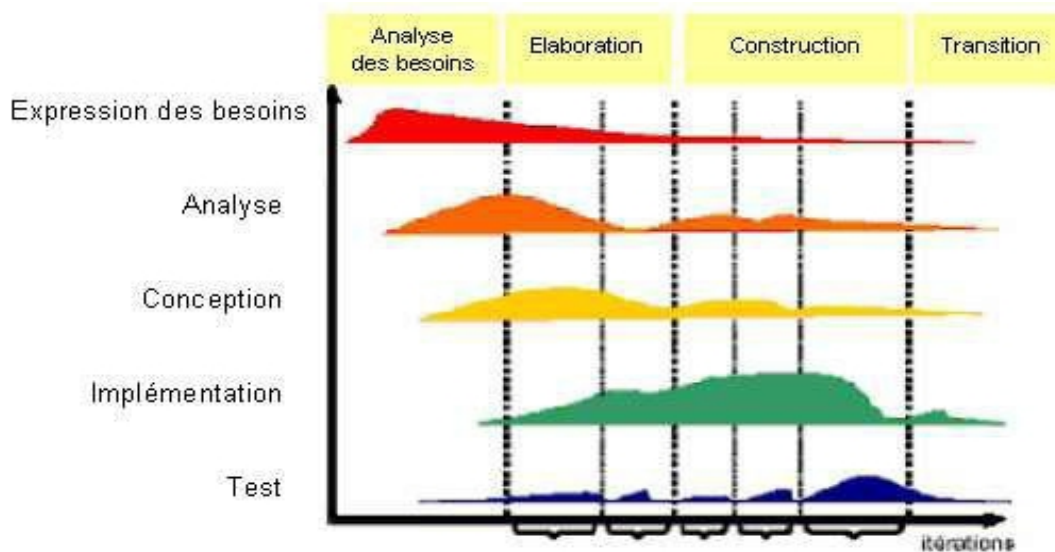


FIGURE I.7 – Les étapes du processus UP

## I.5 Sécurité des applications web et mobiles

La sécurité des applications n'est ni une fonctionnalité ni un avantage - c'est une nécessité absolue. La sécurité devrait être une priorité à partir du moment où l'on commence à écrire la première ligne de code.[6]

Avec plusieurs types d'informations en jeu, une application mobile doit être conçue, à tout prix pour protéger ses utilisateurs et clients. La sécurité doit être assurée et intégrée chaque étape et élément de l'application via 10 façons :[6]

### 1) **Sécuriser le code**

Les bugs et les vulnérabilités d'un code sont le point de départ de la plupart des attaques pour pénétrer une application, tout ce dont ils ont besoin est une copie publique de l'application. Des recherches montrent que les codes malveillants affectent à tout moment plus de 11,6 millions d'appareils mobiles. [6] Pour remédier au mieux à cela, il faut minifier le code afin de ne pas être victime d'ingénierie inverse. Il est aussi nécessaire d'exécuter plusieurs tests suivis de correction des bugs au fur et à mesure. Enfin, Il faut renforcer le code en utilisant une signature du code.[6]

### 2) **Chiffrer toutes les données**

Chaque unité de données échangée sur l'application doit être chiffrée. Le cryptage est le moyen de brouiller le texte brut jusqu'à ce qu'il ne soit qu'un bloc de texte sans signification pour personne, sauf pour ceux qui ont la clé. Cela signifie que même si des données sont volées, il n'y a rien que les criminels puissent lire et utiliser à mauvais escient.[6]

### 3) **Etre prudents avec les bibliothèques**

Lors de l'utilisation de bibliothèques tierces, il faut soigneusement tester le code avant de l'utiliser dans l'application. Aussi utiles qu'elles soient, certaines bibliothèques peuvent être extrêmement peu sûres pour l'application. La bibliothèque GNU C, par exemple, avait une faille de sécurité qui permettait aux attaquants d'exécuter à distance du code malveillant et de planter un système. Et cette vulnérabilité est restée inconnue pendant plus de sept ans.[6]

### 4) **Utilisation d'APIs autorisées**

Les APIs qui ne sont pas autorisées peuvent accorder involontairement à un pirate des privilèges malveillants.[6]

### 5) **Utilisation d'une authentification de haut niveau**

Certaines des plus grandes violations de sécurité se produisent en raison d'une authentification faible. Une grande partie de l'authentification dépend des utilisateurs finaux de l'application, mais les développeurs peuvent encourager les utilisateurs à être plus sensibles à l'authentification, et ce en concevant une authentification n'acceptant que des mots de passe alphanumériques forts qui doivent être renouvelés tous les trois ou six mois. L'authentification multi-facteur gagne aussi en importance, utilisant ainsi plusieurs facteurs d'authentification.[6]

### 6) **Déployer des technologies de détection de sabotage**

Il existe des techniques pour déclencher des alertes lorsque quelqu'un essaie de falsifier le code ou d'injecter du code malveillant. Une détection de sabotage active peut être déployée pour s'assurer que le code ne fonctionnera pas du tout s'il est modifié.[6]

### 7) **Utiliser le principe du moindre privilège**

Le principe du moindre privilège dicte qu'un code doit s'exécuter avec uniquement les autorisations dont il a absolument besoin et pas plus.[6]

### 8) **Déployer une gestion de session appropriée**

Les sessions sur mobile durent beaucoup plus longtemps que sur ordinateur. Cela rend la gestion des sessions plus difficile pour le serveur. Il est préférable d'utiliser des jetons au lieu des identifiants d'appareil pour identifier une session. Les jetons peuvent être révoqués 'a tout moment, ce qui les rend plus sûrs en cas de perte ou de vol d'appareils.[6]

9) **Utiliser les meilleurs outils et techniques de cryptographie**

Il ne faut pas coder les clés de chiffrement en hard code car cela permet aux attaquants de les voler facilement. Il est préférable de stocker les clés dans des conteneurs sécurisés et jamais localement sur l'appareil.[6]

10) **Tester a plusieurs reprises**

La sécurisation est un processus qui ne s'arrête jamais. De nouvelles menaces émergent et de nouvelles solutions sont nécessaires. Il faut investir dans les tests de pénétration, la modélisation des menaces et les émulateurs pour tester en continu les vulnérabilités et ainsi les corriger avec chaque mise à jour.[6]

## I.6 Conclusion

A travers ce chapitre nous avons vu quelque notion sur les évolutions des applications web et les applications progressives web, ainsi que les Frameworks back-end et front-end.

Dans le chapitre suivant on vous présente le cahier des charges après on passe à l'objectif de notre projet et a la description du système de notre plateforme.

# Chapitre II

## Analyse et Spécification des besoins

## II.1 Introduction

Le cahier des charges est la base de la conception d'un projet ; c'est un document qui permet de comprendre et d'expliquer un projet dans son ensemble, avec toutes les contraintes, les besoins, les objectifs où encore les intervenants qui y sont liés. Nous allons, dans ce chapitre, présenter notre projet ainsi que l'analyse des besoins fonctionnels et non-fonctionnels. On passe à la phase d'analyse et conception qui représente deux étapes essentielles à la démarche, ces étapes nous permettent de passer d'une structure fonctionnelle via les cas d'utilisation à une structure objet sous forme de modèle de classe. Au cours de ce qui suit, nous représenterons une vue dynamique du système par la modélisation des diagrammes de séquences et une vue statique par la modélisation du diagramme de classe.

## II.2 Problématique

Le gaspillage alimentaire est le fait de perdre ou jeter de la nourriture généralement destinée à la consommation humaine. Il se produit tout au long de la chaîne d'approvisionnement, de la production agricole jusqu'à la consommation, Le gaspillage alimentaire mondial est estimé par la FAO (Organisation des Nations unies pour l'alimentation et l'agriculture) à près d'un tiers de la production totale, ce qui représente environ 1.3 milliard de tonnes de nourriture perdue ou jetée chaque année l'équivalent de 1000 milliards de dollars. Nous on va s'intéresser au gaspillage au niveau des points de vente ou les magasins en général, et plus, précisément les produits périssables.

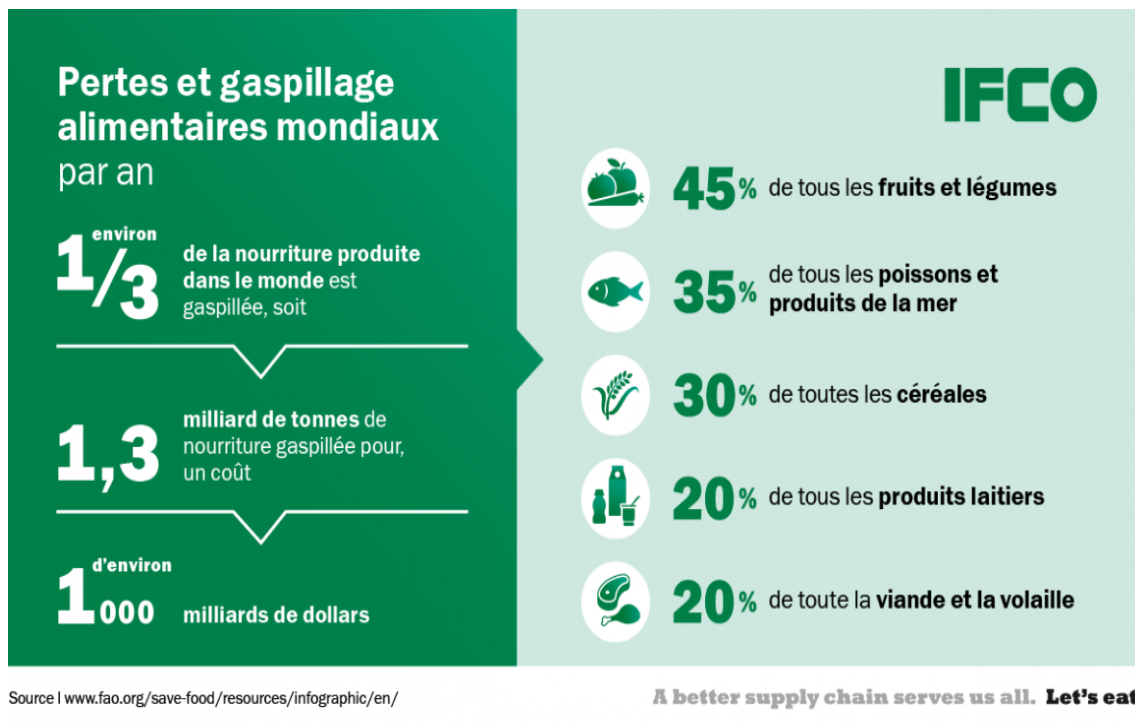


FIGURE II.1 – Le gaspillage alimentaire mondiale

### II.2.1 Gaspillage alimentaire au niveau des magasins

D'après les statistiques de la FAO (Organisation des Nations unies pour l'alimentation et l'agriculture) qui on était fait en France durant l'année 2019, il est estimé que chaque jour, chaque supermarché ou hypermarché français jette en moyenne 20 kg de nourriture. Cela représente un gaspillage alimentaire de 218 tonnes par jour ou 10 millions de tonnes par an. Cela représente 16 milliards d'euros perdus chaque année et 15.3 millions de tonnes d'équivalent CO2 émis pour rien !.

En 2019, la quantité de gaz à effet de serre produits par le gaspillage alimentaire se classe au troisième rang mondial des pays pollueurs, derrière la Chine et les États-Unis.

Parmi les aliments les plus gaspillés, l'on trouve les fromages(30.3% ), le pain frais complet. (27.8%), le lait (25.2%), le yaourt (24.5%).

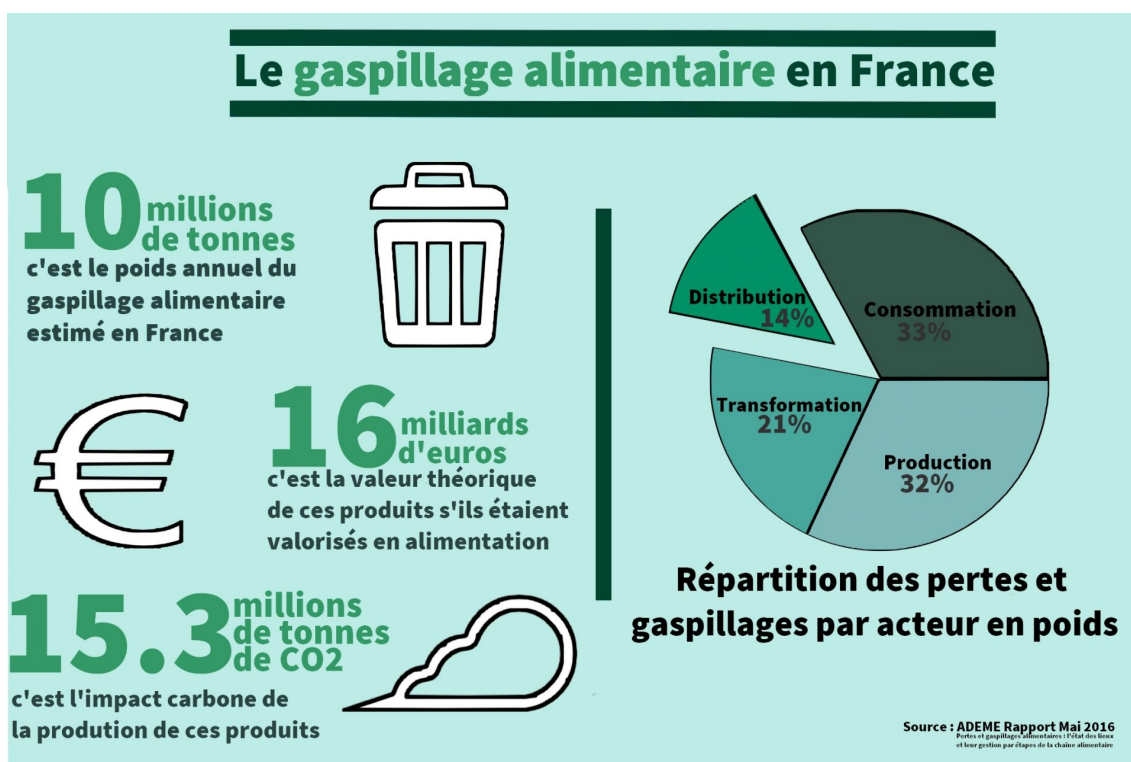


FIGURE II.2 – Diagramme de séquence gérer un jaimé

## II.3 Objectif du projet

À fin de participer à la réduction du gaspillage alimentaire au niveau des points de vente ou les magasins on propose une application progressive web de vente promotionnel pour faire de la publicité à ces produits alimentaire périssable qui sont généralement en promotion dans les différentes magasins pour leur assuré une vaste visibilité et attirer plus de gens à les acheter vu qu'ils sont à prix réduits.

Notre projet s'inscrit dans le cadre de e-commerce, ceci pour optimiser la vente des produit périssable dans les magasins ou ceux avec leur date de péremption proche en général, qui sont mis à prix réduits, mais leur client sont les seuls à les voir qui sont en général très peu pour acheter tous ces produits ou ils n'ont pas besoin de les consommés dans cette période.

Cela peut se faire grâce à notre application qui consiste à partager ces produits à prix réduit sur



une plateforme sociale qui permettra une vaste visibilité de ces produits de différents magasins par des individus clients et non clients de ces magasins.

L'application assure également de fournir différentes informations sur le magasin tel que le nom du magasin, la localisation, le numéro mobile pour faciliter le déplacement vers ce dernier.

Aussi des informations sur les produits telles que leur date de péremption une petite description, le prix avant la promotion et le prix après promotion ainsi une photo du produit.

## II.4 Spécification des besoins

### II.4.1 Analyse de l'existant

#### II.4.1.1 Identification des acteurs

**Définition** : Un acteur représente un rôle joué par une entité externe (utilisateur, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Dans le cas de notre application, nous avons quatre acteurs qui interagissent avec le système.

#### II.4.1.2 Description des rôles de chaque acteur :

Acteur	Description
Visiteur	C'est toute personne ou consommateur potentiel qui n'est pas inscrit à l'application. Il peut voir les publications, mais il ne peut pas faire des j'aimes ou des commentaires et il ne peut pas consulter la page de training.
Client	Le client est un visiteur inscrit à l'application, il peut voir toutes les publications, il peut faire des j'aime ou des commentaires et il peut consulter la page de training.
utilisateur	L'utilisateur est un client inscrit à l'application qui possède un magasin et il peut partager des produits alimentaires qui sont en promotion dans son magasin avec un prix réduit consultable par les clients.
Modérateur	Charger de superviser les signalisations et filtrer les publications. À accès à certaines fonctionnalités de l'application, qui permettent d'aider l'admin à gérer l'application.
Admin	Charger de la gestion de l'application.

#### II.4.1.3 Spécification des besoins fonctionnels

- Inscription.
- L'authentification.
- Faire une recherche.
- Gérer une publication.
- Gérer un commentaire.
- Gérer les signalisation .
- Gérer les comptes.

TABLE II.1 – Différents acteurs et leur descriptions

#### II.4.1.4 Spécification des besoins no fonctionnels

Notre future application doit répondre aux critères suivants :

**Convivialité** : L'application doit être facile à utiliser, les interfaces doivent être simples et ergonomiques.

**Rapidité de traitement** : Le temps d'exécution des opérations doit s'approcher du temps réel.

**Sécurité** : L'application doit assurer la sécurité et ses données doivent être protégées.

**Compatibilité** : Notre application est un jeu mobile fonctionnant sur Android et iOS.

#### II.4.1.5 Identification des cas d'utilisation

**Définition du cas d'utilisation** : Un cas d'utilisation (en anglais use case) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié.

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système.

L'objectif poursuivi par les cas d'utilisations est de permettre de décrire la finalité des interactions du système et de ces utilisateurs.

#### Liste des cas d'utilisations

N	Cas d'utilisations	Acteur
1	- Effectuer une Recherche	client/ utilisateur/ admin /modérateur
2	- S'inscrire	visiteur
		client
		utilisateur
3	- S'authentifier	visiteur
		client
		Modérateur
		admin
4	- Gérer une publication	modifier
		ajouter
		signaler
		archiver
5	- Gérer un commentaire	utilisateur
		client/admin/ modérateur
		modérateur
6	-Gestion des Comptes	ajouter
		modifier
		archiver
7	-Gestion des signalisations	bloquer
		réactiver
		admin
		client/ utilisateur/ admin /modérateur

TABLE II.2 – Liste des cas d'utilisations

### II.4.1.6 Description textuelle des cas d'utilisations

Dans ce qui suit, nous allons décrire l'ensemble des cas d'utilisation

1) Description de cas d'utilisation " **Effectuer une Recherche** "

Titre	Effectuer une recherche
Acteur	client/ utilisateur/ admin /modérateur
Description	L'acteur à la possibilité de faire une recherche dans le site.
Pré-condition	L'utilisateur doit être connecté au site.
Scénario nominal	1-L'acteur doit se connecter a l'application. 2-Saisir le(s) mot(s) clé. 3-Valider la recherche. 4-Le système affiche le résultat de la recherche.
Scénario Alternatif	Si le résultat non trouvé, alors le système affiche un message de non existence de résultat pour recherche.

2) Description de cas d'utilisation " **S'inscrire** "

Titre	S'inscrire.
Acteur	Utilisateur/ Client
Description	Permet à l'acteur de créer un compte sur l'application.
Pré-condition	Consulter l'application.
Scénario nominal	1-L'acteur choisir de s'inscrire. 2-Le système affiche le formulaire correspondant. 3- L'acteur remplit le formulaire. 4-Le système vérifie les données saisies. 5-Le système envoie un email de confirmation. 6-Après la confirmation par email, le système affiche l'espace du membre. 7-L'instance de cas d'utilisation se termine.
Scénario Alternatif	-Si un champ lui manque ou présente une erreur de saisie, le système affiche un message d'erreur. -L'acteur reviens a l'étape 3.

TABLE II.3 – Cas d'utilisations s'inscrire

3) Description de cas d'utilisation " **S'authentifier** "

Titre	S'authentifier.
Acteur	Utilisateur/ Client/Admin/Modérateur.
Description	Permet a l'acteur de s'identifié en saisissant son e-mail et mot de passe.
Pré-condition	L'acteur doit s'inscrire pour créer un compte.
Scénario nominal	1-L'acteur ouvre l'application. 2-Le système affiche la page d'authentification. 3-L'acteur saisit son e-mail et le mot de passe. 4-Le système vérifie l'existence des données. 5-Le système affiche la page d'accueil.
Scénario Alternatif	-Si l'e-mail ou mot de passe non valide. Le système affiche un message d'erreur. -L'acteur reviens a l'étape 3.

TABLE II.4 – Cas d'utilisations s'authentifier

4) Description de cas d'utilisation " **Gérer une publication** " : Ce cas d'utilisation permet à l'utilisateur de faire « modifier » ou « ajouter » ou « supprimer » la publication

**A. Modifie une publication**

Titre	Modifier une publication.
Acteur	Utilisateur/Admin/Modérateur.
Description	Permet a l'acteur de modifier la publication.
Pré-condition	L'utilisateur doit s'authentifier en succès.
Scénario nominal	1-L'acteur ouvre l'application. 2-Le système affiche la liste des publications apres on choisie celle qu'on veut. 3-Le système affiche le formulaire des informations de publication. 4-L'acteur modifie la publication. 5-Le système demande la validation de modification. 6-Le système verifie les données. 7-Le système enregistre la modification dans la base de données
Scénario Alternatif	1-Champ obligatoire vide ou non valide affiche le message d'erreur . 2-L'acteur reviens a l'étape 3.

TABLE II.5 – Cas d'utilisations modifier une publication

**B. Archiver une publication**

Titre	Archiver une publication .
Acteur	Utilisateur/Admin/Modérateur.
Description	Permet à l'Acteur de supprimer la publication de son profil, mais elle sera archiver dans la base de données.
Pré-condition	L'utilisateur doit s'authentifie en succès.
Scénario nominal	1-L'acteur ouvre l'application. 2-Le système affiche la liste des publications. 3- L'acteur choisir de supprimer une publication. 4-Le système demande la validation de suppression. 5-Le système supprime la publication du profil, et l'archive dans la base de données.
Scénario Alternatif	1-Si l'acteur annule la suppression. 2-L'acteur reviens a l'étape 2.

TABLE II.6 – Cas d'utilisations supprimer une publication

**C. Ajouter une publication**

Titre	Ajouter une publication.
Acteur	Utilisateur/Admin/Modérateur.
Description	Permet à l'acteur d'ajouter une publication.
Pré-condition	L'utilisateur doit s'authentifie en succès.
Scénario Nominal	1-L'acteur choisit l'ajout d'une nouvelle publication. 2-Le système affiche le formulaire à remplir. 3-L'acteur saisit les informations à remplir sur la nouvelle publication. 4-L'acteur valide l'ajout. 5-Le système vérifie les données. 6-Le système enregistre la publication dans la base de données.
Scénario Alternatif	1-Champ obligatoire vide ou non valide affiche le message d'erreur . 2-L'acteur reviens a l'étape 3.

TABLE II.7 – Cas d'utilisations ajouter une publication

**D. Signaler une publication**

Titre	Signaler une publication
Acteur	Client/Utilisateur
Description	Permet à l'acteur de signaler une publication.
Pré-condition	L'utilisateur doit s'authentifier en succès.
Scénario Nominal	1-L'acteur ouvre l'application. 2-Le système affiche la liste des publications. 3-L'acteur choisit de signaler une publication. 4-Le système demande la validation de signalisation. 5-Le système ajoute le signalement dans la base de données
Scénario Alternatif0	1-Si l'acteur annule la signalisation. 2-Le scénario reprend au point 2.

TABLE II.8 – Cas d'utilisations signaler une publication

5) Description de cas d'utilisation " **Gérer un commentaire**"**A. Ajouter un commentaire**

Titre	Ajouter un commentaire.
Acteur	Client/Utilisateur/Admin/Modérateur.
Description	Permet à l'utilisateur d'ajouter un commentaire.
Pré-condition	l'acteur doit s'authentifier en succès.
Scénario Nominal	1-L'acteur choisit l'ajout d'un commentaire. 2-Le système affiche le champ d'ajout . 3- L'acteur saisit sont commentaire. 4 -l'acteur valide sont commentaire. 5-Le système enregistre le commentaire dans la base de données.
Scénario Alternatif	1-Si L'acteur annule l'écriture de commentaire. 2-L'acteur reviens a l'étape 1.

TABLE II.9 – Cas d'utilisations ajouter un commentaire

**B. Modifier un commentaire**

Titre	Modifier un commentaire.
Acteur	Client/Utilisateur/Admin/Modérateur.
Description	Permet à l'acteur de modifier un commentaire.
Pré-condition	L'utilisateur doit s'authentifier en succès.
Scénario Nominal	1-L'acteur cherche sont commentaire. 2-L'acteur modifié le commentaire. 3- L'acteur validé le commentaire. 4-Le système enregistre le commentaire dans la base de données.
Scénario Alternatif	1-Si l'acteur annule la modification de commentaire. 2-L'acteur reviens à l'étape 1.

TABLE II.10 – Cas d'utilisations modifier un commentaire

### C. Archiver un commentaire

Titre	Archiver un commentaire
Acteur	Client/Utilisateur/Admin /Modérateur.
Description	Permet à l'Acteur de supprimer un commentaire d'une publication mais il sera archiver dans la base de données.
Pré-condition	L'utilisateur doit s'authentifie en succès.
Scénario nominal	1-L'acteur cherche sont commentaire. 2-L'acteur supprime le commentaire. 3-L'acteur valide la suppression. 4-Le système supprime le commentaire et l'archive dans la base de données.
Scénario Alternatif0	1-Si l'acteur annule la suppression de commentaire. 2-Cet enchaînement démarre au point 1.

TABLE II.11 – Cas d'utilisations Archiver un commentaire

- 6) Description de cas d'utilisation " **Gestion des Compte** " : Ce cas d'utilisation permet à l'admin de supprimer des comptes

### Modifier un compte

Titre	Modifier un compte .
Acteur	Utilisateur/Client/Admin/Modérateur.
Description	Permet a l'acteur de modifie sont profile.
Pré-condition	L'utilisateur doit s'authentifie en succès. .
Scénario nominal	1- L'acteur ouvre l'application. 2- L'acteur se pointe sur la page profile . 3-Le système affiche le formulaire de profile. 3- L'acteur modifie sont profile. 4-Le système verifie les données modifiée . 5-Le système valider les données. 6-Le système enregistrer les données modifiées dans la base données.
Scénario Alternatif	-Si il ya un champ obligatoire vide ou non valide affiche un message d'erreur . -L'acteur reviens a l'étape 3. .

TABLE II.12 – Cas d'utilisations Modifier un compte

**Bloquer un compte**

Titre	Bloquer un compte.
Acteur	Admin/Modérateur.
Description	Permet à l'acteur de Bloquer un compte.
Pré-condition	l'Acteur doit s'authentifier en succès.
Scénario Nominal	1-L'acteur ouvre l'application. 2-L'acteur se pointe sur la page profile d'ou il peut gérer les comptes. 3-Le système affiche la liste des comptes. 4-l'Acteur choisir le compte a bloquer. 5-Le système demande la validation de blocages. 6-Le système bloque le compte dans la base de données
Scénario Alternatif	1-Si l'acteur annule le blocages. 2-Le scénario reprend à l'étape 3.

TABLE II.13 – Cas d'utilisations Bloquer un compte

**Réactiver un compte**

Titre	Réactiver un compte.
Acteur	Admin/Modérateur.
Description	Permet à l'acteur de réactiver un compte.
Pré-condition	l'Acteur doit s'authentifier en succès.
Scénario Nominal	1-L'acteur ouvre l'application. 2-L'acteur se pointe sur la page profile d'ou il peut gérer les comptes. 3-Le système affiche la liste des comptes. 4-l'Acteur choisir le compte à réactiver. 5-Le système demande la validation de réactivation. 6-Le système réactive le compte dans la base de données
Scénario Alternatif	1-Si l'acteur annule la réactivation. 2-Le scénario reprend à l'étape 3.

TABLE II.14 – Cas d'utilisations Réactiver un compte



- 7) Description de cas d'utilisation " **Gestion des signalisations** " : Ce cas d'utilisation permet à l'admin de bloquer des comptes ou des publications

Titre	Gestion des signalisations
Acteur	Admin/ modérateur
Description	Permet à l'acteur de bloquer une publication signaler, ou un compte de client.
Pré-condition	L'acteur doit s'authentifier en succès.
Scénario Nominal	1-L'acteur ouvre l'application. 2-Le système affiche la liste des publications signalé. 3-L'acteur choisir de bloquer une publication ou un compte. 4-Le système demande la validation de blocage. 5-Le système ajoute le blocage dans la base de données
Scénario Alternatif0	1-L'acteur annule le blocage. 2-Le scénario reprend au point 2.

TABLE II.15 – Cas d'utilisations Gestion des signalisations

## II.5 Conclusion

Ce chapitre nous a permis de détailler les spécifications du projet, d'identifier les acteurs et leurs cas d'utilisation et nous avons spécifié les différents besoins, fonctionnels et non-fonctionnels. Ces derniers ont été décrits sous forme textuelle nécessaire pour entamer la conception dans la prochaine partie de notre projet, tout , ça nous allons l'aborder dans le prochain chapitre.

# Chapitre III

## Conception de l'application

## III.1 Introduction

Nous allons, dans ce chapitre, présenter la conception détaillée de notre application, où nous présentons les différents diagrammes des cas d'utilisation, de séquence système, et de classes.

## III.2 Diagramme de cas d'utilisation

Les principales fonctionnalités de notre application sont illustrées par les diagrammes de cas d'utilisation suivants :

### III.2.1 Diagramme de cas d'utilisation associé à l'Utilisateur

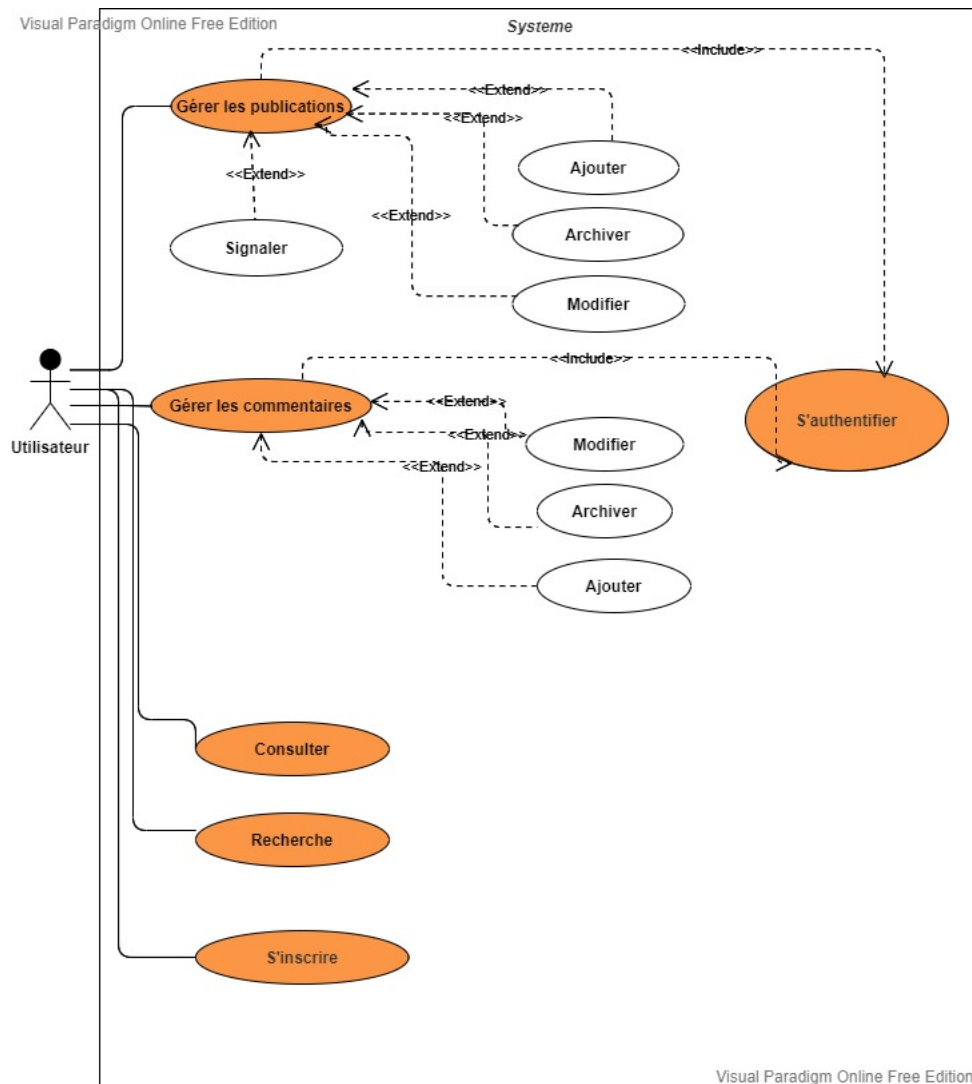


FIGURE III.1 – Diagramme de cas d'utilisation Utilisateur

### III.2.2 Diagramme de cas d'utilisation associé à l'Admin et le modérateur

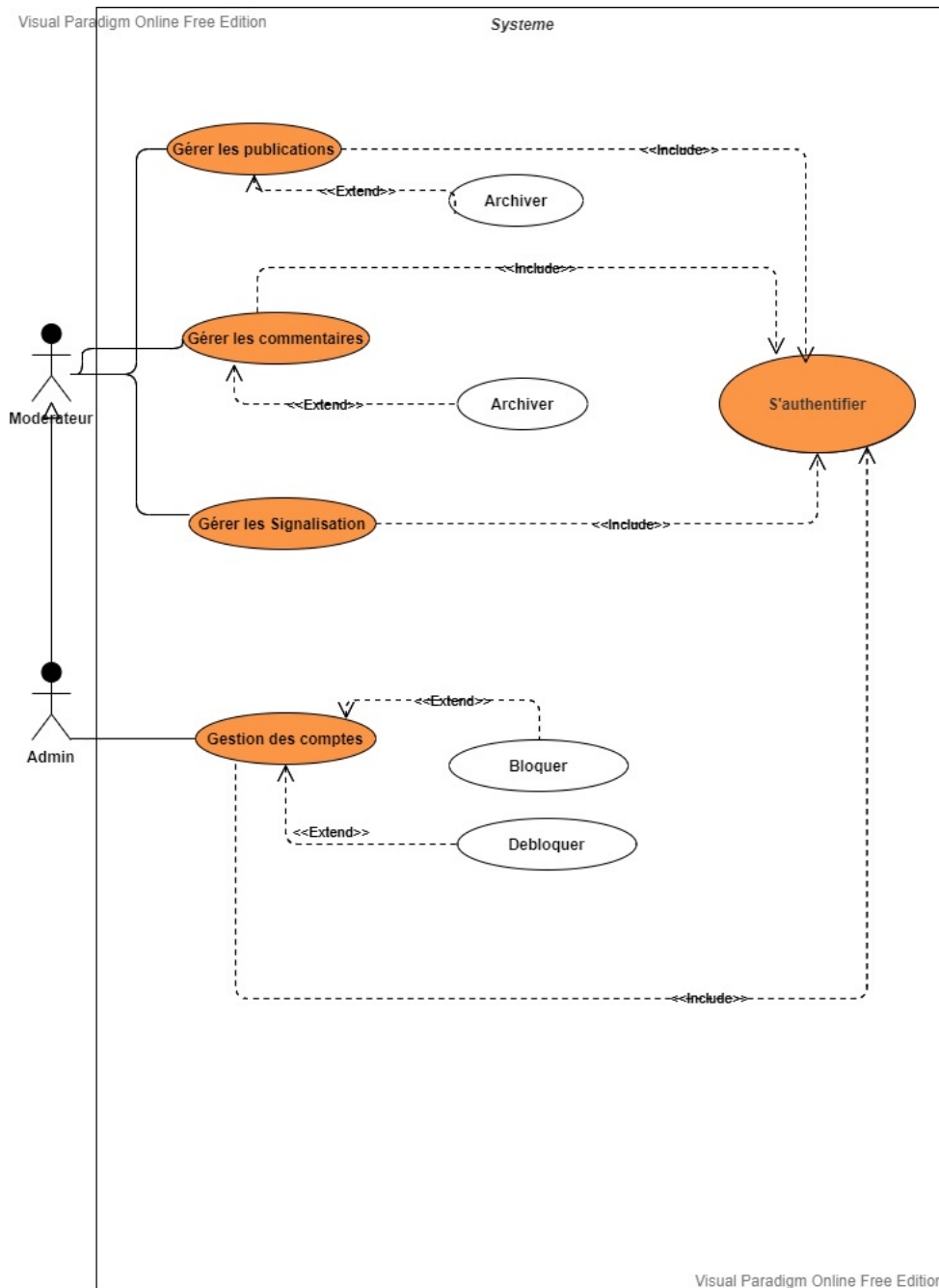


FIGURE III.2 – Diagramme de cas d'utilisation admin et modérateur

### III.2.3 Diagramme de cas d'utilisation associé à l'Client et le visiteur

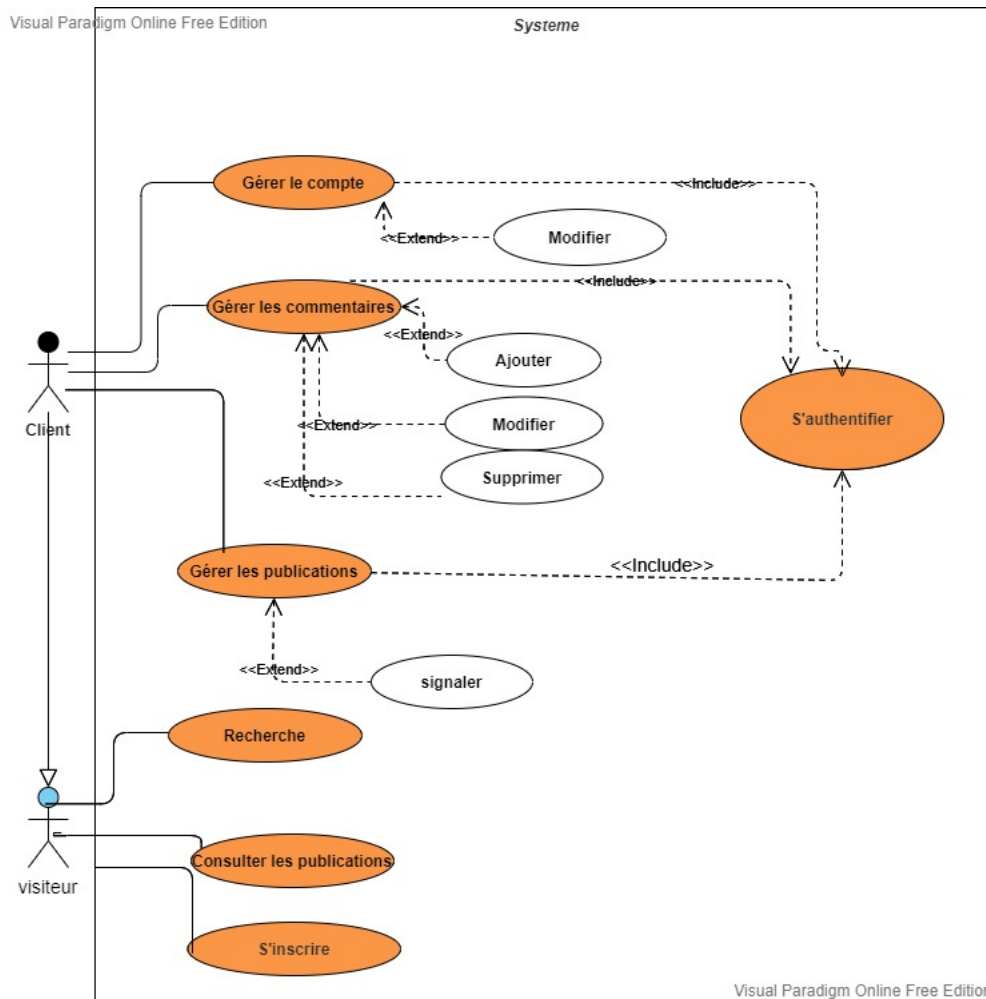


FIGURE III.3 – Diagramme de cas d'utilisation client et visiteur

## III.3 Diagramme de séquence

### III.3.1 Étude de cas d'utilisation

Dans ce qui suit nous allons faire une étude détaillée des cas d'utilisation décrits au paravent à travers le diagramme de séquence, nous allons présenter un diagramme de séquence pour chaque cas d'utilisation.

III.3.1.1 Diagramme de séquence «s'authentifie»

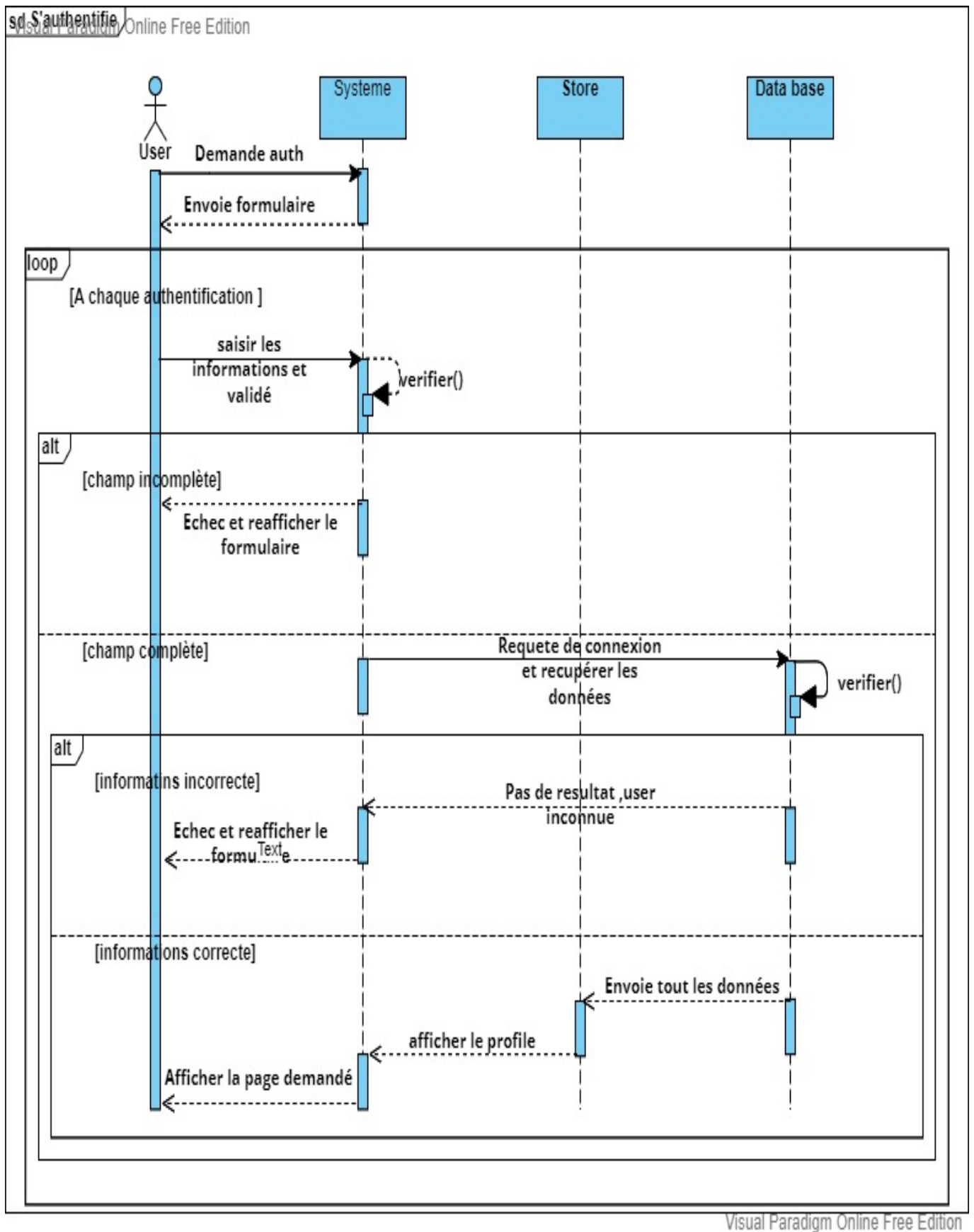


FIGURE III.4 – Diagramme de séquence s'authentifie

III.3.1.2 Diagramme de séquence «gérer un produit»

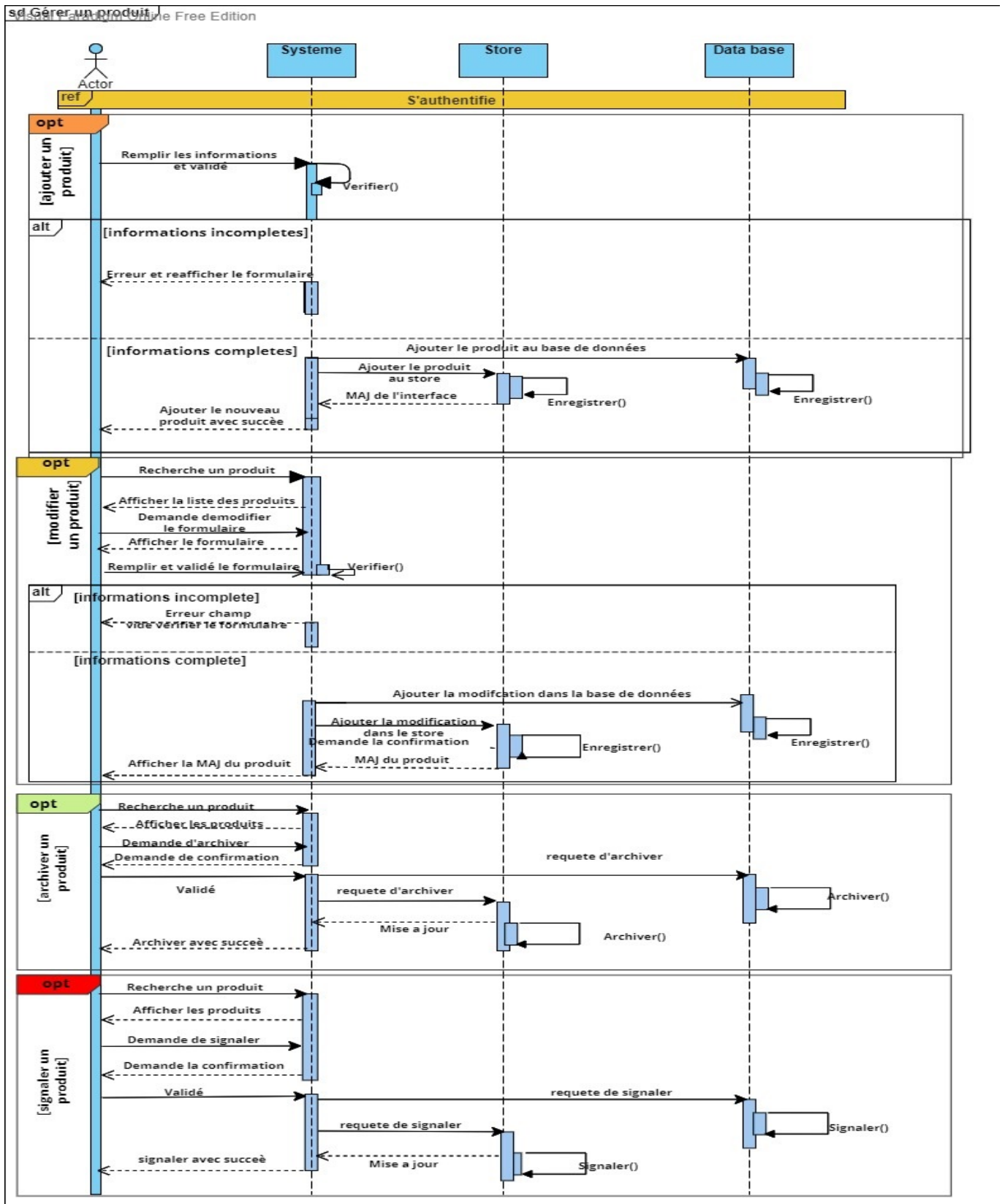


FIGURE III.5 – Diagramme de séquence gérer un produit

III.3.1.3 Diagramme de séquence « gérer un compte »

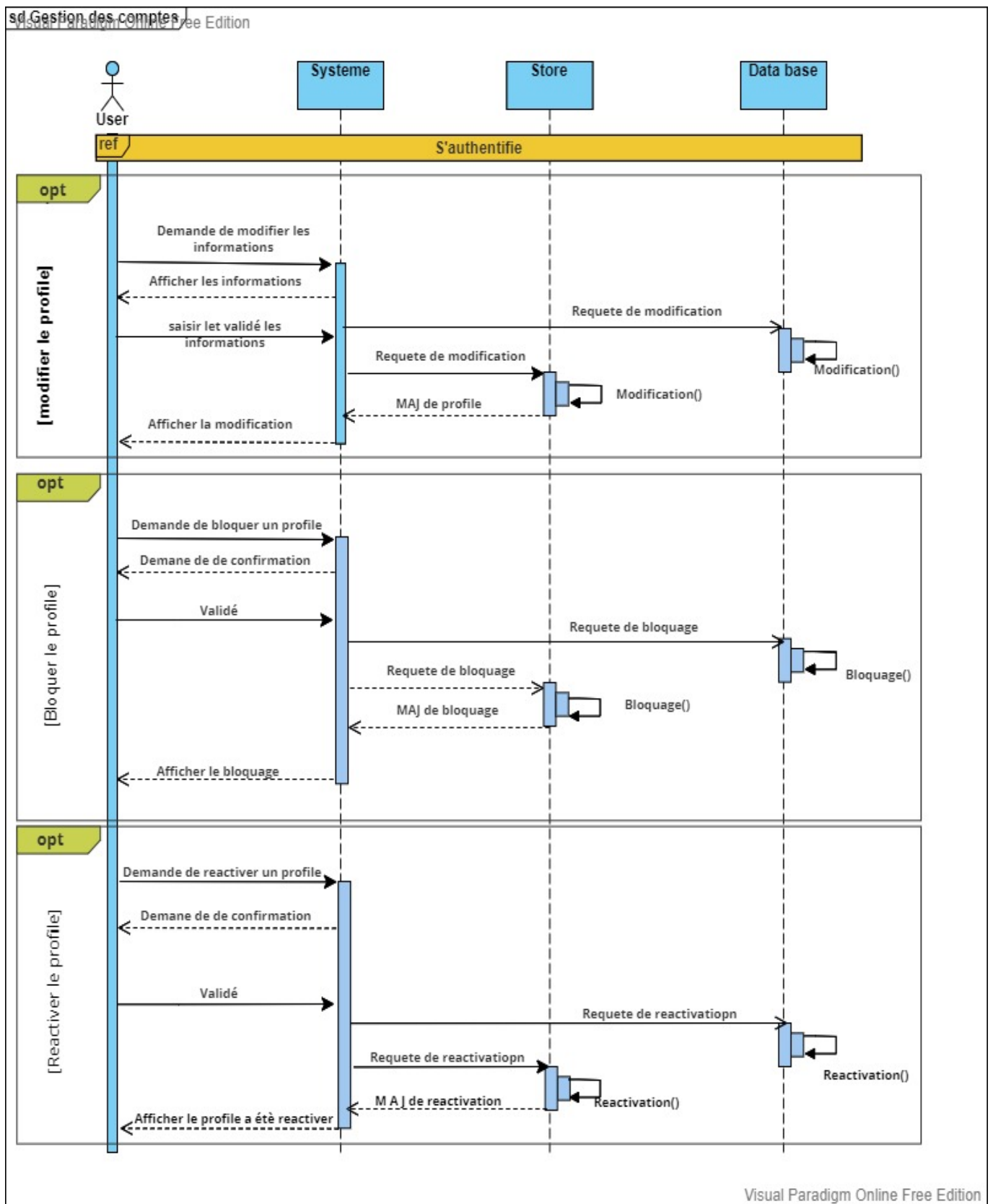


FIGURE III.6 – Diagramme de séquence gérer un compte



III.3.1.4 Diagramme de séquence «gérer un commentaire»

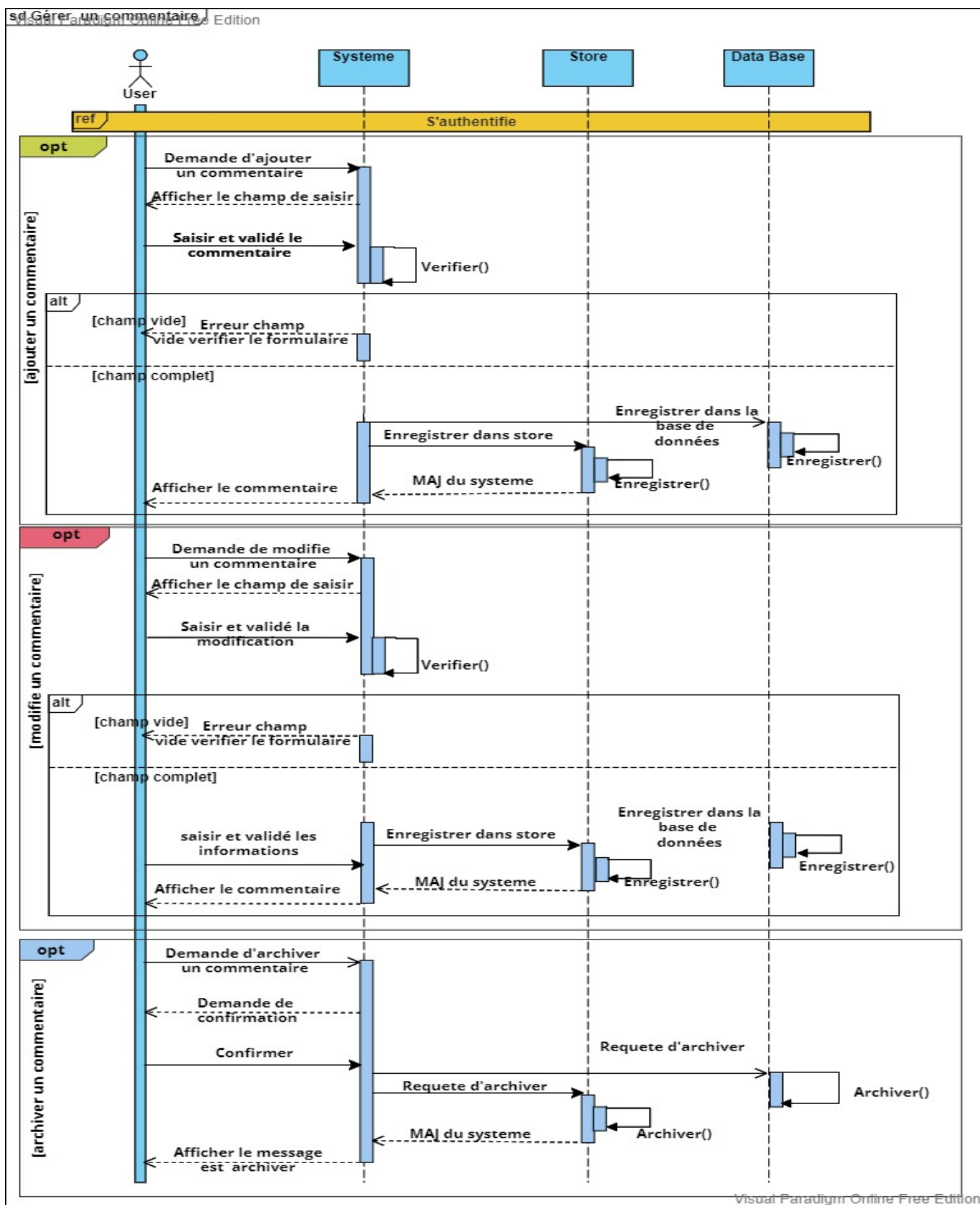


FIGURE III.7 – Diagramme de séquence gérer un commentaire

III.3.1.5 Diagramme de séquence «gérer les signalisation»

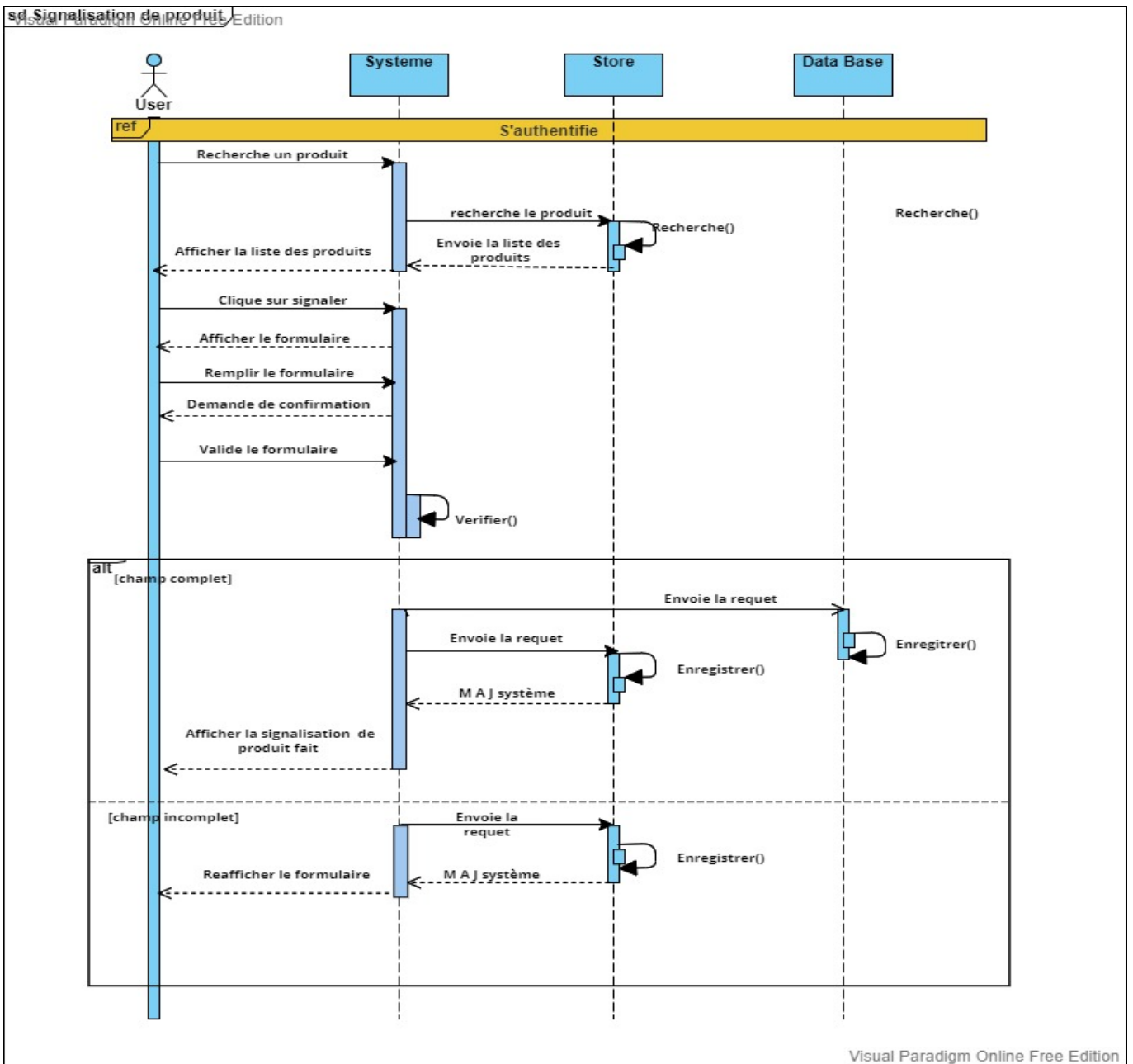


FIGURE III.8 – Diagramme de séquence gérer les signalisation

III.3.1.6 Diagramme de séquence «Rechercher un produit »

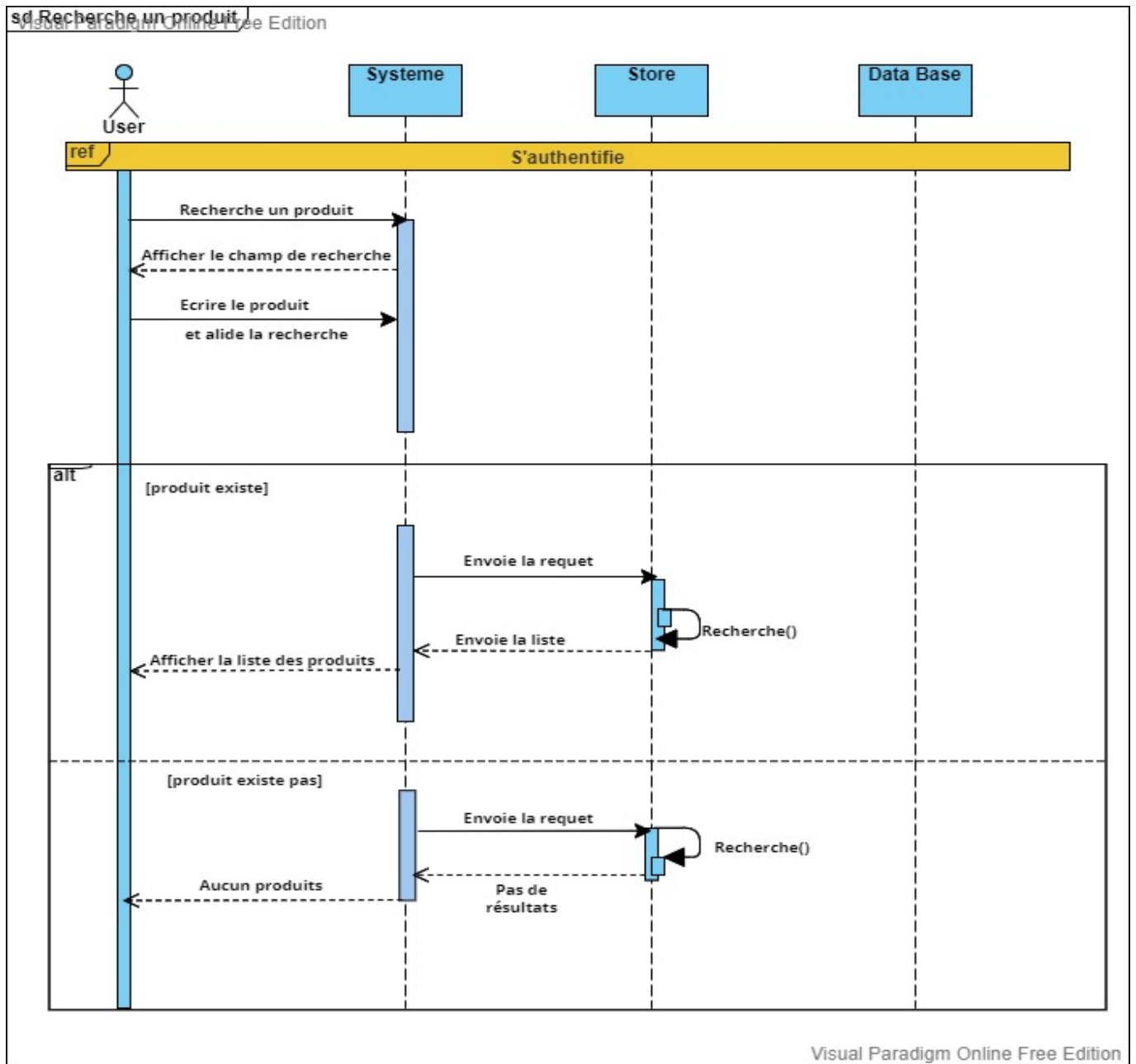


FIGURE III.9 – Diagramme de séquence Rechercher un produit

III.4 Conclusion

Dans ce chapitre, nous avons suivi une démarche de modélisation pour développer notre application, en se basant sur le langage UML.

En premier lieu, nous avons pu spécifier et analyser les besoins et les différents cas d'utilisation, Par la suite en phase de conception nous avons défini la conception des diagrammes de séquences et nous avons élaboré le diagramme de classes.

Dans le chapitre suivant nous entamerons l'étape de réalisation de notre application.

# Chapitre IV

## Réalisation

## IV.1 Introduction

Dans ce dernier chapitre, nous allons déterminer tous les outils de développement de notre application, nous allons aussi présenter les grandes étapes qui ont servi à la réalisation de notre application et ses interfaces principales.

## IV.2 Environnement et outils de programmation :

### IV.2.1 Html

HTML signifie HyperText Markup Language qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure.



FIGURE IV.1 – Logo de HTML

### IV.2.2 CSS

Les CSS (Cascading Style Sheets en anglais, ou « feuilles de style en cascade ») sont le code utilisé pour mettre en forme une page web.

### IV.2.3 JavaScripts cote client

JavaScript est un langage de script ou de programmation qui permet d'implémenter des programmes complexes sur les pages web, ces scripts peuvent être écrits directement sur une page web HTML et exécutés automatiquement quand la page se charge.



FIGURE IV.2 – Logo de css

### IV.2.4 JavaScript (ES6)

ES6 fait référence à la version 6 du langage de programmation ECMA Script. ECMA Script est le nom standardisé de JavaScript, et la version 6 est la prochaine version après la version 5, qui a été publiée en 2011. Il s'agit d'une amélioration majeure du langage JavaScript et ajoute de nombreuses autres fonctionnalités destinées à faciliter le développement de logiciels à grande échelle.[20]



FIGURE IV.3 – Logo de JavaScript

### IV.2.5 Node JS

Node.js est un environnement open source permettant de développer des applications multiplateformes. Basé sur le V8 de Google, le moteur d'exécution JavaScript utilisé dans Chrome, Node.js traite donc les données en JavaScript. Node.js vous permet d'utiliser JavaScript côté serveur, en dehors du navigateur avant de pouvoir exécuter votre JavaScript sur le serveur, le code a besoin d'être interprété. C'est précisément ce que fait Node.js en utilisant le moteur V8 de Google, celui qui est utilisé par le navigateur Chrome. D'autre part, Node.js intègre un nombre important de modules utiles, ainsi, vous n'avez pas à écrire tout le code à partir de zéro. Ainsi, Node.js comprend deux éléments distincts : un contexte d'exécution et une bibliothèque.[9]



FIGURE IV.4 – Logo de Node JS

### IV.2.5.1 Npm

NPM (Node Package Manager) comme son nom l'indique est le "package manager" officiel de l'univers JavaScript (frontend / backend). Il est installé automatiquement lors de l'installation de NodeJS.



FIGURE IV.5 – Logo de Npm

### IV.2.5.2 Node-sass

Liaisons Node.js à libsass \*. Il s'agit d'une bibliothèque qui fournit la liaison de Node.js à LibSass, la version C du préprocesseur de feuille de style populaire, Sass. Il vous permet de compiler nativement des fichiers .scss en css à une vitesse incroyable et automatiquement via un middleware de connexion.[10]



FIGURE IV.6 – Logo de node-sass

### IV.2.5.3 Js-cookie

Une API JavaScript simple et légère pour gérer les cookies.



FIGURE IV.7 – Logo de js-cookie

### IV.2.5.4 Multer

Multer est un middleware node.js pour la gestion des données multipart/form, qui est principalement utilisé pour télécharger des fichiers.[12]



FIGURE IV.8 – Logo de multer



### IV.2.6 Redux

Redux est une bibliothèque JavaScript open source utilisée pour gérer l'état de l'application. React utilise Redux pour créer l'interface utilisateur. Il permet aux composants React de lire les données d'un magasin Redux et d'envoyer des actions au magasin pour mettre à jour les données. Redux aide les applications à évoluer en fournissant un moyen judicieux de gérer l'état via un modèle de flux de données unidirectionnel. [11]



FIGURE IV.9 – Logo de Redux

### IV.2.7 Axios

Axios est une bibliothèque basée sur des promesses utilisée avec Node.js et votre navigateur pour effectuer des requêtes HTTP JavaScript asynchrones.

Axios nous permet de communiquer facilement avec les API dans nos applications React. Bien que cela puisse également être réalisé par d'autres méthodes comme la récupération ou AJAX, Axios peut fournir un peu plus de fonctionnalités qui vont très loin avec les applications qui utilisent React.[9]



FIGURE IV.10 – Logo de axios

## IV.2.8 React

React est une bibliothèque JavaScript frontale à code source ouvert permettant de créer des application web monopage. Développée par Facebook depuis 2013.[11]

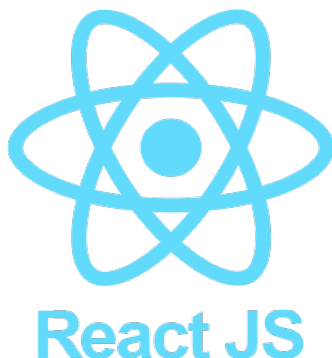


FIGURE IV.11 – Logo de react

## IV.2.9 Github

GitHub est un service basé sur le cloud qui héberge un système de contrôle de version (VCS) appelé Git. Il permet aux développeurs de collaborer et d'apporter des modifications aux projets partagés tout en gardant un suivi détaillé de leur progression.[13]



FIGURE IV.12 – Logo de Github

## IV.3 Serveur d'application

### IV.3.1 ExpressJS

Express est un framework qui s'exécute dans Node.js et qui permet à un développeur de créer et de maintenir des serveurs robustes. Il est livré avec de nombreuses fonctionnalités intégrées, mais dispose également d'une gigantesque suite de modules complémentaires tiers que les développeurs peuvent utiliser pour fournir de meilleures fonctionnalités, une sécurité accrue et une vitesse améliorée.[16]



FIGURE IV.13 – Logo de expressJS

### IV.3.2 MongoDB

MongoDB est une base de données NoSQL orientée document. Elle se distingue des bases de données relationnelles par sa flexibilité et ses performances.

### IV.3.3 MongoDB compass

MongoDB Compass est une interface graphique pour MongoDB. Il est également connu sous le nom d'interface graphique MongoDB. MongoDB permet aux utilisateurs d'analyser le contenu de leurs données stockées sans aucune connaissance préalable de la syntaxe des requêtes MongoDB. Lorsque nous explorons l'exploration de nos données dans l'environnement visuel, nous pouvons utiliser l'interface graphique de Compass pour optimiser les performances, gérer les index et implémenter la validation des documents.[14]



FIGURE IV.14 – Logo de mongoDB

### IV.3.4 JSON

JSON (JavaScript Object notation) : un format léger qui permet de décrire des objets en JavaScript, il est principalement utilisé pour faire des échanges entre JS et un serveur web, JSON se base sur deux structures : Une collection de couple nom/valeur : nom : valeur ; Un tableau qui est une collection de valeurs ordonnées [valeur, valeur] JSON est un format de données. Autrement dit, c'est une façon de stocker des informations, un peu comme une base de données, il est maintenant lié à JavaScript qui inclut un objet JSON, et de nombreux développeurs l'incorporent quasiment comme un sous-ensemble du langage.[21]



FIGURE IV.15 – Logo de JSON

### IV.3.5 Postman

Postman est un logiciel permettant de créer et de tester des requêtes HTTP. Il vous permet de les personnaliser dans les plus fins détails grâce à une interface ergonomique et intuitive. Vous pouvez choisir la méthode de la requête, entrer l'URL du serveur que vous voulez interroger, et rajouter tous les paramètres possibles pour une requête HTTP. Le logiciel tient un historique de vos requêtes. Il est très utile pour tester une api.[15]



POSTMAN

FIGURE IV.16 – Logo de postman

## IV.4 Sécurité d'application

### IV.4.1 Cors

CORS est une technique qui permet le partage de ressources entre des scripts exécutés sur un client de navigateur et des ressources d'origine différente. Le partage des ressources cross-origin utilise des en-têtes HTTP supplémentaires pour demander au navigateur d'autoriser une application Web s'exécutant sur un domaine à accéder à des ressources spécifiques à partir d'un serveur d'une origine différente. Il y a quelques années, les requêtes AJAX et le partage des ressources des polices Web étaient limités par la politique de même origine. Cependant, aujourd'hui, les développeurs peuvent utiliser CORS pour faciliter la communication entre domaines à partir du navigateur. Le protocole CORS prend en charge les transferts de données d'origine croisée et les demandes entre les serveurs Web et les navigateurs. CORS détermine si

des demandes d'origine sont faites ou non. Et avant d'aller plus loin, examinons les deux types de requêtes différents.[17]



FIGURE IV.17 – Logo de Cors

#### IV.4.2 Bcrypt

Bcrypt est une fonction de hachage de mot de passe conçue par Niels Provos et David Mazières, basée sur le chiffrement Blowfish, et présentée à USENIX en 1999.

Bcrypt est un utilitaire de chiffrement de fichiers multiplateforme. Les fichiers cryptés sont portables sur tous les systèmes d'exploitation et processeurs pris en charge. Les phrases secrètes doivent comporter entre 8 et 56 caractères et sont hachées en interne sur une clé de 448 bits. Cependant, tous les caractères fournis sont significatifs. Plus votre phrase secrète est forte, plus vos données sont sécurisées.[18]



FIGURE IV.18 – Logo de bcrypt

### IV.4.3 Jsonwebtoken

Le jeton Web JSON (JWT), prononcé "jot", est une norme ouverte (RFC 7519) qui définit un moyen compact et autonome de transmettre en toute sécurité des informations entre les parties en tant qu'objet JSON. Encore une fois, JWT est une norme, ce qui signifie que tous les jetons sont des jetons, mais que tous les jetons ne sont pas des JWT.

En raison de sa taille relativement petite, un JWT peut être envoyé via une URL, via un paramètre POST ou dans un en-tête HTTP, et il est transmis rapidement. Un JWT contient toutes les informations requises sur une entité pour éviter d'interroger une base de données plusieurs fois. Le destinataire d'un JWT n'a pas non plus besoin d'appeler un serveur pour valider le jeton.[19]

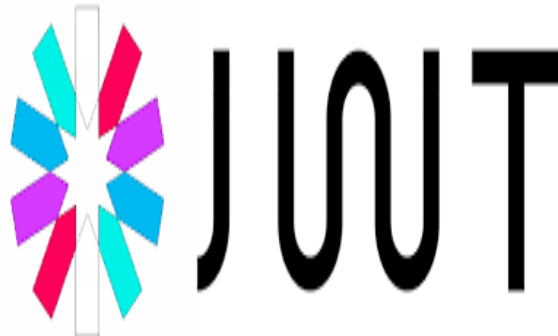


FIGURE IV.19 – Logo de Jsonwebtoken

## IV.5 Architecture de l'Application

Cette figure nous montre comment les différents outils de l'environnement MERN s'interagissent entre eux.

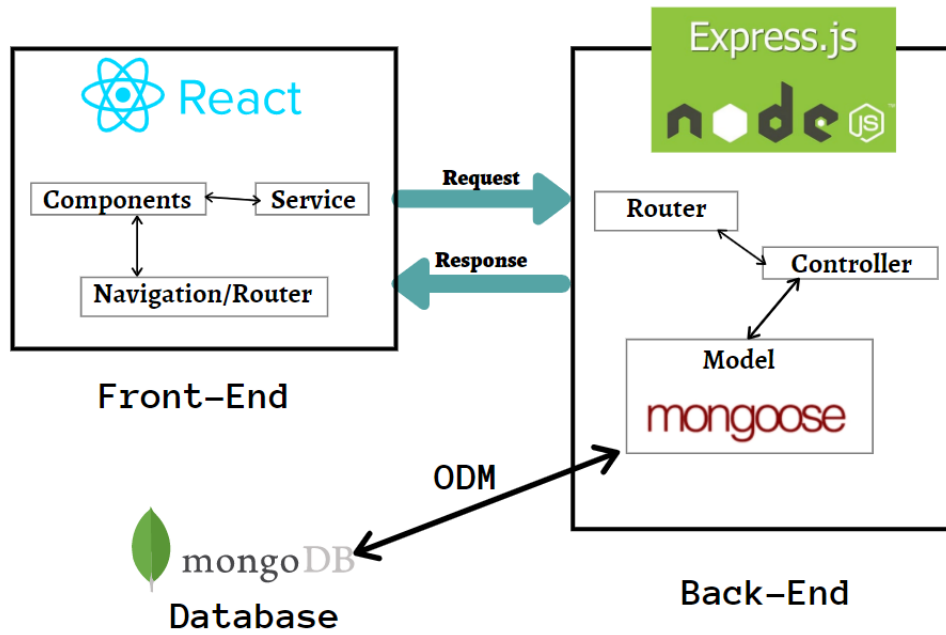


FIGURE IV.20 – Architecture de l'Application

### IV.5.1 Architecture de redux

Cette figure nous montre comment les différents outils de l'environnement MERN s'interagissent avec Redux.

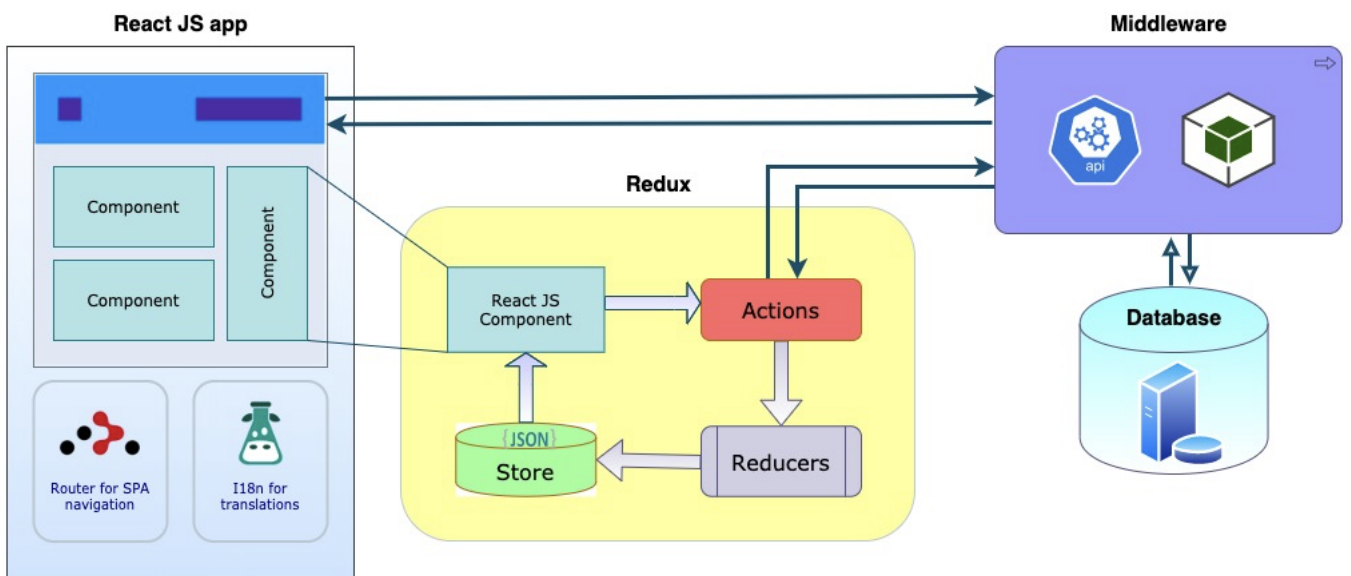
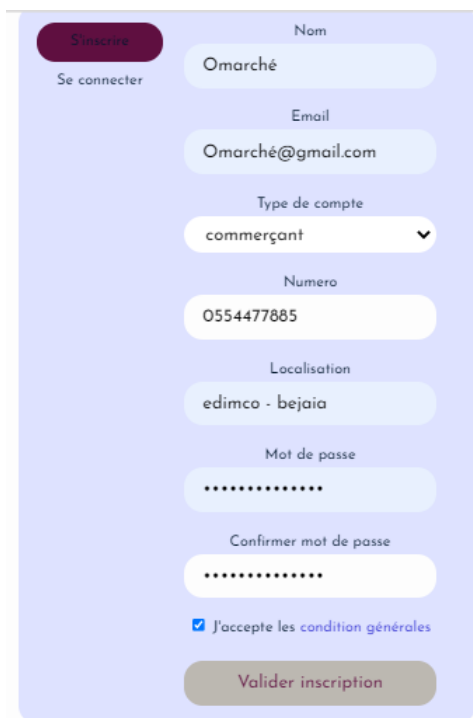


FIGURE IV.21 – Architecture de redux

## IV.6 Principales captures d'interfaces

### IV.6.1 Interface Inscription :

La figure ci-dessous représente l'interface d'inscription pour le commerçant et l'utilisateur



The screenshot shows a registration form for a merchant. It features a dark red button labeled 'S'inscrire' and a light blue button labeled 'Se connecter'. The form fields are: 'Nom' (Omarché), 'Email' (Omarché@gmail.com), 'Type de compte' (commerçant), 'Numero' (0554477885), 'Localisation' (edimco - bejaia), 'Mot de passe' (masked with dots), and 'Confirmer mot de passe' (masked with dots). There is a checkbox for 'J'accepte les condition générales' and a 'Valider inscription' button at the bottom.

FIGURE IV.22 – Captures d'interfaces d'nscription

### IV.6.2 Interface d'authentification :

La figure ci-dessous représente l'interface d'authentification.



The screenshot shows an authentication interface. It features a dark red button labeled 'S'inscrire' and a light blue button labeled 'Se connecter'. The form fields are: 'Eamil' (Omarché@gmail.com) and 'Mot de passe' (masked with dots). There is a 'se connecter' button at the bottom.

FIGURE IV.23 – Captures d'interfaces d'authentification



### IV.6.3 Interface profile marchand

La figure ci-dessous représente l'interface de profile pour le marchand, où il peut changer ses informations personnelles comme le numero et la localisation de son magasin.



FIGURE IV.24 – Captures d'interfaces profile de marchand

#### IV.6.4 Interface Profile d'utilisateur :

La figure ci-dessous représente l'interface de profile pour le utilisateur, où il peut changer ses information personnelles comme la photo et la biographie.

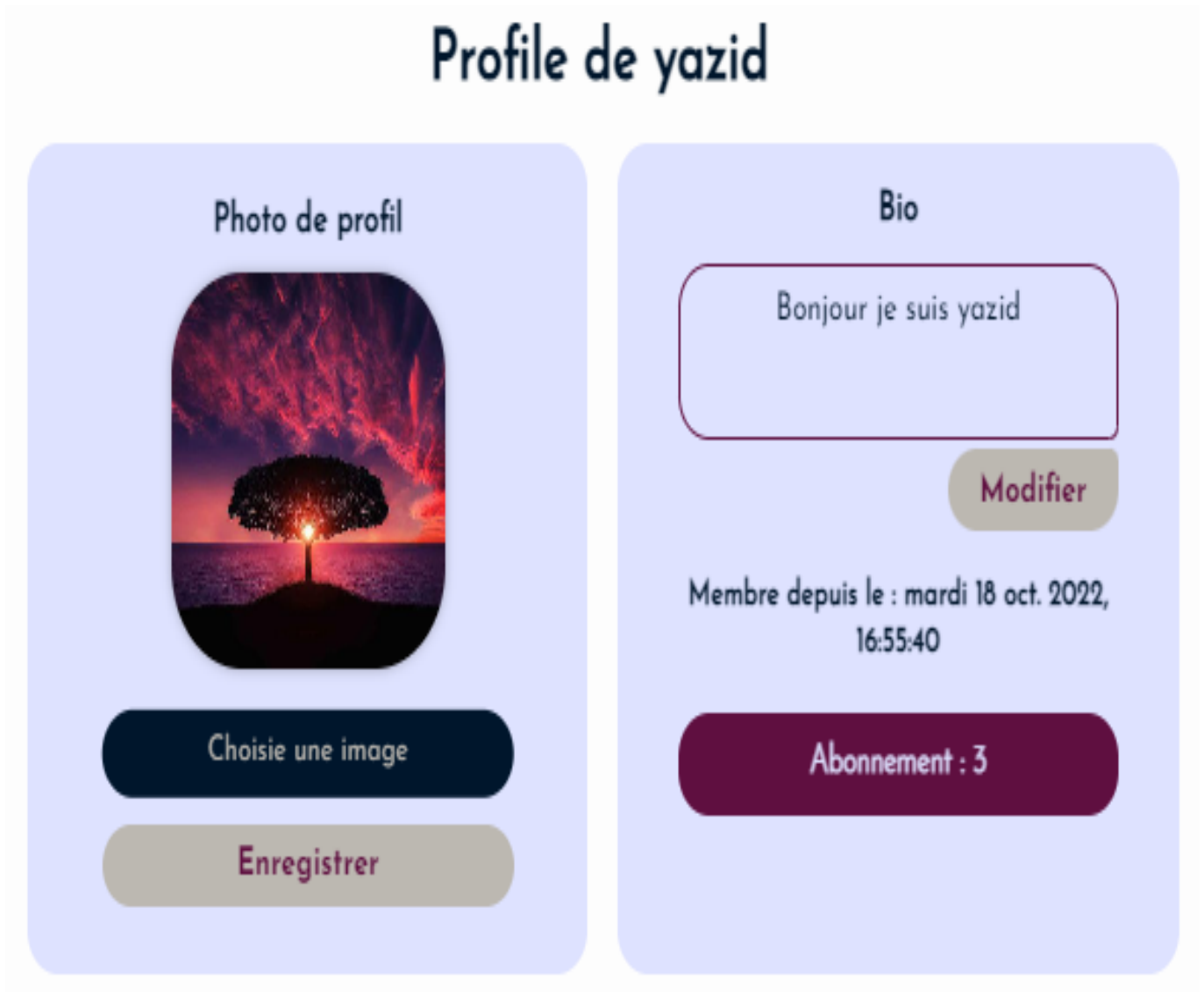


FIGURE IV.25 – Captures d'interfaces profile d'utilisateur

### IV.6.5 Interface page d'accueil d'utilisateur :

La figure ci-dessous représente l'interface de page d'accueil d'utilisateur où il peut voir les différents produits partager par les commerçants.

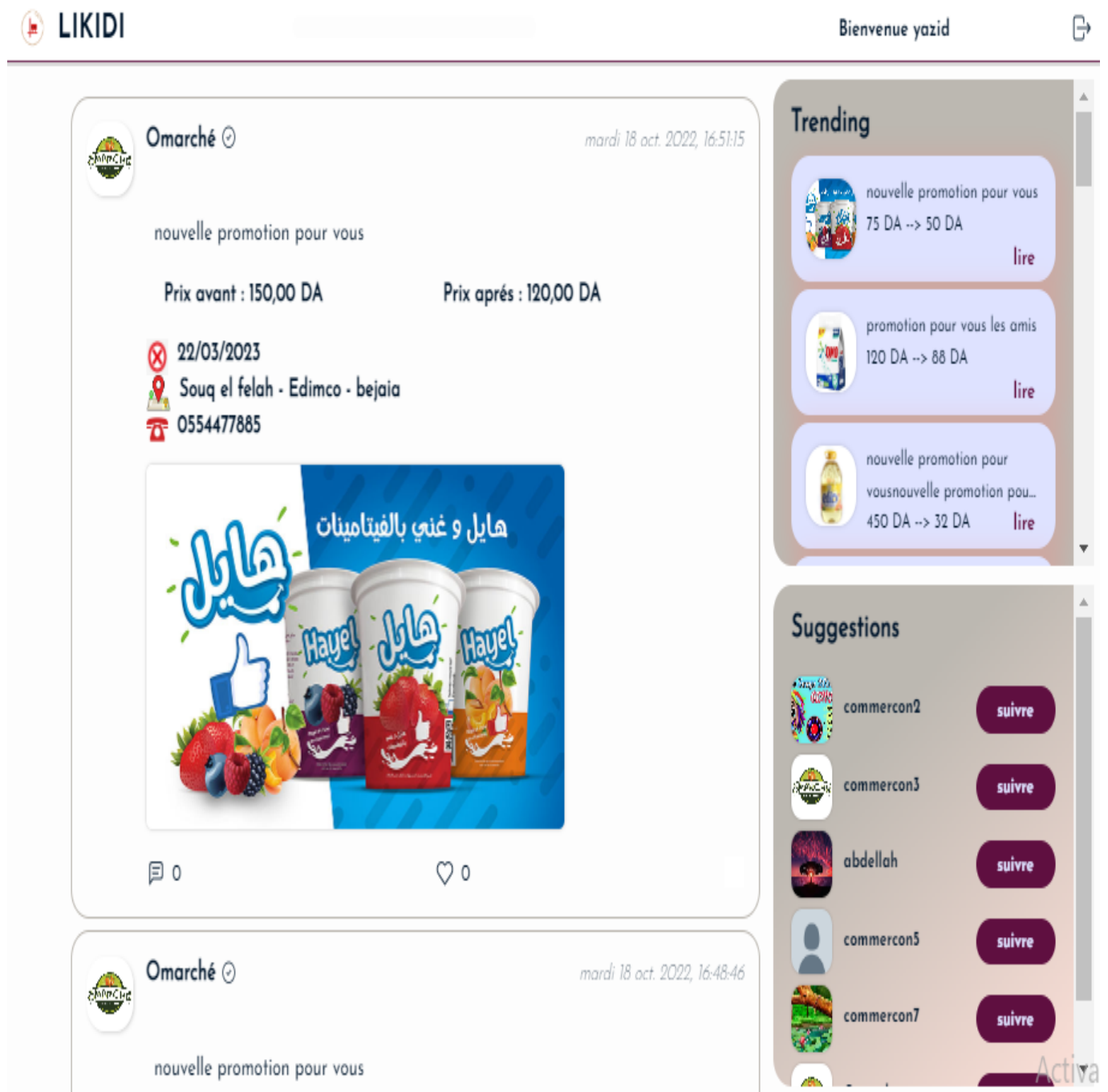


FIGURE IV.26 – Captures d'interfaces page d'accueil d'utilisation

### IV.6.6 Interface age d'accueil marchand

La figure ci-dessous représente l'interface de page d'accueil d'utilisateur où il peut partager différents produits en promotion.

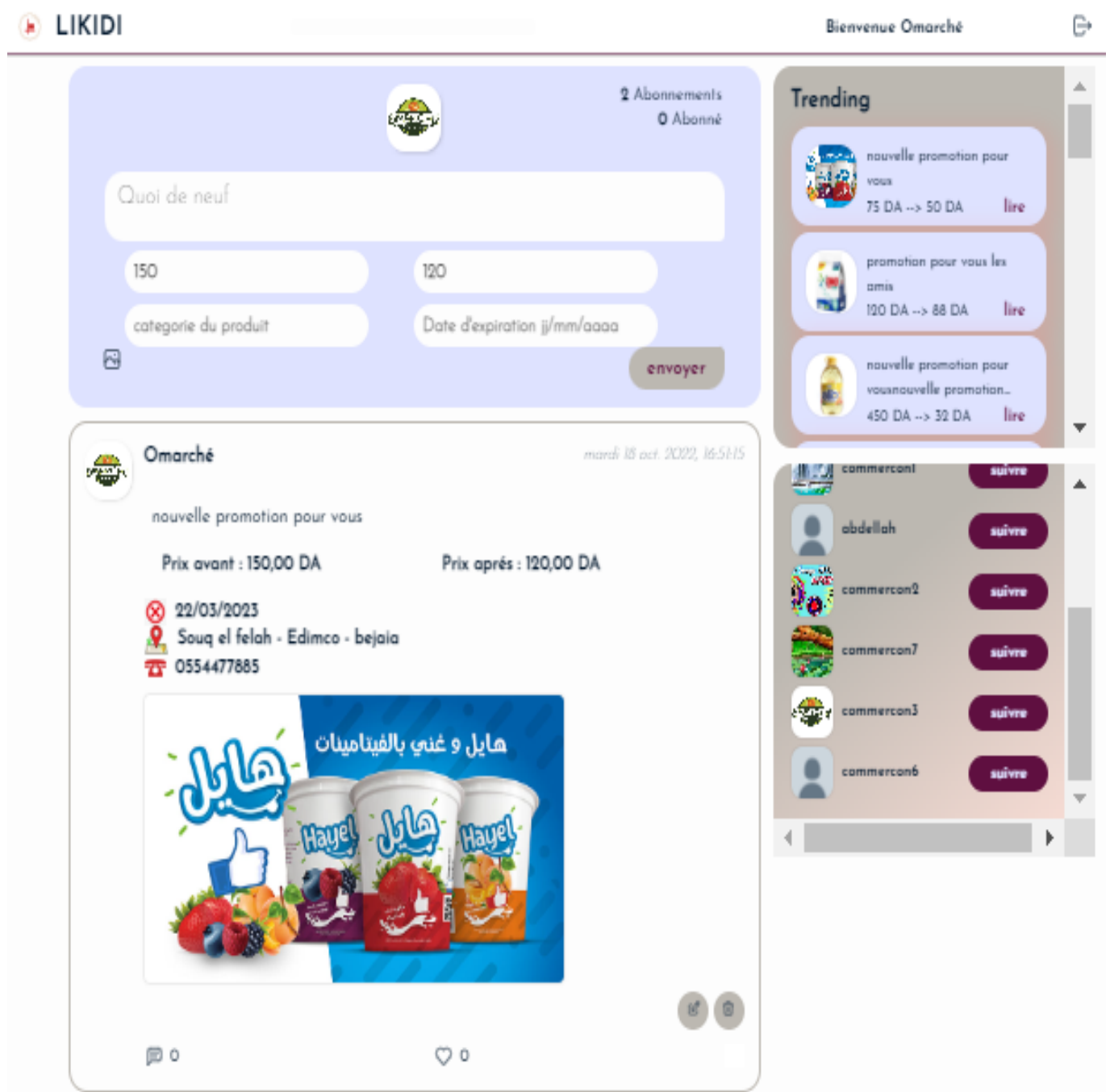


FIGURE IV.27 – Captures d'interfaces page d'accueil marchand

### IV.6.7 Interface gestion des comptes :

La figure ci-dessous représente l'interface de profile pour l'admin d'où il peut supprimer Les comptes indésirables.



FIGURE IV.28 – Captures d'interfaces gestion des comptes

### IV.6.8 Captures d'interfaces accueil commerçant sur smartphone

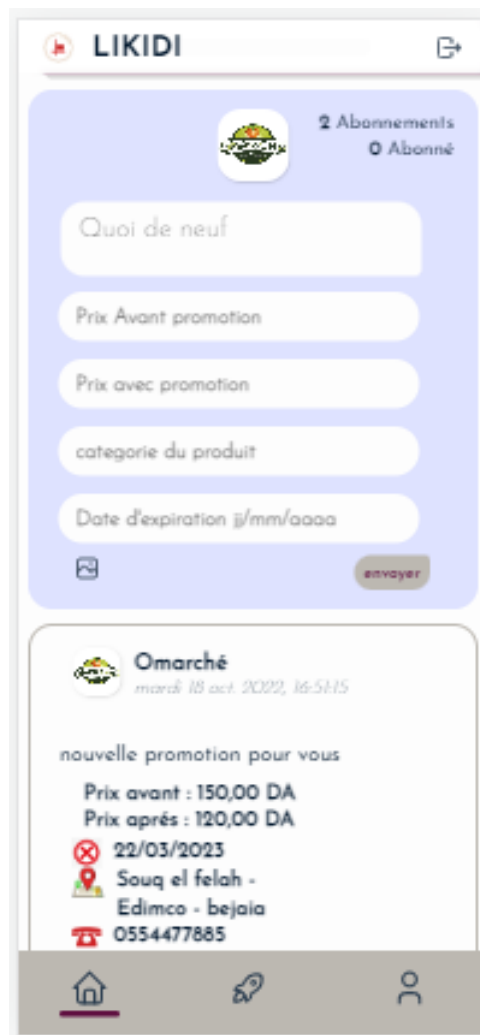


FIGURE IV.29 – Captures d'interfaces accueil commerçant sur smartphone

### IV.6.9 Interface profil commerçant sur smartphone

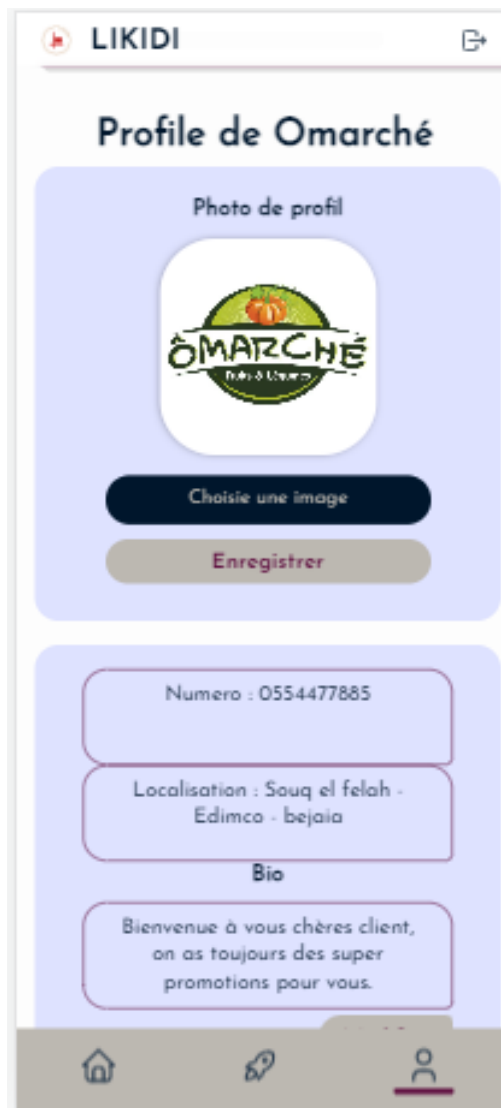


FIGURE IV.30 – Captures d'interfaces profil commerçant sur smartphone

### IV.6.10 Interface accueil utilisateur sur smartphone



FIGURE IV.31 – Captures d’interfaces accueil utilisateur sur smartphone



## IV.7 Lighthouse test

Lighthouse est un outil open source automatisé pour améliorer la qualité des pages Web. Vous pouvez l'exécuter sur n'importe quelle page Web, publique ou nécessitant une authentification. Il a des audits pour les performances, l'accessibilité, les applications Web progressives, le référencement et plus encore.

Nous pouvons Lighthouse utiliser pour donner un score de préparation à notre application Web. Nous exécuterons le test de l'application Web progressive et interpréterons les résultats. En particulier, il est essentiel qu'une PWA soit exécutée sur HTTPS en production - sinon, elle ne fonctionnera pas du tout comme une PWA. La figure en bas montre que notre application, à passer avec succès le teste de PWA fourni par Lighthouse.

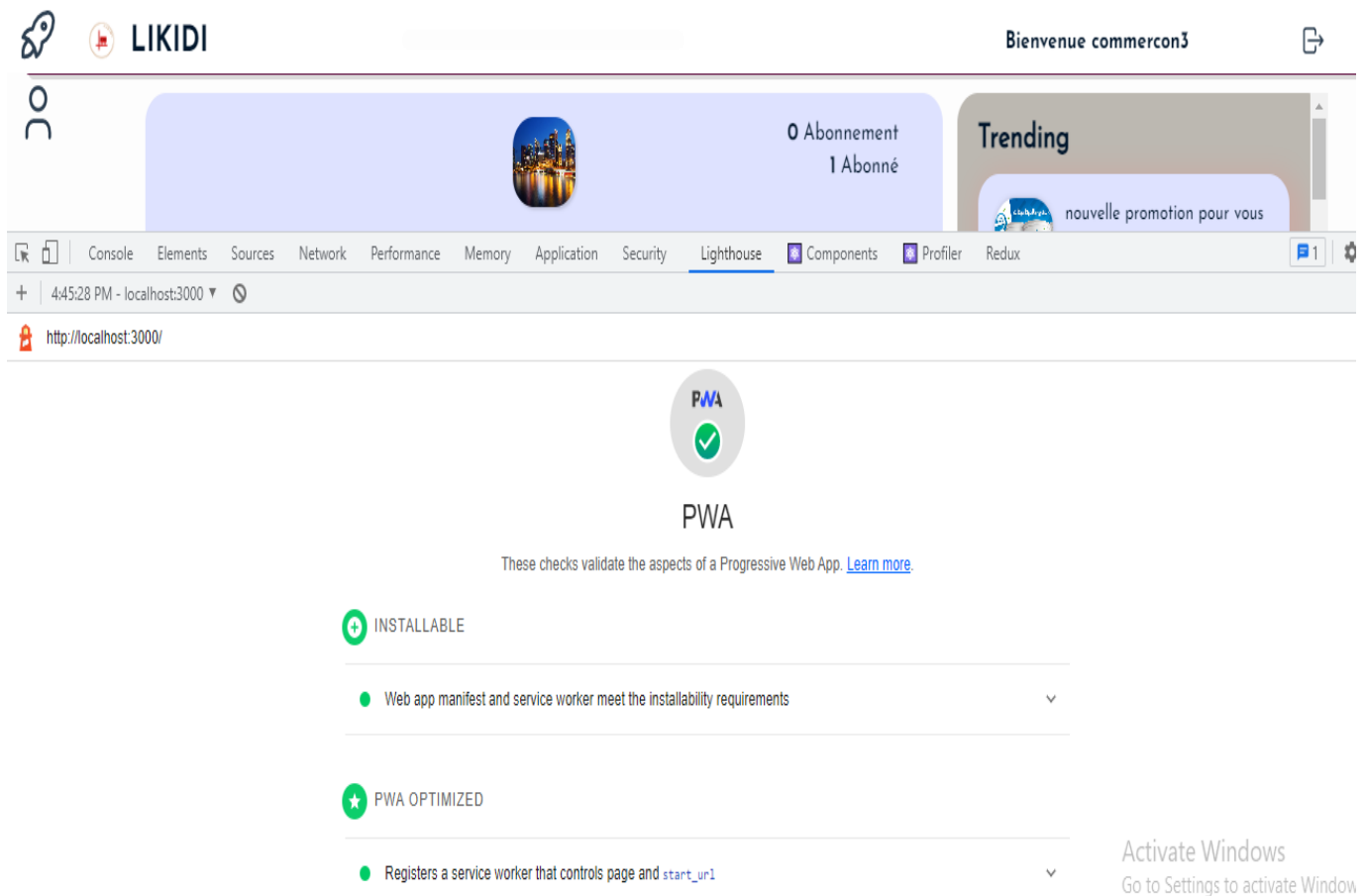


FIGURE IV.32 – Captures d'interfaces accueil utilisateur sur smartphone

## IV.8 Conclusion

Dans ce dernier chapitre, qui représente la phase finale du processus de développement logiciel, nous avons présenté les différents outils et langages informatiques utilisés pour développer notre application. Nous avons clôturé ce chapitre par la présentation de quelques interfaces des utilisateurs.

# Conclusion Générale

Le présent rapport est le résultat de notre travail que nous avons effectué dans le cadre de la réalisation de notre projet de fin d'études de Master 2 en Informatique option : Génie Logiciel. Lors de ce travail, nous avons pu mettre en pratique nos connaissances théoriques acquises durant notre formation, de plus, nous sommes arrivés à réaliser les objectifs que nous avons mis au début de cette période bien que nous ayons vécu des difficultés, les méthodologies que nous avons utilisées pour les dépasser sont des signes de satisfaction.

Tout au long de ce travail, nous avons abordé le problème de gaspillage alimentaire et ces conséquences sur l'économie mondiale ainsi que sur l'environnement. En vue de participer à la réduction de ce gaspillage, nous avons proposé une solution à travers notre application, qui consiste à faire de la publicité aux produits alimentaires périssables, qui sont généralement en promotion dans les différents magasins pour leur assurer une vaste visibilité et attirer plus de gens à les acheter vu qu'ils sont à prix réduits.

Ensuite, nous avons décrit les applications web et les PWA avec leurs différents avantages. L'application que nous avons réalisée a été conçue et modélisée de manière à ce qu'elle soit accessible pour tous, ergonomique, conviviale, simple d'utilisation sont autant de qualités consacrées à faciliter la navigation des différents utilisateurs.

Pour la réalisation de l'application, nous avons eu recours à différents langages et outils de développement et de programmation telle que React js et Redux pour la mise en place des différents traitements et fonctionnalités de l'application ainsi qu'HTML, CSS, JavaScript pour la réalisation de son aspect graphique, sans oublier MongoDB comme système de gestion de bases de données.

À l'issue de ce projet, notre application est réalisée dans sa première version. Elle reste en cours de développement et ouverte à des perspectives d'amélioration. On a réussi à implémenter la majorité des fonctionnalités de l'application comme « la publication des produits », « faire des commentaires », etc. Les deux fonctionnalités « faire une recherche » et « la gestion de signalisation » ne sont pas encore implémentées.

Ce projet a fait l'objet d'une expérience très intéressante, car elle nous a permis de nous familiariser avec de nouvelles notions d'une part, et d'améliorer nos connaissances et nos compétences dans le domaine de la programmation, et d'autre part de renforcer notre sens de responsabilité dans la gestion des projets qui permet l'insertion dans le domaine professionnel.

Enfin, on est convaincue que notre application est une solution efficace pour lutter contre le gaspillage, tout en faisant des économies. On espère que ce modeste travail servira de référence et de motivation pour ceux qui voudront s'engager dans des gestes écologiques pour protéger notre planète.

# Bibliographie

- [1] FRAMEWORK *Framework ou infrastructure logicielle*, <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/>. consulter le,02/08/2022
- [2] FRONTEND, BACKEND *Framework frontend, backend*, <https://www.vigicorp.fr/blog/qu-est-ce-qu-un-framework/>. consulter le,02/08/2022
- [3] AGILE *MÉTHODE AGILE*, <https://www.amalo-recrutement.fr/blog/methode-agile/>. consulter le,03/08/2022
- [4] . AGILE *Valeurs principales d'agile*, <https://www.amalo-recrutement.fr/blog/methode-agile/>. consulter le,03/08/2022
- [5] *Processus unifié*, G.Picard. Conduite de projet, Méthode d'analyse et de conception, Processus unifié. PhD thesis, [Ecole Nationale Supérieure des Mines Saint-Étienne, 2008](#). consulter le,03/08/2022
- [6] *Supported Codes et Platforms*, <https://www.peerbits.com/blog/mobile-app-security-coding-practices.html>. consulter le,03/08/2022
- [7] *progressive web*, [https://en.wikipedia.org/wiki/Progressive\\_web\\_app\\_History](https://en.wikipedia.org/wiki/Progressive_web_app_History). consulter le,05/08/2022
- [8] *progressive web*, [https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps). consulter le,05/08/2022
- [9] *Node.js Guide*, URL : <https://nodejs.org>. consulter le,07/08/2022
- [10] *node-sass*, <https://stackshare.io/stackups/node-sass-vs-sass>. consulter le,07/08/2022
- [11] *react redux*, <https://www.javatpoint.com/react-redux>. consulter le,07/08/2022
- [12] *Multer*, <https://www.npmjs.com/package/multer>. consulter le,07/08/2022
- [13] *github*, <https://www.hostinger.fr/tutoriels/github-cest-quoi-et-comment-lutiliser>. consulter le,07/08/2022
- [14] *mongodb compass*, <https://www.javatpoint.com/mongodb-compass>. consulter le,07/08/2022  
*Mémoire de licence, université ABDE RAHMANE MIRA, 2014/2015.* consulter le,08/08/2022
- [15] *postman*, <https://explorweb.github.io/cours2018/cours/postman.html>. consulter le,08/08/2022

- [16] *Express JS : A Guide for Beginners* , <https://www.coursereport.com/blog/what-is-express> .  
consulter le,08/08/2022
- [17] *cors*,<https://rapidapi.com/blog/api-glossary/cors/> . consulter le,08/08/2022
- [18] *bcrypt* ,<https://www.quora.com/What-is-bcrypt> . consulter le,08/08/2022
- [19] *json web tokens* ,<https://auth0.com/docs/secure/tokens/json-web-tokens> .  
consulter le,08/08/2022
- [20] *java scripte ES6* ,<https://www.programiz.com/javascript/ES6> . consulter le,08/08/2022
- [21] *JSON* ,  
[\[AIT HATRIT Fatima et all, Conception et réalisation d'un site web dynamique pour l'aéroport « ABANE RAMDANE » De Bejaia, Mémoire de licence, université ABDERAHMANE MIRA, 2014/2015.\]](#). consulter le,09/08/2022
- [22] *pwa vs app native* ,  
<https://www.teknicks.com/blog/pwa-progressive-web-app-vs-native-apps-which-is-better/> .  
consulter le,09/08/2022