

2022 - 2023 PROJET DE FIN D'ÉTUDES

DIPLÔME NATIONAL D'INGÉNIEUR

SPÉCIALITÉ: SIM

SUJET DU RAPPORT DE STAGE

Réalisé par:

Encadrant ESPRIT:

Encadrant Entreprise:









Résumé

Dans le cadre de notre projet de fin d'études à l'Ecole Sup Privée d'ingénierie et de

Technologies (ESPRIT) et en vue de l'obtention du Diplôme National D'ingénieur en

Informatique, j'ai effectué mon stage de fin d'études au sein de la société « DoT-IT ».

Ce projet consiste à concevoir et développer une plateforme de commerceélectronique

web et mobile en marque blanche pour les entreprises souhaitant se lancer dans le

commerce en ligne. Notre plateforme permettra aux entreprises de personnaliser leur

boutique en ligne avec leurpropre marque, tout en offrant des fonctionnalités avancées

telles que la gestion des stocks, la gestion des commandes, le traitement des paiements, la

gestion des expéditions et des retours, ainsi que des outils marketing pour attirer de

nouveaux clients et développer leur activité en ligne. Nous visons à offrir des

opportunités importantes pour les entreprises souhaitant se développer dans le secteur

de l'e-commerce en Tunisie, en offrant une solution clé en main pour les aider à atteindre

leur objectif.

Mots clé: Node JS, Flutter, NoSQL, MongoDB, Firebase, UML...

Abstract

As part of our final year project at the Private Higher School of Engineering and

Technologies (ESPRIT) and in order to obtain the National Engineering Degree in

Computer Science, I completed my final year internship at the company "DoT-IT".

This project consists of designing and developing a white-label web and mobile e-

commerce platform for businesses wishing to venture into online commerce. Our

platform will allow businesses to customize their online store with their own brand,

while offering advanced features such as inventory management, order management,

payment processing, shipping and returns management, as well as marketing tools to

attract newcustomers and grow their online business. We aim to provide significant

opportunities for businesses looking to expand in the e-commerce sector inTunisia, by

offering a turnkey solution to help them achieve their goal.

Keywords: Node JS, Flutter, NoSQL, MongoDB, Firebase, UML.

Dédicace

Remerciements

Tableau des Matières

Introduction générale	
CHAPITRE I : Contexte et Objectifs du Projet	2
Introduction	3
1. Contexte général	3
1.1. Cadre du projet	3
1.2. Présentation de l'entreprise d'accueil	3
1.3. Concept principal	4
2. Problématique	5
3. Etude de l'existant	5
3.1. Solution Existantes	5
3.1.1. Solutions en Tunisie	5
3.1.2. Solution à l'étranger	7
3.2. Critique de l'existant	9
4. Solution proposée	10
5. Méthodologie de développement	11
5.1. Etude de Méthodologie de développement	11
5.2. Présentation de quelques méthodologies	11
5.2.1. Agile	11
5.2.1.1. Scrum	12
5.2.1.2. La méthode XP	13
5.2.2. Processus unifié	14
5.2.2.1. Two Track Unufied Process (2TUP)	15
5.2.2.2. Rational Unified Process(RUP)	15
5.3. Etude comparative et évaluation	15
5.4. Choix de la méthodologie de travail	16
5.5. Diagramme de Gantt	18
Conclusion:	19
CHAPITRE II : Analyse et spécification des Besoins	20
Introduction	21
1. Besoins fonctionnels	21
2. Besoins non fonctionnels	23

2.1.	Exigence de qualité	23
2.3.	Contraintes de conception	24
3. Beso	oins techniques	25
3.1.	Choix technologique	25
3.1.1	. Front-End	25
3.1.2	2. Back-End	27
3.1	1.2.1. Serveur	27
3.1	1.2.2. Base de données	27
3.1.3	3. Technologies retenues	28
3.2.	Choix architectural	28
3.2.1	. Modèle MVVM	29
3.2.2	2. Modèle MVC	29
3.2.3	3. Architecture retenue	30
Conclusio	on	31
CHAPIT	RE III : CONCEPTION	32
Introduct	tion	33
1. Con	ception générique	33
1.1.	Le langage UML	33
1.2.	Les Diagrammes d'UML	33
2.1.	Les vues de Kruchten	34
2.2.	Modèles d'UML utilisés	36
2.3.	Vue des cas d'utilisation : Diagramme de cas d'utilisation	36
2.3.1	. Cas d'utilisation général	36
2.3.2	2. Description détaillée des cas d'utilisation	39
2.3	3.2.1. cas d'utilisation Créer compte et S'authentifier	39
2.3	3.2.2. Les Cas d'utilisation consulter produits et gérer panier	41
2.3	3.2.3. Le cas d'utilisation gérer ses commandes	43
2.3	3.2.4. Les cas d'utilisation gérer les catégories et gérer les produits	44
2.3	3.2.5. Le cas d'utilisation Gérer utilisateurs	47
2.4.	Vue des processus : diagramme de séquence	49
2.4.1		
2.4.2	2. Diagramme de séquence Gérer panier	50
2.4.3		
	Vue logique : Diagramme de classes	
2.6.	Vue de Composants : Diagramme de Composants	52
Conclusio	on	53

CHAP	ITRE IV : Réalisation	54
Introdu	ction:	55
1. Er	nvironnement logiciel	55
1.1.	Environnement et technologies de développement	55
1.2.	Environnement de design	57
2. Pr	ésentation de la solution mobile	58
2.1.	Interface d'accueil	58
2.2.	Interface menu	59
2.3.	Interface produit	60
2.4.	Interface login et registre	60
2.5.	Interface information de livraison	61
2.6.	Interface de Paiement	62
2.7.	Interface favoris	63
Conclu	sion	63
Conclu	sion générale	64
Bibliog	raphie	65

Liste des abréviations

PME: Petites et moyennes entreprises

PMI: Petites et moyennes industries

XP: eXtreme Programming

MVC: Modèle-Vue-Contrôleur

MVVM: Modèle-Vue-Vue Modèle

JS: JavaScript

ERP: Entreprise Resource Planning

UML: Unified Modeling Language

SPA : Single Page Application

HTML: Hypertext Markup Language

DOM: Document Object Model

PHP: Hypertext Preprocessor

DLA: Data Access Layer

IDE : Environnement de développement intégré

IHM: Interface Homme Machine

Liste des Figures

Figure 1-1: Logo DoT-IT	4
Figure 1-2 : Page d'accueil du site web Tayara.tn ⁽²⁾	6
Figure 1-3 : Page d'accueil du site web Wiki.Tn (3)	7
Figure 1-4: Page d'accueil du site web «eBay.com» ⁽⁴⁾	8
Figure 1-5: Illustration de la solution	
Figure 1-6: Les étapes du processus Agile ⁽⁵⁾	12
Figure 1-7: Les étapes du processus Scrum (6)	13
Figure 1-8 : principe et fonctionnement de la méthode XP	14
Figure 1-9 : Les étapes du processus unifié (7)	
Figure 1-10 : Les étapes du processus 2TUP	
Figure 1-11 : Diagramme de Gantt et planning prévisionnelle	19
Figure 2-1 : architecture typique d'une application Flutter ⁽⁹⁾	26
Figure 2-2 : Choix technologique de notre application	28
Figure 2-3 : Architecture MVVM	29
Figure 2-4 : Architecture MVC	30
Figure 3-1: Vue 4+1 (Kruchten)	35
Figure 3-2 : Diagramme de cas d'utilisation général	37
Figure 3-3 : Diagramme de cas d'utilisation Créer compte et S'authentifier	39
Figure 3-4 : Diagramme des cas d'utilisation Consulter les produits et Gérer panier	41
Figure 3-5 : Diagramme de cas d'utilisation Gérer ses Commandes	43
Figure 3-6 : Diagramme des cas d'utilisation « Gérer les catégories » et « Gérer lesproduits »	44
Figure 3-7 : Diagramme de cas d'utilisation « Gérer les Utilisateurs »	47
Figure 3-8 : Diagramme de séquence « S'authentifier »	
Figure 3-9 : Diagramme de séquence « Gérer panier »	50
Figure 3-10 : Diagramme de séquence « Gérer les produits »	51
Figure 3-11 : Diagramme de classe	52
Figure 3-12 : Diagramme de composant	53
Figure 4-1: maquette I de l'application	57
Figure 4-2: Maquette II de l'application	58
Figure 4-3: Interface d'accueil application mobile	59
Figure 4-4: Interface menu application mobile	59
Figure 4-5 : Interface produit application mobile	60
Figure 4-6 : Interfaces SignUp & LogIn application mobile	61
Figure 4-7: Interface information de livraison	
Figure 4-8 : Interface de paiement	62
Figure 4-9: Interface favoris	63

Liste des tableaux

Tableau 1-1 : Identité de la société	3
Tableau 1 -2 : les points forts et faibles de tayara.tn	6
Tableau 1 -3: les points forts et faibles de tayara.tn	7
Tableau1-4: les points forts et faibles d'eBay	8
Tableau 1-5: Tableau comparatif des applications web existantes	10
Tableau 1-6 : Tableau comparatif des méthodologies de développement	15
Tableau2-1 : Avantages et inconvénients de l'utilisation de Flutter	26
Tableau 2-2 : Bases de données Relationnelles vs NoSQL	27
Tableau 3-1: Tableau des cas d'utilisation	37
Tableau 3-2 : Description textuelle de cas d'utilisation Créer compte	40
Tableau 3-3: Description textuelle de cas d'utilisation « S'authentifier »	40
Tableau 3-4 : Description textuelle des cas d'utilisation « Consulter les produits » et «GérerPanier »	42
Tableau 3-5 : Description textuelle de cas d'utilisation « Gérer ses commandes »	43
Tableau 3-6 : Description textuelle de cas d'utilisation « Gérer les catégories »	45
Tableau 3-7 : Description textuelle de cas d'utilisation « Gérer les produits »	46
Tableau 3-8 : Description textuelle de cas d'utilisation « Gérer les Utilisateurs »	48

Introduction générale

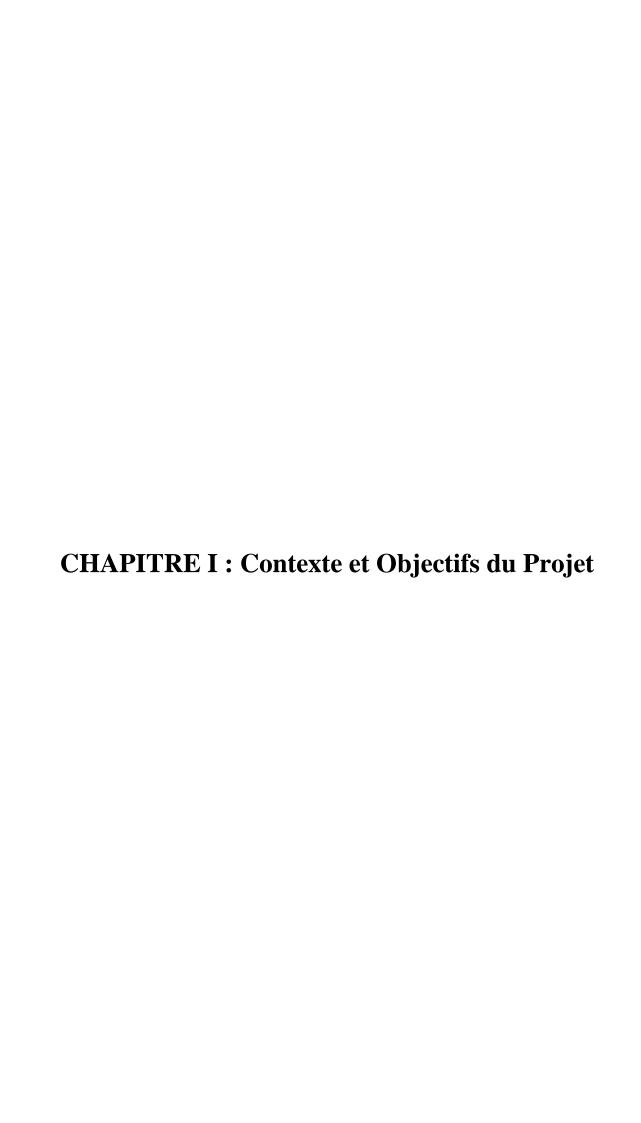
Le commerce électronique, ou e-commerce, est l'un des secteurs les plus dynamiques de l'économie numérique. Devenu un moteur essentiel de la croissance économique mondiale, il offre de nouvelles opportunités aux entreprises, aux consommateurs et aux gouvernements. La pandémie de Covid-19 a accentué l'importance du commerce électronique, les consommateurs ayant dû se tourner vers les achats en ligne pour éviter les contacts physiques et la propagation du virus.

C'est dans ce contexte que nous avons réalisé notre stage dans une entreprise spécialisée dans le e-commerce. Notre mission était de développer une plateforme de vente en ligne visant à répondre aux besoins des clients et à améliorer leur expérience d'achat en ligne. L'objectif de notre projet était de fournir une plateforme de commerce électronique facile à utiliser, sécurisée et offrant une grande variété de produits.

Ce rapport détaille les différentes étapes de notre projet de développement de plateforme de ecommerce. Il est structuré en plusieurs chapitres :

- ❖ Le premier chapitre présente le contexte et les objectifs du projet, ainsi qu'une analyse de l'état du marché du e-commerce.
- ❖ Le deuxième chapitre aborde la collecte des besoins, les spécifications fonctionnelles et non-fonctionnelles, ainsi que les choix de technologies et d'outils de développement, en plus de la gestion du projet via le processus de développement 2TUP.
- **❖ Le troisième chapitre**, intitulé « Conception », décrit l'architecture générale de notre application web et présente l'environnement de conception.
- ❖ Le quatrième chapitre présente l'environnement de développement matériel et logiciel adopté pour la partie web, avec quelques captures d'écran du site.
- ❖ Le cinquième chapitre traite de l'environnement de développement matériel et logiciel, ainsi que des technologies adoptées pour la partie mobile, incluant des captures d'écran de l'application.

Enfin, le rapport se conclut par une synthèse du travail réalisé



Introduction

Dans ce chapitre introductif, la première partie se consacre à la présentation de l'organisme d'accueil, « DoT-IT », en mettant en avant son expertise et ses atouts. La seconde partie est dédiée à l'idée générale de notre projet et aux problématiques qui ont motivé l'organisme à entreprendre ce projet. Ensuite, nous examinerons les solutions existantes en les critiquant, avant de présenter les solutions proposées. Enfin, le chapitre se conclut par une analyse des différentes méthodes de développement et une comparaison visant à choisir la méthode la plus appropriée pour l'implémentation de notre plateforme.

1. Contexte général

Cette partie présente le cadre général du projet ainsi que ses principaux objectifs.

1.1.Cadre du projet

Le présent travail s'inscrit dans le cadre du projet de fin d'étude en vue de l'obtention du Diplôme National D'ingénieur en Informatique délivré par Ecole Sup Privée d'ingénierie et de Technologies (ESPRIT). Notre stage s'est déroulé au sein de la société " Dot-IT " pour une durée de six mois du 01 février 2023 au 31 juillet 2023.

1.2. Présentation de l'entreprise d'accueil

DoT-IT e-business solutions, voir Figure I.1, est basée à Sousse et comporte une équipe dynamique, des consultants et un large réseau de partenaires permettant d'offrir un service informatique global dans différents domaines en utilisant la majorité des technologies connues du marché. Elle a pour métier un service informatique de qualité auprès des PME/PMI, à très forte valeur ajoutée misant sur l'innovation et l'expertise. La société qui s'engage à garantir la satisfaction de ses clients.

Tableau 1-1 : Identité de la société

Raison sociale	DoT-IT			
Adresse	Technopole de Sousse, pépinière d'entreprise BA22_A23			
Téléphone	70 834 163			
Courriel	contact@dotit.com.tn			
Site web	www.dotit-corp.com			

Compétences	E-commerce solutions, NodeJS, Flutter, MySQL, Laragon, MongoDB, Firebase, UML
-------------	---

DoT-IT est une entreprise de développement de logiciels qui propose une gamme de solutions dans divers domaines liés à :

- ✓ Les systèmes ERP.
- ✓ Les solutions logicielles de gestion industrielle et manufacturière.
- ✓ L'assistance technique dans divers processus industriels.
- ✓ Le conseil et l'analyse dans la mise en place des systèmes d'information.
- ✓ L'expertise en préparation de certifications.
- ✓ L'externalisation informatique (IT outsourcing).
- ✓ Le développement de logiciels personnalisés adaptés aux besoins des clients.



Figure 1-1: Logo DoT-IT

1.3. Concept principal

Le monde actuel se distingue par sa transformation numérique, qui constitue l'une de ses évolutions les plus significatives. Nous sommes témoins d'une véritable révolution en constante évolution. En 2010, une étude réalisée par IMS Research et rapportée par Le Monde Informatique (1) a recensé 5 milliards d'appareils connectés à Internet, et ce chiffre devrait dépasser les 71 milliards d'ici 2025.

Cette révolution est désormais bien ancrée dans notre quotidien et son impact se fait de plus en plus fort sur la façon dont nous, consommateurs, nous approprions l'information, la publicité et donc les marques.

Face à ce constat et à cette croissance, nous ne pouvons ignorer le déplacement des priorités

des opérations de vente sur le terrain aux ventes virtuelles, cela nous oblige à donner plus

d'accent sur les ventes électroniques. Les e-boutiques s'accélèrent et faciliter le traitement

traditionnel, comme les processus de commande et de facturation en offrant un accès rapide et

facile au produit. Le prochain défi est d'amener les internautes d'aujourd'hui nos futurs clients

à travers une interface agréable et une plateforme performante satisfaire les besoins des clients

et faciliter le processus d'achat de biens et des services.

2. Problématique

La vente en ligne est en constante progression, avec de plus en plus de consommateurs optant

pour l'achat de produits en ligne. Les sites de e- commerce en marque blanche offrent une

opportunité pour les entreprises de proposer leur propre plateforme de vente en ligne sans avoir

à développer leur propre site. Cependant, malgré cette facilité, il y a des défis à relever pour les

sites de e-commerce en marque blanche afin de satisfaire les besoins des consommateurs. Une

problématique qui se pose est donc : Comment développer une plateforme de e-commerce en

marque blanche web et mobile répondant aux attentes des consommateurs en termes de

navigation, d'expérience utilisateur, de sécurité des données et de service client ?

3. Etude de l'existant

Avant de réaliser un projet, il est essentiel de mener une étude préalable pour évaluer les forces

et les faiblesses des systèmes existants ainsi que les besoins du client. Cette section présente

une étude basée sur l'analyse de différentes applications web spécialisées dans le domaine de la

vente en ligne en Tunisie et à l'étranger. Cette analyse nous permet d'identifier les avantages et

les inconvénients de chaque système afin de déterminer les besoins et de les prendre en compte

dans la conception et la réalisation de notre projet.

3.1. Solution Existantes

3.1.1. Solutions en Tunisie

Solution 1 : Tayara

Créé en 2012, Tayara.Tn est un site web de petites annonces en Tunisie qui permet aux

utilisateurs de vendre et d'acheter des produits et services en ligne, voir Figure 1.2. Il propose

une large gamme de catégories, notamment l'immobilier, l'automobile, l'électronique, la mode,

5

les services, etc. Les utilisateurs peuvent également publier des annonces gratuitement et rechercher des produits ou services selon leurs préférences.

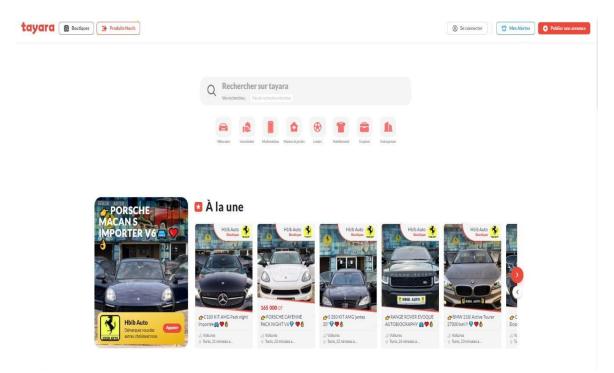


Figure 1-2: Page d'accueil du site web Tayara.tn⁽²⁾

Le tableau ci-dessous présente une synthèse des points forts et des points faibles du site Tayara

Tableau 1 -2 : les points forts et faibles de tayara.tn

Points forts	Points faibles	
Tayara est le site le plus connu en	Il n'y a pas de gestion de produits sur le site, ce qui	
Tunisie grâce aux publicités dans	signifie qu'il est nécessaire de créer une annonce	
les radios, TV	distincte pour chaque produit.	
	Il n'y a pas d'espace admin.	
	Il n'y a pas de procédures de paiement, livraison	

♣ Solution 2 : Wiki

WIKI est une chaîne de magasins basés en Tunisie, voir Figure 1.3, qui se spécialise dans la commercialisation des produits électroménagers, des téléphones mobiles et du matériel informatique.

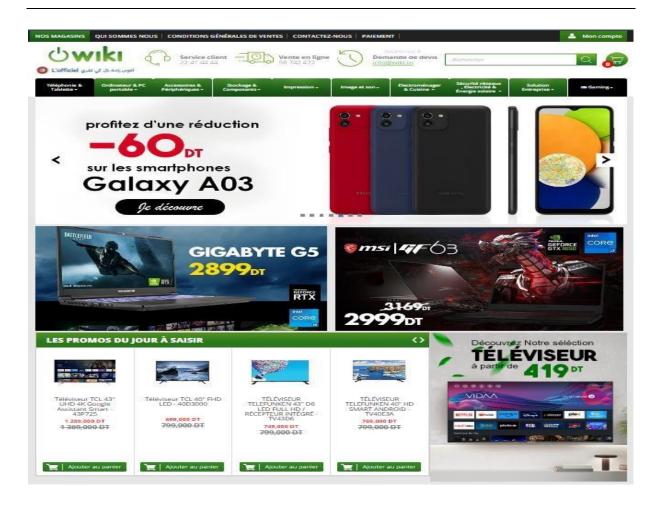


Figure 1-3 : Page d'accueil du site web Wiki. $Tn^{(3)}$

Le tableau ci-dessous présente une synthèse des points forts et des points faibles du site wiki.tn

Tableau 1 -3 : les points forts et faibles de tayara.tn

Points forts	Points faibles	
- Propose une large gamme de produits High	- Le portail est dédié uniquement aux	
Tech en Tunisie.	produits High-Tech.	
- Organise les procédures de paiement et de	- Le portail n'est pas une plateforme en	
livraison.	marque blanche.	
	- Le service clientèle est un peu lent.	

3.1.2. Solution à l'étranger

♣ eBay:

eBay, une plateforme de vente en ligne populaire, permet aux particuliers et aux entreprises de vendre des produits et services dans le monde entier. Les vendeurs peuvent créer des annonces

et les acheteurs peuvent enchérir ou acheter à prix fixe. Disponible dans de nombreux pays et langues, eBay est l'une des plus grandes plateformes de vente en ligne, comme illustré à la Figure 1.4.

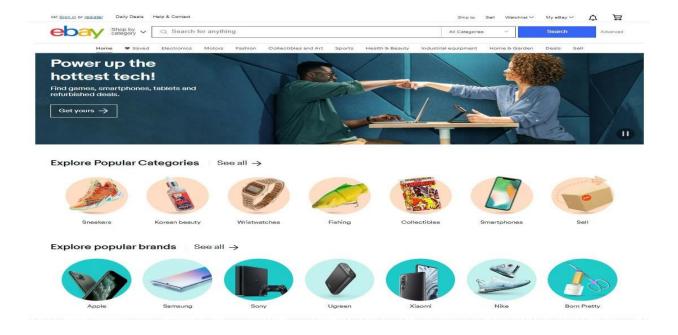


Figure 1-4: Page d'accueil du site web « eBay.com » (4)

Le tableau ci-dessous résume les points forts et les points faibles d'eBay comme plateforme de vente en ligne

Tableau1-4: les points forts et faibles d'eBay

Points forts	Points faibles
- Le vendeur peut construire une boutique en	- Il faut payer une commission sur le prix
ligne au sein du site eBay pour vendre ses	final et des frais d'insertion et de mise en
produits.	vente.
- Paiement facile et sécurisé.	- En Tunisie, il y a des problèmes avec les
	cartes bancaires internationales, rendant
	l'achat et la vente sur des sites
	internationaux comme eBay difficiles.
- eBay est une grande entreprise mondiale	
spécialisée dans le commerce électronique,	
ce qui évite les problèmes liés à	

l'hébergement et aux performances du site web.

3.2. Critique de l'existant

Nous pouvons classer le résultat de l'analyse des applications web existantes mentionnées précédemment selon six critères [Cx] pris en considération dans le processus d'évaluation de ses applications :

[C1] Interface utilisateur et Attirance : Pour que le visiteur soit attiré par le site et puisse naviguer facilement, il est important de cibler son attention et de fournir des orientations claires sur le site.

[C2] Espace Clients: La mise à disposition d'un espace client permet aux utilisateurs de gérer facilement leurs coordonnées, préférences, adresses...

[C3] Gestion des produits : La renommée d'un site web repose sur ses fonctionnalités, et dans notre cas, la gestion des produits est essentielle. Ainsi, nous avons comparé les sites en fonction de la disponibilité de cette fonctionnalité.

[C4] Application mobile associée : Posséder un site web n'est pas une tâche facile, surtout avec des clients et des internautes de plus en plus exigeants.

Il est donc essentiel de leur offrir une expérience améliorée et une disponibilité du contenu accrue. C'est pourquoi nous avons inclus dans notre évaluation la disponibilité d'une version mobile de l'application.

[C5] Paiement en ligne en Tunisie: Le paiement en ligne est aujourd'hui considéré comme la méthode la plus simple, rapide et sécurisée par rapport aux méthodes de paiement traditionnelles. C'est la raison pour laquelle nous avons choisi d'inclure le paiement en ligne comme un critère dans notre évaluation des sites.

[C6] Temps de réponse : La rapidité de chargement des données.

	[C1]	[C2]	[C3]	[C4]	[C5]	[C6]
Tayara.Tn	Faible	Non	Oui	Oui	Non	Lent
	****					****
Wiki.Tn	Moyenne	Oui	Non	Non	Oui	Moyenne
WIKI. I II	****	Oui	Non	Non	Oui	
	~~~~					XXXXX
eBay.com	Bien	Oui	Oui	Oui	Non	Rapide
	****					****

Tableau 1-5: Tableau comparatif des applications web existantes

#### 4. Solution proposée

Après avoir pris en compte les critiques sur l'existant et les résultats de notre étude, il est apparu que les solutions existantes ne répondent pas adéquatement à notre problématique. Par conséquent, nous sommes dans l'obligation de proposer une solution fiable qui répondra à nos besoins et objectifs. Cette nouvelle solution doit combler les lacunes observées dans les solutions existantes. Pour y parvenir, nous avons opté pour la conception et la réalisation de notre propre portail web et mobile qui soutiendra le système d'information du secteur commercial en offrant les fonctionnalités suivantes :

- Interface utilisateur conviviale et responsive pour permettre une expérience utilisateur optimale sur ordinateur, tablette et mobile.
- Un système de panier d'achat pour permettre aux utilisateurs de sélectionner et d'acheter des produits.
- Un système de gestion des stocks pour permettre aux vendeurs de gérer efficacement leur inventaire et de mettre à jour les niveaux de stock en temps réel.
- Un moteur de recherche avancé pour permettre aux utilisateurs de rechercher des produits par catégorie, mot-clé ou par scan.
- Un système de gestion des utilisateurs pour permettre à l'administrateur de gérer les comptes des utilisateurs et de suivre leur historique d'achats.
- Des outils de marketing et de promotion pour permettre aux vendeurs de créer des offres spéciales, des coupons de réduction, des programmes de fidélité, etc.

Notre solution proposée est illustrée dans la figure 1.5 suivante :

# recherche des produits passe des commandes paiement en ligne Commande Statut de la commande Livraison de la commande Livraison de la commande

#### E_commerce en marque blanche

Figure 1-5: Illustration de la solution

#### 5. Méthodologie de développement

La méthodologie de développement est un cadre qui permet de structurer, planifier et contrôler le processus de développement d'une application. Elle consiste à modéliser le système avant sa réalisation pour en comprendre le fonctionnement et garantir sa cohérence. L'utilisation de modèles permet de réduire les coûts et les délais de développement, tout en assurant un niveau de qualité élevé pour le produit final. En effet, un modèle bien conçu est un élément clé pour assurer une maintenance efficace du produit dans le temps.

#### 5.1. Etude de Méthodologie de développement

Avec la multitude de méthodes de développement disponibles, il est devenu difficile de faire un choix. Afin de sélectionner la méthode la plus adaptée, nous avons étudié plusieurs méthodologies de développement objet. Après analyse, nous avons opté pour celle qui répond le mieux à nos besoins.

#### 5.2. Présentation de quelques méthodologies

#### **5.2.1.** Agile

La méthode Agile est une approche itérative et incrémentale de la gestion de projet qui permet de mieux s'adapter aux changements et aux évolutions du projet tout au long de son cycle de vie. Elle favorise la collaboration et la communication entre les membres de l'équipe de développement, ainsi que la réactivité aux feedbacks des utilisateurs. Cette méthode est souvent utilisée pour des projets de développement de logiciels, car elle permet d'optimiser le processus de création de logiciels en fournissant des résultats plus rapides et plus pertinents, tout en limitant les risques et les coûts.

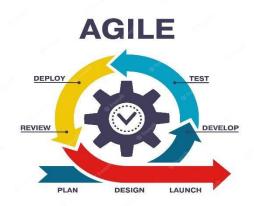


Figure 1-6: Les étapes du processus Agile⁽⁵⁾

#### 5.2.1.1.Scrum

Scrum est une méthodologie de gestion de projet Agile spécifiquement utilisée dans le domaine du développement logiciel. Elle se base sur un processus itératif et collaboratif pour organiser les tâches à réaliser par l'équipe de développement. Le processus de Scrum se déroule en cycles appelés sprints, qui durent généralement entre deux et quatre semaines.

Au début de chaque sprint, l'équipe de développement définit les tâches à réaliser et les ajoute à un tableau appelé tableau de bord Scrum (ou "Scrum board"). Pendant le sprint, l'équipe se réunit quotidiennement pour discuter de l'avancement du projet et des éventuels obstacles rencontrés, lors d'une réunion appelée "Daily scrum".

À la fin du sprint, l'équipe présente les résultats de leur travail, sous forme de fonctionnalités ou d'éléments de travail finalisés, lors d'une revue de sprint. Enfin, l'équipe se réunit pour une rétrospective, qui permet de discuter des aspects positifs et négatifs du sprint et d'identifier des pistes d'amélioration pour le prochain sprint.

La méthode Scrum permet d'optimiser le processus de développement en favorisant la communication et la collaboration entre les membres de l'équipe, tout en permettant une plus grande flexibilité et une réactivité accrue aux changements et aux feedbacks des utilisateurs.

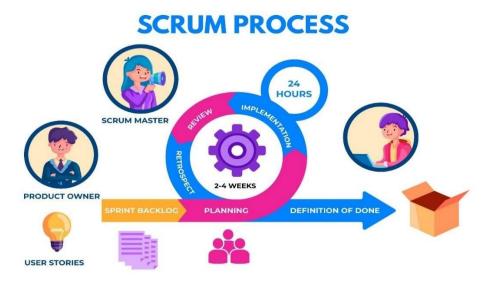


Figure 1-7: Les étapes du processus Scrum (6)

#### 5.2.1.2. La méthode XP

La méthode XP (eXtreme Programming) est une approche de développement logiciel agile qui met l'accent sur la qualité, la flexibilité et la collaboration au sein de l'équipe de développement. XP a été créé dans les années 1990 par Kent Beck et est basé sur une série de principes et de pratiques. Voici certains aspects clés de la méthode XP :

- Communication constante
- Tests automatisés
- Programmation en binôme
- Intégration continue
- Planification itérative
- Client impliqué

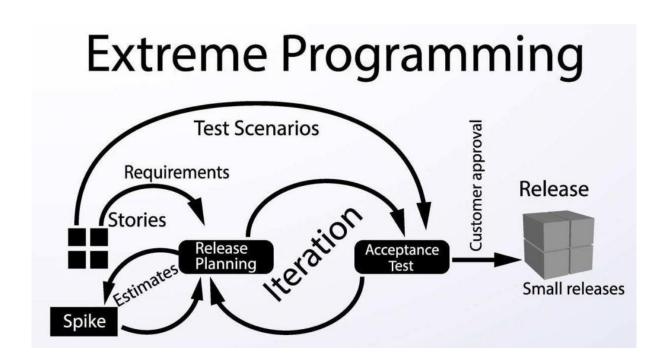


Figure 1-8: principe et fonctionnement de la méthode XP

#### 5.2.2. Processus unifié

Le Processus Unifié, voir Figure 1.9, est une méthodologie de gestion de projet de développement logiciel qui se concentre sur la production d'un produit de qualité dans un délai et un budget défini. Elle se base sur un processus itératif et incrémental qui s'adapte aux besoins et aux changements du projet tout au long du cycle de vie. Le Processus Unifié est divisé en quatre phases principales : l'Inception, l'Élaboration, la Construction et la Transition. Chaque phase est subdivisée en itérations ou sprints, qui sont des périodes courtes (2 à 6 semaines) de travail intensif sur un ensemble de fonctionnalités à développer.

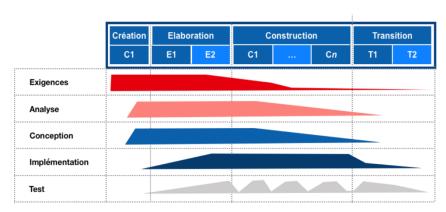


Figure 1-9: Les étapes du processus unifié (7)

#### **5.2.2.1.Two Track Unufied Process (2TUP)**

Le processus unifié Two Track Unified Process (2TUP) se base sur le principe que toute évolution imposée à un logiciel peut être décomposée et traitée en parallèle, en suivant deux axes distincts : l'axe fonctionnel et l'axe technique. La réalisation du logiciel consiste ensuite à fusionner les résultats de ces deux branches du processus.

#### 5.2.2.2. Rational Unified Process(RUP)

Le Rational Unified Process (RUP) est un processus de développement de logiciel qui a été créé par Rational Software Corporation, qui a ensuite été acquis par IBM. Le RUP est un processus itératif et incrémental qui se concentre sur l'implémentation de fonctionnalités du logiciel à travers des cycles de développement courts et successifs. Il est basé sur une approche de développement centrée sur l'architecture et sur des principes de modélisation UML (Unified Modeling Language). Le RUP propose des phases, des activités et des rôles définis pour chaque membre de l'équipe de développement, et il met l'accent sur la communication et la collaboration entre les membres de l'équipe.

#### 5.3. Etude comparative et évaluation

Tableau 1-6: Tableau comparatif des méthodologies de développement

Méthodologies	Avantages	Inconvénients		
Agile	Agile présente de nombreuses opportunités pour engager les parties prenantes et les équipes. En impliquant le client à chaque étape du projet, cela favorise une forte collaboration entre le client et l'équipe de projet, permettant à l'équipe de mieux comprendre la vision du client.	<ul> <li>L'intégration des tests et des opérations n'a pas été aussi réussie que prévu dans ce mélange.</li> <li>La technologie n'a pas été suffisamment mise en avant, ce qui peut rendre difficile la vente du concept aux cadres supérieurs qui ne comprennent pas le rôle que la culture joue dans le développement de logiciels.</li> </ul>		

Scrum	<ul> <li>Le projet est centré sur le client.</li> <li>Esprit d'équipe favorisé.</li> <li>Communication ouverte et transparente.</li> <li>Priorité donnée à la simplicité, l'efficacité et la qualité.</li> <li>Capacité d'adaptation aux changements.</li> <li>Avancement basé sur des résultats tangibles.</li> </ul>	Le manque de pratique pour atteindre l'état "DONE" dans le contexte logiciel peut être un inconvénient de Scrum.
ХР	-Communication accrue entre les membres de l'équipe. - Adaptabilité aux changements fréquents. - Qualité du code améliorée grâce aux tests automatisés.	-Peut nécessiter une équipe hautement qualifiée et expérimentée. - Peut être moins adapté aux projets avec des exigences strictes et prédictibles.

#### 5.4. Choix de la méthodologie de travail

Notre projet repose sur un processus de développement rigoureux, qui commence par l'analyse des besoins fonctionnels du système et se poursuit jusqu'à la conception et la programmation finale. Pour cela, nous avons besoin d'un cycle de développement qui sépare les aspects techniques des aspects fonctionnels, en commençant par une étude préliminaire. Après une étude comparative approfondie, nous avons choisi la méthode 2TUP, car elle propose une approche innovante qui répond à nos besoins spécifiques tout en respectant le cadre de notre projet.

#### ➤ Mise en pratique du processus 2TUP

2TUP est un processus unifié (c'est-à-dire construit sur UML, itératif, centré sur l'architecture conduit par les cas d'utilisation). Les étapes de ce processus sont clairement représentées dans la Figure 1.10.

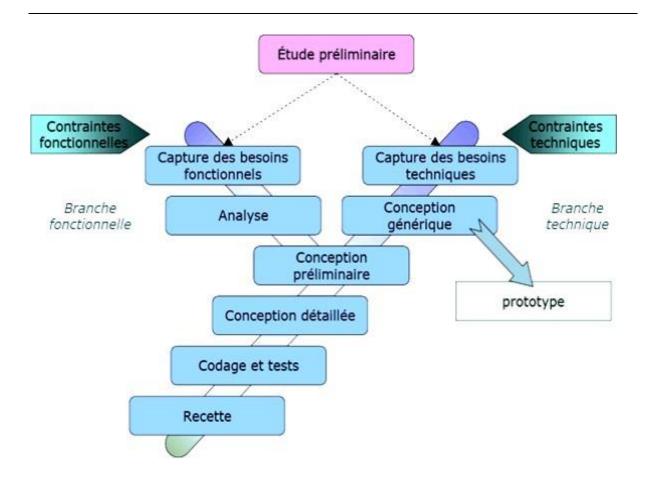


Figure 1-10: Les étapes du processus 2TUP

Le principe fondateur du 2TUP est que toute évolution imposée à un logiciel peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique.

La réalisation du logiciel consiste à fusionner les résultats de ces deux branches du processus. Selon le processus 2TUP que nous suivons, nous passons obligatoirement par les phases suivantes à la suite de notre travail :

- ❖ L'étude préliminaire : qui contient une description du service et les cas d'utilisation principaux, c'est une première version de la spécification générale.
- ❖ La capture des besoins fonctionnels : qui définit le quoi faire à travers une spécification générale qui décrit le service à développer d'un point de vue fonctionnel et une spécification détaillée qui précise les traitements qui concernent chaque scénario des cas d'utilisation présent en spécification générale tout en respectant les contraintes fonctionnelles et non fonctionnelles.
- ❖ L'analyse : où on effectue simultanément l'étude des données et l'étude des traitements à effectuer.

- ❖ La capture des besoins techniques : qui permettent de satisfaire les contraintes techniques présentes dans le cahier des charges et donc répondre aux attentes de client.
- **La conception générique :** qui définit le comment faire.
- ❖ La conception détaillée : qui précise l'implémentation technique de l'application. Elle consiste en la fusion de la spécification détaillée et de la conception générique, pour déterminer comment faire le quoi faire dans le détail.

On y trouve le schéma de base de données, les diagrammes de classes et les diagrammes de séquence supplémentaires qui détaillent les interactions entre les composants du logiciel.

❖ Le Codage et les Tests : qui décrivent comment est l'application est réalisée et installée sur un environnement d'exécution et comment la stratégie de validation est effectuée.

#### 5.5.Diagramme de Gantt

Le diagramme de Gantt est un outil de gestion de projet qui permet de visualiser les différentes tâches à réaliser ainsi que leur durée et leur ordonnancement dans le temps. Le diagramme de Gantt est souvent utilisé dans le cadre de la planification de projets et permet de suivre l'avancement du projet en temps réel. Les tâches sont représentées par des barres horizontales, dont la longueur correspond à la durée de la tâche, et leur positionnement sur l'axe temporel indique leur ordre d'exécution. Les dépendances entre les tâches peuvent également être représentées pour faciliter la gestion des priorités.

Le diagramme de Gantt est un outil visuel qui permet de communiquer efficacement sur le planning du projet avec toutes les parties prenantes.

La Figure 1.11 du diagramme de Gantt suivante représente les taches principales à réaliser dans notre projet :

#### **DIAGRAMME DU GRANTT**

Application mobile e-commerce marque blanche chez Dot It



Figure 1-11 : Diagramme de Gantt et planning prévisionnelle

#### **Conclusion:**

Au cours de ce premier chapitre, nous avons effectué une analyse préliminaire pour entamer notre projet avec une vision claire de la solution à développer. Nous avons examiné différentes approches de développement et nous avons finalement opté pour la méthodologie 2TUP pour la réalisation de notre projet. Dans le prochain chapitre, nous aborderons la phase de capture des besoins.

CHAPITRE II : Analyse et spécification des	Besoins

#### Introduction

Le choix de la méthodologie 2TUP a été effectué en tenant compte de ses avantages en termes de développement de logiciels de qualité. Ce chapitre est donc consacré à la phase de capture des besoins, qui constitue une étape cruciale dans le processus de développement. Nous nous concentrerons sur l'identification des fonctionnalités de notre futur système en recensant les besoins fonctionnels, les exigences traduites par les besoins non fonctionnels et les besoins techniques. La collecte de ces informations est essentielle pour garantir une description sans ambiguïté du portail à développer et pour s'assurer que les attentes des utilisateurs finaux seront pleinement satisfaites.

#### 1. Besoins fonctionnels

Dans tout système interactif, il est essentiel de prendre en compte les différents acteurs impliqués et de définir leurs rôles et responsabilités pour assurer une interaction efficace avec le système. Pour notre plateforme de e-commerce en marque blanche, nous avons identifié les acteurs suivants :

- ➤ Internaute : L'internaute est un acteur qui représente un utilisateur non enregistré sur le système. Il a accès à la page d'accueil et peut consulter les différents produits proposés par le site, mais il ne peut pas acheter de produits ou accéder à certaines fonctionnalités réservées aux utilisateurs enregistrés.
- Visiteur : Il possède toutes les fonctions de l'internaute mais il peut créer un compte client.
- ➤ Client : c'est un internaute qui peut accéder au catalogue de produits, ajouter des produits à son panier, passer des commandes, effectuer des paiements, suivre l'état de ses commandes, laisser des commentaires et des avis sur les produits...
- Administrateur: C'est un abonné au site qui peut accéder à l'espace d'administration du site. Il peut gérer les produits, les catégories de produits, les commandes, les clients et les boutiques. L'administrateur doit également s'assurer que le site est facilement accessible et convivial pour les clients et que les transactions se déroulent sans heurts.
- ➤ Prestataire : C'est un internaute qui possède une boutique dans le site. Il peut proposer ses produits ou services à la vente sur la plateforme. Il est responsable de la gestion de ses produits (ajout, modification, suppression), de la gestion des commandes passées sur sa boutique, de la gestion des stocks et de la facturation de ses ventes.

Dans la suite, nous allons détailler les besoins fonctionnels de notre plateforme de e-commerce en marque blanche, en mettant en relation les différents acteurs impliqués. Il est important de souligner que ces besoins doivent répondre aux exigences du système à venir.

#### Concernant « l'internaute », il a le droit de :

 Consulter le site : L'internaute peut consulter les différents produits proposés par la plateforme, mais il ne peut pas acheter de produits ou accéder à certaines fonctionnalités réservées aux utilisateurs enregistrés.

#### Le « client » détient le droit de :

- Créer un compte : Le client peut créer un compte sur la plateforme pour accéder à toutes les fonctionnalités, y compris la possibilité de commander des produits et de gérer ses informations personnelles.
- Consulter les produits : Le client peut consulter les différents produits proposés par la plateforme et effectuer des recherches pour trouver les produits qui l'intéressent.
- Ajouter des produits au panier : Le client peut ajouter des produits à son panier pour préparer une commande future.
- Passé une commande : Le client peut passer une commande en utilisant le panier, en sélectionnant les produits qu'il souhaite acheter et en effectuant le paiement.
- **Gérer les commandes :** Le client peut suivre l'état de ses commandes et avoir accès à l'historique de ses achats.
- Laisser des commentaires et des avis : Le client peut laisser des commentaires et des avis sur les produits pour aider les autres clients à faire leur choix.

#### « L'administrateur » détient le droit de :

- **Gérer les produits :** L'administrateur peut ajouter, modifier ou supprimer des produits et des catégories de produits, et mettre à jour les informations relatives aux produits.
- **Gérer les commandes :** L'administrateur peut consulter et gérer les commandes passées sur la plateforme, en traitant les paiements, engérant les retours et en résolvant les problèmes de livraison.
- **Gérer les clients :** L'administrateur peut gérer les comptes des clients, en supprimant ou en bloquant les comptes qui ne respectent pas les conditions d'utilisation.

• **Gérer les prestataires :** L'administrateur peut gérer les comptes des prestataires, en vérifiant et en validant les informations relatives à leur boutique.

Le « prestataire » détient le droit de :

- **Créer une boutique :** Le prestataire peut créer une boutique sur la plateforme pour proposer ses produits ou services à la vente.
- **Gérer sa boutique :** Le prestataire peut gérer les produits de sa boutique, en ajoutant, en modifiant ou en supprimant des produits et en gérant les stocks.
- **Gérer les commandes :** Le prestataire peut gérer les commandes passées sur sa boutique, en traitant les paiements, en préparant les colis et en les expédiant.
- **Gérer les commentaires et les avis :** Le prestataire peut consulter et répondre aux commentaires et aux avis laissés par les clients sur ses produits ou services.

#### 2. Besoins non fonctionnels

Les besoins non fonctionnels sont des exigences qui ne concernent pas directement les fonctionnalités du système, mais qui ont un impact sur la qualité, la performance et la sécurité de la plateforme. Pour notre site web, on a défini les besoins non fonctionnels suivants :

#### 2.1. Exigence de qualité

- ➤ Ergonomie: L'expérience utilisateur lors de l'achat d'un produit sur le site web doit être fluide et simple. La mise en page du site doit être claire et intuitive afin de faciliter la navigation et l'achat. Un site trop complexe et trop riche en informations peut décourager l'utilisateur et entraver son processus d'achat. Il est donc important de simplifier autant que possible la démarche d'achat et de proposer une interface conviviale.
- Formulaire de commande simple : Il est fréquent que les internautes rencontrent des difficultés lorsqu'ils doivent remplir un bon de commande lors de l'achat d'un produit en ligne. Par conséquent, il est important de concevoir et de présenter le bon de commande de manière claire et attrayante afin de faciliter cette étape pour le client et de minimiser le risque de découragement.

#### 2.2.Exigences de performance

➤ Besoin de sécurité : L'application doit garantir la sécurité et la confidentialité des données des utilisateurs connectés, en utilisant un système d'authentification par login et mot de passe crypté.

Voici les exigences de sécurité supplémentaires :

- Les mots de passe doivent respecter une longueur minimale et contenir des caractères spéciaux et des chiffres, avec une politique de réutilisation bien déterminée.
- Une déconnexion automatique doit être effectuée après un certain temps d'inactivité, avec des durées et des actions spécifiées.
- Les utilisateurs doivent être classés en fonction de leurs privilèges d'accès. Un administrateur doit avoir le droit de tout faire, tandis que les autres utilisateurs ont des droits limités.
- ➤ **Performance :** L'application doit fournir tous les statuts et informations en temps réel et d'une manière optimale.
- Temps de réponse minimal : En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.
- Chargement rapide de l'application : le rafraîchissement des données doit être en temps réel, souple, et rapide.
- L'intégrité: Le système doit être capable de gérer les erreurs de manière efficace, en détectant et en signalant les échecs et les erreurs d'entrées-sorties. Il doit être en mesure de traiter les mauvaises données de manière à garantir l'intégrité des informations dans les tables de la base de données et des interfaces.
- La portabilité: L'application doit être compatible avec tous les navigateurs courants, sans aucune restriction de version ou de plateforme. Il est également crucial qu'elle soit conçue pour s'exécuter sans problème sur les appareils Android et sur toutes les tailles d'écran, afin d'offrir une expérience utilisateur cohérente et optimale.
- La flexibilité: Il est essentiel que notre application soit conçue avec une architecture évolutive et extensible afin de permettre une intégration facile avec d'autres systèmes ou domaines. Cette ouverture vers d'autres domaines serait un atout majeur pour notre application et doit donc être considérée dès la conception de l'architecture du projet.

#### 2.3. Contraintes de conception

Mise à jour des données: Les informations relatives aux produits présentés sur le site proviendront de deux sources complémentaires. La première servira à alimenter la base avec tous les nouveaux produits, tandis que la seconde permettra de mettre à jour les données qui concernent le prix, l'état et la disponibilité des produits du catalogue. Ces

informations seront saisies manuellement à travers un espace administrateur (backoffice) dédié à la gestion des produits, des catégories et des utilisateurs de l'application.

Les formulaires du site : Les données personnelles des clients saisies sur le site web et enregistrées dans la base de données seront stockées en toute sécurité et leur confidentialité sera garantie. Les informations relatives aux commandes, telles que les coordonnées des clients, les caractéristiques des produits commandés, les dates de livraison prévues et les factures, seront mémorisées et traitées par le service clientèle. Les clients auront accès à leur historique de commandes pour suivre l'évolution de leurs achats.

#### 3. Besoins techniques

Nous nous concentrons sur la phase de capture des besoins techniques, qui vise à capitaliser le savoir-faire technique. Pour expliquer nos choix technologiques, nous avons mené une étude comparative des différentes technologies qui pourraient être utilisées dans notre projet.

#### 3.1. Choix technologique

Dans la section suivante, nous présenterons les technologies utilisées pour Front-End et Back-End sélectionnés pour le développement de notre application mobile.

#### 3.1.1. Front-End

Parmi les différents langages et frameworks disponibles pour le développement mobile, nous avons choisi Flutter pour notre projet.

Flutter est un framework open-source développé par Google pour la création d'applications mobiles, web et de bureau. Il est largement utilisé pour la construction d'interfaces utilisateur interactives et dynamiques. Ce framework se distingue par sa facilité d'utilisation, ses performances élevées et sa capacité à créer des composants réutilisables. La Figure 2.1 illustre bien l'architecture d'une application utilisant Flutter. Grâce à ces avantages, Flutter est devenu l'un des choix les plus populaires pour le développement d'applications modernes.

Le tableau ci-dessous présente les avantages et les inconvénients de l'utilisation de Flutter :

Tableau2-	l : Avantages	et inconvénients de	l'utilisation de Flutter
-----------	---------------	---------------------	--------------------------

Avantages	Inconvénients
- Haute performance grâce à son moteur de	- Long à maîtriser
rendu natif	
- Interface utilisateur flexible et réactive	- Nécessite une connaissance approfondie du
	langage Dart
- Facile à apprendre pour les développeurs	- Taille des applications plus grande par
avec une expérience en développement	rapport à d'autres frameworks
mobile	
- Framework open source soutenu par	- Communauté et écosystème encore en
Google	croissance, ce qui peut limiter les ressources
	disponibles



La Figure 2.1 illustre l'architecture typique d'une application utilisant Flutter

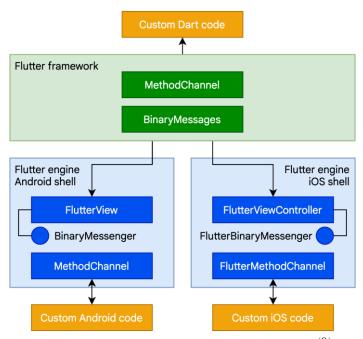


Figure 2-1 : architecture typique d'une application  $\mathit{Flutter}^{(9)}$ 

#### 3.1.2. Back-End

Le Back-End, qui est invisible pour le client, représente une grande partie du développement d'un projet web. On peut décomposer le Back-End en deux parties essentielles :

- Serveur
- Base de données

#### 3.1.2.1.Serveur

Parmi les différents langages de programmation côté serveur, nous avons choisi Node JS avec son Framework Express JS pour notre projet. Node.js est une plateforme logicielle en JavaScript orientée vers les applications réseau hautement concurrentes qui doivent pouvoir monter en charge. Express.js est un Framework de Node JS qui permet de créer des applications web rapidement et facilement. Bien que d'autres langages tels que Python, Java, C# et PHP soient également largement utilisés pour les applications web côté serveur, notre choix s'est porté sur Node.js pour sa haute scalabilité et sa rapidité d'exécution, ainsi que sur Express.js pour sa simplicité et sa facilité d'utilisation pour la création d'applications web.

### 3.1.2.2.Base de données

Une base de données permet de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une activité. Lorsqu'il s'agit de choisir une base de données, la décision la plus importante est de choisir une structure de données relationnelle (SQL) ou non relationnelle (NoSQL). Le tableau ci-dessous dresse une comparaison entre ces deux structures :

Tableau 2-2 : Bases de données Relationnelles vs NoSQL

	BD Relationnelle	BD NoSQL
Туре	Relationnelle	Non relationnelle
	Structurées, stockéesdans	Non structurées
Données	des tables	stockées dans des
		fichiers JSON
Schéma	Prédéfini, rigide	Non prédéfini, flexible
	Verticale (augmenter la	Horizontale (partage oul'ajout
Évolutivité	charge sur un seul serveur)	de serveurs)
Transaction	ACID	Théorème CAP

Selon cette comparaison, les systèmes de base de données NoSQL ont l'avantage de fournir une grande évolutivité horizontale et une capacité de stockage flexible, sans sacrifier les performances. Ils permettent de stocker des données non structurées, semi-structurées ou fortement structurées, ce qui les rend plus adaptés aux besoins actuels des entreprises en matière de données. De plus, les bases de données NoSQL sont conçues pour fonctionner dans des environnements distribués et fournir des performances élevées avec une faible latence.

Donc nous avons choisi MongoDB comme base de données, car il offre une grande flexibilité pour stocker différents types de données et peut être facilement mis à l'échelle horizontalement pour gérer des volumes de données élevés. De plus, MongoDB fournit des fonctionnalités avancées telles que le support de l'indexation géospatiale et la réplication pour une haute disponibilité des données. Enfin, MongoDB permet une évolution continue de la structure des données, ce qui peut être un avantage pour les projets qui évoluent rapidement.

### 3.1.3. Technologies retenues

Notre application mobile sera développée en utilisant la pile technologique gratuite et open source, composée de Flutter, Node.js, Express.js et MongoDB. Cette pile offre une approche de développement moderne et efficace. La synergie entre ces technologies est illustrée dans la Figure 2.2.



 $Figure\ 2-2: Choix\ technologique\ de\ notre\ application$ 

#### 3.2.Choix architectural

Avant de commencer le développement de notre projet, il est important de choisir un modèle architectural adapté à nos besoins. Parmi les modèles les plus couramment utilisés, nous avons

choisi de nous concentrer sur deux d'entre eux : le modèle Modèle-Vue-Vue Modèle (MVVM) et le modèle Modèle-Vue-Contrôleur (MVC).

#### 3.2.1. Modèle MVVM

Le pattern MVVM sépare l'interface utilisateur (View) de la logique métier (Model) en introduisant une couche intermédiaire appelée ViewModel. Le ViewModel est responsable de la gestion des interactions entre l'interface utilisateur et la logique métier. Il expose également les données et les commandes nécessaires à l'interface utilisateur, ce qui permet à celle-ci de rester indépendante de la logique métier sous-jacente.

Le ViewModel communique avec le Model pour récupérer et enregistrer les données. Il expose ensuite ces données sous forme d'objets, qui sont liés aux contrôles de l'interface utilisateur de manière déclarative. Cela signifie que toute modification de l'état du ViewModel est immédiatement reflétée dans l'interface utilisateur, sans que celle-ci ait besoin de gérer les détails de la mise à jour des données.

Le pattern MVVM est souvent utilisé en conjonction avec des Framework de développement d'applications tels que AngularJS, React, Vue.js, ou encore Xamarin. La Figure 2.3 suivante illustre la structure générique d'une architecture MVVM.

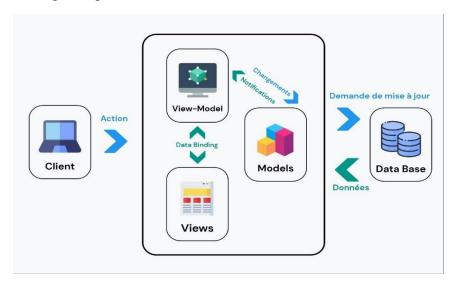


Figure 2-3: Architecture MVVM

## 3.2.2. Modèle MVC

MVC (Modèle-Vue-Contrôleur) est un modèle de conception qui permet de séparer les responsabilités dans une application en trois parties distinctes :

- Le modèle : représente les données et leur traitement.
- La vue : représente l'interface utilisateur, c'est-à-dire ce que l'utilisateur voit à l'écran et avec quoi il interagit.
- Le contrôleur : gère les interactions entre la vue et le modèle, et est responsable de la coordination du flux de l'application.

L'idée derrière MVC est de séparer les différentes responsabilités de manière à rendre l'application plus facile à maintenir et à évoluer. En effet, si l'on change le modèle, cela n'affectera pas la vue ou le contrôleur, et vice versa. Le modèle MVC est souvent utilisé dans le développement de sites web, mais il peut également être utilisé pour d'autres types d'applications. La Figure 2.4 suivante illustre la structure générique d'une architecture MVC.

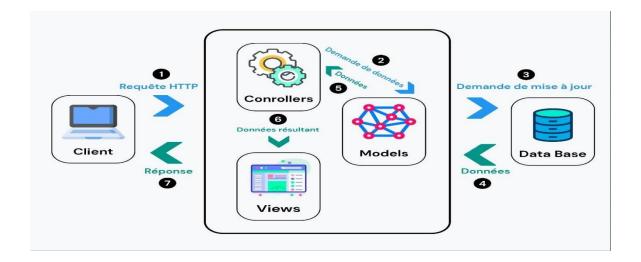


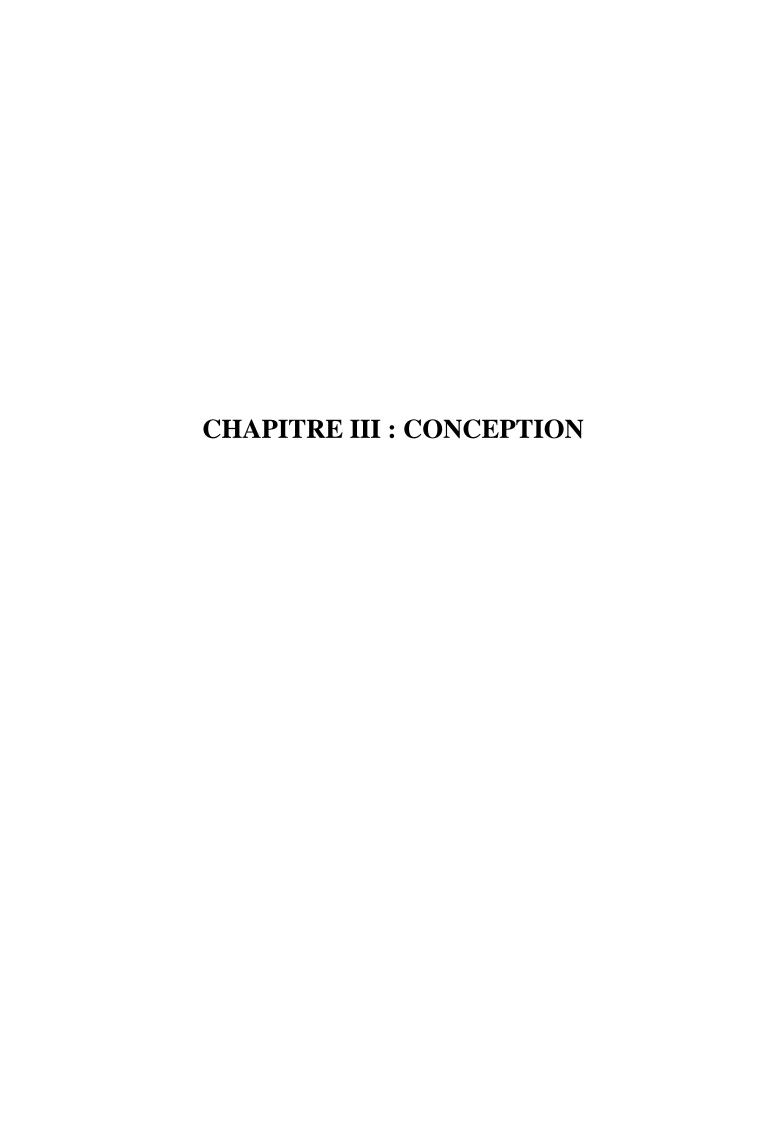
Figure 2-4: Architecture MVC

### 3.2.3. Architecture retenue

D'après l'étude comparative réalisée, nous avons choisi de travailler avec l'architecture MVC pour notre projet. Cette architecture est adaptée à la nature et à la taille de notre application, car elle permet une séparation claire entre la logique de l'application et la partie présentation. Cette séparation facilite l'ajout et la modification du code d'une section sans affecter le reste de l'application. Le patron MVC convient particulièrement bien pour le développement des applications de taille moyennes et petites.

## **Conclusion**

Au cours de ce chapitre, nous avons appliqué le processus 2TUP pour la capture et la gestion des besoins de notre projet. Nous avons classé les différents besoins en trois catégories : fonctionnelles, non fonctionnelles et techniques. Après avoir analysé les exigences et choisi les technologies à utiliser, il est maintenant temps de passer à la conception de notre application dans le prochain chapitre.



## Introduction

Dans ce chapitre, nous entamons la phase de conception de notre application mobile. La première partie sera dédiée à la présentation de la conception générale utilisant le formalisme UML, qui se base sur des diagrammes. Enfin, nous aborderons la création des différents diagrammes UML nécessaires à la conception de notre application

## 1. Conception générique

La phase de conception a pour objectif de décrire de manière précise et détaillée le fonctionnement futur du système en utilisant généralement un langage de modélisation. Cette étape est cruciale pour faciliter la réalisation du système en fournissant une représentation claire et structurée de toutes les fonctionnalités et interactions requises.

### 1.1.Le langage UML

Nous avons opté pour l'utilisation du langage de modélisation unifié «UML» (Unified Modeling Language) afin de faciliter notre tâche de conception. Ce langage standard permet de modéliser un problème de façon précise et est devenu une référence en termes de modélisation objet. Les avantages de l'utilisation d'UML sont multiples :

- Il permet un gain de temps et encourage l'utilisation d'outils spécifiques, ce qui garantit la stabilité de notre application.
- Il permet de cadrer l'analyse et facilite la compréhension de représentations abstraites complexes.
- UML est normalisé et formel, ce qui le rend polyvalent et universellement reconnu.

### 1.2.Les Diagrammes d'UML

UML est composé de treize types de diagrammes, chacun dédié à la représentation de concepts particuliers d'un système logiciel. Ces diagrammes sont regroupés en deux grandes catégories :

### 🖶 Sept diagrammes comportementaux :

• **Diagramme de cas d'utilisation :** Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

- Diagramme de vue d'ensemble des interactions : Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.
- **Diagramme de séquence :** Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.
- **Diagramme de communication :** Il montre la communication entre objets dans le plan au sein d'une interaction.
- **Diagramme de temps :** Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.
- **Diagramme d'activité :** Il montre l'enchaînement des actions et décisions au sein d'une activité.
- **Diagramme d'états :** Il montre les différents états et transitions possibles des objets d'une classe.

### Six diagrammes structurels :

- **Diagramme de classes :** Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.
- **Diagramme d'objets :** Il montre les instances des éléments structurels et leurs liens à l'exécution.
- **Diagramme de packages :** Il montre l'organisation logique du modèle et les relations entre packages.
- **Diagramme de structure composite :** Il montre l'organisation interne d'un élément statique complexe.
- **Diagramme de composants :** Il montre des structures complexes, avec leurs interfaces fournies et requises.
- **Diagramme de déploiement :** Il montre le déploiement physique des 'artefacts' sur les ressources matérielles.

### 2. Conception détaillée

### 2.1. Les vues de Kruchten

Kruchten offre des perspectives distinctes et complémentaires pour décrire la méthodologie 2TUP que nous avons choisie. Ainsi, en optant pour l'approche de Kruchten, nous pourrons passer à la phase de conception de notre projet. Le langage de modélisation UML sera utilisé

pour cette étape et ses 13 types de diagrammes seront employés en fonction des besoins et répartis selon les vues de Kruchten, comme présenté dans la Figure 3.1.

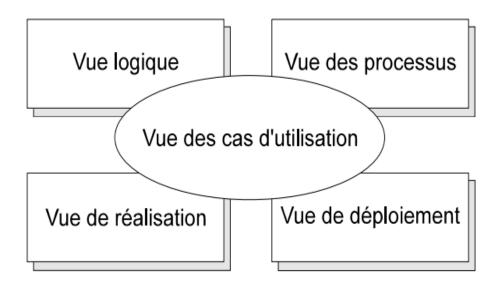


Figure 3-1: Vue 4+1 (Kruchten)

- ➤ La vue des cas d'utilisation : est un guide pour toutes les autres vues. Elle définit les besoins des clients du système ainsi que l'architecture nécessaire pour répondre à ces besoins. Cette vue unifie les autres vues et motive les choix en identifiant les interfaces et en mettant l'accent sur les besoins des utilisateurs. Elle présente les diagrammes de cas d'utilisation et de package.
- ➤ La vue logique : décrit les aspects statiques et dynamiques d'un système en termes de classes, d'objets, de connexions et de communications. Elle se concentre sur l'abstraction et l'encapsulation. Cette vue présente les diagrammes de classe et d'objet.
- La vue de processus : décrit les interactions entre les différents processus (threads, tâches) et exprime la synchronisation et l'allocation des objets. Elle vérifie également le respect des contraintes de fiabilité, d'efficacité et de performances des systèmes multitâches. Cette vue présente les diagrammes de séquence, d'activité, de collaboration, d'état de transition et de temps.
- ➤ La vue des composants : également appelée vue de réalisation, est de bas niveau et permet de visualiser l'organisation des composants (bibliothèque dynamique et statique, code source...) dans l'environnement de développement. Elle permet également de gérer la configuration (auteurs, versions...). Cette vue présente les diagrammes de composants et de structure composite.

La vue de déploiement : est utile pour la maintenance régulière du système pour la tolérance aux fautes et aux pannes. Elle permet également de traiter les contraintes de temps de réponse et de bande passante. Elle présente le diagramme de déploiement.

### 2.2. Modèles d'UML utilisés

Dans la partie précédente, nous avons présenté différents diagrammes UML qui permettent de décrire les objets et les liens qui les unissent. En particulier, pour répondre aux vues de Kruchten, nous avons choisi les diagrammes suivants :

- Le diagramme de cas d'utilisation pour la vue des cas d'utilisation.
- Le diagramme de séquence pour la vue des processus.
- Le diagramme de classe pour la vue logique.
- Le diagramme de composants pour la vue de composants et de déploiement.

Ces quatre diagrammes ont été soigneusement sélectionnés pour répondre aux besoins spécifiques de notre projet et couvrir les aspects essentiels de la modélisation.

### 2.3. Vue des cas d'utilisation : Diagramme de cas d'utilisation

Un cas d'utilisation correspond à une unité distincte d'interaction entre un utilisateur (qu'il soit humain ou machine) et un système. Il représente une tâche significative à réaliser pour l'utilisateur. Dans un diagramme de cas d'utilisation, les utilisateurs sont représentés en tant qu'acteurs, qui interagissent avec les cas d'utilisation. Ce diagramme permet d'identifier les différentes interactions possibles entre le système et les acteurs, ainsi que toutes les fonctionnalités que le système doit fournir.

### 2.3.1. Cas d'utilisation général

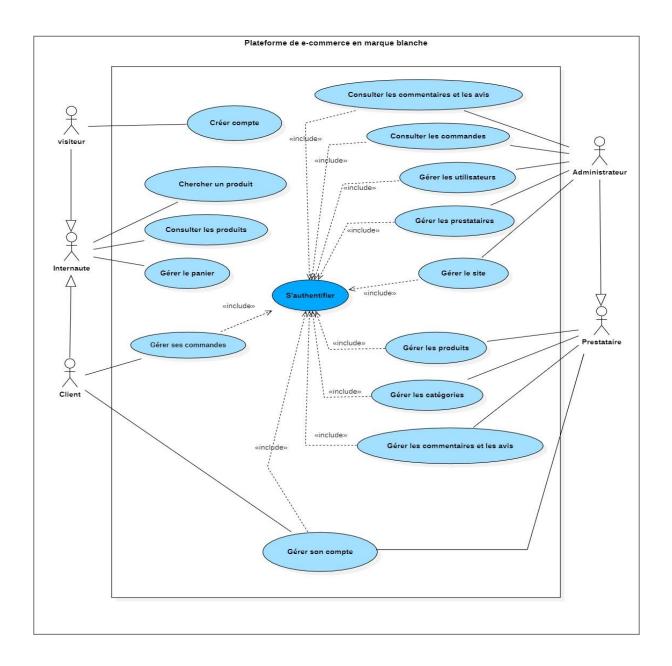


Figure 3-2 : Diagramme de cas d'utilisation général

Le tableau 3.1 représente les différents cas d'utilisation, les acteurs impliqués et une description pour chaque cas.

Tableau 3-1: Tableau des cas d'utilisation

Cas d'utilisation	Acteur(s)	Description
Créer compte	Visiteur	Un visiteur peut créer un compte.

Chercher un produit	Internaute-Client	L'internaute ou le client a la possibilité de chercher un produit
Consulter les produit	Internaute-Client	L'internaute ou le client a la possibilité de visualiser la liste des produits disponibles sur le site.
Gérer panier	Internaute-Client	L'internaute ou le client a la possibilité d'enregistrer les produits qui l'intéressent dans un panier virtuel
Gérer ses commandes	Client	Le client finalise son achat en ligneen validant sa commande.
Gérer son compte	Client-Prestataire- Administrateur	Les acteurs ont la possibilité de gérer leurs comptes et de modifierdes informations telles que l'e-mail,le mot de passe, etc.
Gérer les commandes	Prestataire	Le prestataire peut gérer les commandes de ses clients : confirmer commande, remise
Gérer les commentaires et les avis	Prestataire	Le prestataire peut voir et modifier tous les commentaires et les avis deses clients
Gérer les catégories	Prestataire- Administrateur	Un prestataire ou un administrateur dispose du droit degérer les catégories.
Gérer les produit	Prestataire- Administrateur	Un prestataire ou unadministrateur a le droit de gérer les produits.
Gérer le site	Administrateur	L'administrateur peut gérer le site dans la partie « BackOffice »

		L'administrateur peut ajouter,
Gérer les prestataire	Administrateur	modifier et supprimer les prestataires
Gérer les utilisateurs	Administrateur	L'administrateur a la possibilité de gérer les comptes des utilisateurs.
Consulter les commandes	Administrateur	L'administrateur peut consulter et manipuler les commandes effectuées.
Consulter les commentaires et les avis	Administrateur	L'administrateur peut consulter et manipuler les commentaires et les avis effectuées.

## 2.3.2. Description détaillée des cas d'utilisation

Dans cette section, nous présenterons une description détaillée de quelques cas d'utilisation importants.

## 2.3.2.1. cas d'utilisation Créer compte et S'authentifier

La Figure 3.3 illustre les cas d'utilisation « Créer compte » et « S'authentifier », montrant les interactions entre les utilisateurs et le système pour créer un nouveau compte et se connecter à l'application.

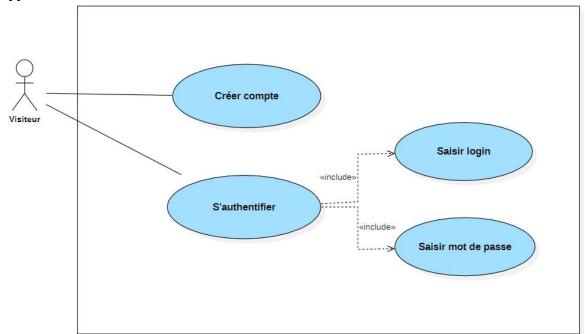


Figure 3-3 : Diagramme de cas d'utilisation Créer compte et S'authentifier

Description textuelle de cas d'utilisation Créer compte :

Tableau 3-2 : Description textuelle de cas d'utilisation Créer compte

Sommaire d'identifiant	
Acteur(s)	Visiteur
Objectif	Le visiteur est tenu de créer un compte afin d'effectuer desactions réservées aux utilisateurs enregistrés.
	Description d'enchainements
Pré_conditions	Le visiteur ne dispose pas d'un compte utilisateur.
Post_conditions	Le visiteur est soit inscrit ou non.
Scénario nominal	<ol> <li>Le visiteur accéder au site et demande de créer un compte.</li> <li>Le Système affiche un formulaire d'inscription.</li> <li>Le visiteur remplir le formulaire par les champs nécessaire.</li> <li>Le Système créer un compte client.</li> </ol>
Alternatives	<ul><li>3.1- Le visiteur ne peut être inscrit : un champ vide.</li><li>3.2- Le Système revient sur l'affichage de la page de création de compte.</li></ul>

Description textuelle de cas d'utilisation s'authentifier :

Tableau 3-3 : Description textuelle de cas d'utilisation « S'authentifier »

Sommaire d'identifiant	
Acteur(s)	Internaute
Objectif	L'internaute doit s'authentifier pour effectuer des tâchesseul l'utilisateur peut les faire
Description d'enchainements	

Pré_conditions	Néant.
Post_conditions	L'internaute est soit authentifié ou non.
Scénario nominal	<ol> <li>L'internaute accède à la page d'identification.</li> <li>Le Système valide l'authenticité de l'internaute. 3- L'internaute est authentifié.</li> </ol>
	<ul><li>2.1- L'internaute ne peut être authentifié : un champ invalide.</li><li>2.2- Le Système revient sur l'affichage de la page d'identification.</li></ul>
Alternatives	

## 2.3.2.2. Les Cas d'utilisation consulter produits et gérer panier

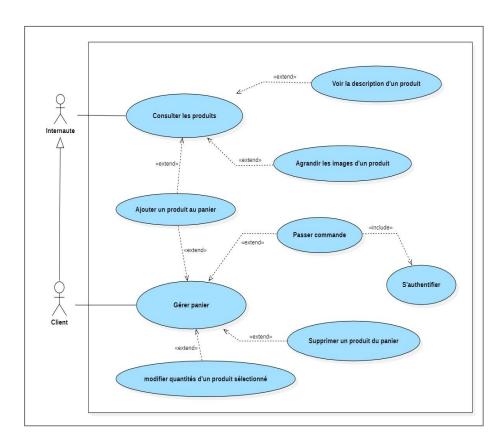


Figure 3-4 : Diagramme des cas d'utilisation Consulter les produits et Gérer panier

Description textuelle des cas d'utilisation Consulter les produits et Gérer Panier :

 $Tableau \ 3-4: Description \ textuelle \ des \ cas \ d'utilisation \ «\ Consulter \ les \ produits \ »\ et \ «G\'erer Panier \ »$ 

Sommaire d'identifiant	
Acteur(s)	Internaute et client
Objectif	L'internaute a la liberté de parcourir les différentes catégories, consulter les produits disponibles, et s'il est intéressé par un produit, il peut l'ajouter à son panier virtuel. Le panier virtuel lui permet de gérer les produits sélectionnés à tout moment.
	Description d'enchainements
Pré_conditions	Les produits existent.
Post_conditions	Un panier peut contenir des produits sélectionnés en vuede passer à l'étape suivante, ou bien être vide si aucun produit n'a été ajouté.
Scénario nominal	<ol> <li>L'internaute ou le client a accès à la page contenant les catégories et les produits disponibles.</li> <li>L'internaute ou le client a la possibilité d'enregistrer les produits qui l'intéressent dans un panier virtuel.</li> <li>Le système affiche à l'internaute ou au client l'état de sonpanier. Chaque produit sélectionné est présenté sur une ligne, avec des informations telles que son nom, son image,son prix unitaire et la quantité. Le système calcule également le total général du panier, affichant ainsi le nombre total de produits sélectionnés et le montant total à payer.</li> <li>L'Internaute ou le client continue ses achats.</li> </ol>
Alternatives	3.1- le panier est vide.  + Le Système affiche un message « Votre panier est vide ». 3.2- L'internaute ou le client modifie les quantités deslignes du panier, ou le supprime.  + Le cas d'utilisation reprend à l'étape 3 du scénario nominal.  3.3-L'internaute fait passer une commande, le total à payerest affiché.  + L'internaute s'identifie en tant que Client + L'internaute doit créer un compte client.

## 2.3.2.3. Le cas d'utilisation gérer ses commandes

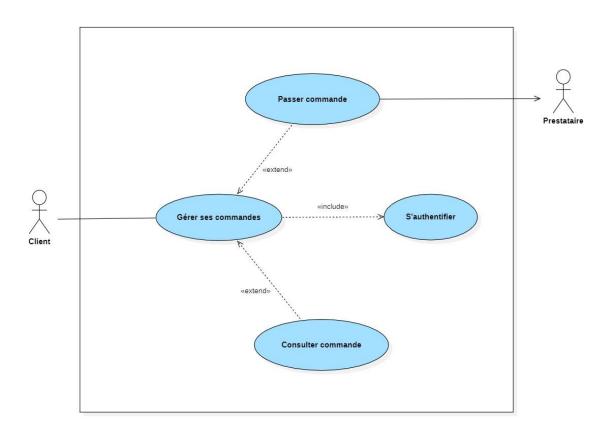


Figure 3-5 : Diagramme de cas d'utilisation Gérer ses Commandes

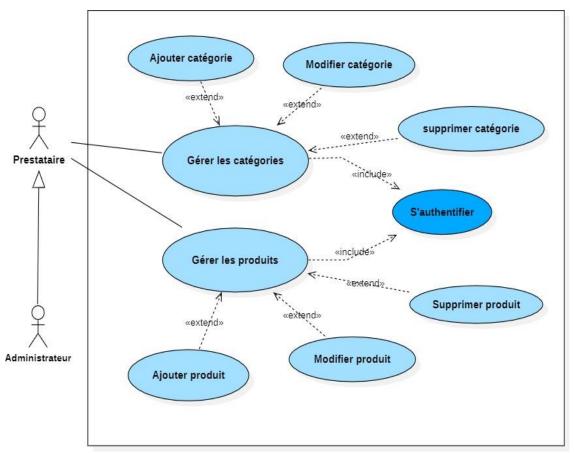
Description textuelle de cas d'utilisation « Gérer ses commandes » :

Tableau 3-5 : Description textuelle de cas d'utilisation « Gérer ses commandes »

Sommaire d'identifiant		
Acteur(s)	Client	
Objectif	Le client a la possibilité d'effectuer une commande à tout moment.	
Description d'enchainements		
Pré_conditions	Pré_conditions  L'acteur doit s'authentifier.  Panier non vide.	
Post_conditions	Commande effectué et enregistrée	

2- Le système affiche une page contenant les produitsprésents dans le panier du client.  Scénario nominal  3- Le client confirme sa commande.		1- Le client prend la décision de passer une commande.
		2- Le système affiche une page contenant les produitsprésents
Scénario nominal 3- Le client confirme sa commande.		dans le panier du client.
	Scénario nominal	3- Le client confirme sa commande.
4- Le système confirme et enregistre la commande, et offre		4- Le système confirme et enregistre la commande, et offre
également la possibilité au client d'imprimer le reçud'achat.		également la possibilité au client d'imprimer le reçud'achat.
3.1- Le client annule sa commande, ce qui entraîne leretour à		3.1- Le client annule sa commande, ce qui entraîne leretour à
l'affichage du panier, et le cas d'utilisation setermine avec un		•
Alternatives échec.	Alternatives	échec.

## 2.3.2.4. Les cas d'utilisation gérer les catégories et gérer les produits



 $Figure \ 3-6: Diagramme \ des \ cas \ d'utilisation \ « \ Gérer \ les \ catégories \ » \ et \ « \ Gérer \ les produits \ »$ 

## Description textuelle de cas d'utilisation « Gérer les catégories » :

Tableau 3-6 : Description textuelle de cas d'utilisation « Gérer les catégories »

Sommaire d'identifiant		
Acteur(s)	Prestataire et administrateur	
Objectif	L'acteur a le droit de gérer les catégories tel qu'ajouter une catégories, supprimer une catégories	
Description d'enchainements		
Pré_conditions	L'acteur doit s'authentifier.	
Post_conditions	Base de données mise à jour.	
Scénario nominal	<ol> <li>Le système affiche les catégories qui existe dans le site.2-L'acteur choisit de modifier quelque catégories.</li> <li>Le système affiche le formulaire pour modifier une catégories.</li> <li>L'acteur remplit le formulaire selon son besoin etconfirme la modification.</li> <li>Le système valide la modification.</li> <li>L'acteur choisit de supprimer quelque catégorie.</li> <li>Le système valide la suppression.</li> <li>L'acteur choisit d'ajouter des nouvelles catégories, il saisit le nom de catégorie, le catégorie parente (s'il y a une)et valide l'ajout.</li> <li>Le système valide l'ajout.</li> </ol>	
Alternatives	3.1- le client annule sa commande, le système revient sur l'affichage du panier et le cas d'utilisation se termine avecéchec.	

Description textuelle de cas d'utilisation « Gérer les produits » :

 $Table au\ 3-7: Description\ textuelle\ de\ cas\ d'utilisation\ «\ G\'erer\ les\ produits\ »$ 

Sommaire d'identifiant		
Acteur(s)	Prestataire et administrateur	
Objectif	L'acteur a le droit de gérer les produits tel qu'ajouter un produit, supprimer un produit	
Description d'enchainements		
Pré_conditions	L'acteur doit s'authentifier.	
Post_conditions	Base de données mise à jour.	
Scénario nominal	<ol> <li>Le système affiche les produits qui existe dans le site.2- L'acteur choisit de modifier quelque produits.</li> <li>Le système affiche le formulaire pour modifier unproduit.</li> <li>L'acteur remplit le formulaire selon son besoin etconfirme la modification.</li> <li>Le système valide la modification.</li> <li>L'acteur choisit de supprimer quelque produit.</li> <li>Le système valide la suppression.</li> <li>L'acteur choisit d'ajouter des nouvelles produits, il saisitle nom de produit, leur catégorie et valide l'ajout.</li> <li>Le système valide l'ajout.</li> </ol>	

2.1 - l'acteur modifie le produit, il va remplir un formulaire(Nom produit, catégorie de produit...):

+ s'il laisse un champ vide alors le système affiche un message «

Champ obligatoire est vide! » et réafficher lapage.

+ S'il saisit des informations invalide le système affiche un message « Champ invalide! » et réafficher la page.

8.1 - l'acteur ajout un produit, il va remplir un formulaire(Nom produit, catégorie de produit...):

+ S'il laisse un champ vide alors le système affiche un message «

Champ obligatoire est vide! » et réafficher lapage.

+ S'il saisit des informations invalide le système affiche un message « Champ invalide! » et réafficher la page.

### 2.3.2.5.Le cas d'utilisation Gérer utilisateurs

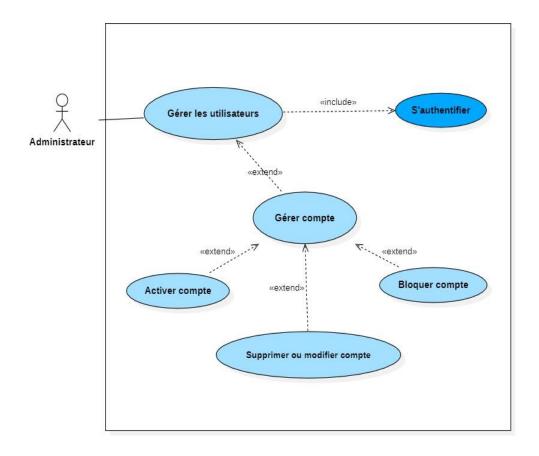


Figure 3-7 : Diagramme de cas d'utilisation « Gérer les Utilisateurs »

## Description textuelle de cas d'utilisation « Gérer les Utilisateurs » :

Tableau 3-8 : Description textuelle de cas d'utilisation « Gérer les Utilisateurs »

Sommaire d'identifiant		
Acteur(s)	Administrateur	
Objectif	L'administrateur peut de gérer les utilisateurs	
Description d'enchainements		
Pré_conditions	L'administrateur doit s'authentifier	
Post_conditions	Base de données mise à jour.	
Scénario nominal	1- L'administrateur demande la liste des utilisateurs.2- Le système affiche la liste de tous les utilisateurs. 3 - L'administrateur peut choisir une action entre: + Ajouter un utilisateur 3.1.1 - Le système affiche un formulaire. 3.1.2 - L'administrateur remplir le formulaire par leschamps nécessaire et valide l'ajout. 3.1.3 - Le système valide l'ajout. + Gérer un Compte 3.2.1 - Le système affiche un formulaire. 3.2.2 - L'administrateur choisit de modifier quelques informations sur l'utilisateur, supprimer le compte, désactiver le compte ou le réactiver. 3.2.3 - Le système valide la modification.	
Alternatives	3.1 - l'administrateur ajoute un utilisateur : + champ invalide alors le système affiche un message « Champ Invalide ! »  3.2 - l'administrateur modifie un utilisateur : + si champ invalide alors le système affiche un message « Champ Invalide ! »	

### 2.4. Vue des processus : diagramme de séquence

Le diagramme de séquence "système" est une représentation visuelle des interactions entre les acteurs et le système, organisées chronologiquement, conformément à la notation UML. Ce diagramme est utilisé pour décrire les aspects dynamiques d'un système, que ce soit dans le cadre d'une opération, d'un système global, d'un sous-système ou d'un cas d'utilisation spécifique (scénario). Il permet de visualiser les échanges et les séquences d'actions entre les différents acteurs et le système, en mettant l'accent sur la dimension temporelle. Dans la suite, nous présenterons quelques scénarios à l'aide des diagrammes de séquences système.

### 2.4.1. Diagramme de séquence s'authentifier

L'internaute initie une demande d'authentification auprès du système, qui lui renvoie le formulaire à remplir. L'internaute saisit alors ses identifiants. Le système procède à la vérification de ces identifiants. Si les identifiants sont valides, le système affiche la page d'accueil. En revanche, si les identifiants sont invalides, le système envoie un message d'erreur à l'internaute.

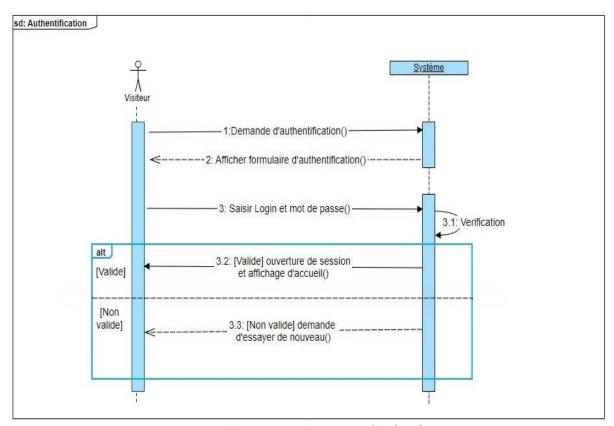


Figure 3-8 : Diagramme de séquence « S'authentifier »

### 2.4.2. Diagramme de séquence Gérer panier

Lorsque l'internaute visite la page du panier, les produits qu'il a sélectionnés sont affichés. Si aucun produit n'a été trouvé, un message indiquant que le panier est vide est affiché à l'internaute. En revanche, s'il y a des produits dans le panier, l'internaute peut visualiser la liste des produits et modifier la quantité d'un produit, supprimer un produit ou même vider complètement son panier. Toute modification ou mise à jour effectuée sur un produit du panier est immédiatement affichée à l'internaute.

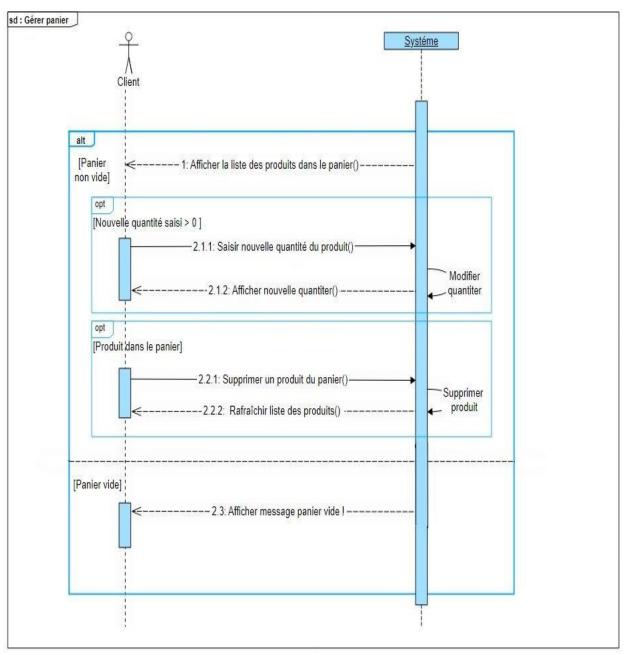


Figure 3-9 : Diagramme de séquence « Gérer panier »

### 2.4.3. Diagramme de séquence Gérer les produits

Après l'authentification, le prestataire consulte la page des produits dans la partie backoffice. Il a la possibilité d'ajouter un nouveau produit. Dans ce cas, le système affiche un formulaire où le prestataire doit remplir les informations du nouveau produit. Une fois le formulaire validé, le produit est ajouté à la base de données. Le prestataire peut également choisir de modifier un produit existant en sélectionnant le produit et en saisissant les informations à modifier. De plus, il peut choisir de supprimer un produit en le sélectionnant et en validant la suppression. Une fois la validation effectuée, les changements apportés aux produits sont immédiatement visibles.

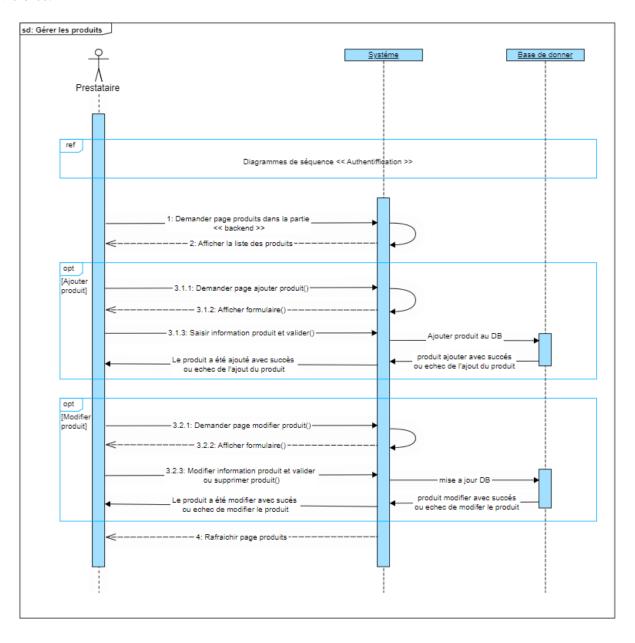


Figure 3-10 : Diagramme de séquence « Gérer les produits »

### 2.5. Vue logique : Diagramme de classes

Un diagramme de classe dans le langage UML est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système de classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes.

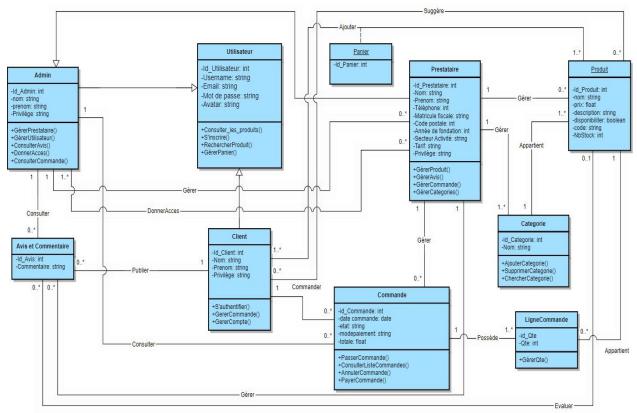


Figure 3-11 : Diagramme de classe

### 2.6. Vue de Composants : Diagramme de Composants

Dans le langage UML, un diagramme de composants est utilisé pour représenter la structure et l'organisation des composants d'un système logiciel. Il permet de visualiser comment les différents composants sont câblés ensemble pour former des composants plus importants ou des systèmes complets.

Dans notre projet, nous avons utilisé le diagramme de composants pour décrire la structure de notre système logiciel. La Figure présente le diagramme de composants correspondant à notre système, illustrant ainsi la manière dont les différents composants sont câblés ensemble pour former le système dans son ensemble. Ce diagramme permet d'avoir une vue claire de l'organisation des composants et de leurs interactions, contribuant ainsi à la compréhension globale de l'architecture du système.

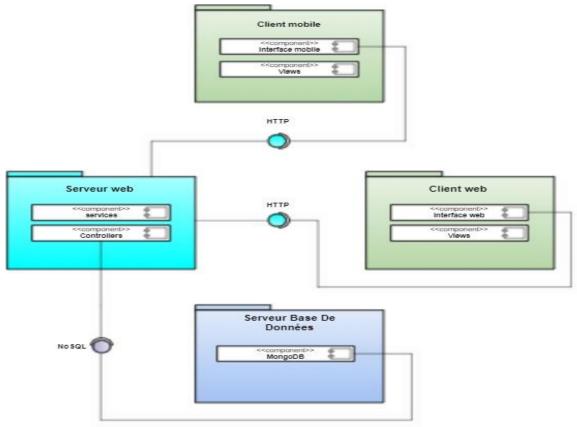


Figure 3-12 : Diagramme de composant

### **Conclusion**

Durant ce chapitre, nous avons abordé à la fois la conception détaillée de notre application mobile ainsi que la modélisation de sa base de données. Ces étapes cruciales marquent la fin d'une phase importante de notre projet : la conception.

Maintenant, nous sommes prêts à passer à la phase de réalisation de l'application. Dans le prochain chapitre, nous explorerons en détail cette étape, en mettant l'accent sur les aspects pratiques de la mise en œuvre de notre solution. Nous aborderons les choix technologiques, les outils utilisés, ainsi que les différentes étapes du développement de l'application.

**CHAPITRE IV : Réalisation** 

## Introduction:

Dans ce chapitre, nous détaillerons la phase de réalisation de notre application mobile. Cette section couvre les différentes étapes du développement, depuis la mise en place de l'environnement de développement jusqu'à l'implémentation des fonctionnalités principales. Nous aborderons les choix technologiques, les outils utilisés, ainsi que les défis rencontrés et les solutions apportées. Chaque sous-section fournira un aperçu des tâches spécifiques réalisées et des résultats obtenus, illustrant ainsi le processus de transformation de la conception en un produit final fonctionnel.

## 1. Environnement logiciel

Dans ce qui suit, nous dresserons la liste des logiciels et des outils qui ont été utilisés pour le développement et l'implémentation de notre application mobile :

### 1.1. Environnement et technologies de développement

Dans ce qui suit, nous dresserons la liste des logiciels et des outils qui ont été utilisés pour le développement et l'implémentation de notre application mobile :

• Visual Studio Code : Nous avons l'utilisé aussi comme environnement de développement.



• **Flutter:** est un Framework open source développé par Google, conçu pour faciliter la création d'applications multiplateformes. Il permet aux développeurs de construire des applications mobiles, web et de bureau à partir d'un code source unique, en utilisant le langage de programmation Dart. Et nous avons l'utilisé pour la développement de notre application mobile coté Frontend.



• Android Studio: Bien que VS Code soit l'IDE principal pour le développement Flutter, Android Studio complète efficacement les fonctionnalités spécifiques à Flutter qui peuvent ne pas être disponibles dans VS Code. Il fournit un ensemble d'outils spécialisés pour le développement, le débogage, le profilage et la gestion des projets Flutter, ce qui en fait un choix courant pour les développeurs travaillant avec Flutter, même s'ils utilisent principalement VS Code.



Android Emulator: est un outil essentiel pour les développeurs d'applications
Android. Il permet d'exécuter des applications Android sur un périphérique virtuel,
reproduisant ainsi les fonctionnalités et les caractéristiques d'un véritable appareil
Android.

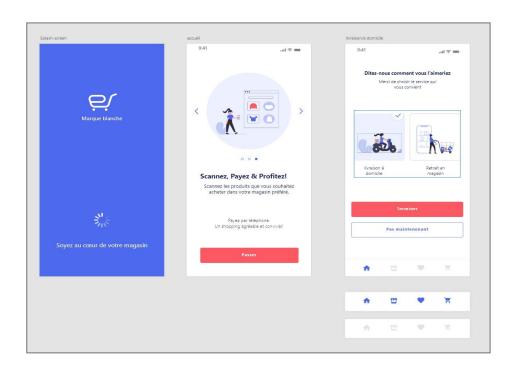


### 1.2. Environnement de design

Adobe XD: est une application de conception et de prototypage d'interfaces utilisateur
 (UI) et d'expériences utilisateur (UX) développée par Adobe. Elle est largement utilisée par les designers pour créer des maquettes interactives, des prototypes et des wireframes d'applications mobiles et web.



Nous avons l'utilisé pour préparer quelques interfaces graphiques de notre application. Les maquettes permettent d'ajuster les besoins et de les concevoir de manière précise. En effet, les interfaces graphiques donnent une vision claire de l'application avant sa réalisation. Les maquettes de notre application sont présentées dans les deux figures Les maquettes de notre application sont présentées dans les deux figures Figure 4.1 et Figure 4.2.



 $Figure\ 4-1: maquette\ I\ de\ l'application$ 

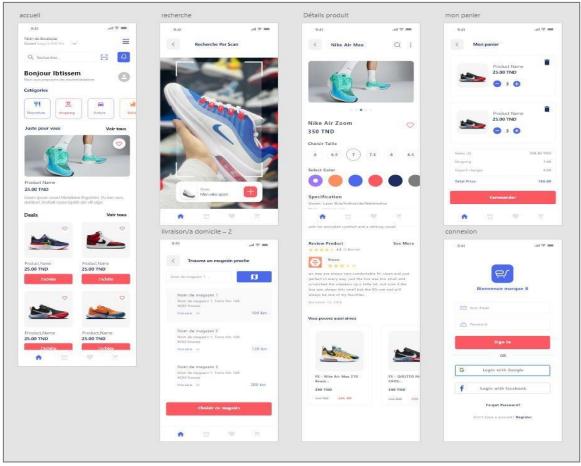


Figure 4-2: Maquette II de l'application

### 2. Présentation de la solution mobile

Dans cette partie de notre rapport, nous allons présenter quelques interfaces de l'application mobile résumant les différentes fonctionnalités offertes.

### 2.1.Interface d'accueil

L'interface d'accueil est la vitrine virtuelle de notre application, offrant aux utilisateurs un point d'entrée convivial et intuitif vers notre vaste gamme de produits. Notre interface d'accueil est conçue pour offrir aux utilisateurs une expérience d'achat fluide et agréable. En combinant une barre de recherche intelligente, une navigation simplifiée par catégories et des mises en avant de produits attractives, nous cherchons à faciliter la découverte, la recherche et l'achat des produits souhaités, tout en offrant une expérience utilisateur immersive et satisfaisante.

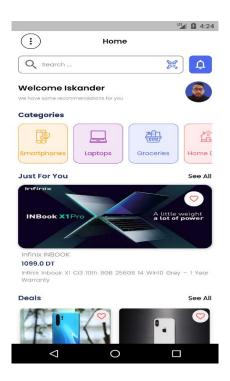


Figure 4-3: Interface d'accueil application mobile

### 2.2.Interface menu

Cette interface expose les différents volets auxquels l'utilisateur peut accéder.

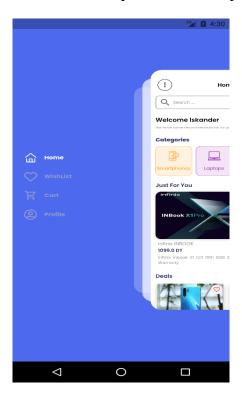


Figure 4-4 : Interface menu application mobile

## 2.3.Interface produit

L'interface produit présente les détails d'un seul produit (description, prix, commentaire, avis, produit similaire...).

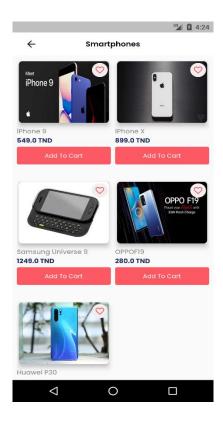


Figure 4-5 : Interface produit application mobile

## 2.4.Interface login et registre

Le processus de connexion et d'inscription est simplifié grâce à notre interface conviviale, permettant aux utilisateurs de créer un compte ou de se connecter en toute simplicité pour accéder à des fonctionnalités personnalisées et profiter d'une expérience utilisateur optimale.

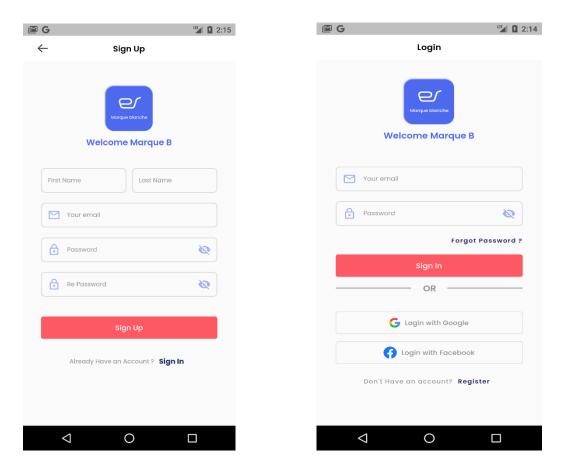


Figure 4-6: Interfaces SignUp & LogIn application mobile

### 2.5. Interface information de livraison

L'interface d'information de livraison vous permet de fournir toutes les informations nécessaires pour assurer une livraison précise et sans problème. Vous pouvez saisir votre adresse de livraison, spécifier des détails supplémentaires et sélectionner une méthode de livraison qui vous convient le mieux.

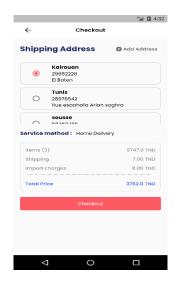


Figure 4-7: Interface information de livraison

### 2.6.Interface de Paiement

En ce qui concerne l'interface de paiement, elle vous permet de choisir parmi une variété de méthodes de paiement sécurisées. Que vous préfériez utiliser une carte de crédit ou un portefeuille électronique, vous pouvez effectuer votre paiement en toute confiance grâce à des mesures de sécurité avancées. Cette interface vous permet de finaliser votre commande en toute simplicité et de garantir une transaction sans souci.

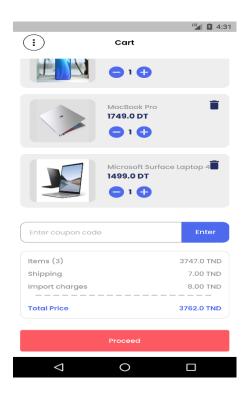
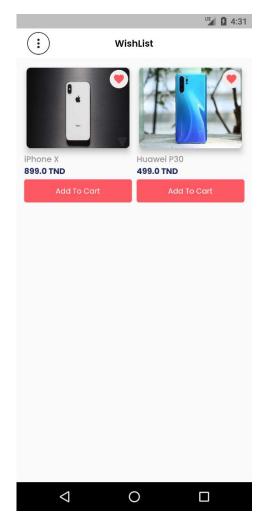


Figure 4-8 : Interface de paiement

### 2.7.Interface favoris

L'interface "Favoris" vous permet de sauvegarder facilement vos produits préférés pour un accès rapide ultérieur



Figure~4-9: Interface ~favor is

## **Conclusion**

Afin de bien conclure ce dernier chapitre, nous avons présenté les différents éléments composant la réalisation de l'application mobile. En effet on a présenté les différentes interfaces ainsi que l'environnement matériel et logiciel favorable au bon fonctionnement de l'application mobile.

## Conclusion générale

Le présent rapport a été réalisé dans le cadre de notre projet de fin d'études en vue de l'obtention du Diplôme National d'Ingénieur en Informatique. Ce travail a pour objectif de concevoir et de réaliser une application mobile de commerce électronique en marque blanche. Notre projet vise à fournir une solution complète pour les entreprises souhaitant établir leur présence en ligne et offrir une expérience d'achat exceptionnelle à leurs clients.

Au cours de ce rapport, nous détaillons les différentes étapes d'analyse, de conception et de réalisation de cette application. Nous avons commencé par une étude de l'existant, en identifiant les lacunes des solutions actuelles et en mettant en évidence l'importance de suivre une méthodologie de travail adaptée à notre projet.

J'ai ensuite spécifié les besoins fonctionnels, non fonctionnels et techniques de l'application, en accordant une attention particulière à cette étape cruciale pour assurer le succès du projet. Après avoir finalisé la phase de spécification des besoins, nous sommes passés à la phase de conception, où nous avons défini l'architecture de la plateforme, élaboré des schémas de base de données et réalisé des maquettes pour l'interface utilisateur.

Puis, j'ai abordé l'implémentation de l'application, en choisissant les environnements de travail appropriés et en développant les interfaces nécessaires pour l'application mobile.

Ce projet a été extrêmement bénéfique pour notre développement professionnel. Il nous a permis de nous familiariser avec les exigences du monde du travail, de développer notre sens des responsabilités, de la discipline et du travail en équipe. De plus, il nous a offert l'opportunité d'améliorer nos compétences dans les environnements de développement et de maîtriser les langages de programmation qui seront certainement utiles dans notre future carrière professionnelle.

## **Bibliographie**

- 1. objets connectés . *Le monde informatique*. [En ligne] [Citation : 28 02 2023.] https://www.lemondeinformatique.fr/objets-connectes-144.html.
- 2. Page d'accueil du site web « Tayara.tn ». *Tayara.tn*. [En ligne] [Citation : 02 03 2023.] https://www.tayara.tn.
- 3. Page d'accueil du site web « Wiki.Tn ». Wiki.tn. [En ligne] [Citation : 02 03 2023.] https://www.wiki.tn.
- 4. Page d'accueil du site web « eBay.com ». *eBay.com.* [En ligne] [Citation : 02 032023.] https://www.ebay.com.
- 5. Les étapes du processus Agile. *freepik.* [En ligne] [Citation : 04 03 2023.] https://fr.freepik.com/photos-vecteurs-libre/methode-agile.
- 6. Les étapes du processus Scrum. *bubble plan*. [En ligne] [Citation : 04 03 2023.] https://bubbleplan.net/pedagogie-projet/methode-scrum.
- 7. Les étapes du processus unifié. *Wikipedia*. [En ligne] [Citation : 04 03 2023.] https://fr.wikipedia.org/wiki/Processus_unifié.
- 8. diagramme de gantt. [En ligne] [Citation : 10 02 2023.] https://www.ganttproject.biz.
- 9. Architecture d'une application utilisant Flutter. [En ligne] https://docs.flutter.dev/resources/architectural-overview



# ECOLE SUPÉRIEURE PRIVÉE D'INGÉNIERIE ET DE TECHNOLOGIES

www.esprit.tn - E-mail : contact@esprit.tn

Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe: Z.I. Chotrana II - B.P. 160 - 2083 - Pôle Technologique - El Ghazala - Tél.: +216 70 685 685 - Fax.: +216 70 685 454