



Ministère de l'Enseignement Supérieur, de la  
Recherche Scientifique



Direction Générale des Études Technologiques

<b>Code projet</b>	.....
------------------------	-------

**Institut Supérieur des Études Technologiques de Djerba**

**Département Technologies de l'informatique**

## **Rapport de**

# **STAGE DE FIN D'ETUDES**

**Présenté en vue de l'obtention du**

**Diplôme de master professionnel**

**Co-Construit en Technologie de l'Information et  
Communication et Innovation Touristique**

**(TICIT)**

**Développement d'une application web et**

**Mobile Pour la gestion d'un restaurant**

**Élaboré par :**

**Moez Oueriemmi**

**Encadré par :**

**Mr. Soufienne Ben Jamaa**

**Effectué à :**

**Entreprise : Slima Tours**

**Encadreur : Mr. Lassad Lourimmi**

**Année universitaire : 2021/2022**

# Dédicaces

A mes chers parents,

Aucun mot, aucune dédicace ne saurait exprimer mon respect, ma considération et l'amour éternel pour les sacrifices que vous avez déployés pour mon instruction et mon bien être dans les meilleures conditions.

Votre générosité et votre bonté ont toujours été un exemple pour moi. Trouvez en ce travail le fruit de votre dévouement et l'expression de ma gratitude et mon profond amour.

A mes frères, mes sœurs et mes proches, à qui je dois ma reconnaissance, je vous remercie vivement pour votre présence et votre soutien.

# Remerciements

Au terme de ce projet de fin d'étude, mes vifs remerciements sont dédiés à tous ceux qui ont contribué, directement ou indirectement à l'élaboration de ce projet. En Premier lieu, je remercie Monsieur Soufienne Ben Jamaa, mon encadrant coté ISET, pour sa disponibilité, ses directives et son sens du détail s'ont permis de soigner et d'améliorer constamment la qualité de ce travail.

Mes remerciements s'adressent également à Monsieur Lassad Lourimmi, mon encadrant coté entreprise, qui a toujours trouvé le temps de faire le suivi de notre travail et d'être à l'écoute lorsque je rencontrais des difficultés dans le projet.

C'est avec une immense fierté que j'adresse mon remerciement les plus distingués à tous les enseignants de l'ISET qui m'ont transmis leur savoir et m'ont assuré la meilleure des formations.

# Sommaire

Introduction générale.....	1
Chapitre I : Etude préalable .....	2
I.1 Introduction.....	2
I.2 Présentation de l'organisme d'accueil.....	2
I.3 Etude de l'existant.....	3
I.3.1 Présentation de l'existant.....	3
I.3.2 Critique de l'existant .....	3
I.3.3 Solution proposée .....	4
I.4 Expression des besoins .....	4
I.4.1 Les besoins fonctionnels.....	4
I.4.2 Les besoins non fonctionnels .....	7
I.5 Conclusion .....	7
II.1 Introduction .....	8
II.2 Choix de la méthodologie de conception .....	8
II.3 Diagramme de cas d'utilisation .....	8
II.3.1. Les acteurs .....	8
II.3.2. Les cas d'utilisation.....	9
II.3.3. Les diagrammes .....	10
II.4. Diagramme de classes .....	11
II.4.1Dictionnaire de données .....	11
II.4.2. Le diagramme .....	15
II.5. Diagramme de séquences .....	15
II.6 Conclusion.....	18
III.1 Introduction .....	19
III.2 Méthodologie de travail .....	19

III.2.1 Méthode SCRUM .....	19
III.2.2 Les sprints.....	19
III.2.3 Réalisation .....	20
III.3 Choix technologiques.....	20
III.3.1 Pourquoi utiliser un Framework ? .....	20
III.3.2 Le Framework Flutter .....	21
III.3.2.1 Pourquoi choisir Flutter ? .....	22
III.3.3 Le Framework Angular.....	23
III.3.3.1 Pourquoi choisir Angular ?.....	24
III.3.4 Le Framework Spring Boot.....	24
III.3.4.1 Pourquoi choisir spring boot .....	24
III.3.5 API.....	25
III.3.5.1 Qu'est-ce qu'une API REST ?.....	25
III.3.5.2 API REST ou API SOAP ? .....	25
III.3.5.3 Pourquoi choisir REST ?.....	26
III.3.6 MySQL.....	26
III.3.6.1 Pourquoi choisir MySQL ?.....	26
III.4 Environnement de travail.....	27
III.4.1 Environnement matériel.....	27
III.4.2 Environnement logiciel.....	27
III.4.2 .1 Visual Studio Code .....	27
III.4.2 .2 Xampp .....	28
III.4.2 .3 Postman .....	28
III.4.2 .4 Eclipse .....	28
III.4.2 .5 PhpMyAdmin (PMA).....	29
III.5 Enchainements des interfaces de l'application.....	29
III.5.1 Application Web.....	29

III.5.2 Application Mobile .....	34
III.6 Conclusion.....	38
Conclusion Générale .....	39
Bibliographie.....	40

# Table des figures

<u>Figure 1 : Organigramme d'entreprise</u> .....	3
<u>Figure 2 : Diagramme de cas d'utilisation « Visiteur &amp; Client »</u> .....	10
<u>Figure 3 : Diagramme de cas d'utilisation « administrateur »</u> .....	10
<u>Figure 4 : Diagramme de cas d'utilisation « Employé »</u> .....	11
<u>Figure 5 : Diagramme de classe</u> .....	15
<u>Figure 6 : Diagramme de séquence « authentification »</u> .....	16
<u>Figure 7 : Diagramme de séquence « Créer un compte »</u> .....	17
<u>Figure 8 : Diagramme de séquence « Gérer les articles »</u> .....	18
<u>Figure 9:Diagramme de séquence « Effectuer une commande »</u> .....	19
<u>Figure 10: Sprint</u> .....	21
<u>Figure 11:Logo flutter</u> .....	22
<u>Figure 12:Tableau comparatif</u> .....	24
<u>Figure 13:Logo angular</u> .....	25
<u>Figure 14:Logo SpringBoot</u> .....	26
<u>Figure 15:Logo MySQL</u> .....	28
<u>Figure 16:LogoFlutter</u> .....	29
<u>Figure 17:Logo Xamp</u> .....	30
<u>Figure 18:Logo Postman</u> .....	30
<u>Figure 19:Logo Eclipse</u> .....	31
<u>Figure 20:Logo PhpMyAdmin</u> .....	31
<u>Figure 21:Interface d'authentification</u> .....	32
<u>Figure 22:Tableau de bord</u> .....	32
<u>Figure 23:Interface client</u> .....	33
<u>Figure 24:Interface de commande</u> .....	33
<u>Figure 25:Interface de livraison</u> .....	34

<u>Figure 26:Interface de livreur</u> .....	34
<u>Figure 27:Interface d'article</u> .....	35
<u>Figure 28:Interface d'utilisateur</u> .....	35
<u>Figure 29:Interface d'employé</u> .....	36
<u>Figure 30:Interface de produit</u> .....	36
<u>Figure 31:Interface d'accueil</u> .....	37
<u>Figure 32:Liste des pizzas</u> .....	37
<u>Figure 33:Liste des boissons</u> .....	38
<u>Figure 34: Liste des plats</u> .....	38
<u>Figure 35 :Panier</u> .....	38
<u>Figure 36:Liste des crêpes</u> .....	39
<u>Figure 37:Interface de contact</u> .....	39
<u>Figure 38:Interface d'authentification</u> .....	40
<u>Figure 39:Consulter la commande</u> .....	40



# Liste des tableaux

<a href="#">Tableau 1:Information d'entreprise</a> .....	2
<a href="#">Tableau 2:Dictionnaire de données</a> .....	14
<a href="#">Tableau 3:Réalisation des sprints</a> .....	20

# Introduction générale

Le tourisme est l'un des secteurs économiques les plus importantes dans le monde, il est considéré comme une industrie et une grande source d'investissement et d'emploi.

Depuis quelques années, l'industrie du voyage a connu un grand bouleversement et une grande transformation sous l'impulsion du digital et l'intégration des nouvelles technologies de l'information et de la communication.

Les agences de voyages jouent un rôle indispensable dans le développement et l'évolution de ce secteur, puisqu'ils présentent l'intermédiaire entre les compagnies aériennes, les prestataires de service, les systèmes d'hébergement et les clients.

Actuellement, la migration vers le digital et le changement de processus de commercialisation des voyages (e-paiement, e-marketing, Smartphone, tablette) offrent une grande flexibilité aux voyageurs et constituent un moyen rapide et efficace pour les voyageurs.

C'est dans ce contexte que s'intègre notre stage de fin d'étude, élaboré au sein de l'agence de voyage **Slima Tours**, qui a pour objectif la mise en place d'une solution permettant d'automatiser le processus et surmonter le désordre et l'encombrement qui survient dans les restaurants.

L'application visée permet à un restaurant, de gérer les commandes, ... Et en ce qui concerne les clients, la solution conçue sera un moyen pour consulter les offres, et réserver.

Les différentes étapes de la réalisation du projet seront résumées dans le présent rapport organisé suivant les chapitres suivants :

- Un premier chapitre introductif décrivant l'organisme d'accueil et expose l'étude de l'existant afin de mettre l'accent sur la solution proposée et l'analyse et la spécification des besoins fonctionnels et non fonctionnels du projet.
- Un deuxième chapitre dédié pour l'étude conceptuelle cette analyse sera traduite par des diagrammes de cas d'utilisation afin de présenter les différentes fonctions et délimiter le périmètre du projet.
- Le troisième chapitre, traite la partie de développement de notre solution. Il comporte la conception architecturale, l'environnement de développement ainsi que la réalisation.
- Nous clôturons ce rapport par une conclusion générale dans laquelle nous évaluons le travail réalisé.

# Chapitre I : Etude préalable

## I.1 Introduction

Dans ce premier chapitre, nous allons présenter l'organisme d'accueil et donner un aperçu général sur le projet que nous comptons réaliser ainsi que son cahier de charges et les contraintes rencontrées sur le matériel de réalisation et les logiciels à utiliser. Nous allons ensuite définir la problématique ainsi que les besoins qui vont guider ce travail. Par la suite, nous décrivons l'existant élaboré et nous présentons nos critiques puis notre solution.

## I.2 Présentation de l'organisme d'accueil

L'agence de voyage Slima Tours est une agence de type « B2C » qui propose et vend des offres de voyages à ses clients. Elle joue le rôle d'intermédiaire entre les consommateurs et les différents prestataires des services présents sur le marché du tourisme : compagnies aériennes, hôteliers, etc....

Slima Tours	Information
Adresse	Slima Tours Située au Houmt Souk Djerba (Avenue Habib Bourguiba en face du stade Municipal)
Fax	75621374
Téléphone	75621375

Tableau 1:Information d'entreprise

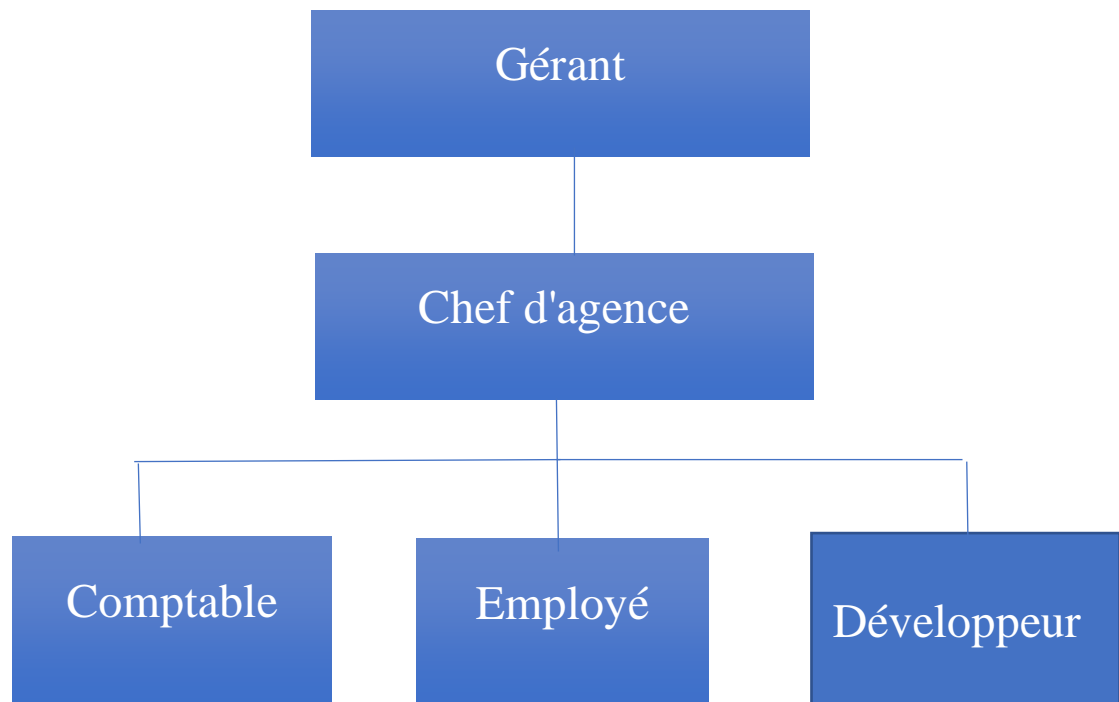


Figure 1 : Organigramme d'entreprise

## **I .3 Etude de l'existant**

### **I .3.1 Présentation de l'existant**

Dans cette partie, nous allons décrire la procédure suivie par un client pour effectuer une commande dans une pizzeria.

Le client doit se présenter à la pizzeria pour sélectionner une commande et il choisit tous les détails (les ingrédients, les suppléments, les boissons) l'employé tape la commande sur la caisse et donne au client un ticket. Le dernier attend jusqu'à sa commande sera prête.

### **I .3.2 Critique de l'existant**

Dans nos jours tout le monde cherche l'efficacité et la rapidité d'un tel produit, donc dans notre cas nos clients ont besoin de la rapidité aussi, ils veulent que ses commandes soient prêtes dès qu'ils arrivent à la pizzeria.

Cette méthode prend beaucoup de temps ainsi qu'elle est difficile à gérer au niveau :

-La mise à jour des articles (prix, ingrédients, image...) : pour changer le prix ou les ingrédients d'un article il faut réimprimer le menu.

- La Gestion des commandes :

- . Difficulté d'agir les commandes pendant L'encombrement.
- . Possibilité d'avoir des erreurs au niveau des ingrédients d'une commande.

### **I .3.3 Solution proposée**

Grâce à la critique ci-dessus, nous sommes guidés pour proposer une solution qui satisfait l'objectif et surmonter le désordre et l'encombrement qui survient dans les restaurants. Dans cette démarche, je propose à l'agence de développer une application qui met en relation le restaurant avec ses clients.

L'agence offre à ses clients la possibilité de consulter les offres. Elle facilite la recherche des bons plans et garantit la fidélité. L'application projetée propose également des recherches rapides et intelligentes.

Notre solution se compose de deux parties : une dédiée aux administrateurs pour gérer le restaurant et une autre aux clients pour chercher un produit et effectuer une commande.

Cette application facilite :

- La mise à jour d'un article
- La gestion des commandes
- L'ajout d'un nouvel article
- La suivie d'une commande par le client
- La proposition d'un article au client
- La fidélisation de client

## **I.4 Expression des besoins**

### **I.4.1 Les besoins fonctionnels**

L'application attendue doit répondre aux besoins fonctionnels suivants :

- **Création du logo**

*Application Mobile (client) :*

- **Consulter les articles :**

Le client a le droit de consulter les articles proposés par la pizzeria.

- **S'inscrire :**

Le client a le droit de créer un compte.

- **Passer une commande :**

Le client peut passer une commande mais il doit remplir un formulaire par ses informations personnel.

- **Consulter la commande :**

Pour consulter la commande le client se connecté à son espace personnel avec son login et un mot de passe.

- **Envoyer des messages :**

Le client ou le visiteur peut envoyer des messages ou des remarques à l'administrateur

### **Application Web (administrateur) :**

- **Authentification :**

Chaque administrateur est caractérisé par un login et un mot de passe utilisé pour accéder à son espace de gestion.

- **Gestion des contacts :**

Un contact est caractérisé par le nom, prénom, le numéro du téléphone, l'email de client.

L'administrateur peut chercher et supprimer les contacts envoyer par les clients ou les visiteurs.

- **Gestion des commandes :**

Le client peut envoyer une ou plusieurs commandes.

Une commande est caractérisée par les informations du client, les articles choisis, le type (livraison, sur place, emporter).

L'administrateur peut ajouter, modifier, chercher et supprimer des commandes.

- **Gestion des utilisateurs :**

Chaque utilisateur est caractérisé par son nom, son prénom, son email, son numéro de téléphone, son rôle, son login et son mot de passe.

L'administrateur peut ajouter, modifier, chercher et supprimer des utilisateurs.

- **Gestion des clients :**

Chaque client a un compte client pour consulter ses commandes.

Le client est caractérisé par son nom, son prénom, son adresse, son numéro de téléphone et son mot de passe.

L'administrateur peut ajouter, modifier, chercher et supprimer des clients.

- **Gestion des produits :**

L'administrateur peut ajouter, modifier, chercher et supprimer des produits.

Un produit est caractérisé par son nom, son image, sa quantité et son prix.

- **Gestion des articles :**

L'administrateur peut ajouter, modifier, chercher et supprimer des articles.

Un article est caractérisé par son nom, son image, ses ingrédients et son prix.

- **Gestion des employés :**

L'administrateur peut ajouter, modifier, chercher et supprimer des employés.

Un employé est caractérisé par son nom, son prénom, son adresse, son poste et son salaire.

- **Gestion des livreurs :**

L'administrateur peut ajouter, modifier, chercher et supprimer des livreurs.

Un livreur est caractérisé par son nom, son prénom, son numéro de téléphone, son adresse et le prix de livraison.

- **Gestion des livraisons :**

L'administrateur peut ajouter, modifier, chercher et supprimer des livraisons.

Une livraison est caractérisée par un client, une commande et un livreur

### **I.4.2 Les besoins non fonctionnels**

Pour que notre solution réponde aux attentes, notre application doit nécessairement assurer ces besoins :

- L'extensibilité : dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- Élever la performance et le temps de réponse des services offerts par l'application.
- Le système doit être sécurisé au niveau des données : authentification et contrôle des droits d'accès.
- Les matériels nécessaires : des tablettes, des écrans.

## **I.5 Conclusion**

Au cours de ce présent chapitre nous avons présenté l'environnement dans lequel notre projet de fin d'études a été effectué le thème de projet, étudié l'existant et spécifié les besoins fonctionnels et non fonctionnels à travers notre cahier des charges, ainsi nous sommes prêts pour la phase conceptuelle qui nous présenterons dans le chapitre suivant.



# Chapitre II : Etude Conceptuelle

## II.1 Introduction

Il convient de dire que la conception est la partie la plus importante pour chaque projet. Pour mieux concevoir notre application on a choisi UML pour réaliser le modèle conceptuel de données.

## II.2 Choix de la méthodologie de conception

UML (Unified Modeling Language) est un langage d'analyse et de conception orienté objet défini par l'OMG (Object Management Group). UML homogénéise les représentations graphiques des objets. Et il peut être intégré à n'importe quel processus de développement logiciel de manière transparente, car UML est une sorte de boîte à outils qui permet d'améliorer progressivement les méthodes de travail tout en préservant les modes de fonction. [1] Pour la modélisation de notre projet nous avons choisi UML comme langage vu ses avantages :

- UML est un langage formel et normalisé.
- UML est un support de communication performant : il cadre l'analyse, facilite la Compréhension des représentations abstraites complexes.

Le principal avantage d'UML est qu'il est devenu le standard en termes de modélisation objet, son caractère polyvalent et performant et sa souplesse en fait un langage universel.

Dans le cas de notre projet, nous avons choisi UML comme langage de modélisation.

## II.3 Diagramme de cas d'utilisation

Les cas d'utilisation (use cases) sont une technique qui permet de capturer les exigences fonctionnelles d'un système lui-même et constitue en quelque sorte le « récit » de la façon dont ce dernier est utilisé. [2]

### II.3.1. Les acteurs

- Visiteur : Inconnu, qui peut néanmoins consulter les articles, envoyer des messages et créer un compte client.
- Client : Connu par l'application, qui peut consulter sa commande et en suivre son état.

- Administrateur : C'est le responsable de la gestion (des comptes d'utilisateurs, des employés, des articles, des produits, des livreurs, des livraisons, des contacts, des commandes).

### **II.3.2. Les cas d'utilisation**

- Le Visiteur : Ces cas d'utilisation principaux sont :

- Consulter les articles
- Envoyer des messages
- Effectuer une commande

- Le Client : Ces cas d'utilisation principaux sont :

- Consulter les articles
- Effectuer une commande
- Consulter l'état de ses commandes
- Envoyer des messages
- S'authentifier

- L'administrateur : Ces cas d'utilisation principaux sont :

- Gérer les commandes des clients
- Gérer les employés
- Gérer les articles
- Gérer les produits,
- Gérer les livreurs
- Gérer les livraisons
- Gérer les messages
- Gérer les comptes clients
- Gérer les comptes utilisateurs
- S'authentifier

### II.3.3. Les diagrammes

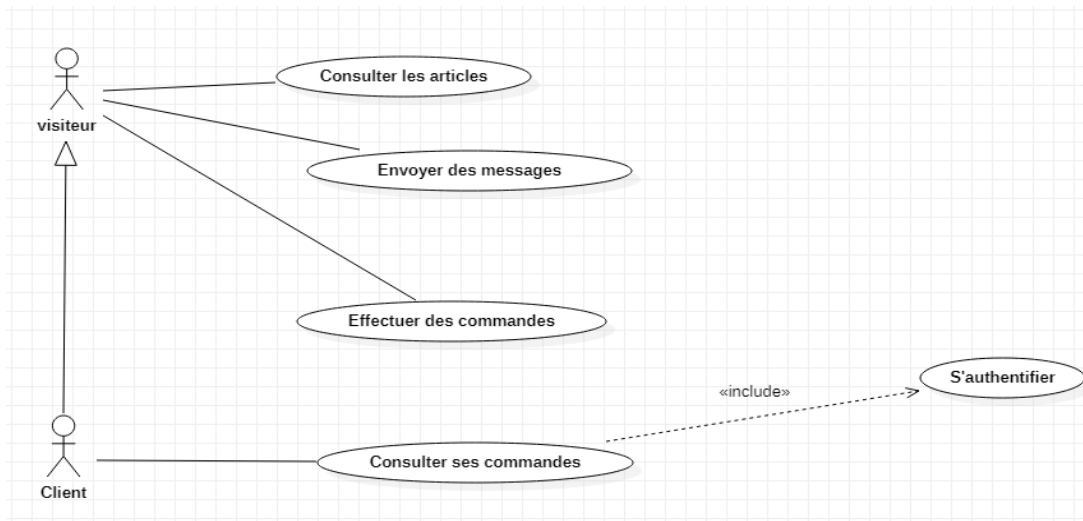


Figure 2 : Diagramme de cas d'utilisation « Visiteur & Client »

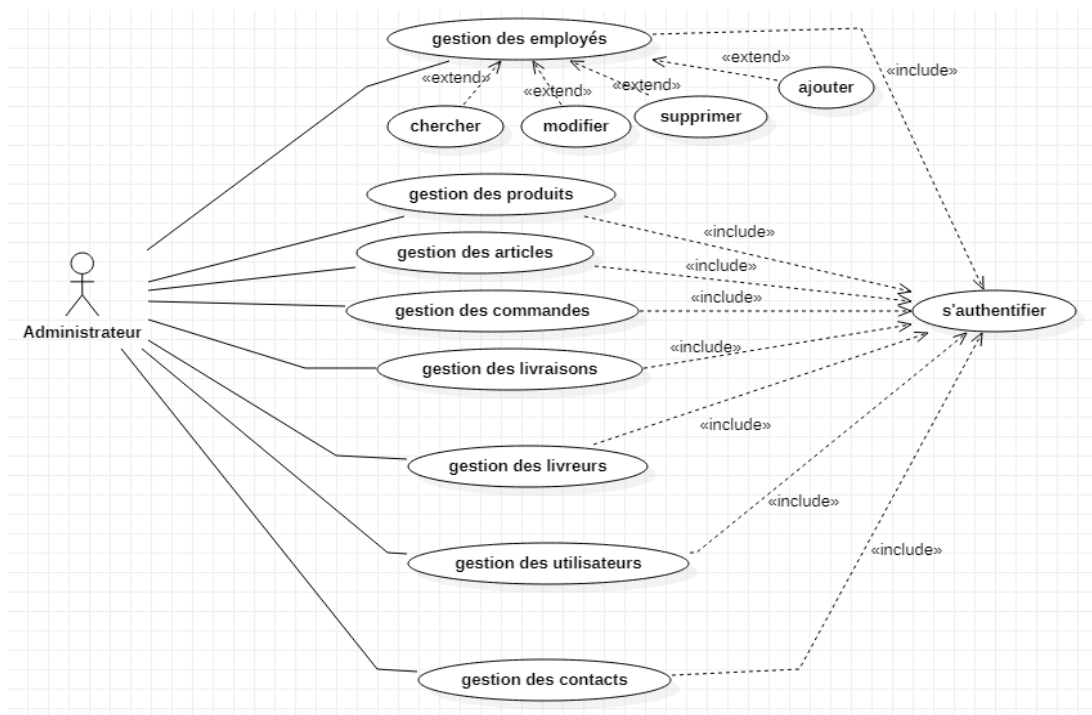


Figure 3 : Diagramme de cas d'utilisation « administrateur »

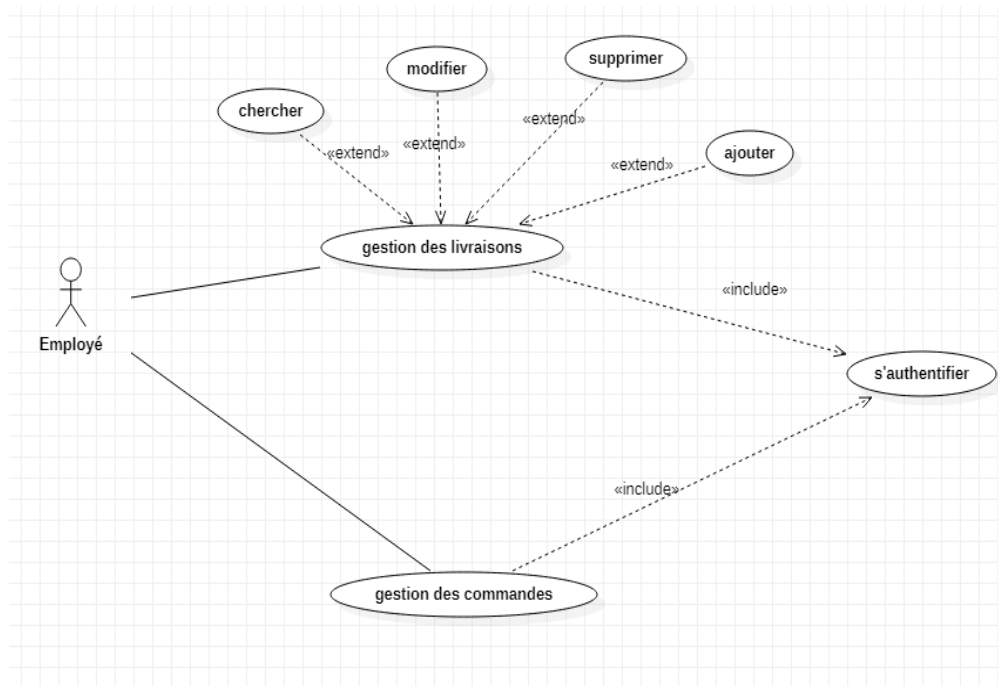


Figure 4 : Diagramme de cas d'utilisation « Employé »

## II.4. Diagramme de classes

Quand un concepteur décide d'utiliser UML pour la modélisation de ses projets objets, visualiser l'ensemble des classes et des relations entre elles est primordiale pour une gestion optimale du système d'information. Ainsi le diagramme des classes devient inévitable. [\[3\]](#)

### II.4.1 Dictionnaire de données

Classe	Attributs	Désignation
Employé	Idemp	L'identifiant d'un employé
	dateentree	La date entrée d'un employé
	jour repos	Le jour de repos d'un employé
	Nom	Le nom d'un employé

	Prénom	Le prénom d'un employé
	Poste	La poste d'un employé
	Salaire	Le salaire d'un employé
	Tel	Le téléphone d'un employé
Livreur	Livreur	L'identifiant d'un livreur
	Adresse	L'adresse d'un livreur
	Nom	Le nom d'un livreur
	Prénom	Le prénom d'un livreur
	Prix	Le prix de livraison d'un livreur
	Tel	Le téléphone d'un livreur
Produit	Id	L'identifiant d'un produit
	Image	L'image d'un produit
	Nom	Le nom d'un produit
	Prix	Le prix d'un produit
	Qte	La quantité d'un produit
User	Id	L'identifiant d'un utilisateur
	Adresse	L'adresse d'un utilisateur
	Email	L'email d'un utilisateur
	Login	Le login d'un utilisateur

	Pwd	Le mot de passe d'un utilisateur
	Nom	Le nom d'un utilisateur
	Tel	Le téléphone d'un utilisateur
	Rôle	Le rôle d'un utilisateur
	Prénom	Le prénom d'un utilisateur
Contact	Id	L'identifiant d'un client
	Email	L'email d'un client
	Message	Le message d'un client
	Nom-prénom	Le nom et le prénom d'un client
	Telc	Le téléphone d'un client
Article	Idart	L'identifiant d'un article
	Image	L'image d'un article
	Ingrédients	Les ingrédients d'un article
	Nom	Le nom d'un article
	Prix	Le prix d'un article
Commande	Id-com	L'identifiant d'une commande
	Date-com	La date d'une commande
	Etat	L'état d'une commande
	Article-id	L'identifiant d'un article

	User-id	L'identifiant d'un utilisateur
	Prix	Le prix d'une commande
Ligne-de-commande	Id	L'identifiant d'une ligne de commande
	Idcom	L'identifiant d'une commande
	Remarque	La remarque de client sur l'article
	Article-idart	L'identifiant d'un article
	Qte	La quantité d'un article
Client	Nom	Le nom d'un client
	Prénom	Le prénom d'un client
	Adresse	L'adresse d'un client
	Téléphone	Le téléphone d'un client
Livraison	id livraison	L'identifiant d'une livraison
	Commande-id	L'identifiant d'une commande
	Livreur-id	L'identifiant d'un livreur

Tableau 2:Dictionnaire de données

## II.4.2. Le diagramme

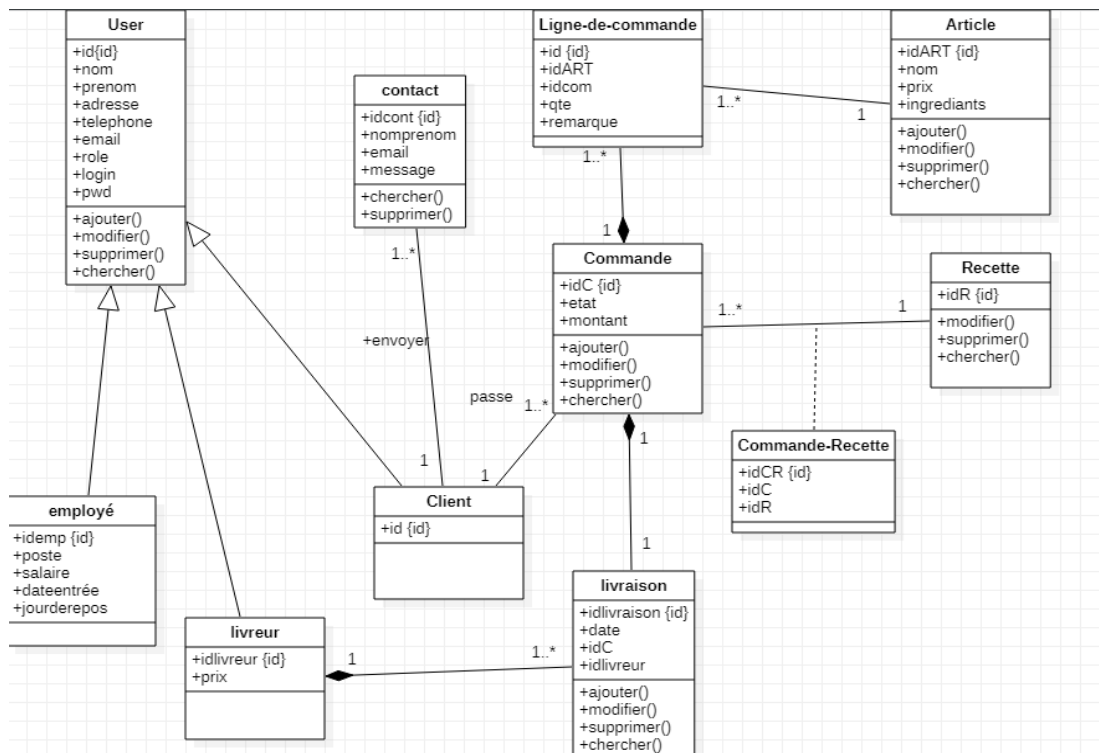


Figure 5 : Diagramme de classe

## II.5. Diagramme de séquences

Les diagrammes d'interaction décrivent la façon dont les groupes d'objets collaborent à la réalisation d'un comportement donné. UML définit plusieurs formes de diagrammes d'interaction dont le plus courant est le diagramme de séquence.

En général, un diagramme de séquence capture le comportement d'un seul scénario. Il contient un certain nombre d'objets et les messages transmis entre ces objets dans le cadre d'un cas d'utilisation. [4]



- **S'authentifier :**

Ce diagramme de séquence du cas d'utilisation « authentification ».

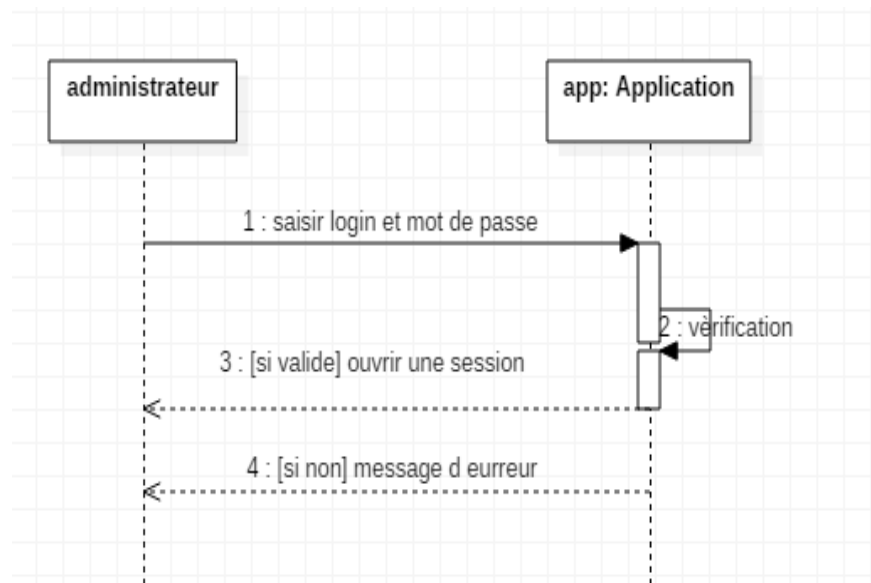


Figure 6 : Diagramme de séquence « authentification »

- **Créer un compte :**

Ce diagramme de séquence du cas d'utilisation « Créer un compte ».

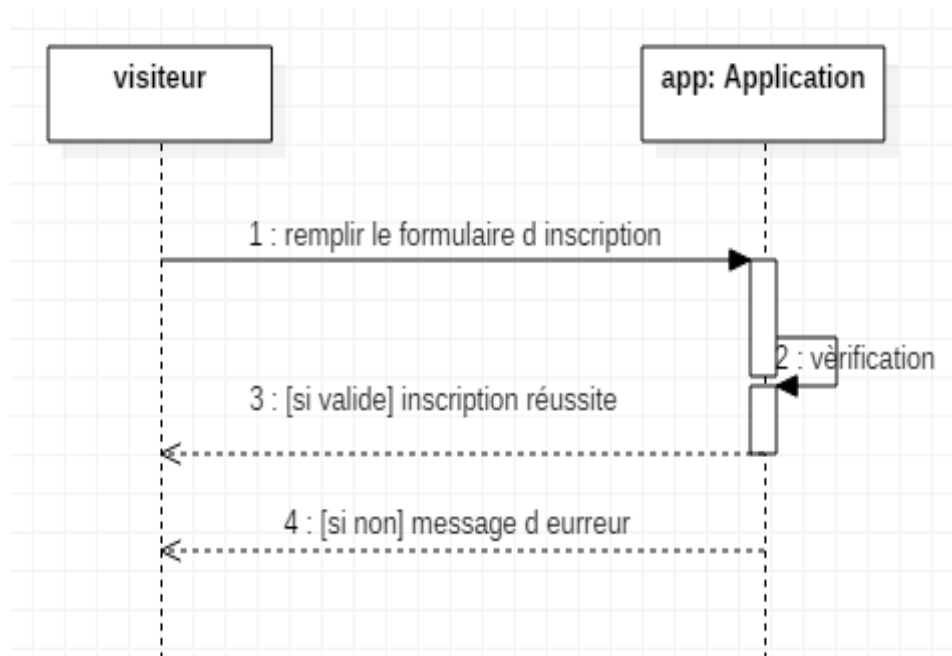


Figure 7 : Diagramme de séquence « Créer un compte »

- **Gérer les articles :**

Ce diagramme de séquence du cas d'utilisation « **Gérer les articles** »

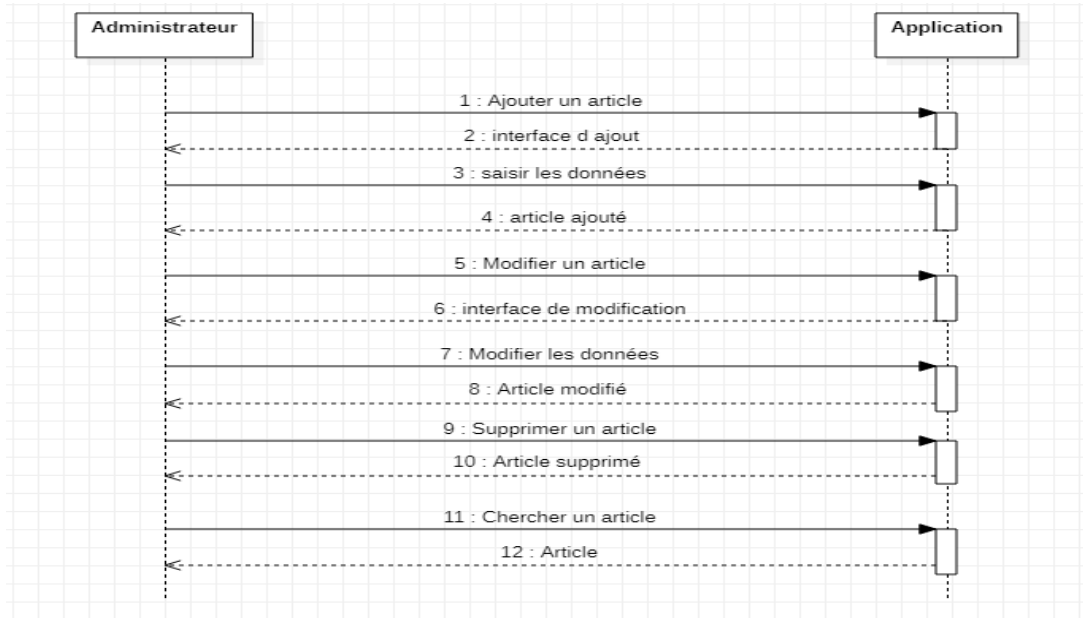


Figure 8 : Diagramme de séquence « Gérer les articles »

- **Effectuer une commande :**

Ce diagramme de séquence du cas d'utilisation « Effectuer une commande ».

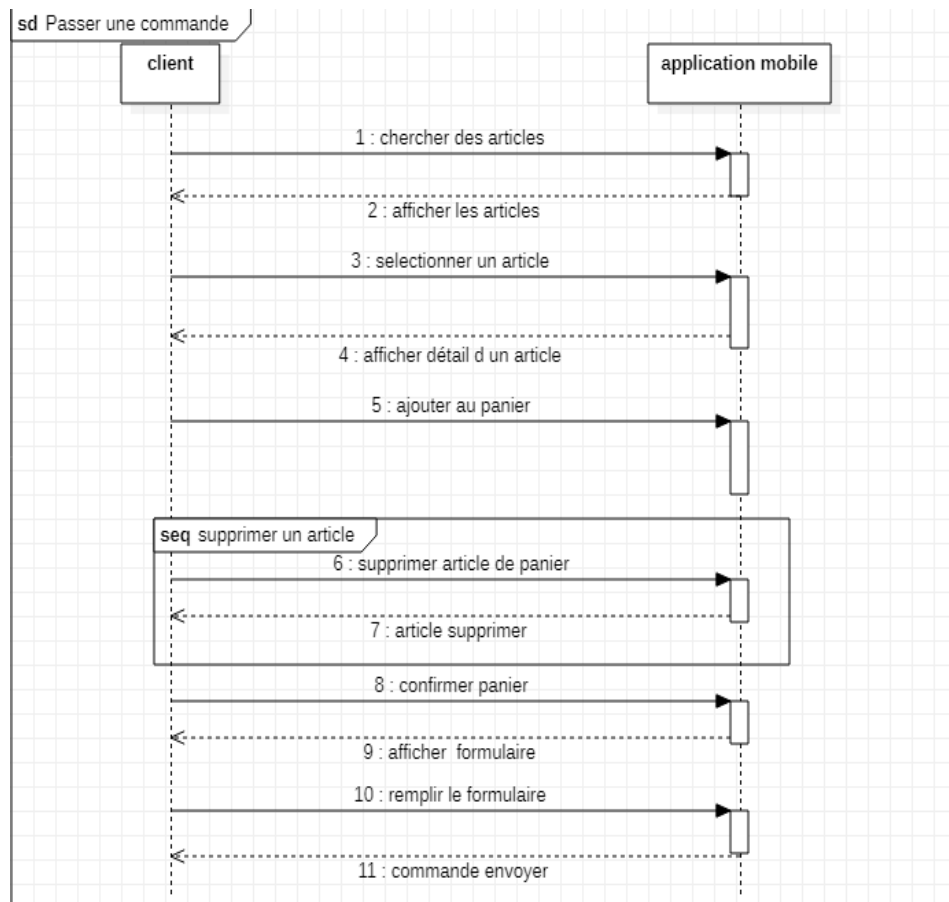


Figure 9 : Diagramme de séquence « Effectuer une commande »

## II.6 Conclusion

Tout au long de ce chapitre, nous avons présenté une étude conceptuelle basée sur des diagrammes permettant une mise au point des différents traitements et données. Après l'étude préalable, nous passons par la suite à la phase suivante qui sera consacrée à la réalisation de notre projet

# Chapitre III : Réalisation

## III.1 Introduction

Ce chapitre représente le dernier volet de ce rapport, il sera consacré à l'implémentation de mon système. Je vais commencer par la présentation la méthodologie de travail, les ressources matérielles et logicielles utilisées, puis je vais passer à la présentation des captures d'écran dans le but de mettre en évidence l'aspect ergonomique et fonctionnel des interfaces développées.

## III.2 Méthodologie de travail

### III.2.1 Méthode SCRUM

La méthode Scrum est une méthode agile de gestion de projets informatiques privilégiant la communication, et facilitant les réorientations opportunes. C'est désormais la méthode privilégiée pour les démarches dites « agiles ». Elle est basée sur les sprints.

### III.2.2 Les sprints

Un sprint désigne une période pendant laquelle un travail spécifique doit être mené à bien avant de faire l'objet d'une révision.

Chaque sprint commence par une réunion de planification. Au cours de la réunion, le propriétaire du produit (la personne qui porte la vision du travail à réaliser) et l'équipe chargée du développement se mettent d'accord sur les tâches précises à accomplir pendant le sprint. En règle générale un sprint se déroule entre 2 et 4 semaines

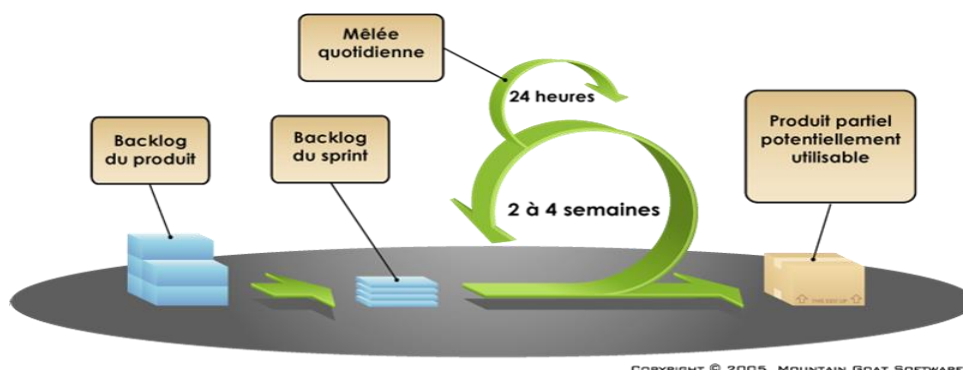


Figure 10 : Sprint

### III.2.3 Réalisation

Sprints	Taches	Durée
Sprint 1	Conception	Deux semaines
	Création d'un logo	
Sprint 2	Préparer l'environnement de travail(web)	Un mois
	Préparer les interfaces coté administrateur	
	Développer	
	Les APIS	
	Use case : Authentification	
	Use case : Gérer les articles	
	Use case : Gérer les clients	
	Use case : Gérer les utilisateurs	
	Use case : Gérer les commandes	
Sprint 3	Préparer les interfaces coté client(mobile)	Trois semaines
	Développer	
	Use case : Effectuer une commande	
	Use case : Envoyer un message	
	Use case : Authentification	
	Use case : Consulter une commande	

Tableau 3:Réalisation des sprints

## III.3 Choix technologiques

### III.3.1 Pourquoi utiliser un Framework ?

Commençons par une petite définition. Un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture bien défini (squelette). Il est souvent fourni sous la forme d'une bibliothèque logicielle et accompagné d'une documentation sur l'architecture du Framework.

L'objectif essentiel d'un Framework est d'aider les développeurs dans la réalisation de leurs tâches. L'organisation et la structure du Framework visent la productivité maximale du développeur qui va l'utiliser et cela facilite la maintenance du logiciel.

Voici les trois raisons à utiliser un Framework :

- **Rapidité** : le Framework permet un gain de temps et une livraison beaucoup plus rapide qu'un développement à zéro.
- **Organisation** : l'architecture d'un Framework favorise une bonne organisation du code source (modèle MVC par exemple).
- **Maintenabilité** : l'organisation du Framework facilite la maintenance du logiciel et la gestion des évolutions.

Les Framework permettent un développement des applications sécurisées est facile grâce aux :

- Systèmes d'authentification
- Gestion des injections SQL
- La protection CSRF (Cross-Site RequestForgery) gérée par la plupart des Framework.

### III.3.2 Le Framework Flutter



Figure 11 : Logo flutter

Flutter [\[5\]](#) est un Framework gratuit et open source de développement natif multiplateforme créé par Google. Flutter prend en charge le mobile (Android et iOS).

Flutter est encore jeune mais il a déjà commencé à faire ses preuves. Le Framework a déjà séduit certains grands noms de la technologie : Alibaba, le géant de l'e-commerce chinois l'a utilisé pour son application Xianyu. Les applications comme Reectliy, Hamilton Musical et bien d'autres utilisent Flutter. Google l'utilise également dans ses produits (Google Ads et Google Greentea), il y a une forte communauté qui se bâtit autour du Framework. Flutter est basé sur Dart, un langage de programmation moderne (développé par Google) qui est très facile à apprendre si vous connaissez déjà Java ou JavaScript. Vous pouvez commencer à vous familiariser avec Dart en lisant sa documentation qui est de qualité, facile à suivre et disponible sur le site officiel. Dart avait été conçu au départ par Google pour le web. En plus de pouvoir

utiliser Dart avec Flutter, vous pouvez créer une application Web, un script de ligne de commande ou une application côté serveur. Dart compile en Javascript et en langage machine.

Flutter est livré avec un grand nombre de widgets UI prêts à l'emploi. Toute l'interface utilisateur d'une application Flutter est écrite en Dart. Vous n'avez pas besoin d'utiliser des DSL UI externes (c'est-à-dire des fichiers HTML ou XML).

### III.3.2.1 Pourquoi choisir Flutter ?

Voici une comparaison [6] entre les multiplateformes de développement d'applications mobiles.

Attribute	 React Native	 Xamarin	 Ionic	 Flutter
 Programming Language	JavaScript + Swift/Objective-C or Java	C# with .net environment	HTML5, CSS, and JavaScript + Typescript	Dart
 Performance	Close-to-native ★★★★★	<div>Xamarin</div> <div>ios/Android</div> <div>Close-to-native</div> <div>★★★★★</div>	<div>Xamarin</div> <div>Forms</div> <div>Moderate</div> <div>★★</div>	Amazing ★★★★★
 GUI	Use Native UI Controllers	Use Native UI Controllers	HTML/CSS	Use Proprietary Widgets and deliver amazing UI
 Market and Community Support	Very Strong 👑	Strong	Strong	Not very popular
 Use Cases	All apps	Simple apps	Simple apps	All apps
 Code Reusability	90% of code is reusable	96% of code is reusable	98% of code is reusable	50-90% (approx.) of code is reusable
 Popular Apps	Facebook, Instagram, Airbnb, UberEats	Ola, the World Bank, Storyo.	JustWatch, Pacifica, and Nationwide.	Hamilton
 Pricing	Open-source	Open-source + Paid as well	Open-source + Paid as well	Open-source

Figure 12 : Tableau comparatif

D'après le tableau ci-dessus, on distingue l'importance d'utilisation de Flutter dans les critères suivants :

- **Très bonnes performances**
- **Possibilité d'intégrer plus facilement des animations**
- **Beaucoup de composants sont disponibles (un grand nombre de widgets)**
- **Les éléments qui n'existent pas en flutter peuvent être créés via des ponts vers du code natif en Swift ou Kotlin**
- **Riche documentation**
- **Compilation extrêmement rapide**
- **Les corrections des bugs sont rapides et fréquentes**
- **Fortement poussé par Google**

Je trouve que Flutter est le meilleur outil de développement des applications mobile multiplateforme, ce qui explique mon choix pour l'outil de développement mobile de mes applications.

### **III.3.3 Le Framework Angular**



Figure 13 : Logo angular

Angular [\[7\]](#) est un Framework JavaScript développé par Google et reconnu pour être l'un des Framework les plus complets. Il embarque en effet nativement nombre de fonctionnalités le qualifiant pour le développement d'applications web, même les plus complexes. Grâce à son évolution, Angular est cependant également adapté à tout projet d'application de bureau ou mobile. Rapide à déployer et puissant, il sait convaincre petites et grandes entreprises.



### III.3.3.1 Pourquoi choisir Angular ?

L'importance d'utilisation angular dans les critères suivants :

- **Des applications responsives** : Les applications développées sont responsives et s'adaptent à tous les écrans :desktop, mobile et tablette.
- **Interfaces haut de gamme** : Il est un excellent choix pour créer des solutions aux interfaces graphique complexes. Il est également choisi pour créer d'excellentes animations graphiques.
- **La facilité de prise en main des projets** : La structure du code est facile à lire et permet de bien se retrouver dans le projet
- **Une forte maintenabilité** : Le Framework facilite la maintenance des applications web ou mobile. Il est plus facile avec cette technologie de se projeter dans des évolutions et dans le temps.

### III.3.4 Le Framework Spring Boot



Figure 14 : Logo SpringBoot

Le Framework Spring Boot [\[8\]](#) permet de créer rapidement des API REST solides selon une architecture de code respectant le modèle MVC. Dans ce document, la structure principale ainsi que ses éléments principaux seront présentées de façon à construire graduellement le squelette d'une API Rest.

#### III.3.4.1 Pourquoi choisir spring boot

- **Légèreté** : Spring Boot a la particularité d'être très léger et d'embarquer avec lui le strict minimum pour faire tourner votre service.
- **Intégration facilitée** : Spring Boot s'intègre particulièrement bien dans une architecture orientée micro services et c'est l'un des seuls. En effet, l'adoption des architectures micro services au sein des organisations étant relativement récentes, il n'existait

pas dans l'univers de Java de Framework capable de créer des services suffisamment légers et performants.

- **Simplicité de prise en main** : Spring Boot permet donc de créer une API de services très simplement.

### **III.3.5 API**

API signifie « Application Programming Interface ». C'est un ensemble de moyens utilisés par un logiciel pour donner accès à certaines de ses fonctionnalités ou ressources. En d'autres termes, c'est un moyen mis en place par un logiciel pour que d'autres logiciels puissent interagir simplement avec lui. On parle d'API à partir du moment où une entité informatique cherche à agir avec ou sur un système tiers, et que cette interaction se fait de manière normalisée en respectant les contraintes d'accès définies par le système tiers.

#### **III.3.5.1 Qu'est-ce qu'une API REST ?**

Une API REST se doit d'être sans état. La communication entre le client et le serveur ne doit pas dépendre d'un quelconque contexte provenant du serveur ainsi, chaque requête doit contenir l'ensemble des informations nécessaires à son traitement. Cela permet de traiter indifféremment les requêtes de plusieurs clients de multiples instances de serveurs.

#### **III.3.5.2 API REST ou API SOAP ?**

Il existe actuellement deux types d'architecture très utilisés pour les API : Simple Object Access Protocol (SOAP) et Représentationnel State Transfer (REST).

Les deux techniques ont des problèmes à prendre en compte au moment de décider quel protocole utiliser.

Il est important de préciser que même si SOAP et REST présentent des similitudes en utilisant le protocole HTTP, SOAP est un ensemble plus rigide que REST.

REST a une architecture qui ne nécessite pas de traitement et qui est naturellement plus flexible. SOAP et REST reposent sur des règles bien établies que tout le monde a accepté de respecter dans l'intérêt de l'échange des informations.

### III.3.5.3 Pourquoi choisir REST ?

L'Api avec laquelle l'application mobile interagit avec le Back-end est développé avec REST pour les raisons suivantes :

- **En raison de son évolutivité** : Ce protocole se distingue par son évolutivité. Grâce à la séparation client / serveur, le produit peut être mis à l'échelle par une équipe de développement sans trop de difficultés.
- **En raison de son indépendance** : En raison de la séparation entre le client et le serveur, le protocole facilite le développement indépendant des différents domaines d'un projet. De plus, l'API REST s'adapte à tout moment à la syntaxe et à la plateforme de travail. Cela offre la possibilité d'essayer plusieurs environnements en cours de développement.

### III.3.6 MySQL



Figure 15 : Logo MySQL

MySQL [\[9\]](#) est un Système de Gestion de Base de Données (SGBD) parmi les plus populaires au monde. Il est distribué sous double licence, une licence publique générale GNU et une propriétaire selon l'utilisation qui en est faite. La première version de MySQL est apparue en 1995 et l'outil est régulièrement entretenu.

Ce système est particulièrement connu des développeurs pour faire partie des célèbres quatuors: WAMP (Windows, Apache, MySQL et PHP), LAMP (Linux) et MAMP (Mac). Ces packages sont si populaires et simples à mettre en œuvre que MySQL est largement connu et exploité comme système de gestion de base de données pour des applications utilisant PHP. C'est d'ailleurs pour cette raison que la plupart des hébergeurs web proposent PHP et MySQL.

#### III.3.6.1 Pourquoi choisir MySQL ?

- **Rapidité**

Le serveur MySQL est très rapide. Des tests de performances sont disponibles sur le site de MySQL.

- **Facile à utiliser**

MySQL est beaucoup plus simple à utiliser que la plupart des serveurs de bases de données commerciaux.

- **Coût**

Jusqu'à la version 3.20.32a, MySQL était sous licence GPL. Il faut une licence pour faire tourner MySQL sous Windows. Sous les autres plates-formes, l'utilisation de MySQL est gratuite dans un but non commercial.

## **III.4 Environnement de travail**

### **III.4.1 Environnement matériel**

Pour le développement de mon application, j'ai utilisé un PC portable dont les caractéristiques suivantes :

- Processeur : Intel core TM i5 CPU
- Système d'exploitation : Windows 7 64 bits
- Mémoire vive : 6 Go

### **III.4.2 Environnement logiciel**

#### **III.4.2 .1 Visual Studio Code**

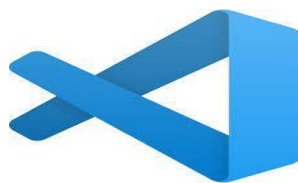


Figure 16 : Logo Flutter

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code.[\[10\]](#)

### **III.4.2 .2 Xampp**



Figure 17 : Logo Xampp

XAMPP est un ensemble de logiciels libres. Le nom est un acronyme venant des initiales de tous les composants de cette suite. Ce dernier réunit donc le serveur Web Apache, la base de données relationnelle et système d'exploitation MySQL ou Maria DB ainsi que les langages scripts Perl et PHP.[\[11\]](#)

### **III.4.2 .3 Postman**



Figure 18 : Logo Postman

Postman est une application permettant avec un navigateur Web de lancer des appels d'API et de les tester. Postman permet d'envoyer des requêtes vers l'API de site en lui ajoutant des entêtes clés / valeurs puis il permet de formater le résultat sur plusieurs formats tels que JSON. XML. HTML et autres.[\[12\]](#)

### **III.4.2 .4 Eclipse**



Figure 19 : Logo Eclipse

Eclipse est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation.[\[13\]](#)

### III.4.2 .5 PhpMyAdmin (PMA)



Figure 20 : Logo PhpMyAdmin

PhpMyAdmin c'est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.[\[14\]](#)

## III.5 Enchainements des interfaces de l'application

### III.5.1 Application Web

Dans cette partie, je vais vous présenter les différentes interfaces de l'application web

- Interface d'authentification

Permet à l'utilisateur de Connecter à l'aide de son identification (login mot passe).

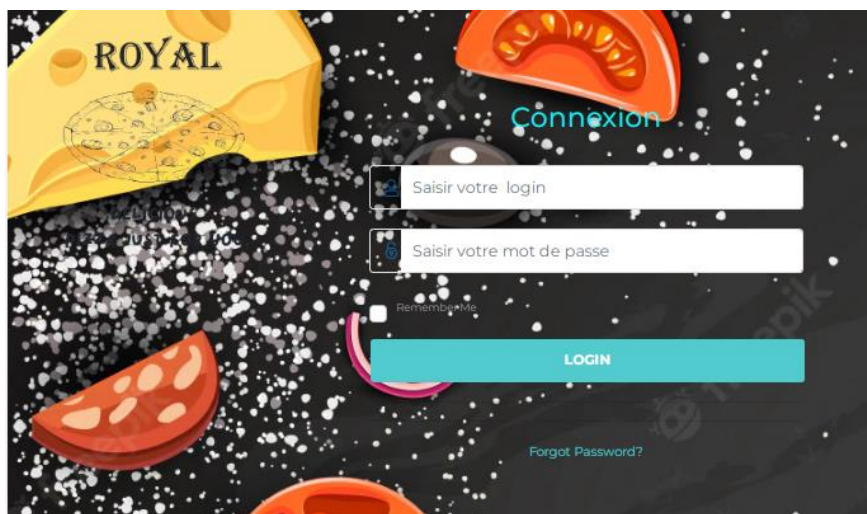


Figure 21 : Interface d'authentification

- Tableau de bord :

Cette interface de tableau de bord est l'accueil de l'application, où l'utilisateur peut naviguer dans toute l'application.

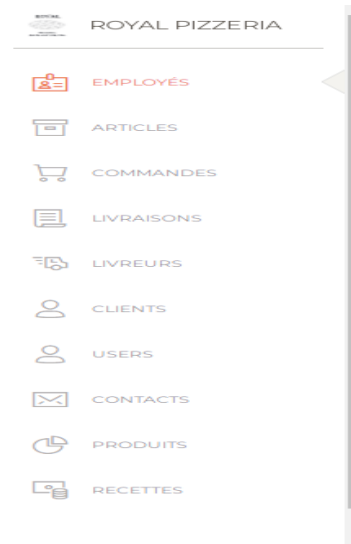


Figure 22 : Tableau de bord

- Interface client :

Dans cette partie, l'administrateur peut ajouter, modifier, supprimer les clients.













Clients				
Search...				
MOEZ OUERIMMI GÉRER				
ID	Nom	Prenom	Tel	Actions
2	oueghlani	nacer	25888777	 
72	ouederni	sami	23415627	 
125	souabni	wael	20450765	 
126	azzabi	hanen	29071286	 
124	chrigui	mohamed	27815656	 
123	oueghlaniiii	nacer	25888777	 

Figure 23 : Interface client

- Interface client :

Interface commande : Dans cette partie, l'administrateur peut ajouter, modifier, valider, supprimer les commandes.

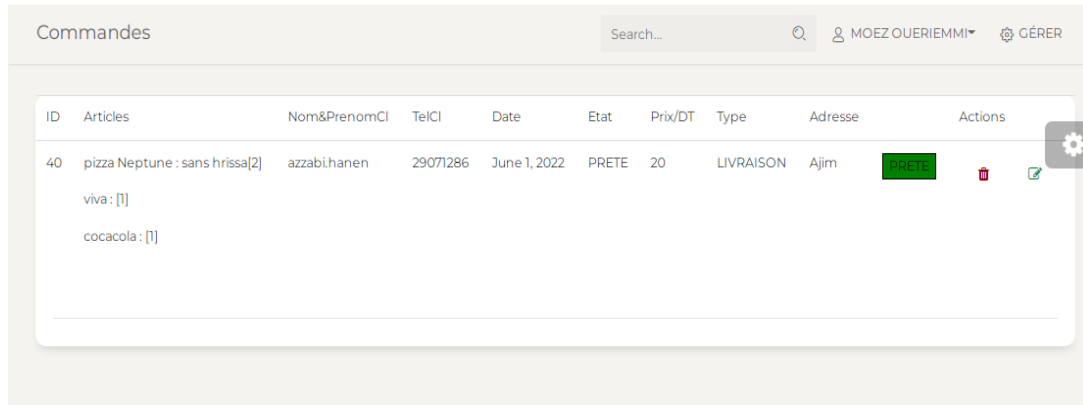


Figure 24 : Interface de commande

- Interface livraison :

Dans cette partie, l'administrateur peut ajouter, modifier, supprimer les livraisons.

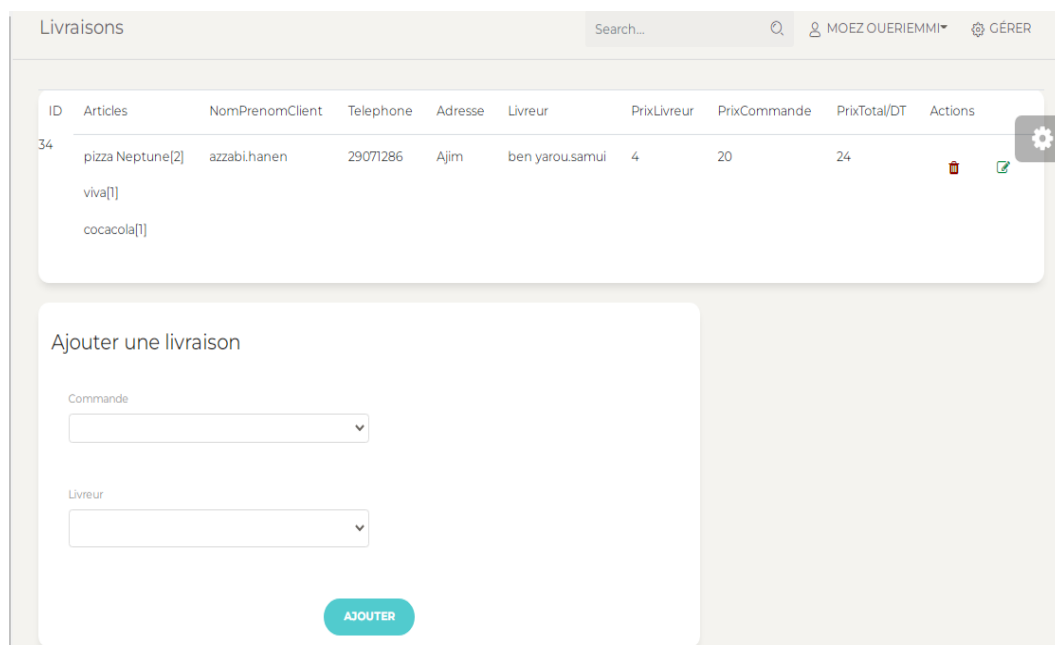


Figure 25 : Interface de livraison



- Interface livreur :

Dans cette partie, l'administrateur peut ajouter, modifier, supprimer les livreurs.

ID	Nom	Prenom	Tel	Adresse	prix(DT)/Livraison	Actions
3	ben yarou	samui	41758496	houmtsouk	4	
127	ben chikh	ayoub	24356443	sedouikech	5	

Ajouter un Livreur

Figure 26 : Interface de livreur

- Interface article :

Dans cette partie, l'administrateur peut ajouter, modifier, supprimer les articles.

ID	image	Nom	Ingredients	Categorie	Prix/DT	Actions
35		Plat escalope	(salade,hrissa,escalope,oeuf,frites)	PLAT	8	
33		pizza Neptune	(sauce tomate,fromage,thon)	PIZZA	10	
34		Sandwich	(hrissa,salade,escalope,oeuf,frites)	SANDWICH	5	
45		TACOS	(hrissa,salade,escalope,oeuf,frites)	SANDWICH	8	
69		Boga		BOISSON	2	
107		viva		BOISSON	2	
108		cocacola		BOISSON	2	

Figure 27 : Interface d'article

- Interface user :

Dans cette partie, l'administrateur peut ajouter, modifier, supprimer les utilisateurs.

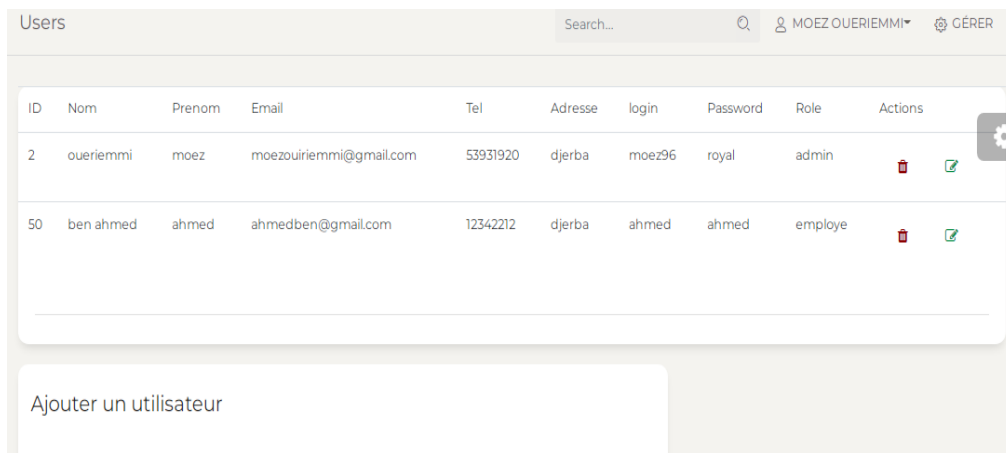


Figure 28 : Interface d'utilisateur

- Interface employée :

Dans cette partie, l'administrateur peut ajouter, modifier, supprimer les employés.

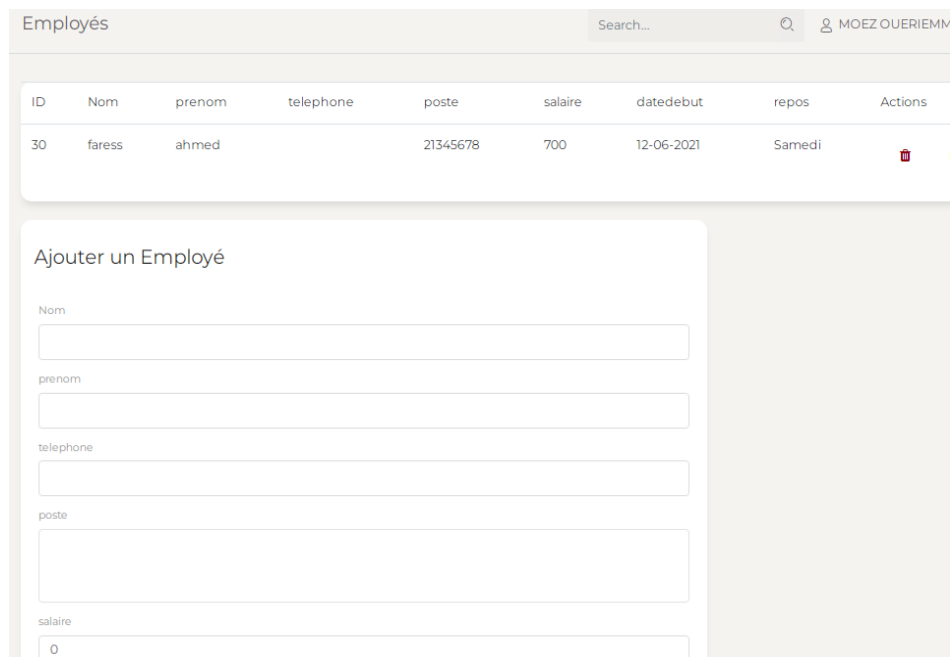


Figure 29 : Interface d'employé

- Interface produit :

Dans cette partie, l'administrateur peut ajouter, modifier, supprimer les produits.

ID	image	Nom	Quantité(KG/L)	Prix/DT	Actions
5		farine	30	30	
118		huile	20	30	

Ajouter un Produit

Figure 30 : Interface de produit

### III.5.2 Application Mobile



Figure 31 : Interface d'accueil



Figure 32 : Liste des pizzas



Figure 33 : Liste des boissons



Figure 34 : Liste des plats

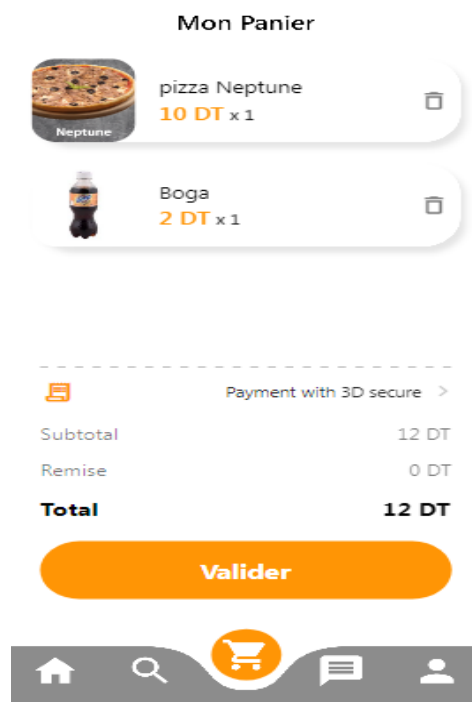


Figure 35 : Panier



Figure 36 : Liste des crêpes

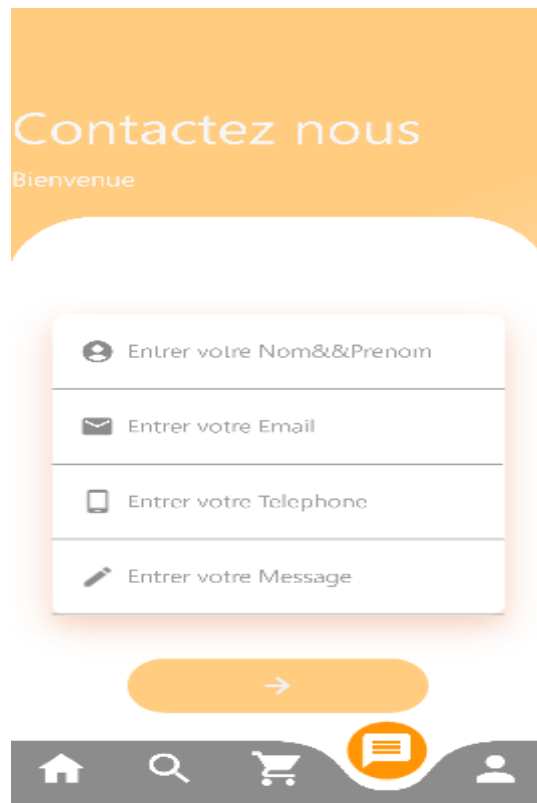


Figure 37 : Interface de contact

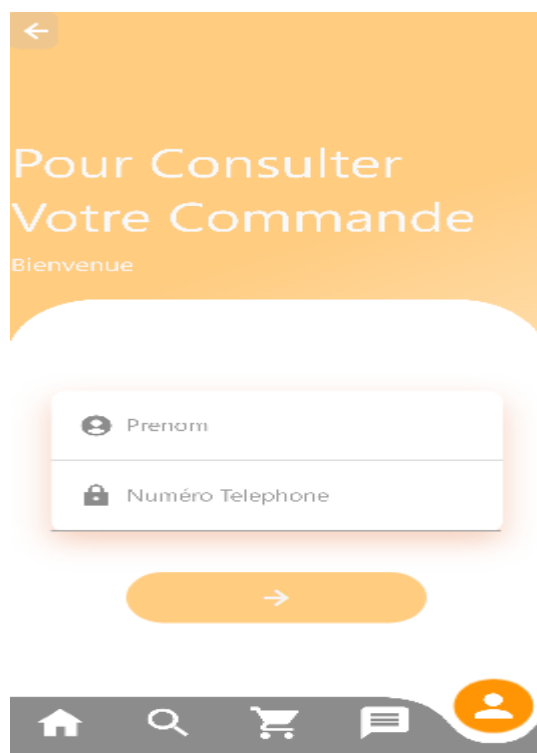


Figure 38 : Interface d'authentification



Figure 39 : Consulter la commande

## III.6 Conclusion

Comme titre de conclusion, dans ce dernier chapitre nous avons présenté la méthodologie de travail et l'environnement de travail (l'environnement matériel et logiciel, choix technique). Aussi nous avons fait un défilement des interfaces de notre application.

## Conclusion Générale

Au terme de ce projet, j'ai réalisé une application informatique qui a pour but la mise en place la gestion d'un restaurant. Par ailleurs, cette application a été une bonne opportunité pour approfondir mes connaissances aux niveaux professionnel et technique. Dans ce rapport, j'ai suivi une démarche structurée et bien déterminée.

Il faut avouer que la tâche n'a pas été facile, mais après deux années de formation en développement web et mobile, dans le cadre de mastère TICIT, j'avais tous les outils nécessaires pour répondre au problème qui a été posé. C'était l'occasion pour moi de consolider mes acquis en matière de développement. La résolution du problème est passée par les étapes suivantes :

- L'analyse des besoins, la conception de mon système et la réalisation. Pour améliorer mon application, il faut d'abord la mettre sur play store.
- Ensuite intégrait un module de paiement en ligne.

Riche de cette nouvelle expérience, cela m'a réconforté dans l'idée, de chercher des autres projets en relation avec le développement web et mobile.



# Bibliographie

- [1] <https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/>
- [2] <https://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/use-case-diagramme/>
- [3] <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes>
- [4] <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/diagrammes-de-sequence/>
- [5] <https://dan.com/fr-fr/buy-domain/flutter.dev>
- [6] <https://www.apptunix.com/blog/frameworks-cross-platform-mobile-app-development/>
- [7] <https://www.50a.fr/0/angular>
- [8] <https://www.infoq.com/fr/articles/microframeworks1-spring-boot/>
- [9] <https://www.futura-sciences.com/tech/definitions/internet-mysql-4640/>
- [10] <https://blog.webnet.fr/visual-studio-code/>
- [11] <http://www.standard-du-web.com/xampp.php>
- [12] <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-api-web/>
- [13] <https://www.techno-science.net/definition/517.html>
- [14] <https://docs.phpmyadmin.net/fr/latest/intro.html>