



ECOLE NATIONALE SUPEÉRIEURE D'INFORMATIQUE ET
D'ANALYSE DES SYSTÈMES

PROJET D'UNE PLATE-FORMES DE DEVELOPPEMENT / JEE

Développement d'une plateforme e-Commerce

Réalisé par :

- Mohamed Amine IBRAHIMI
- Mohamed GHARBI
- Taha NACER
- Hajar ATBIB

Encadré par :

- Pr.Mahmoud EL-HAMLAOUI

REMERCIEMENT

AU TERME DE CE TRAVAIL, J'AVOUE MA RECONNAISSANCE À TOUTES LES PERSONNES QUI ONT CONTRIBUÉ, DE PRÈS OU DE LOIN, AU BON DÉROULEMENT DE NOTRE PROJET. ON TIENT À EXPRIMER TOUT PARTICULIÈREMENT NOTRE GRATITUDE À PR.EL HAMLAOUI MOHAMOUD POUR SON ACCUEIL, POUR AVOIR ACCEPTÉ LA CHARGE DE NOUS ENCADRER, IL A SU NOUS DONNER UNE GRANDE LIBERTÉ D'INITIATIVE TOUT EN RESTANT TOUJOURS PRÉSENT POUR DISCUTER LES PROBLÈMES RENCONTRÉS TOUTE LA DURÉE DU PROJET..

NOUS SAISISSEZ AUSSI L'OCCASION POUR REMERCIER NOS ENSEIGNANTS DE L'ENSIAS POUR NOUS AVOIR OFFERT UNE FORMATION ACTUALISÉE TRÈS COMPLÈTE.

ABSTRAIT

Notre projet se concentre sur l'étude, la conception et la réalisation d'un site de e-commerce électronique, qui permet aux clients du site d'enrichir de plus en plus sa base de données clientèle, ayant pour cible dans notre cas, ce qu'on appelle le cyberconsommateur, c'est à dire le client distant sur le net.

Les objectifs majeurs de ce site sont : la possibilité de présenter des produits dans une boutique virtuelle à la disposition de tout le monde, de faire des transactions commerciales, faire des achats en ligne et de faciliter la tâche du paiement.

Mot-clés : Plateforme, Produit, Achat, vente, Products, Distant, Connexion.

ABSTRACT

Our project focuses on the study, design and creation of an electronic e-commerce site, which allows the site's customers to enrich more and more on a customer database, targeting in our case, what is called the cyber consumer, tell the remote client on the net.

The main objectives of this site are : the possibility of presenting products in a virtual store available to everyone, do business transactions, shop online and make payment easier.

Mot-clés : Store, Plateforme, Items, Products, Distant Shop, Connexion.

ABBREVIATION

SQL : Structured Query Language

UML : Unified Modeling Language

MCD : Modèle conceptuel des données

HTML : Hypertext Markup Language

JEE : Java Enterprise Edition

CSS : Cascading Style Sheets

JSON : JavaScript Object Notation

LMS : Teaming Management System

REST : Representational state transfer

HTTP : Hypertext Transfer Protocol

TCP : Transmission Control Protocol

XML : Extensible Markup Language

API : application programming interface

Table des matières

1	Contexte Générale du Projet	1
1.1	Objectif du projet	1
1.2	Plan de travail	2
1.2.1	Organisation du rapport :	2
1.3	Méthodologie de travail : DevOps	2
1.3.1	Définition	2
1.3.2	Fonctionnement de DevOps	2
1.3.3	Les avantages de DevOps	3
1.4	Conclusion	4
2	Architecture Micro-Service	5
2.1	Définition :	5
2.2	Explication de la définition de l'API REST	6
2.3	Comment fonctionnent les API REST ?	7
2.4	Mots clés :	7
2.5	Pourquoi devriez-vous utiliser l'API REST ?	8
2.6	Directeur de la conception d'API REST : Serveur / Client	8
2.7	Conclusion	8
3	Etude de l'existant spécification des besoins	9
3.1	Introduction	9
3.2	Etude de l'existant	9
3.2.1	Description de l'existant	9
3.2.2	Critique de l'existant	9
3.2.3	Solution proposée	10
3.3	Etude des besoins	10
3.3.1	Besoins fonctionnels	10
3.3.1.1	L'exposition des produits :	11
3.3.1.2	L'inscription du client :	11
3.3.1.3	Ajout des produits au panier :	11
3.3.1.4	Mode de livraison :	11
3.3.1.5	Boutique de livraison :	11
3.3.1.6	La livraison à domicile :	11
3.3.1.7	La confirmation de la commande :	11
3.3.1.8	Le paiement :	12
3.3.1.9	La fin de l'opération d'achat :	12
3.4	Conclusion	12
4	Conception	13
4.1	Langage de modélisation	13
4.2	Diagrammes des Cas d'utilisation	14
4.2.1	Définition	14
4.2.2	Les acteurs de notre projet	14
4.2.3	Nos diagrammes des cas :	15
4.3	Diagramme de Classe	16
4.3.1	Définition	16
4.3.2	Notre diagramme des classes	16
4.4	Diagramme de séquence	17
4.4.1	Définition	17

4.4.2	Les diagrammes de séquences de notre site web	17
4.5	Conclusion	18
5	Réalisation du projet	19
5.1	Environnement de développement	19
5.1.1	Langages de développement	19
5.1.1.1	Java	19
5.1.1.2	Python	19
5.1.1.3	JavaScript	20
5.1.1.4	TypeScript	20
5.1.1.5	HTML & CSS	20
5.1.2	Librairies	21
5.1.2.1	NLTK	21
5.1.2.2	Numpy	21
5.1.2.3	TensorFlow	21
5.1.3	Logiciels	22
5.1.3.1	Atom	22
5.1.3.2	mySQL Admin	22
5.1.3.3	Docker	22
5.1.4	Frameworks	23
5.1.4.1	Framwork Angular : Front-End	23
5.1.4.2	Framwork Spring Boot : Backend	23
5.2	La partie de la réalisation	24
5.2.1	Coté Application web	24
5.2.1.1	Interface : Chercher une montre	24
5.2.1.2	Interface : Consulter le panier	24
5.2.1.3	Interface : Ajouter une adresse de paimenet	25
5.2.1.4	Interface : Ajouter une adresse de livraison	25
5.2.1.5	Fichier Docker	26
5.2.2	Coté Chatbot	26
5.2.2.1	Ajouter une catégorie	26
5.2.2.2	Ajouter un produit	27
5.2.2.3	Ajouter/Supprimer un produit	27
5.3	Apports	28
5.3.1	Apports au niveau des connaissances techniques	28
5.3.2	Apports au niveau de la conception et du développement	28
5.4	Conclusion	28
6	Conclusion	29

Table des figures

1.1	Image	1
1.2	Image : Principe de fonctionnement	2
1.3	Image : Rapidité	3
1.4	Image : Fiabilité	3
1.5	Figure : Collaboration	3
1.6	Figure : Sécurité	3
1.7	Image : Évolutivité	3
2.1	Figure : Rest API	5
2.2	Figure : Fonctionnement des APIs	6
2.3	Figure : Méthodes HTTP	7
4.1	Logo UML	13
4.2	Figure : Cas d'utilisation : Gestion des produits	15
4.3	Figure : Cas d'utilisation : Gestion des catégories	15
4.4	Figure : Diagramme de classe	16
4.5	Figure : Choisir des produits	17
4.6	Figure : Séquence de Payer une commande	18
4.7	Figure : Séquence d'ajouter un Produit	18
5.1	Logo JAVA	19
5.2	Logo JAVA	19
5.3	Logo JavaScript	20
5.4	Logo TypeScript	20
5.5	Logo HTML et CSS	20
5.6	Logo NLTK	21
5.7	Logo Numpy	21
5.8	Logo TensorFlow	21
5.9	Logo Atom	22
5.10	Logo mySQL	22
5.11	Logo mySQL	22
5.12	Logo Angular	23
5.13	Logo Spring Boot	23
5.14	Interface : Chercher une montre	24
5.15	Interface : Consulter le panier	24
5.16	Interface : Ajouter une adresse de paiement	25
5.17	Interface : Ajouter une adresse de livraison	25
5.18	Fichier Docker	26
5.19	Figure : Ajout d'une catégorie	26
5.20	Figure : Ajout d'un produit	27
5.21	Figure : Ajout/Suppression d'un produit	27

INTRODUCTION

Des ventes de mains en mains, vers des ventes virtuelles, passent les priorités des opérations de ventes des biens et des services, ce qui nous rend obligés de donner plus d'importance à la vente électronique.

Les boutiques en ligne sont depuis des années, largement conseillés pour les sociétés qui se basent sur la vente des produits et même des services Ces types de sites web représentent un dispositif global fournissant aux clients un pont de passage à l'ensemble des informations, des produits, et des services à partir d'un portail unique en rapport avec son activité.

Les sites de vente en ligne permettent aux clients de profiter d'une foire virtuelle disponible est quotidiennement mise à jours sans la moindre contrainte, ce qui leur permettrai de ne jamais rater les coups de cœur, ainsi Une foire sans problèmes de distance géographique, ni d'horaire de travail ni de disponibilité de transport. D'une autre part ces sites offrent à la société de profiter de cette espace pour exposer ses produits à une plus large base de clientèle.

Chapitre 1

Contexte Générale du Projet

Ce chapitre a pour objectif de situer le projet dans son contexte général, à savoir les caractéristiques de notre plateforme qui nous a inspiré pour sa création et de décrire les objectifs à atteindre.

1.1 Objectif du projet

L'objectif de ce projet est concevoir et développer un site web commercial qui doit permettre l'inscription des visiteurs pour devenir clients, le suivi des commandes effectuée, la gestion des paiements en lignes et le suivi des livraisons..



FIGURE 1.1 – Image

1.2 Plan de travail

1.2.1 Organisation du rapport :

Pour un bon travail il nous faut un rapport bien structuré qui peut être exploité après la mise en place de ce site, pour cela nous allons organiser notre présent rapport de la manière suivante :

— Dans ce premier chapitre « **Contexte Générale du Projet** », nous allons mettre notre projet dans son cadre général en définissant l’objectif du projet.

— Dans le deuxième chapitre intitulé « **Etude de l’existant et spécification des besoins** » nous allons en premier lieu, étudier les Procédures de vente utilisées actuellement en relevant les manques et les insuffisances et proposant les solutions convenables. En deuxième lieu, nous précisons les principales solutions offertes par notre projet en tenant compte de ses besoins fonctionnels et non fonctionnels. Et enfin, nous présentons le contexte global de notre projet.

— Dans le troisième chapitre intitulé « **Conception** » nous abordons la phase de conception. Nous spécifions d’abord la méthode de conception adaptée, après, nous présentons les différents diagrammes de notre site web.

— Enfin et au niveau du quatrième et dernier chapitre intitulé « **Réalisation** », nous allons présenter notre site web, en mentionnant les différents environnements de travail matériels et logiciels utilisés pour entamer le projet, ainsi qu’en citant les principales interfaces réalisées.

1.3 Méthodologie de travail : DevOps

1.3.1 Définition

DevOps est une combinaison de philosophies culturelles, de pratiques et d’outils qui améliore la capacité d’une entreprise à livrer des applications et des services à un rythme élevé. Il permet de faire évoluer et d’optimiser les produits plus rapidement que les entreprises utilisant des processus traditionnels de développement de logiciels et de gestion de l’infrastructure. Cette vitesse permet aux entreprises de mieux servir leurs clients et de gagner en compétitivité.

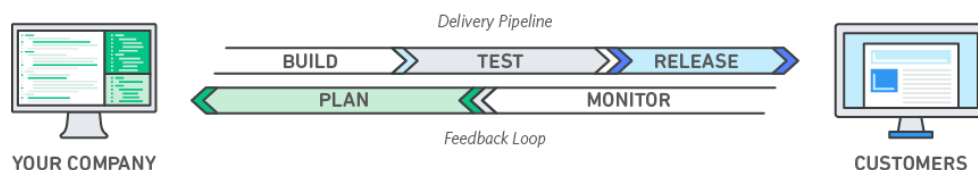


FIGURE 1.2 – Image : Principe de fonctionnement

1.3.2 Fonctionnement de DevOps

Dans un modèle DevOps, les équipes de développement et d’opérations ne sont plus isolées. Il arrive qu’elles soient fusionnées en une seule et même équipe. Les ingénieurs qui la composent travaillent alors sur tout le cycle de vie d’une application, de la création à l’exploitation, en passant par les tests et le déploiement, et développent toute une gamme de compétences liées à différentes fonctions.

1.3.3 Les avantages de DevOps



FIGURE 1.3 – Image : Rapidité



FIGURE 1.4 – Image : Fiabilité



FIGURE 1.5 – Figure : Collaboration



FIGURE 1.6 – Figure : Sécurité

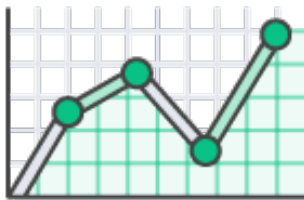


FIGURE 1.7 – Image : Évolutivité

1.4 Conclusion

Dans ce chapitre nous avons défini la problématique qui est d'implémenter un système de recommandation qui mène à prédire des recommandations qu'un utilisateur peut aimer ainsi que l'objectif principal du projet qui est la mise en œuvre de cet application web définie dans la problématique.

Chapitre 2

Architecture Micro-Service

2.1 Définition :

Afin qu'un client Web, tel qu'un client Angular , puisse interagir avec le Back-End, il a été décidé de créer un RESTful API. Ainsi, toutes opérations exécutées par le client auront un point d'accès à l'API.

An API (interface de programme d'application) est un ensemble de règles permettant à dif-

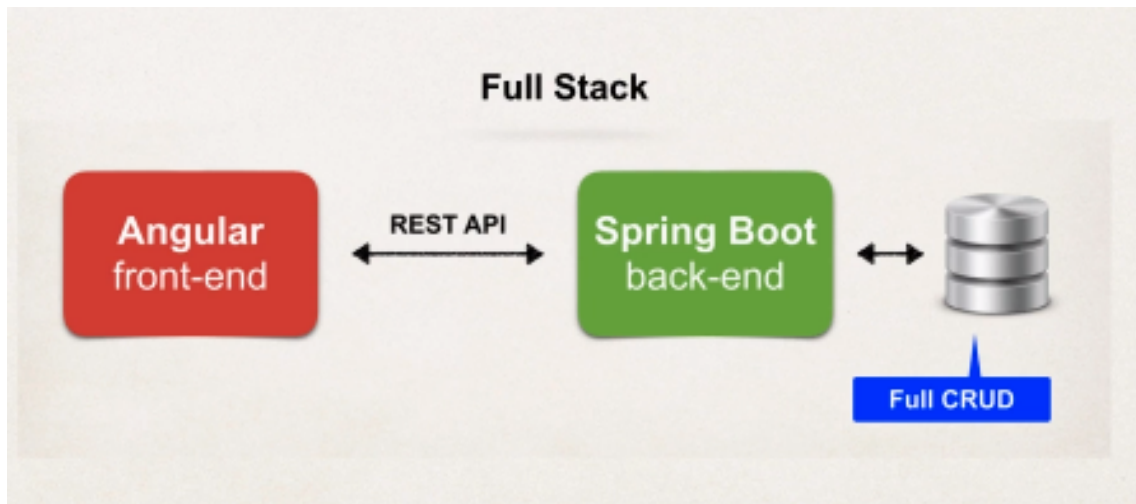


FIGURE 2.1 – Figure : Rest API

férents programmes de communiquer entre eux. Il décrit la manière appropriée pour un développeur de logiciels de composer un programme sur un serveur qui communique avec diverses applications clientes.

2.2 Explication de la définition de l'API REST

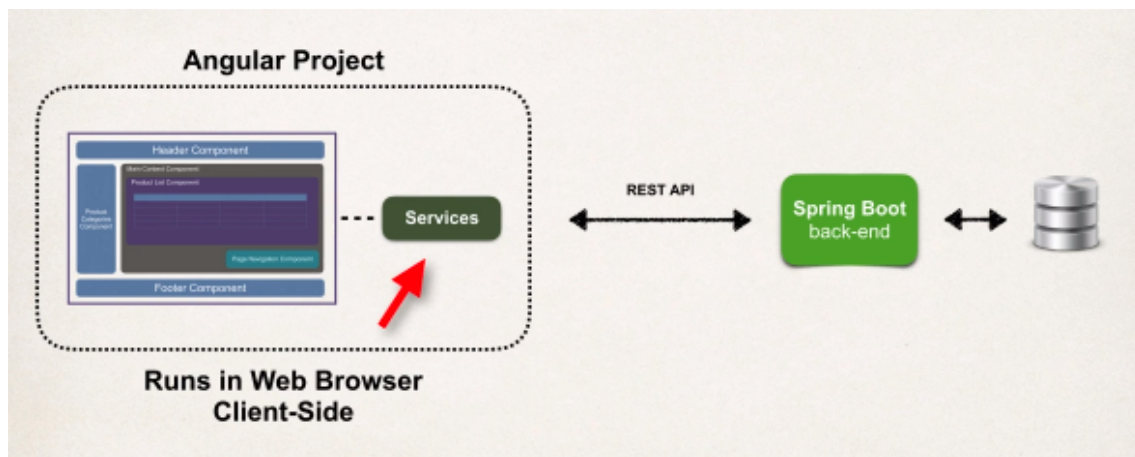


FIGURE 2.2 – Figure : Fonctionnement des APIs

REST est un style architectural et une méthodologie fréquemment utilisés dans le développement de services Internet, tels que les systèmes hypermédias distribués.

REST signifie REpresentational State Transfer, ce qui signifie que lorsqu'une API REST est appelée, le serveur transfère la représentation des ressources demandées à l'état au client.

Par exemple, lorsqu'un développeur demande à l'API Twitter de récupérer l'objet d'un utilisateur (une ressource), l'API renvoie l'état de cet utilisateur, son nom, ses abonnés et les publications partagées sur Twitter.

Cette représentation de l'état peut être au format JSON, XML ou HTML.

Actuellement, plusieurs applications et projets utilisent des API REST pour transférer des données, et les entreprises adoptent de plus en plus les services Web RESTful pour profiter d'une croissance horizontale.

2.3 Comment fonctionnent les API REST ?

REST détermine la structure du API. Les développeurs sont tenus de respecter un certain ensemble de règles lors de la construction de l'API. Par exemple, une règle stipule que la liaison à une URL doit renvoyer des informations.

Chaque URL est connue sous le nom de demande, tandis que les données renvoyées sont appelées réponse.

L'API REST décompose une transaction pour générer une séquence de petits composants. Chaque composant aborde un aspect fondamental spécifique de la transaction. Cette modularité en fait une approche de développement flexible.

Une API REST exploite les méthodes HTTP décrites par le Protocole RFC 2616. Il utilise les requêtes suivantes :

- **ÉCONOMISEZ** récupérer des données
- **PUT** pour modifier l'état des données (comme un objet, un fichier ou un bloc)
- **POSTEZ** pour créer des données
- **EFFACER** méthodes pour l'éliminer.

HTTP Method		CRUD Action
POST	/products	<u>C</u> reate a new product
GET	/products	<u>R</u> ead a list of products
GET	/products/{id}	<u>R</u> ead a single product
PUT	/products/{id}	<u>U</u> pdate an existing product
DELETE	/products/{id}	<u>D</u> eleme an existing product

FIGURE 2.3 – Figure : Méthodes HTTP

2.4 Mots clés :

Client : Le client est le matériel ou le logiciel qui utilise l'API rendue accessible par le serveur. Par exemple, lorsque vous visitez le site Web de Facebook, votre navigateur est le client qui appelle l'API Facebook et utilise les données renvoyées pour afficher des informations sur votre écran.

Ressource : Une ressource peut être n'importe quel objet sur lequel l'API peut offrir des informations. Par exemple, dans le cas d'une API Twitter, une ressource peut être un utilisateur, un hashtag ou tout autre type de média comme une image. Chaque ressource a un identifiant distinct qui peut être un nom ou un numéro.

La ressource est la principale abstraction des informations dans REST. L'API REST utilise un identifiant de ressource pour reconnaître la ressource spécifique impliquée dans la communication entre les différents éléments.

2.5 Pourquoi devriez-vous utiliser l'API REST ?

Évolutivité : L'API REST offre une grande évolutivité. Comme les clients et les serveurs sont séparés, le produit peut être mis à l'échelle par l'équipe de développeurs sans trop de problèmes.

Facilité : Il est également plus facile d'intégrer REST aux sites actuels sans refactoriser l'infrastructure du site Web. Cela permet aux développeurs de travailler plus rapidement au lieu de passer du temps à retravailler sur un site Web à partir de zéro. Comme alternative, ils peuvent simplement ajouter des fonctionnalités supplémentaires.

Flexibilité et portabilité Grâce aux API REST, les utilisateurs peuvent facilement communiquer même si le client et les serveurs sont hébergés sur différents serveurs, offrant un avantage important du point de vue de la gestion.

Indépendance Grâce à la séparation entre le client et le serveur, le protocole REST permet aux développements dans les différentes zones d'un projet de se produire de manière autonome. De plus, l'API REST est ajustable à la syntaxe et à la plate-forme opérationnelles, offrant la possibilité de tester de nombreux environnements pendant le développement.

2.6 Directeur de la conception d'API REST : Serveur / Client

Ce principe fonctionne sur le concept selon lequel le client et le serveur doivent être isolés l'un de l'autre et autorisés à se développer indépendamment. De cette façon, vous pouvez améliorer la gestion sur de nombreuses plates-formes et augmenter l'évolutivité en rationalisant les composants du serveur car les problèmes d'interface utilisateur sont distincts des problèmes de stockage de données.

2.7 Conclusion

Une définition d'API REST permet d'étendre les fonctionnalités du client en téléchargeant et en implémentant le codage sous la forme d'applets ou de scripts. Cela rationalise les clients en diminuant le nombre de fonctionnalités essentielles à être pré-implémentées.

La plupart du temps, un serveur renvoie une représentation statique des ressources au format XML ou JSON. Mais en cas de besoin, les serveurs peuvent fournir du code exécutable au client.

Chapitre 3

Etude de l'existant spécification des besoins

3.1 Introduction

Dans ce troisième chapitre, nous allons mettre le sujet dans son cadre général. Par la suite, nous aborderons l'étude de la manière de vente actuelle, suivie d'une critique pour pouvoir concentrer sur les problèmes à résoudre pendant la réalisation de notre projet.

Ainsi, ce chapitre présente un ensemble des besoins fonctionnels et autres non fonctionnels.

3.2 Etude de l'existant

3.2.1 Description de l'existant

la manière de présenter et de commercialiser ses produits est divisée en deux étapes principales, la première étape c'est l'exposition des produits par des affiches publicitaires, des dépliants, les spots publicitaires dans les radios et la télévision et aussi par des vitrines qui se trouvent au sein de la société, la deuxième étape consiste à vendre les produits à guichet ou par l'intermédiaire des agents commerciaux.

3.2.2 Critique de l'existant

Depuis sa mise en place la procédure existante atteint ces objectifs avec une fréquence limitée et non extensible voir qu'elle ne concerne qu'un nombre limité de clients qui sont très proches de la société pour pouvoir visiter les vitrines, voir les produits exposés et savoir la disponibilité de ces derniers ainsi que leurs prix et leurs caractéristiques techniques, tout ça représente une entrave devant la commercialisation des produits.

3.2.3 Solution proposée

Afin de pallier aux défaillances, nous proposons d’informatiser la commercialisation de nos produits par la création d’une boutique virtuelle sur Internet. Il nous est indispensable de préciser à cette étape que notre projet de fin d’étude prendra en considération toutes ces contraintes en essayant de présenter les solutions nécessaires tout en respectant les règles des jeux d’un site web tels que la simplicité de navigation entre les pages, la bonne ergonomie et la sécurité des données confidentielles des clients.

3.3 Etude des besoins

Dans cette section du chapitre, nous nous intéressons aux besoins des utilisateurs traités dans notre projet c’est à dire l’inscription du client, le choix des produits, le lancement des commandes enfin la confirmation et donc le paiement en ligne à travers les spécifications fonctionnelles et non fonctionnelles pour aboutir à un site de qualité qui répond aux besoins des clients.

3.3.1 Besoins fonctionnels

Les besoins fonctionnels se présentent en huit grandes parties :

- Exposition des produits ainsi que leurs prix et caractéristiques.
- Inscription des clients.
- Ajout des produits choisis au panier.
- Choix du mode de livraison.
- Choix de la boutique de livraison.
- Confirmation de la commande.
- Le paiement en ligne.
- Confirmation de l’opération d’achat et la réception de la facture.

3.3.1.1 L'exposition des produits :

Notre site doit disposer d'une vitrine virtuelle à travers laquelle le client peut consulter une grande variété des produits, il sera donc indispensable d'y présenter les prix et les caractéristiques techniques de chaque produit pour faciliter la sélection du produit à acheter.

3.3.1.2 L'inscription du client :

Jusqu'à ce stade, le client est toujours anonyme mais pour pouvoir passer à un stade plus rigoureux, il faut qu'il s'inscrive, ce la se fait uniquement pour la première commande mais après, notre client peut s'authentifier avec son E-mail et son mot de passe pour passer d'autres commandes.

3.3.1.3 Ajout des produits au panier :

Après le choix d'un produit le client doit mentionner la quantité qui s'ajoute automatiquement à son panier avec le prix unitaire et le prix total.

3.3.1.4 Mode de livraison :

Un client qui a déjà confirmé sa commande il est libre de choisir le mode de livraison de sa marchandise soit à domicile ou chez une boutique selon une liste de choix mentionnée sur notre site web.

3.3.1.5 Boutique de livraison :

Si le mode de livraison choisi est la boutique il faut que le client indique cette boutique avec une précision qui permet aux livreurs d'être sûrs que la marchandise sera dans le bon lieu et dans les rendez-vous, ayant une panoplie de boutiques réelles, le client pourra choisir la plus proche.

3.3.1.6 La livraison à domicile :

En choisissant cette option comme mode de livraison, le client devrait remplir soigneusement un formulaire contenant les informations nécessaires telles que :
- Le nom du destinataire qui peut être le client même ou une autre personne. - L'adresse précise de livraison. - Le numéro de la pièce d'identité du destinataire. - Le jour et l'heure de la livraison estimés.

3.3.1.7 La confirmation de la commande :

Jusqu'à cette phase on a un client, une commande et une adresse de livraison le chemin maintenant est plus clair, la commande ne passera qu'après la validation de toutes les informations qui sont affichées dans une seule interface avant de passer à la phase de paiement.

3.3.1.8 Le paiement :

C'est une phase très sensible, pour cela il faut qu'elle soit très sécurisée, pour terminer la procédure de paiement avec succès le client doit choisir un type de carte dans une liste de choix des cartes proposées sur notre site web, indiquer le numéro de sa carte et sa valeur de vérification dite CVV.

3.3.1.9 La fin de l'opération d'achat :

La page finale représente un petit message de remerciement à nos clients avec une idée sur l'adresse, la date, le temps de la livraison en question et bien sur la possibilité d'imprimer la facture du client.

3.4 Conclusion

Dans ce chapitre on a présenté une étude du système existant, les lacunes qu'il comprend ainsi que les solutions que nous proposons pour pallier ces problèmes, nous avons aussi cité les besoins fonctionnels et non fonctionnels qui sont indispensables pour mieux faciliter le travail à réaliser.

Dans le chapitre suivant nous allons aborder l'étude conceptuelle de notre site, tout en mentionnant tous les scénarios possibles, les acteurs, les diagrammes ...

Chapitre 4

Conception

Dans ce chapitre, nous étendrons la représentation des diagrammes des cas d'utilisation effectués au niveau de l'analyse des besoins puis l'élaboration du modèle conceptuel des données et on termine par le modèle logique de données.

4.1 Langage de modélisation



FIGURE 4.1 – Logo UML

UML est un langage de modélisation graphique. Il est apparu dans le monde du génie logiciel, dans le cadre de la conception orientée objet. Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes. En effet, l'UML nous permet une meilleure conception

du côté de l'application avec ses notions d'objets et de classes, et nous donne une décomposition claire et simple afin de dégager les entités et les classes nécessaires.

4.2 Diagrammes des Cas d'utilisation

4.2.1 Définition

Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit.

Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Ainsi ces cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

4.2.2 Les acteurs de notre projet

Le Client : cet acteur est un visiteur ayant déjà créé un compte sur notre site, il peut donc suivre le processus d'achat des produits en toute sécurité sachant que notre système doit être l'unique responsable de la confidentialité des données personnelles de ses clients.

L'administrateur : pour les sites web on l'appelle généralement « le webmaster ». C'est celui qui assure le dynamisme du site et veille sur les mises à jour des produits, de leurs prix, de leurs disponibilités, de la gestion des paiements et la gestion des livraisons.

4.2.3 Nos diagrammes des cas :

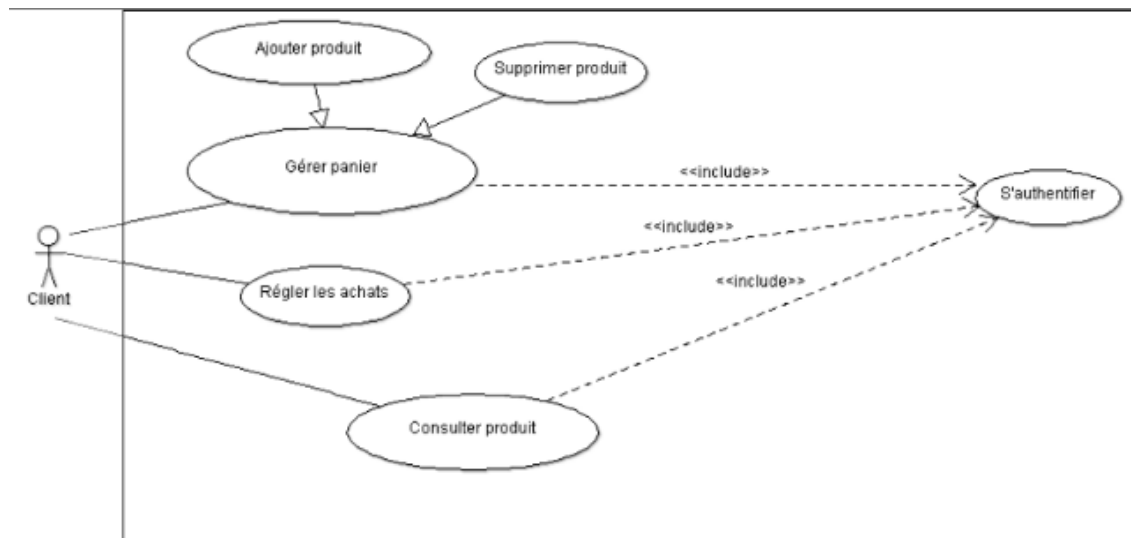


FIGURE 4.2 – Figure : Cas d'utilisation : Gestion des produits

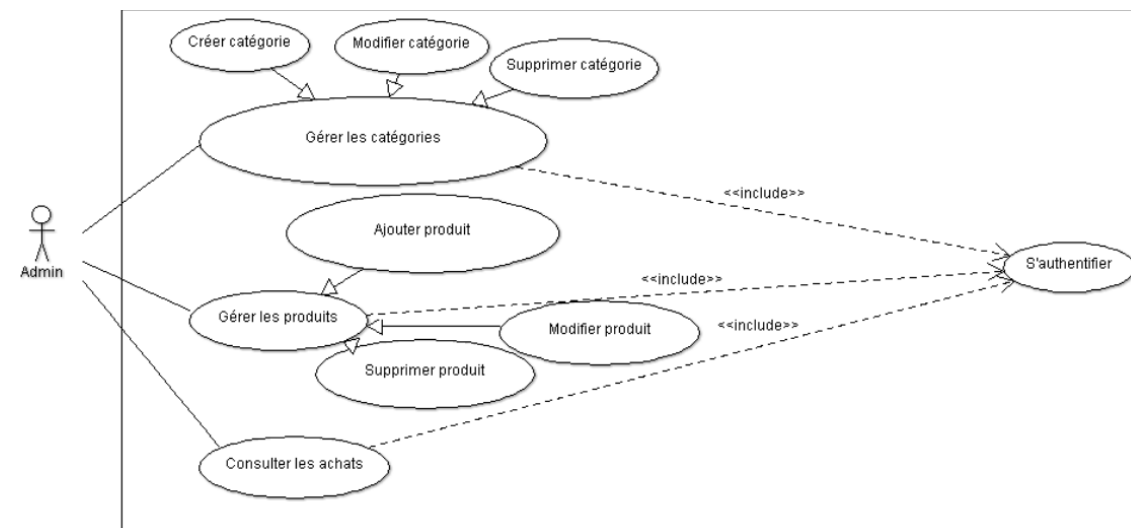


FIGURE 4.3 – Figure : Cas d'utilisation : Gestions des catégories

4.3 Diagramme de Classe

4.3.1 Définition

Un diagramme de classes UML décrit les structures d'objets et d'informations utilisées sur notre site web, à la fois en interne et en communication avec ses utilisateurs.

Il décrit les informations sans faire référence à une implémentation particulière. Ses classes et relations peuvent être implémentées de nombreuses manières, comme les tables de bases de données, les nœuds XML ou encore les compositions d'objets logiciels.

4.3.2 Notre diagramme des classes

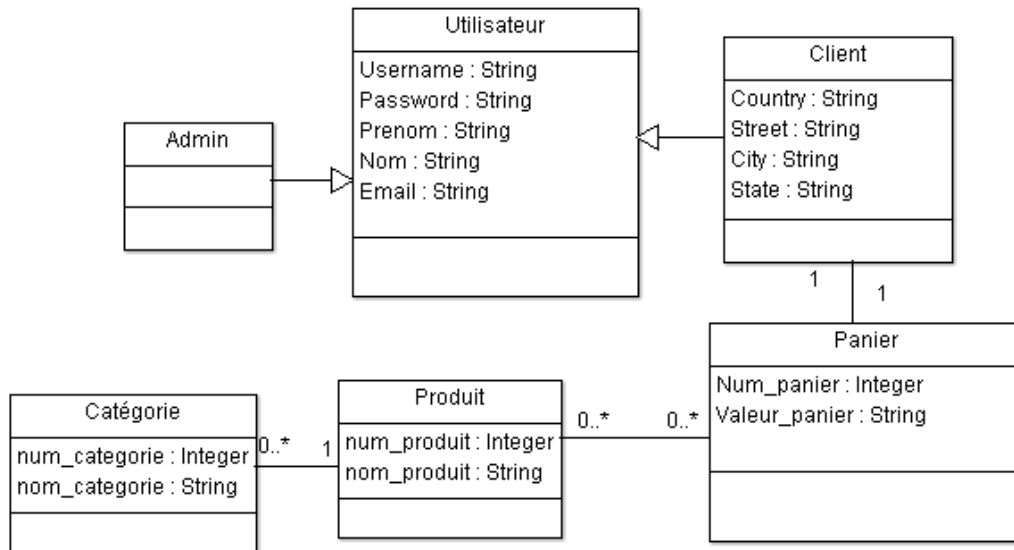


FIGURE 4.4 – Figure : Diagramme de classe

4.4 Diagramme de séquence

4.4.1 Définition

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont.

Les diagrammes de séquences sont organisés en fonction du temps qui s'écoule au fur et à mesure que nous parcourons la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence.

4.4.2 Les diagrammes de séquences de notre site web

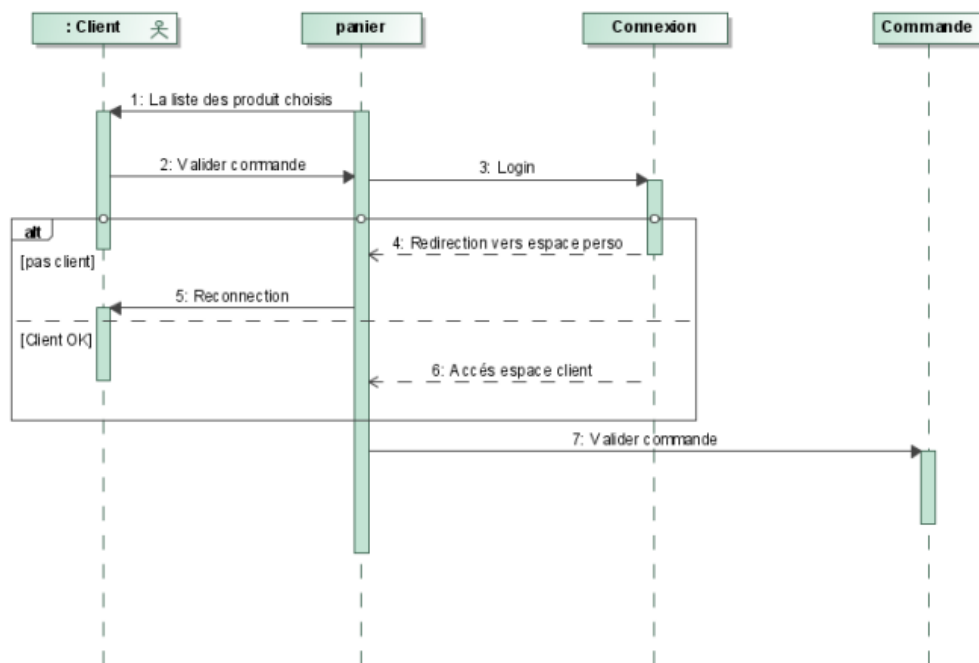


FIGURE 4.5 – Figure : Choisir des produits

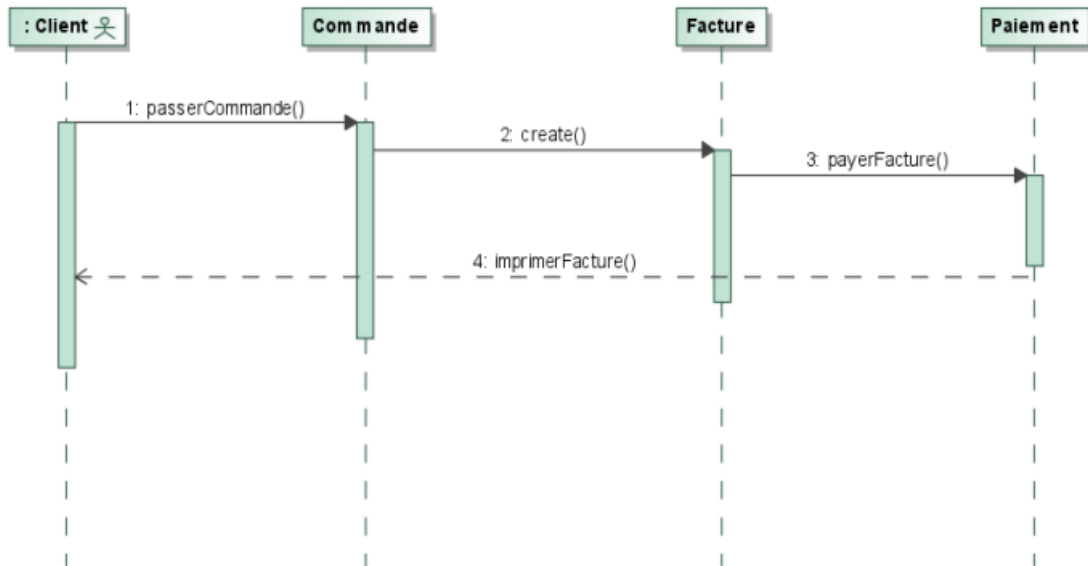


FIGURE 4.6 – Figure : Séquence de Payer une commande



FIGURE 4.7 – Figure : Séquenced 'ajouter un Produit

4.5 Conclusion

Tout au long de ce chapitre, nous avons mis notre projet dans son cadre général, à savoir la présentation du projet et ses objectifs, et nous allons spécifier dans le prochain chapitre les différents besoins auxquels doit répondre notre application.

Dans le chapitre suivant nous allons aborder la dernière partie qui représente la partie réalisation de notre site web, en se basant sur les mécanismes et les solutions déterminés dans la phase de conception.

Chapitre 5

Réalisation du projet

Dans ce chapitre nous allons élaborer la phase de réalisation de l'application , en vous montrant la partie de développement Web que nous avons implementé en utilisant Django, ainsi cette partie présente l'interface homme-machine.

5.1 Environnement de développement

5.1.1 Langages de développement

5.1.1.1 Java



FIGURE 5.1 – Logo JAVA

Java est une technique informatique développée initialement par Sun Microsystems puis acquise par Oracle suite au rachat de l'entreprise. Défini à l'origine comme un langage de programmation, Java a évolué pour devenir un ensemble cohérent d'éléments techniques et non techniques.

5.1.1.2 Python



FIGURE 5.2 – Logo PYTHON

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

5.1.1.3 JavaScript



FIGURE 5.3 – Logo JavaScript

JavaScript : Un langage de programmation de scripts principalement employé dans les pages web interactives

5.1.1.4 TypeScript



FIGURE 5.4 – Logo TypeScript

TypeScript : Un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript.

5.1.1.5 HTML & CSS



FIGURE 5.5 – Logo HTML et CSS

HTML : HyperText Markup Language, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom.

CSS : Les feuilles de style en cascade, ils forment un langage informatique qui décrit la présentation des documents HTML et XML.

5.1.2 Librairies

5.1.2.1 NLTK



FIGURE 5.6 – Logo NLTK

Natural Language Toolkit est une bibliothèque logicielle en Python permettant un traitement automatique des langues, développée par Steven Bird et Edward Loper du département d'informatique de l'université de Pennsylvanie.

5.1.2.2 Numpy



FIGURE 5.7 – Logo Numpy

NumPy est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

5.1.2.3 TensorFlow



FIGURE 5.8 – Logo TensorFlow

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Le code source a été ouvert le 9 novembre 2015 par Google et publié sous licence Apache. Il est fondé sur l'infrastructure DistBelief, initiée par Google en 2011, et est doté d'une interface pour Python et Julia.

5.1.3 Logiciels

5.1.3.1 Atom



FIGURE 5.9 – Logo Atom

Est un éditeur de texte libre pour MacOS, GNU/Linux et Windows développé par GitHub. La plupart des extensions sont sous licence libre et sont maintenues par la communauté.

5.1.3.2 mySQL Admin



FIGURE 5.10 – Logo mySQL

MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire.

5.1.3.3 Docker



FIGURE 5.11 – Logo mySQL

Docker est un logiciel libre permettant de lancer des applications dans des conteneurs logiciels.

5.1.4 Frameworks

5.1.4.1 Framework Angular : Front-End



FIGURE 5.12 – Logo Angular

Angular est un cadriciel côté client, open source, basé sur TypeScript, et co-dirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète de AngularJS, cadriciel construit par la même équipe.

5.1.4.2 Framework Spring Boot : Backend



FIGURE 5.13 – Logo Spring Boot

Spring Boot est un framework qui facilite le développement d'applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en jar , totalement autonome.

5.2 La partie de la réalisation

5.2.1 Coté Application web

5.2.1.1 Interface : Chercher une montre

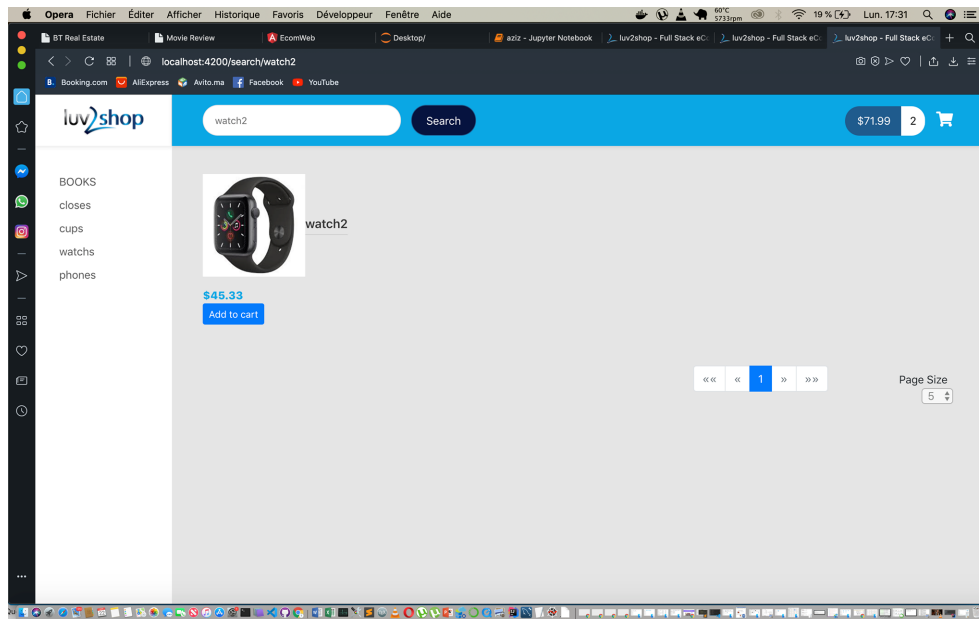


FIGURE 5.14 – Interface : Chercher une montre

5.2.1.2 Interface : Consulter le panier

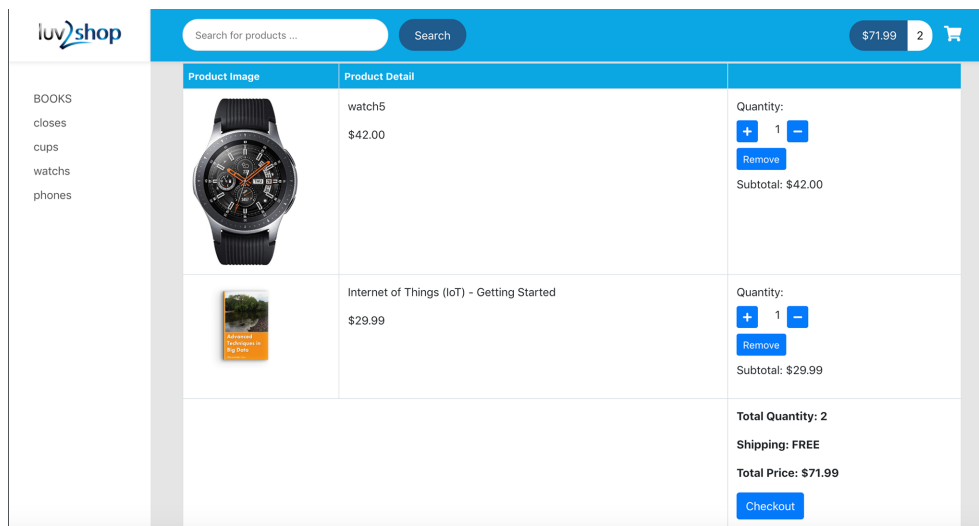


FIGURE 5.15 – Interface : Consulter le panier

5.2.1.3 Interface : Ajouter une adresse de paiement

The screenshot shows the 'luvshop' application interface. The top navigation bar is blue with the 'luvshop' logo on the left, a search bar containing 'watch2' and a 'Search' button in the center, and a shopping cart icon on the right displaying '\$71.99' and '2' items. A left sidebar lists categories: BOOKS, closes, cups, watches, and phones. The main content area is titled 'Billing Address' and contains the following form fields: Country (dropdown), Street (text input), City (text input), State (dropdown), and Zip Code (text input). Below this, the 'Credit Card' section includes Card Type (dropdown), Name on Card (text input), and Card Number (text input).

FIGURE 5.16 – Interface : Ajouter une adresse de paiement

5.2.1.4 Interface : Ajouter une adresse de livraison

The screenshot shows the 'luvshop' application interface. The top navigation bar is blue with the 'luvshop' logo on the left, a search bar containing 'watch2' and a 'Search' button in the center, and a shopping cart icon on the right displaying '\$71.99' and '2' items. A left sidebar lists categories: BOOKS, closes, cups, watches, and phones. The main content area is titled 'Customer' and contains the following form fields: First Name (text input), Last Name (text input), and Email (text input). Below this, the 'Shipping Address' section includes Country (dropdown), Street (text input), City (text input), and State (dropdown).

FIGURE 5.17 – Interface : Ajouter une adresse de livraison

5.2.1.5 Fichier Docker

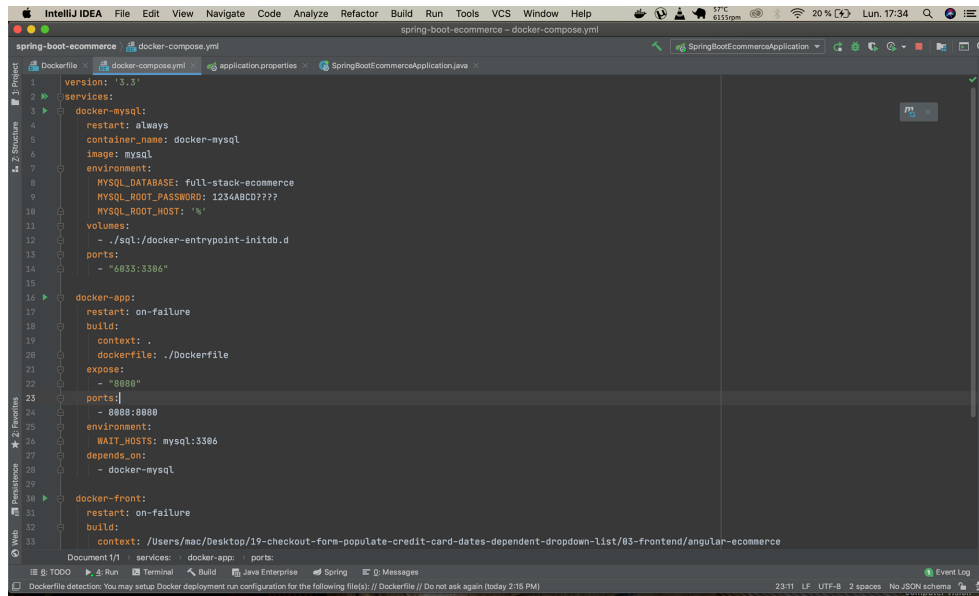


FIGURE 5.18 – Fichier Docker

5.2.2 Coté Chatbot

5.2.2.1 Ajouter une catégorie

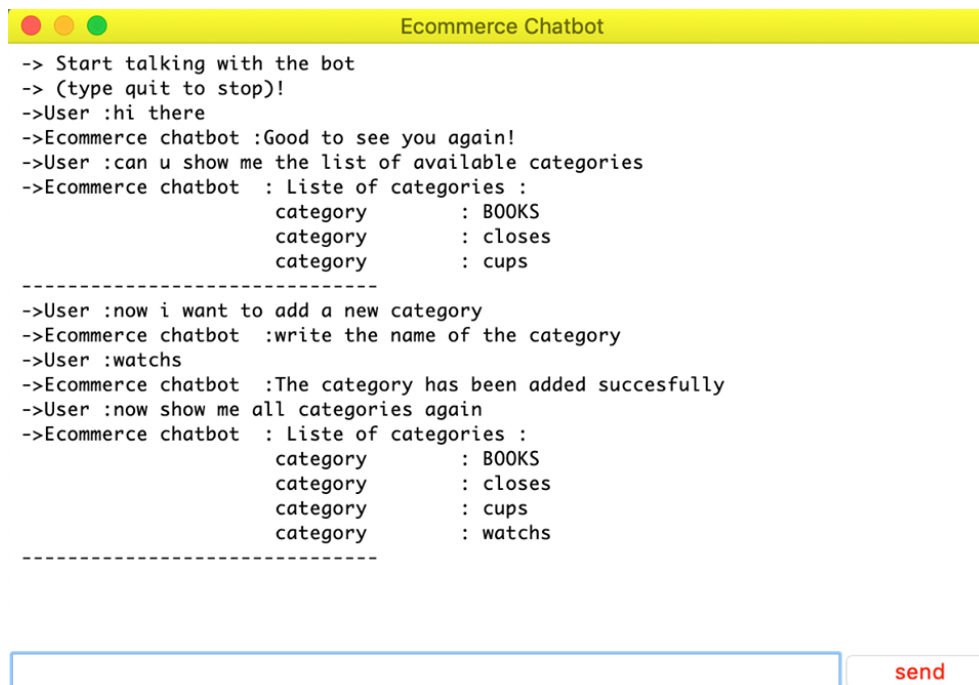


FIGURE 5.19 – Figure : Ajout d'une categorie

5.2.2.2 Ajouter un produit

Ecommerce Chatbot

```
-----
->User :now how about adding a new product
->Ecommerce chatbot :give the product's title
->User :black watch
->Ecommerce chatbot :add the description of the black watch product
->User :a 2020 version one
->Ecommerce chatbot :add the name of category of the black watch product
->User :watches
->Ecommerce chatbot :add the unit price of the black watch product
->User :50
->Ecommerce chatbot :add the units in stock of the black watch product
->User :100
->Ecommerce chatbot :The product has been added successfully
->User :display a product informations
->Ecommerce chatbot :give me the title
->User :Spring Framework Tutorial
->Ecommerce chatbot :
Product      : Spring Framework Tutorial
Description  : Learn Spring
unit price   : 29.99
unit in stock : 100
Category     : BOOKS
```

send

FIGURE 5.20 – Figure : Ajout d'un produit

5.2.2.3 Ajouter/Supprimer un produit

Ecommerce Chatbot

```
->Ecommerce chatbot :add the name of category of the black watch product
->User :watches
->Ecommerce chatbot :add the unit price of the black watch product
->User :50
->Ecommerce chatbot :add the units in stock of the black watch product
->User :100
->Ecommerce chatbot :The product has been added successfully
->User :display a product informations
->Ecommerce chatbot :give me the title
->User :Spring Framework Tutorial
->Ecommerce chatbot :
Product      : Spring Framework Tutorial
Description  : Learn Spring
unit price   : 29.99
unit in stock : 100
Category     : BOOKS

->User :now how about deleting a product
->Ecommerce chatbot :give the product's title
->User :black watch
->Ecommerce chatbot : Product deleted successfully
```

send

FIGURE 5.21 – Figure : Ajout/Suppression d'un produit

5.3 Apports

Ce projet de fin d'études nous a permis de se familiariser avec un certain nombre de concepts tout en se basant sur nos connaissances acquises au cours de notre formation au sein de notre institut. Ainsi, le fait d'avoir travaillé avec nos encadreurs, les personnels de la sock. nous a mis sur la voie professionnelle. Notre projet de fin d'étude a été une occasion, pour développer et exercer nos capacités d'observation, d'analyse, de conception, de développement et de rédaction.

5.3.1 Apports au niveau des connaissances techniques

Ce projet de fin d'étude nous a permis de :

- Mettre en oeuvre les notions et les connaissances acquises au sein d. modules " Methodologie de conception des bases de données "
- Manipuler des logiciels, qui nous ont aidé pour développer l'application.
- Améliorer nos connaissances théoriques concernant la communication client/serveur.
- Apprendre comment gérer un projet.

5.3.2 Apports au niveau de la conception et du développement

Au niveau de la conception et du développement, cette étude nous a permis de :

- Mener une conception représentée avec le langage UML.
- Apprendre à maîtriser la méthodologie de conception et de développement des applications clientes.
- Maîtriser les étapes de développement et de conception d'une application Web.

5.4 Conclusion

Dans ce chapitre, nous avons pu présenter l'environnement et le processus de développement. Nous avons exposé ainsi le résultat de développement à l'aide des aperçus d'écran. Nous avons clôturé par une validation du travail réalisé.

Chapitre 6

Conclusion

Nous sommes appelés dans ce travail de concevoir et réaliser une boutique virtuelle pour la vente en ligne des produits.

Au cours de développement de ce projet, nous avons appris à rechercher des informations et à faire une synthèse, notamment pour ce rapport, ce qui nous a permis de progresser rapidement dans l'apprentissage d'une technologie que nous ne connaissons pas.

En termes d'humain ce projet, une réelle opportunité de vivre en étroite collaboration dans un groupe de travail. Cela nous a permis d'améliorer nos compétences en communication.

Pour arriver à ce point, nous avons procédé en trois étapes principales. Nous avons commencé par une étude des besoins. Nous présentons les besoins fonctionnels et non fonctionnels, ainsi qu'un diagramme global des cas d'utilisation, des diagrammes de séquence et d'autres activités. Nous sommes ensuite passés à l'étude conceptuelle et organisationnelle.

Finalement, nous avons l'architecture générale de l'application et nous avons déployé les interfaces graphiques de la solution que nous avons implémentée, Les objectifs du projet ont été atteints à long terme. En effet, les fonctionnalités sont mentionnées dans les spécifications sur les implémentations et sont fonctionnelles.

Bibliographie

- [1] By Rouâa Ben Hammouda.
<https://fr.slideshare.net/rouaabenhammouda/rapport-conception-et-ralisation-dune-plateforme-social-learning>
- [2] By : Fotolia / Rawf8.
<https://www.proactiveacademy.fr/blog/formation/formation-en-ligne-e-learning/>
- [3] By : Onel Harrison.
<https://moodle.org/?lang=fr>
- [4] By : Unknown.
<https://www.udemy.com>
- [5] B : Kevin Liao
<https://coursera.com>