



# **RAPPORT DE STAGE TECHNICIEN**

EN VUE DE LA VALIDATION DU STAGE D'ÉTÉ

---

## **Développement d'un site de commerce électronique**

---

*Élaboré par :*

**ROUA SOUSSOU**

**Encadreur entreprise : Mr. Habachi Mahmoud**

**Année Universitaire : 2024-2025**

## REMERCIEMENTS

Je tiens à dédier cette section pour exprimer ma profonde gratitude envers tous ceux qui ont joué un rôle essentiel dans la réussite de cette expérience enrichissante.

À la fin de cette période de stage, je tiens à exprimer mes sincères remerciements envers mon superviseur, M. HABACHI MAHMOUD, le directeur général de l'entreprise Flesk.

Je lui suis reconnaissante de m'avoir offert l'opportunité de réaliser mon stage au sein de son organisation et pour la confiance constante dont j'ai bénéficié tout au long de cette expérience professionnelle.

Enfin, mes remerciements vont à toutes les personnes rencontrées au cours de ce stage, dont les échanges constructifs ont grandement contribué à approfondir ma compréhension du monde professionnel et à enrichir cette période d'apprentissage.

# TABLE DES MATIÈRES

<b>Introduction Générale</b>	<b>1</b>
<b>1 Présentation Générale</b>	<b>3</b>
1.1 Introduction . . . . .	4
1.2 Présentation de l'organisme d'accueil . . . . .	4
1.3 Contexte du projet . . . . .	5
1.4 Problématique . . . . .	5
1.5 Objectifs . . . . .	5
<b>2 Analyse et spécification des besoins</b>	<b>7</b>
2.1 Introduction . . . . .	8
2.2 Choix du langage de modélisation . . . . .	8
2.3 Identification des acteurs . . . . .	8
2.4 Besoins fonctionnels . . . . .	9
2.4.1 Besoins fonctionnels de l'Utilisateur . . . . .	9
2.4.2 Diagramme de cas d'utilisation global . . . . .	10
2.5 Besoins non fonctionnels . . . . .	11
2.6 Conclusion . . . . .	11
<b>3 Etude Conceptuelle</b>	<b>12</b>
3.1 Introduction . . . . .	13
3.2 Choix d'architecture . . . . .	13

## TABLE DES MATIÈRES

---

3.3	Architecture des solutions . . . . .	14
3.4	Diagramme de classes . . . . .	14
3.5	Diagrammes des séquences . . . . .	16
3.6	Conclusion . . . . .	18
<b>4</b>	<b>Réalisation</b>	<b>19</b>
4.1	Introduction . . . . .	20
4.2	Environnement et outils de Travail . . . . .	20
4.2.1	Environnement logiciel . . . . .	20
4.2.1.1	Les Framework et les bibliothèques . . . . .	20
4.2.1.2	Les Langages utilisés . . . . .	23
4.2.1.3	Les outils utilisées . . . . .	24
4.3	Interfaces de l'application . . . . .	25
4.3.1	Page d'Accueil (Home) . . . . .	25
4.3.2	Categories(electromenagers) . . . . .	26
4.3.3	Statistiques . . . . .	26
4.3.4	Profil . . . . .	27
4.3.5	creer un compte . . . . .	28
4.3.6	Recherche . . . . .	29
4.3.7	Details produit . . . . .	30
4.3.8	Panier . . . . .	31
4.4	Conclusion . . . . .	31
	<b>Conclusion générale et perspectives</b>	<b>32</b>

## TABLE DES FIGURES

1.1	Flesk . . . . .	4
2.1	Logo UML . . . . .	8
2.2	Diagramme de cas d'utilisation global . . . . .	10
3.1	Architecture Dotnet core MVC . . . . .	13
3.2	Diagramme de classes . . . . .	15
3.3	Diagramme de séquences « Modèle MVC » . . . . .	16
3.4	Diagramme de séquences « Confirmer commande» . . . . .	17
4.1	home page . . . . .	25
4.2	Categories . . . . .	26
4.3	Statistiques . . . . .	26
4.4	Profil . . . . .	27
4.5	creer un compte . . . . .	28
4.6	Recherche . . . . .	29
4.7	Details produit . . . . .	30
4.8	Panier . . . . .	31

## LISTE DES TABLEAUX

2.1	Tableau des acteurs du système . . . . .	9
-----	--	---

## INTRODUCTION GÉNÉRALE

Dans le cadre de mon stage d'été, réalisé au cours de ma deuxième année de formation en ingénierie informatique, j'ai eu l'opportunité d'effectuer un stage de 8 semaines au sein de l'entreprise Flesk.

L'objectif principal de cette expérience résidait dans le perfectionnement de mes compétences en développement web, avec une exploration approfondie des nouvelles techniques telles que "**Angular**" et "**Dotnet core**".

Ce choix de stage s'inscrivait dans ma volonté d'approfondir ma compréhension de l'environnement au sein d'une entreprise de développement, tout en cherchant à améliorer mes connaissances spécifiques dans le domaine du développement web.

Ce rapport synthétise l'ensemble du travail effectué au cours de ce stage de fin d'études, organisé en quatre chapitres distincts :

- Le premier chapitre intitulé « **Présentation générale** », aborde la description de l'organisme d'accueil, le contexte général de travail, la problématique traitée, la solution proposée, ainsi que la méthodologie et la technologie choisies pour la réalisation du projet.
- Le deuxième chapitre sera dédié pour la partie « **analyse et spécification des besoins** » . débutant par le diagramme de cas d'utilisation global pour planifier les exigences identifiées.
- Le troisième chapitre intitulé « **Étude conceptuelle** » explore la conception préliminaire de l'application, avec un accent sur le diagramme de classe global et le découpage du projet.
- Le quatrième et dernier chapitre intitulé « **Réalisation** » met en lumière le travail accompli à travers toutes les interfaces conçues pour l'application en question.

## LISTE DES TABLEAUX

---

- Enfin, ce rapport se clôture par une conclusion générale récapitulant le travail réalisé, mettant en avant les compétences et l'expérience acquises au cours du stage.  
Des perspectives sont également évoquées pour l'amélioration continue du travail effectué.



# CHAPITRE 1

---

## PRÉSENTATION GÉNÉRALE

### 1.1 Introduction

L'étude préliminaire, prélude indispensable à la concrétisation d'un projet réussi, se décline en deux volets distincts.

La première section sera dédiée à la description de l'environnement de travail, incluant la présentation de l'organisme d'accueil.

La seconde partie mettra en lumière les motivations, la problématique, l'analyse de l'existant, la solution proposée, ainsi que la méthodologie de travail.

### 1.2 Présentation de l'organisme d'accueil

Flesk est une société de développement de logiciels qui propose des solutions sur mesure aux startups et aux entreprises dans des secteurs variés tels que la vente au détail, la banque, l'assurance, l'hôtellerie, l'éducation, la mode, et la bijouterie. Elle se distingue comme l'un des leaders du marché dans le développement d'applications mobiles et web, offrant des services allant du cloud computing à la conception UX/UI, en passant par le développement SaaS, le référencement, le conseil informatique et le marketing numérique. Chaque produit est conçu avec soin et passion pour garantir une performance optimale.



FIGURE 1.1 – Flesk

### 1.3 Contexte du projet

Le travail actuel s'inscrit dans le cadre d'un stage d'été réalisé chez Flesk, avec pour objectif principal le développement d'un site e-commerce. Ce projet vise à doter l'entreprise d'une plateforme de vente en ligne efficace, optimisant l'expérience utilisateur, la gestion des stocks, et les commandes.

### 1.4 Problématique

Dans un contexte où les habitudes de consommation évoluent rapidement vers le commerce en ligne, les entreprises sont confrontées à plusieurs défis pour répondre aux attentes des consommateurs. La vente traditionnelle, limitée géographiquement et temporellement, ne suffit plus à satisfaire les clients qui recherchent la praticité, la rapidité et la disponibilité des produits en ligne. De plus, l'absence d'une présence en ligne efficace limite la compétitivité des entreprises face à des concurrents déjà bien implantés sur le web. Cette situation soulève des enjeux importants en matière de gestion des stocks, de suivi des commandes, de logistique, et d'amélioration de l'expérience utilisateur. L'incapacité à s'adapter à ces changements peut entraîner une perte d'opportunités commerciales et une diminution de la satisfaction client.

### 1.5 Objectifs

Pour remédier aux problèmes identifiés, le projet de développement du site e-commerce vise à atteindre les objectifs suivants :

- **Créer une plateforme de vente en ligne intuitive** : Développer un site e-commerce qui offre une navigation fluide et une interface utilisateur conviviale, facilitant ainsi l'expérience d'achat.
- **Optimiser la gestion des stocks** : Mettre en place un système de gestion des stocks efficace permettant de suivre en temps réel la disponibilité des produits, réduisant ainsi les risques de rupture de stock ou de surstock.
- **Améliorer le traitement des commandes** : Élaborer un processus de commande simplifié qui permet aux clients de finaliser leurs achats rapidement et de manière sécurisée, tout en garantissant un suivi efficace des commandes.
- **Développer une stratégie de marketing digital** : Élaborer des outils et des techniques de marketing numérique pour attirer et fidéliser les clients, augmentant ainsi la visibilité

de la plateforme et les ventes.

- **Analyser les performances du site :** Mettre en place des outils d'analyse pour suivre les performances du site e-commerce et identifier les axes d'amélioration continue.

## CHAPITRE 2

# ANALYSE ET SPÉCIFICATION DES BESOINS

### 2.1 Introduction

Dans cette phase de spécification des exigences, l'objectif est de définir les fonctionnalités attendues du système. Ce chapitre commence par présenter les acteurs impliqués, puis se concentre sur l'analyse des besoins fonctionnels et non fonctionnels. Ces exigences seront formalisées à travers des diagrammes de cas d'utilisation, détaillant les divers scénarios que les acteurs peuvent réaliser.

### 2.2 Choix du langage de modélisation

Pour la spécification et la conception de ce travail, j'ai utilisé le langage de modélisation UML, largement utilisé pour décrire les exigences, documenter les systèmes et esquisser l'architecture logicielle, notamment dans le domaine du développement de logiciels et de la conception orientée objet.

UML est structuré autour de trois axes différents, chacun dédié à la présentation d'un concept spécifique du système logiciel.

Dans ce rapport, je vais focaliser sur quelques diagrammes jugés utiles et suffisants pour la compréhension du projet, notamment les diagrammes de cas d'utilisation, de classes et de séquence.



FIGURE 2.1 – Logo UML

### 2.3 Identification des acteurs

Un acteur, dans le contexte de ce projet, est l'idéalisation d'un rôle joué par une personne externe ou un processus interagissant avec le système. Il peut être classifié en tant qu'acteur principal ou acteur secondaire. Un acteur est considéré comme principal pour un cas d'utilisation lorsqu'il bénéficie directement de ce cas. Les autres acteurs sont alors qualifiés de secondaires. Tous les participants sont présentés dans le tableau 2.1 ci-dessous, permettant ainsi une vue d'ensemble des différents rôles impliqués dans l'écosystème du système. :

TABLE 2.1 – Tableau des acteurs du système

Acteur	Role
Utilisateur	C'est la personne qui visite le site pour rechercher, consulter, et acheter des produits. Sa satisfaction est primordiale, car elle détermine le succès de la plateforme. Il interagit avec le site en naviguant à travers les catégories de produits, en utilisant la fonction de recherche, et en finalisant ses achats.

## 2.4 Besoins fonctionnels

L'application à concevoir doit offrir un ensemble de fonctionnalités alignées sur les besoins des utilisateurs.

### 2.4.1 Besoins fonctionnels de l'Utilisateur

L'utilisateur du site e-commerce a plusieurs besoins fonctionnels qui doivent être satisfaits pour garantir une expérience d'achat agréable et efficace. Ces besoins incluent :

- **Navigation intuitive** : L'utilisateur doit pouvoir naviguer facilement sur le site, en accédant rapidement aux différentes catégories de produits et en trouvant les informations souhaitées sans effort.
- **Fonction de recherche** : Une barre de recherche efficace doit être disponible pour permettre à l'utilisateur de rechercher des produits spécifiques en utilisant des mots-clés, des références ou des désignations.
- **Consultation des produits** : L'utilisateur doit pouvoir consulter les détails des produits, y compris les images, descriptions, prix et avis des autres clients, afin de prendre des décisions éclairées.
- **Panier d'achat** : L'utilisateur doit avoir la possibilité d'ajouter des produits à un panier d'achat, de modifier les quantités ou de supprimer des articles, tout en voyant le montant total des achats en temps réel.
- **Processus de commande simplifié** : L'utilisateur doit pouvoir finaliser ses achats en quelques étapes simples, avec des options de connexion pour les utilisateurs enregistrés et une option de paiement pour les invités.

### 2.4.2 Diagramme de cas d'utilisation global

À ce stade, il est possible de représenter de manière globale les différents cas d'utilisation à l'aide du diagramme de cas d'utilisation global, comme illustré dans la figure suivante :

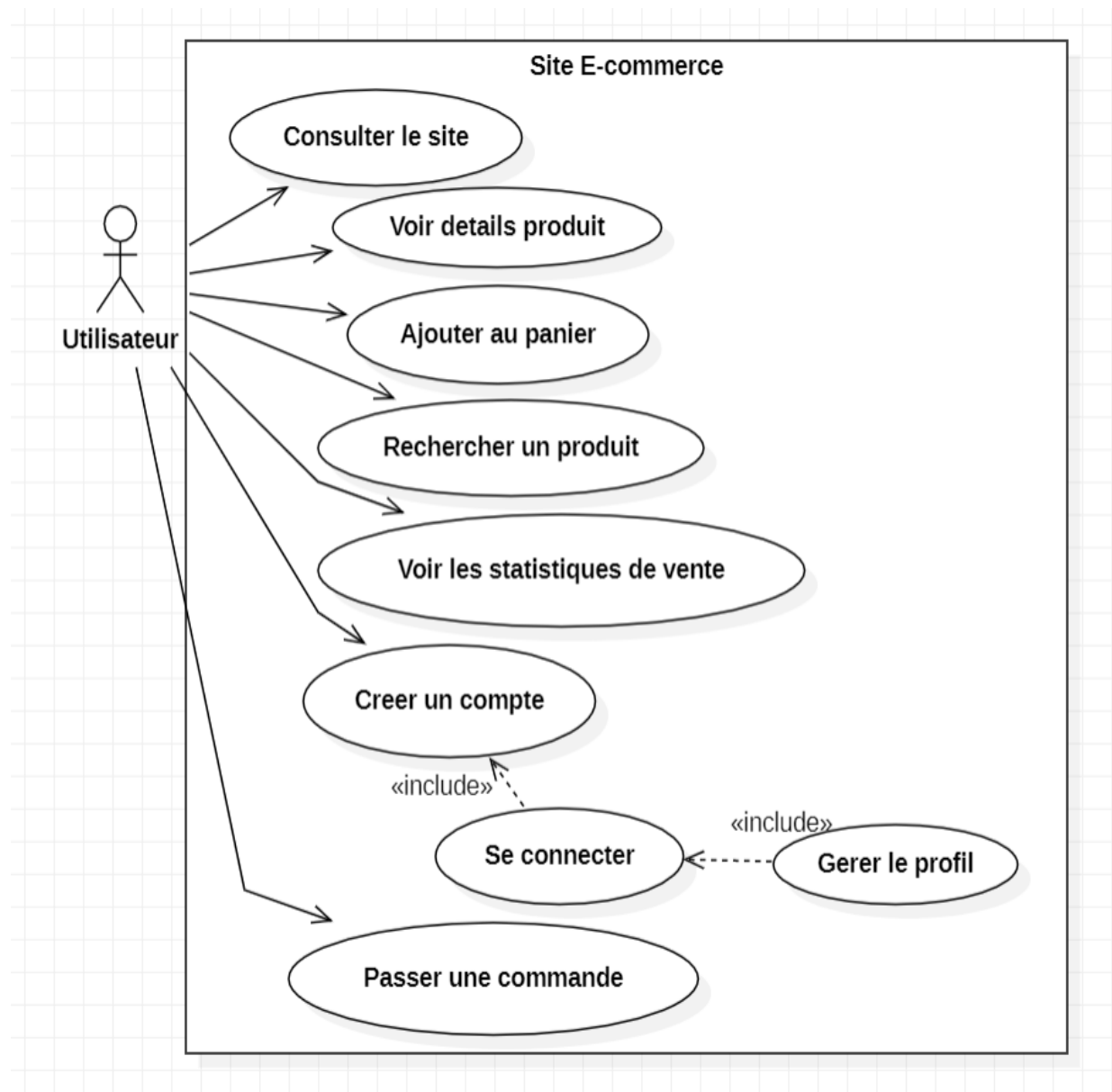


FIGURE 2.2 – Diagramme de cas d'utilisation global



### 2.5 Besoins non fonctionnels

Un besoin non fonctionnel est une exigence qui caractérise une propriété souhaitée (qualité) du système. Mon solution doit répondre aux critères suivants :

**Sécurité** : Ce critère est essentiel pour garantir la protection des données des utilisateurs. Nous mettrons en place des contrôles d'accès rigoureux, en attribuant des autorisations et des droits d'accès aux différentes sections du site. De plus, des protocoles de cryptage seront utilisés pour sécuriser les informations sensibles, notamment lors des transactions financières.

**Performance** : Elle offre des temps de réponse rapides, avec un chargement des pages en moins de 2 secondes, même lors de pics de trafic. Cela assurera une expérience utilisateur agréable et réduira le taux d'abandon lors des achats.

### 2.6 Conclusion

Ce chapitre a été essentiel pour une introduction claire au projet, fournissant une vision détaillée du sujet. Nous avons initié le processus en identifiant les acteurs et en énumérant les besoins fonctionnels et non fonctionnels essentiels pour le service de paie. En outre, j'ai présenté le diagramme de cas d'utilisation global, offrant ainsi une représentation visuelle des interactions clés dans le système. Dans le chapitre suivant, je poursuivrai en détaillant l'étude de conception du projet.

## CHAPITRE 3

ETUDE CONCEPTUELLE

## 3.1 Introduction

Après avoir consacré un temps important à l'analyse et à la clarification des besoins dans le chapitre précédent, je suis maintenant prêt à entamer la phase de conception avec une vision précise des principales fonctions à implémenter. Ce chapitre se centrera sur la méthode de conception utilisée, accompagnée des schémas associés. Cette étape revêt une importance cruciale, car elle me permettra d'établir un modèle solide qui servira de base lors de la mise en place de la solution.

## 3.2 Choix d'architecture

Les architectures utilisées dans le domaine des applications web sont diverses et dépendent de facteurs tels que l'utilisation et le déploiement de l'application. Pour notre solution, nous avons opté pour l'architecture MVC (Modèle, Vue, Contrôleur). Dans cette structure, le modèle, situé au niveau le plus bas, est chargé de la gestion des données. La représentation des données est assurée par le composant de vue, qui génère l'interface utilisateur pour l'utilisateur. Quant au contrôleur, il joue le rôle d'intermédiaire en facilitant la communication entre les vues et le modèle. Il agit comme une pièce maîtresse dans cette architecture.

La figure ci-dessous (3.1a) illustre l'architecture MVC dans le contexte de Dotnet core.

Chaque classe dans le diagramme se traduit en trois dossiers : « controllers », « models » et « Serializer » comme c'est indiqué sur 3.1b ci-dessous où on regroupe tous les modèles, les contrôleurs et les vues de la partie backend de du plateforme.

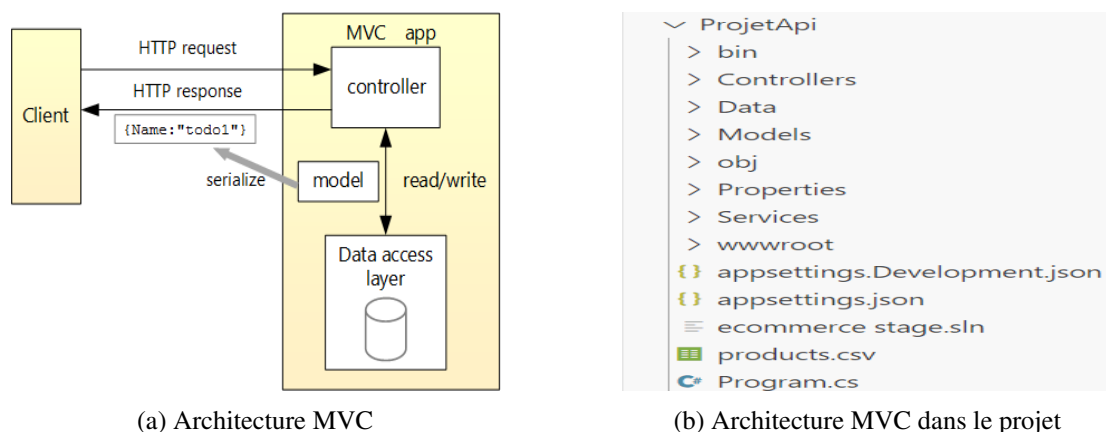


FIGURE 3.1 – Architecture Dotnet core MVC

### 3.3 Architecture des solutions

Notre application repose sur une architecture 3 tiers, illustrée par la figure ??, qui définit clairement trois couches distinctes au sein de notre système. Cette approche logique vise à assurer une séparation nette des responsabilités. La composition de notre architecture comprend :

**- Application Web (Front-end) :**

Développée avec Angular en raison de sa flexibilité, de ses performances optimales et de son écosystème riche.

**- Serveur (Logique Applicative Côté Serveur) :**

Réalisé avec le framework .NET Core, qui a été choisi pour son efficacité, sa robustesse et sa capacité à gérer des réponses en temps réel.

**-Persistance des Données :**

Utilisation de MySQL pour ses fonctionnalités avancées telles que l'atomicité, la cohérence, l'isolation et la durabilité (ACID). MySQL offre des mécanismes de sauvegarde et de restauration efficaces, garantissant ainsi la préservation des données en cas de panne du système ou de suppression involontaire.

En complément, nous avons intégré Postman pour tester les API avant leur intégration avec Angular. Cette combinaison de composants assure une architecture robuste et bien équilibrée pour notre application, permettant de garantir la fiabilité des interactions entre le front-end et le serveur.

### 3.4 Diagramme de classes

La représentation du diagramme de classe constitue une étape clé dans notre processus, illustrant l'ensemble des entités, c'est-à-dire des informations manipulées au sein de l'application à développer. Ce diagramme offre une vision structurée des différentes classes, de leurs attributs, et des relations entre elles.

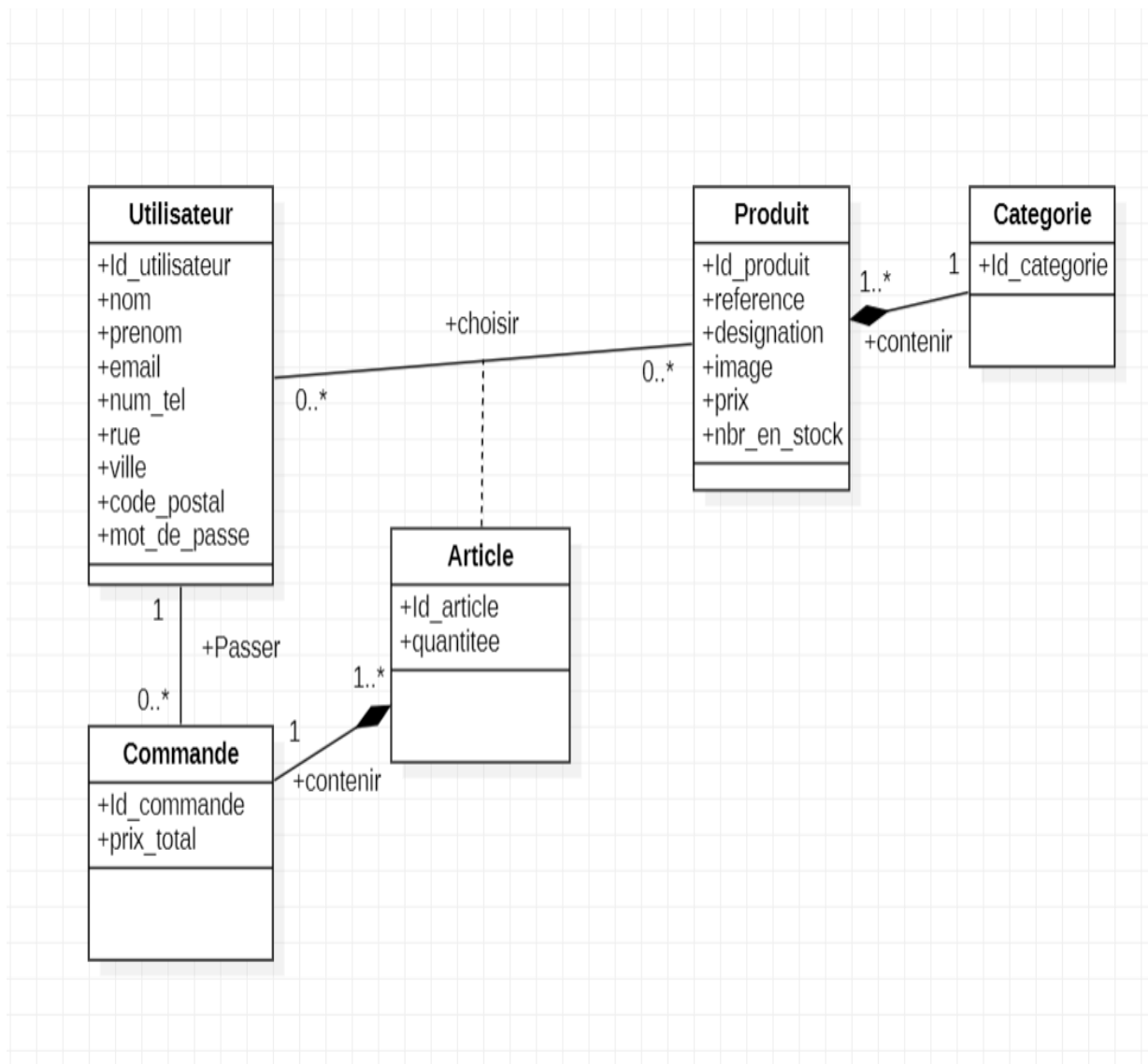


FIGURE 3.2 – Diagramme de classes

### 3.5 Diagrammes des séquences

La vision dynamique de la conception logicielle prend en considération le facteur temps dans la modélisation des interactions entre les utilisateurs et les objets du système.

Cette section présente des diagrammes de séquences visant à décrire le flux interne d'informations échangées par les fonctionnalités du système, ainsi que les diverses interactions entre les acteurs et le système.

Ces diagrammes détaillés de séquences offrent une représentation précise des collaborations entre objets, en mettant l'accent sur la dimension temporelle et en respectant la chronologie de l'envoi des messages.

Cette approche temporelle permet une compréhension approfondie des interactions au sein du système.

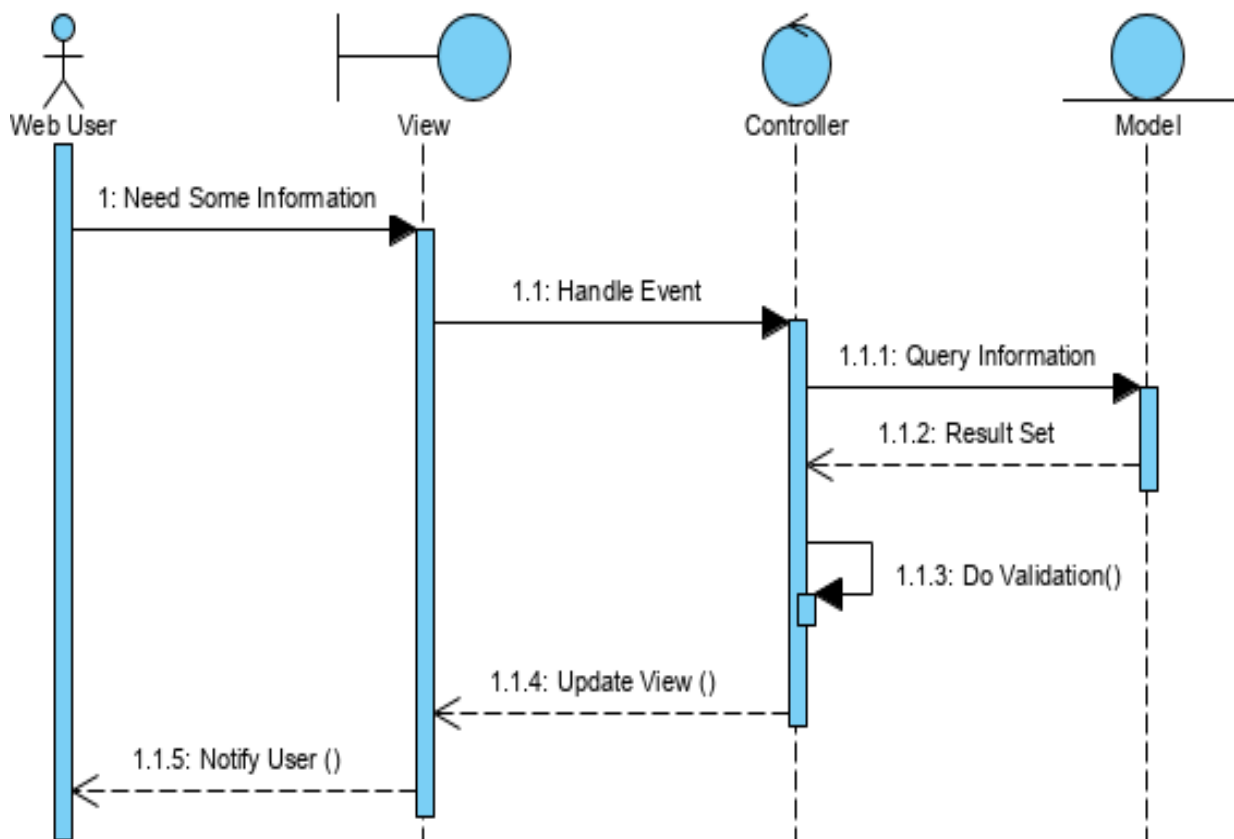


FIGURE 3.3 – Diagramme de séquences « Modèle MVC »

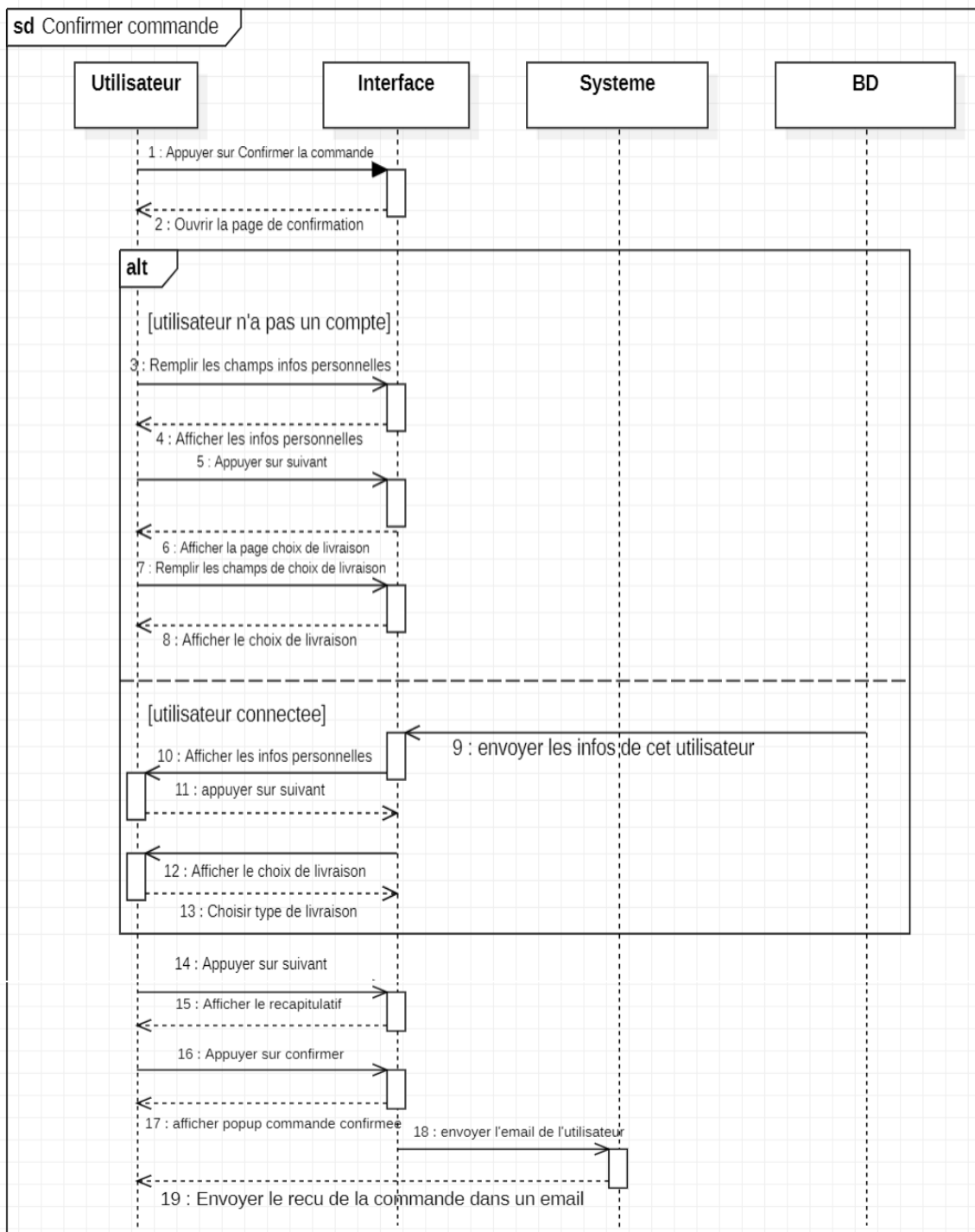


FIGURE 3.4 – Diagramme de séquences « Confirmer commande»

### **3.6 Conclusion**

Dans ce chapitre, j'ai exploré la conception qui sous-tend le fonctionnement de site. J'ai commencé par présenter l'architecture de l'application, détaillant ses composants essentiels. Ensuite, j'ai conclu ce chapitre en décrivant la conception de l'application à travers l'utilisation du diagramme de classes et du diagramme de séquences. Le prochain chapitre abordera la dernière phase du projet, à savoir la phase de réalisation qui englobera le choix des technologies utilisées et les interfaces de l'application.



## CHAPITRE 4

RÉALISATION

### 4.1 Introduction

Après la phase de conception de l'application, ce chapitre se concentre sur la phase de mise en œuvre. Dans un premier temps, je décrirai l'environnement de travail créé spécifiquement pour le développement de l'application. Ensuite, je fournirai un aperçu détaillé du travail accompli en présentant des captures d'écran qui illustrent le fonctionnement concret de l'application. Cette section permettra de visualiser concrètement le résultat de la conception à travers les étapes de développement.

### 4.2 Environnement et outils de Travail

Dans cette section, je consacrerai la présentation de l'environnement logiciel utilisé dans le cadre de mon projet :

#### 4.2.1 Environnement logiciel

Dans cette section consacrée aux technologies, j'ai détaillé les outils, les frameworks et les bibliothèques que j'ai utilisés.

Ces choix ont été faits dans le but de faciliter la phase de conception, la rédaction du rapport, ainsi que la création et la réalisation du projet.

##### 4.2.1.1 Les Framework et les bibliothèques

Dans la section consacrée aux frameworks et bibliothèques, je présente les éléments clés qui ont joué un rôle central dans la réalisation de mon projet.

Ces frameworks et bibliothèques ont été soigneusement sélectionnés pour optimiser la conception, la rédaction du rapport, ainsi que la création et la réalisation du projet dans son ensemble.

##### **Angular**

C'est un framework JavaScript front-end qui permet de créer des applications web dynamiques. Il offre une architecture basée sur les composants, facilitant la réutilisation du code et l'interaction utilisateur. Angular fournit également des outils intégrés pour le routage, la gestion des formulaires et la consommation d'API, ce qui en fait un choix idéal pour construire des interfaces utilisateurs interactives et réactives.

##### **PrimeNG**



PrimeNG est une bibliothèque UI pour Angular avec plus de 80 composants interactifs, offrant des fonctionnalités comme la pagination et le filtrage. Elle permet de créer des interfaces web modernes et personnalisables rapidement.



### **ASP.NET Core**

Ce framework de développement web côté serveur est utilisé pour créer des applications robustes et évolutives. Il permet de développer des API RESTful facilement, facilitant l'intégration avec des bases de données et des applications front-end. Avec sa performance optimisée et sa compatibilité multiplateforme, ASP.NET Core est parfait pour le développement d'applications modernes et hautement disponibles.



### **MySQL**

En tant que système de gestion de base de données relationnelle, MySQL est largement utilisé pour stocker et gérer les données de l'application. Sa fiabilité et sa flexibilité permettent de structurer les données de manière efficace, tout en offrant des fonctionnalités avancées telles que la gestion des transactions et la réplication.



### **Power BI**

est un outil d'analyse de données développé par Microsoft. Il permet de visualiser des données sous forme de rapports interactifs et de tableaux de bord afin d'aider à la prise de décision basée sur des données. Power BI peut se connecter à une variété de sources de données (bases de données, fichiers Excel, services cloud, etc.) pour transformer les données brutes en insights exploitables.



### 4.2.1.2 Les Langages utilisés

les langages que nous avons utilisés dans le développement et dans la partie de conception sont les suivants :



**C#** :C# est un langage de programmation orienté objet développé par Microsoft. Il est principalement utilisé pour le développement d'applications côté serveur, notamment dans des environnements ASP.NET Core, qui ont été employés pour la création de notre API backend. Grâce à sa robustesse, il facilite la gestion de la logique métier, les interactions avec la base de données, et permet d'assurer une communication fluide avec le frontend.



**TypeScript** : TypeScript est un sur-ensemble typé de JavaScript, développé par Microsoft. Il est utilisé dans la partie frontend de notre application, notamment avec Angular, pour apporter des fonctionnalités de typage statique à JavaScript. Ce langage permet de structurer le code de manière plus rigoureuse, d'améliorer la lisibilité et la maintenabilité du projet. TypeScript nous a permis d'interagir avec l'API backend et de gérer efficacement les composants de l'interface utilisateur.



**UML** : est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet. Je l'ai utilisé comme formalisme de conception.

### 4.2.1.3 Les outils utilisés

Les outils utilisés pour réaliser notre projet (pour la conception, pour le développement, rédaction du rapport, etc.) sont les suivants :



**StarUML** : est un logiciel de modélisation UML, permettant de créer les diagrammes de cas d'utilisation, de séquence et de package. Nous avons utilisé cet outil pour créer les diagrammes des séquences.



**Visual Studio Code** : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré . Nous avons utilisé cet IDE pour développer la partie frontend du projet.



**GitHub** : GitHub est une plateforme collaborative basée sur Git qui facilite le développement de projets grâce à des outils de gestion de versions et de soumission de modifications.



**Postman** : est un environnement de développement des API (Application Programming Interface). Il facilite le test des API et des services web, il est utilisé aussi pour le développement et la documentation des API. Nous avons utilisé cet outil pour tester nos APIs après les consommer.



**FileZilla Server** : FileZilla Server est un logiciel open source permettant de configurer un serveur FTP pour transférer des fichiers de manière sécurisée via les protocoles FTP, FTPS, et SFTP.

### 4.3 Interfaces de l'application

Dans cette section, nous présentons les interfaces concrètes de notre application. Celle-ci est constituée de plusieurs interfaces, comprenant notamment :

#### 4.3.1 Page d'Accueil (Home)

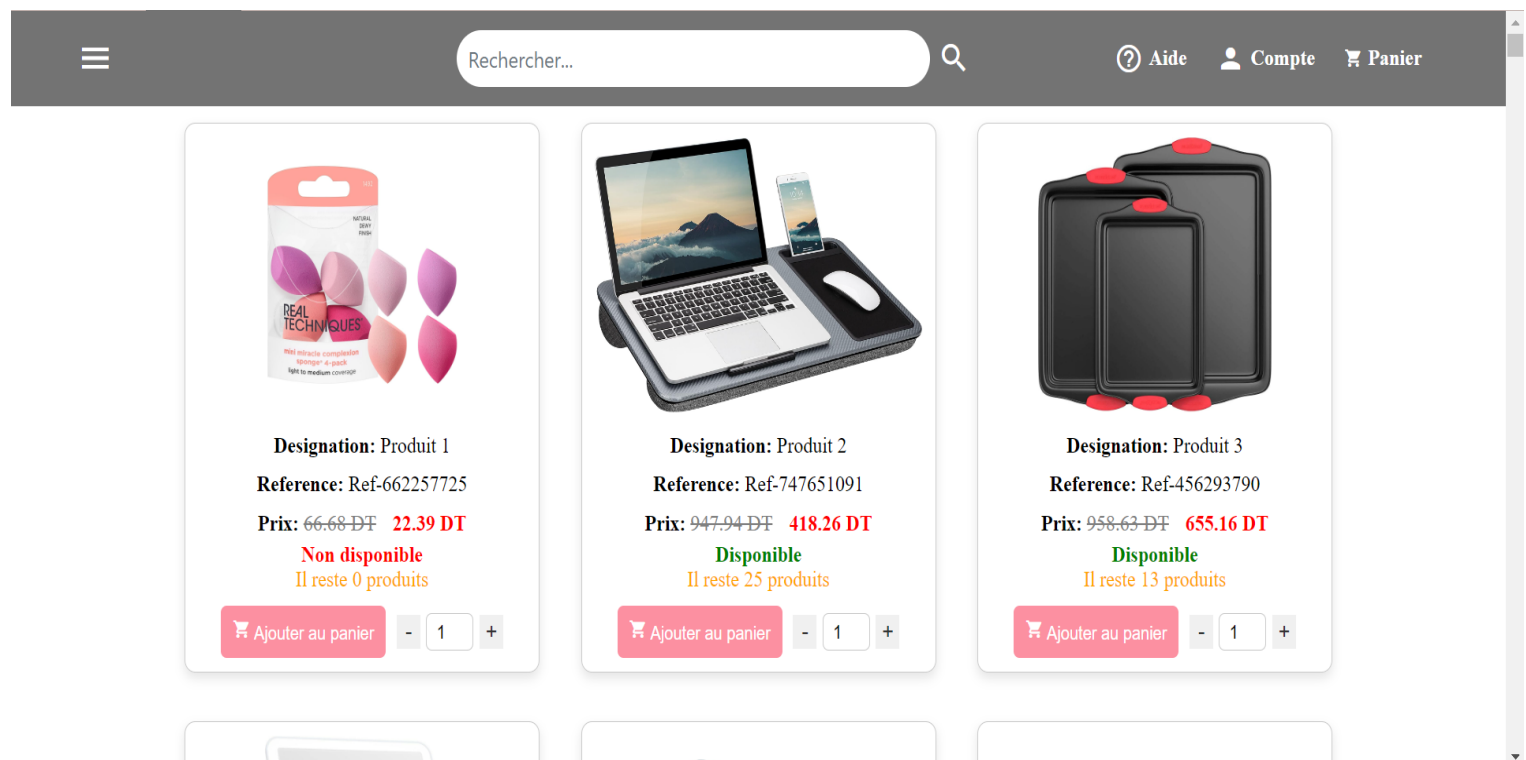


FIGURE 4.1 – home page

### 4.3.2 Categories(electromenagers)

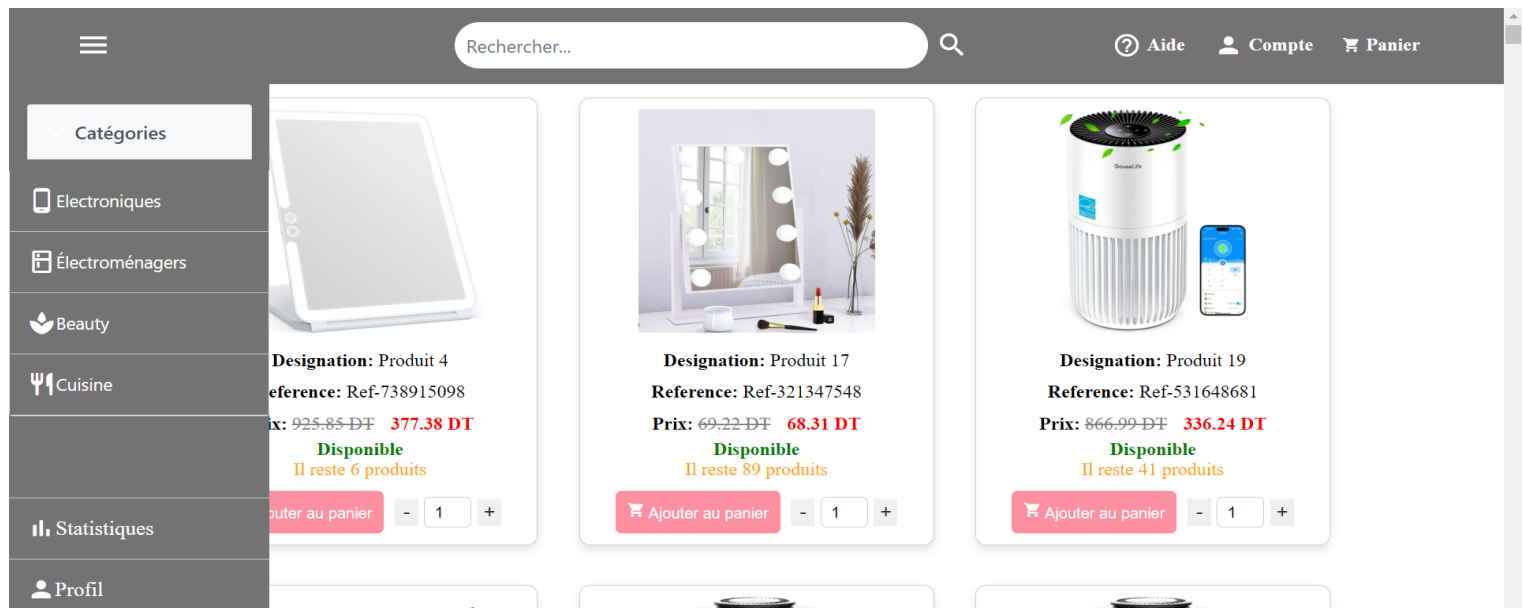


FIGURE 4.2 – Categories

### 4.3.3 Statistiques

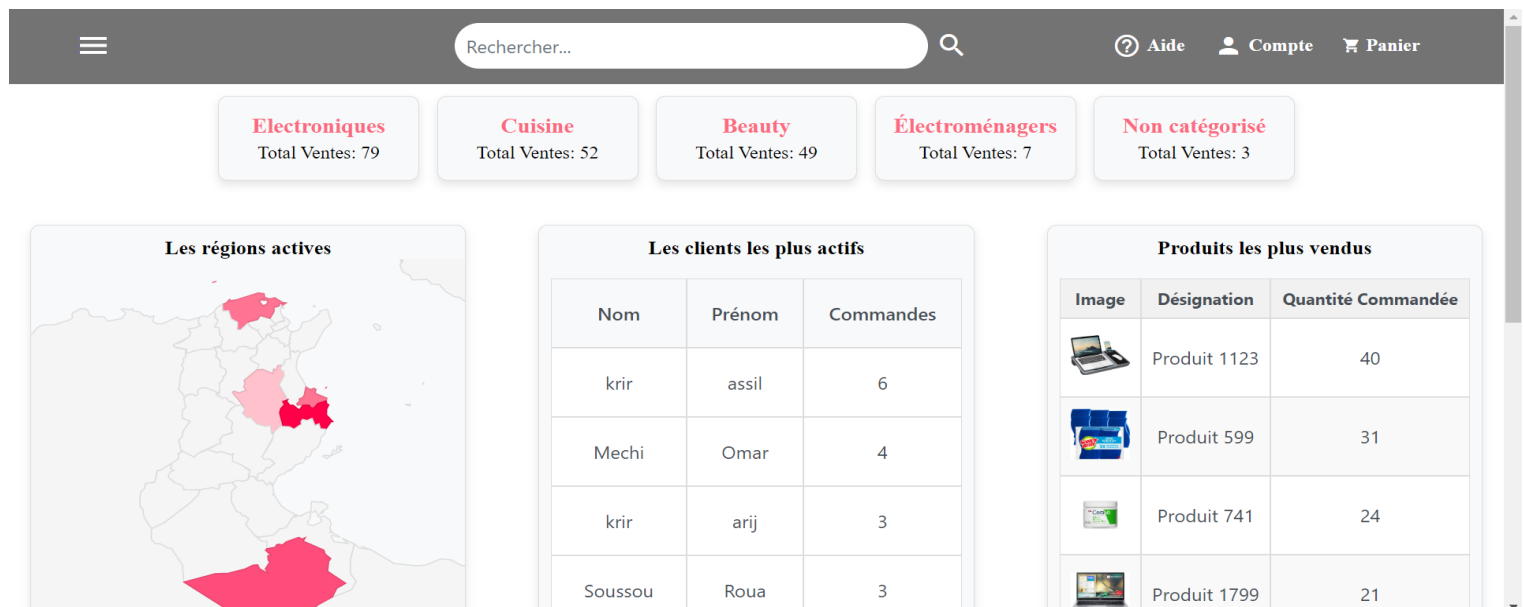


FIGURE 4.3 – Statistiques



### 4.3.4 Profil

The figure displays two screenshots of a web application's user profile page, titled "Mon Profil". The page has a dark grey sidebar on the left with a menu containing "Catégories", "Statistiques", and "Profil". The top navigation bar is dark grey and includes a search bar labeled "Rechercher...", a magnifying glass icon, and links for "Aide", "Compte", and "Panier".

The main content area is light grey and features the heading "Mon Profil" with the instruction "Mettez à jour vos informations personnelles et de contact." Below this, there are two sections:

- Informations Personnelles:** This section contains four input fields: "Nom", "Prénom", "Ville", and "Code Postal".
- Informations de Contact:** This section contains three input fields: "Numéro de téléphone", "Email", and "Mot de Passe". The "Mot de Passe" field has a toggle icon (an eye with a slash) to the right of it. Below these fields is a red button labeled "Enregistrer les modifications".

FIGURE 4.4 – Profil

### 4.3.5 créer un compte

The image shows a web application interface for creating a new account. The header is dark gray with a search bar labeled 'Rechercher...' and navigation links: 'Aide', 'Compte', and 'Panier'. A dropdown menu is open under the 'Compte' link, showing 'Se déconnecter' and 'Créer un compte'. The main content area is titled 'Créer un compte' and contains the following form fields:

- Nom :
- Prénom :
- Adresse Email :
- Numéro de téléphone :

Below these fields, there are additional fields for 'Rue', 'Ville', 'Code Postal', and 'Mot de Passe'. A pink button labeled '✓ Confirmer' is at the bottom of the form.

FIGURE 4.5 – créer un compte

### 4.3.6 Recherche

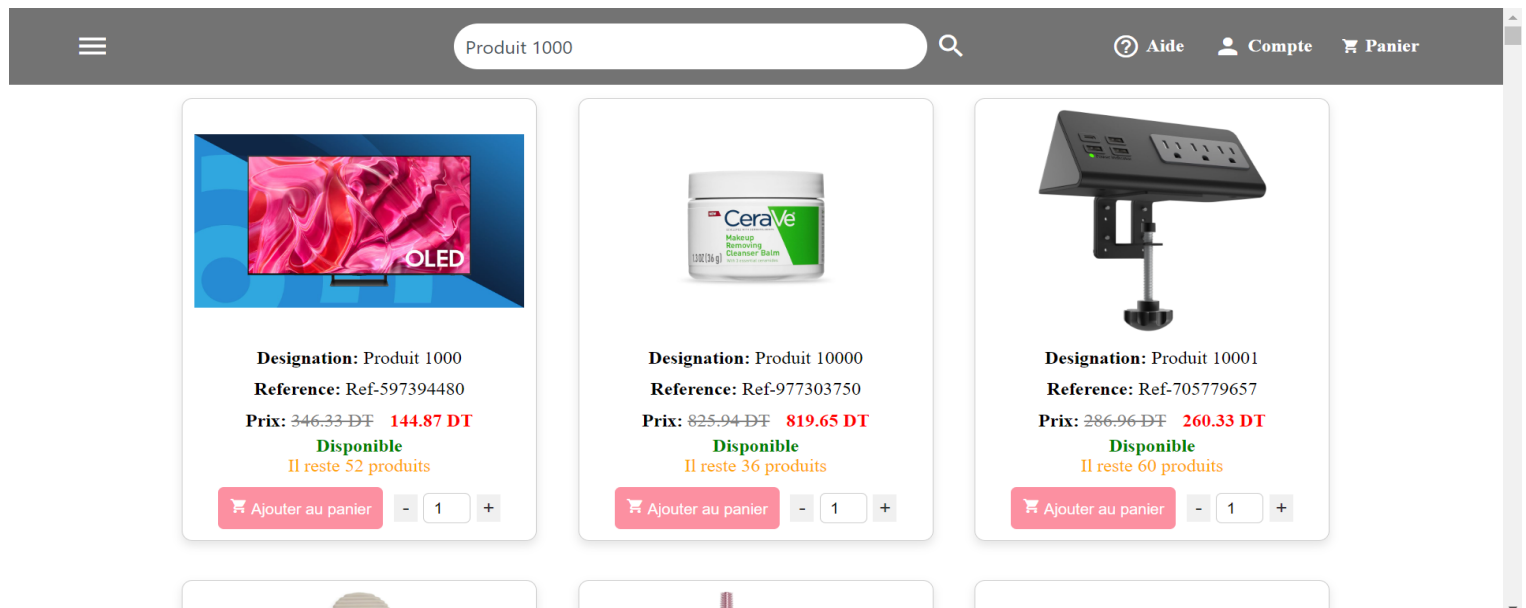


FIGURE 4.6 – Recherche

### 4.3.7 Details produit

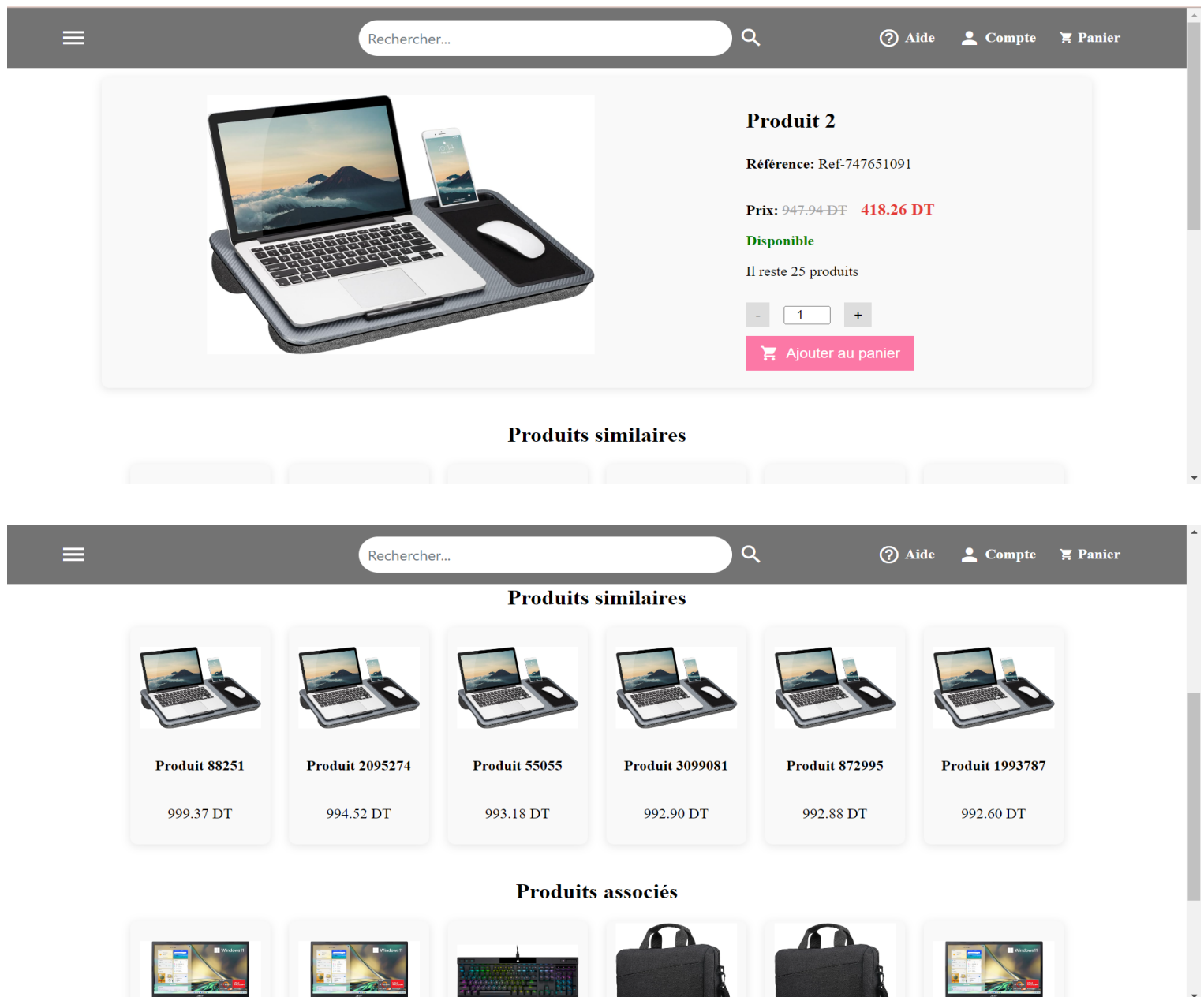


FIGURE 4.7 – Details produit

### 4.3.8 Panier



FIGURE 4.8 – Panier

## 4.4 Conclusion

À ce stade, nous parvenons à la conclusion de l'étude du projet.

Dans ce chapitre, j'ai examiné en détail mon environnement de travail et justifié mes choix technologiques.

J'ai également présenté les outils spécifiques que j'ai employés pour la réalisation du projet.

Enfin, des captures d'écran ont été fournies pour illustrer les interfaces de site.

## CONCLUSION GÉNÉRALE ET PERSPECTIVES

Ce stage a été extrêmement enrichissant d'un point de vue technique, me permettant de me familiariser avec de nouveaux environnements de développement informatique.

En conséquence, j'ai acquis la capacité de m'adapter rapidement à de nouveaux langages de programmation et à des environnements matériels et logiciels variés.

Cette expérience me confère la certitude de pouvoir maîtriser efficacement de nouvelles technologies de programmation.

En somme, ce stage a pleinement répondu à mes attentes.

