

République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



RAPPORT DE PROJET DE FIN D'ÉTUDES

**Pour l'obtention du Diplôme de
Licence Nationale en Informatique :
Génie Logiciel et Système d'Information**

Conception et réalisation d'une plateforme Web e-commerce avec un admin Dashboard

Réalisé par: Mohamed Jaouadi

Encadrent Universitaire: Mme.
Thouraya Menyaoui

Encadrent Entreprise: Mr.
Mohamed Amine Ouchari



Année
Universitaire
2023-2024



ÉCOLE
DU GROUPE

GFI
UNIVERSITY

DEDICATION

I would like to dedicate this project to my parents and my sisters for their unwavering support and encouragement throughout my research.

I also extend this dedication to my cousins and friends, who have been a constant source of support and encouragement. I am deeply grateful for their contributions, particularly those I met during extracurricular activities at the university, where I had the chance to enhance both my hard and soft skills.

I cannot express my gratitude enough to all those who have supported me and those that I dearly cherish.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Mrs. Thouraya Menyaoui, for her invaluable guidance and the autonomy she provided throughout my work. Her faith in me and my aspirations has been truly inspiring.

I extend my heartfelt thanks to Mr. Mohamed Amine Ouchari for his steadfast support and trust. His forward-thinking approach and enthusiasm for exploring innovative solutions and cutting-edge technologies have been greatly appreciated.

I am also profoundly grateful to my school division and all my teachers who have motivated and supported me, contributing significantly to my academic success during this period.

Lastly, I would like to offer a sincere thank you to everyone who participated in this project.

"Your most unhappy customers are your greatest source of learning."

— Bill Gates.

Contents

General Introduction	10
1 General Project presentation	11
1.1 Introduction	12
1.2 Host Company	12
1.2.1 STE LINQUBIT	12
1.2.2 Goals of STE LINQUBIT	12
1.3 E-commerce domain	13
1.3.1 Definition	13
1.3.2 Why E-commerce	13
1.3.3 History of E-commerce	13
1.4 About the project	14
1.5 Study of the existing	14
1.5.1 Description	14
1.5.2 Review	16
1.5.3 Solution	16
1.6 Language and working methodologies	17
1.6.1 Working methodologies	17
1.6.2 Modeling language:UML-Unified Modeling Language .	18
1.7 Conclusion	19
2 Architecture Conception	20
2.1 Introduction	22
2.2 Requirements specification	22
2.2.1 Identification of actors	22
2.2.2 Identification of functional needs	22
2.2.3 Identification of non-functional needs	23
2.3 Presentation of use case diagrams	24
2.3.1 Overall use case diagram	24
2.3.2 "sign in" use case diagram	26
2.3.3 "Manage product" use case diagram	28

2.3.4	”Manage client” use case diagram	30
2.3.5	”Manage complaints” use case diagram	32
2.3.6	”Manage categories” use case diagram	33
2.3.7	”Manage cart” use case diagram	35
2.3.8	”Send complaint” use case diagram	37
2.4	Sequence diagrams	39
2.4.1	Sign In sequence Diagram	39
2.4.2	Add product	40
2.4.3	Delete product	41
2.4.4	Delete category	42
2.4.5	Manage complaint	43
2.5	Overall class diagram	44
2.6	Conclusion	45
3	Realization and implementations	46
3.1	Introduction	47
3.2	Technical Study:	47
3.2.1	Software Environment:	47
3.2.2	Tools for Research and Analysis	47
3.2.3	Tools of development	48
3.3	Architectural Style	53
3.4	Design Pattern	54
4	Overview and future work	55
4.1	Introduction	56
4.2	Design	56
4.2.1	Name	56
4.2.2	Logo	56
4.3	Application overview	56
4.3.1	Sign up UI	57
4.3.2	Sign in UI	58
4.3.3	Home page UI	59
4.3.4	Cart UI	60
4.3.5	ComplaintsUI	61
4.3.6	Admin Dashboard	62
4.3.7	Manage Product	63
4.3.8	Add Product	64
4.3.9	Manage Categories	65
4.3.10	Add Categories	66
4.3.11	Manage complaint	67
4.3.12	Manage Order	68

4.4	Futur work	69
4.5	Conclusion	69
General Introduction		70

List of Figures

1.1	STE LINQUBIT	12
1.2	The history of E-commerce	13
1.3	Amazon logo	14
1.4	Screenshot from Amazon	14
1.5	AliExpress logo	15
1.6	Screenshot from AliExpress	15
1.7	Cascade cycle	18
1.8	UML Logo	18
2.1	Overall use case diagram	25
2.2	Comprehensive use case diagram "sign in"	26
2.3	Comprehensive use case diagram "Manage product"	28
2.4	Comprehensive use case diagram "Manage client"	30
2.5	Comprehensive use case diagram "Manage complaints"	32
2.6	Comprehensive use case diagram "Manage categories"	33
2.7	Comprehensive use case diagram "Manage cart"	35
2.8	Comprehensive use case diagram "Send complaint"	37
2.9	Comprehensive Sequence diagram "sign in"	39
2.10	Comprehensive Sequence diagram "Add product"	40
2.11	Comprehensive Sequence diagram "Delete product"	41
2.12	Comprehensive Sequence diagram "delete category"	42
2.13	Comprehensive Sequence diagram "manage complaint"	43
2.14	Comprehensive class diagram	44
3.1	Google meet logo	47
3.2	Visual Studio Code logo	48
3.3	MongoDB	48
3.4	Next.JS	49
3.5	CSS logo	49
3.6	Java Script logo	50
3.7	Node JS logo	50

3.8	npm logo	51
3.9	GitHub logo	51
3.10	StarUML logo	52
3.11	3 Tiers	53
3.12	MVC pattern	54
4.1	Platform logo	56
4.2	Sign up UI	57
4.3	Sign in UI	58
4.4	Home page UI	59
4.5	Client cart UI	60
4.6	Complaints	61
4.7	Admin Dashboard	62
4.8	Manage Product	63
4.9	Add Product	64
4.10	Manage Category	65
4.11	Add Category	66
4.12	Manage Complaint	67
4.13	Manage Order	68

Acronyms

AWS Amazon Web Service. 15

CSS Cascading style sheets. 49

HTML HyperText Markup Language. 53

MVC Model View Controller. 54

npm node package manager. 51

SEO Search Engine Optimisation. 15

UI User Interface. 5, 55, 57–61

UML Unified Modelling Language. 18, 19, 52

VS Visual Studio. 48

General Introduction

Over the past thirty years, online shopping has grown from almost nothing to a booming industry. It is now more economical and convenient than traditional shopping. With online shopping, you avoid limited store hours, long checkout lines, and difficulty finding the right store. You can shop anytime from your computer with great deals available 24/7. This convenience has made online shopping the preferred choice for many people around the world, leading businesses to see e-commerce as essential.

Because of this shift, it is now crucial for businesses to have an e-commerce website to improve their visibility and reach more customers. An e-commerce website allows businesses to showcase and sell their products and services online.

In this context, our final project at Foreign Job aims to create an e-commerce platform called "MJ Store," where users can publish and sell their products online.

Our report is divided into four chapters. In the first chapter, "General Context and Preliminary Study," we introduce the host company, discuss the project context and its issues, analyze the current situation, identify its shortcomings, and propose a reliable solution, also explaining the Scrum method used for project management. The second chapter, "Specification and Needs Analysis," specifies the functional and non-functional requirements and the project's architecture. The third chapter, "Development of the First Release," covers the initial part of the website's development through sprints, including analysis, design, and implementation. The fourth chapter, "Development of the Second Release," continues with the second part of development, also using sprints for analysis, design, and implementation. Finally, we conclude with a summary of our work and suggest future perspectives.

Finally, we conclude with a general conclusion that presents a summary of the work done and opens up some perspectives.

CHAPTER 1

General Project presentation

Contents

1.1	Introduction	12
1.2	Host Company	12
1.2.1	STE LINQUBIT	12
1.2.2	Goals of STE LINQUIBIT	12
1.3	E-commerce domain	13
1.3.1	Definition	13
1.3.2	Why E-commerce	13
1.3.3	History of E-commerce	13
1.4	About the project	14
1.5	Study of the existing	14
1.5.1	Description	14
1.5.2	Review	16
1.5.3	Solution	16
1.6	Language and working methodologies	17
1.6.1	Working methodologies	17
1.6.2	Modeling language:UML-Unified Modeling Lan- guage	18
1.7	Conclusion	19

1.1 Introduction

This first chapter is organized as follows: Presentation of the host organization, the project context and working domain, study and critique of existing solutions, and the definition of the language and methodology of conception.

1.2 Host Company

1.2.1 STE LINQUBIT



Figure 1.1: STE LINQUBIT

LINQUBIT is a dynamic startup dedicated to the creation of innovative web and mobile applications. Founded on a vision to transform client ideas into digital reality, the company thrives in the ever-evolving landscape of digital solutions. LINQUBIT's mission is to provide cutting-edge software development services, enabling businesses to harness the power of technology to achieve their objectives efficiently.

1.2.2 Goals of STE LINQUIBIT

The company's mission centers on bridging the gap between creative ideas and tangible digital solutions. LINQUBIT aspires to be at the forefront of technological innovation, delivering high quality web and mobile applications that cater to the evolving needs of its clients.

The company's vision is to be recognized as a leading player in the field of digital transformation, offering a diverse range of services and solutions that empower businesses and individuals alike.

1.3 E-commerce domain

1.3.1 Definition

E-commerce (electronic commerce) involves the buying and selling of goods and services, or the transfer of funds or data, over an electronic network, usually the Internet.

1.3.2 Why E-commerce

E-commerce has become an important part of modern businesses and digital marketing. It is a popular way to advertise and sell products and services to people all over the world. This makes e-commerce a key part of the business world.

We chose e-commerce because it has many benefits. It helps businesses find new customers, lower costs, and make the customer experience better.

1.3.3 History of E-commerce

[8]E-commerce has a long history that can be traced back to 1969 when CompuServe was established as the first major e-commerce company.

E-Commerce has also integrated newer technologies, including artificial intelligence and virtual reality, to enhance the shopping experience. As a result, e-Commerce has emerged as a crucial component of the global economy, with an increasing number of businesses taking advantage of online platforms to reach a wider customer base.

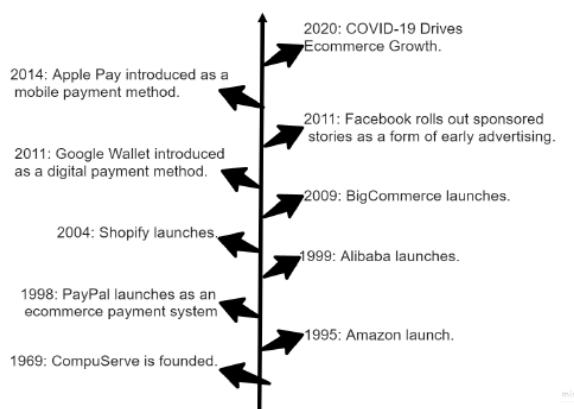


Figure 1.2: The history of E-commerce

1.4 About the project

During this internship, we will develop a comprehensive online store for selling various products, covering all aspects of sales operations. Until the project is completed, our focus will be on achieving customer satisfaction by researching and analyzing the current market offerings, as will be detailed in the future.

1.5 Study of the existing

1.5.1 Description

Due to the huge development that our field of work has witnessed, there are many stores that respond and are partially contain what we seek to achieve.

Amazon



Figure 1.3: Amazon logo

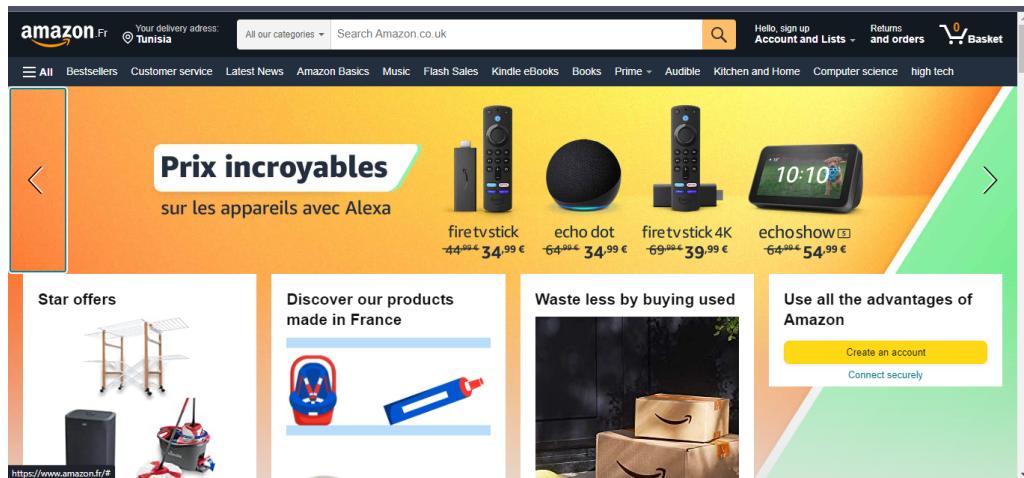


Figure 1.4: Screenshot from Amazon

[2]Amazon is a large-scale platform operates massive e-commerce services scalable infrastructure that focus on offers every customer a personalized shopping experience, all with a user-friendly interface and fast delivery options. Moreover, Amazon provides a reliable cloud computing platform called AWS. But it require some level of technical expertise, and additional services like billing and search engine optimization SEO may require third-party integration.

AliExpress



Figure 1.5: AliExpress logo

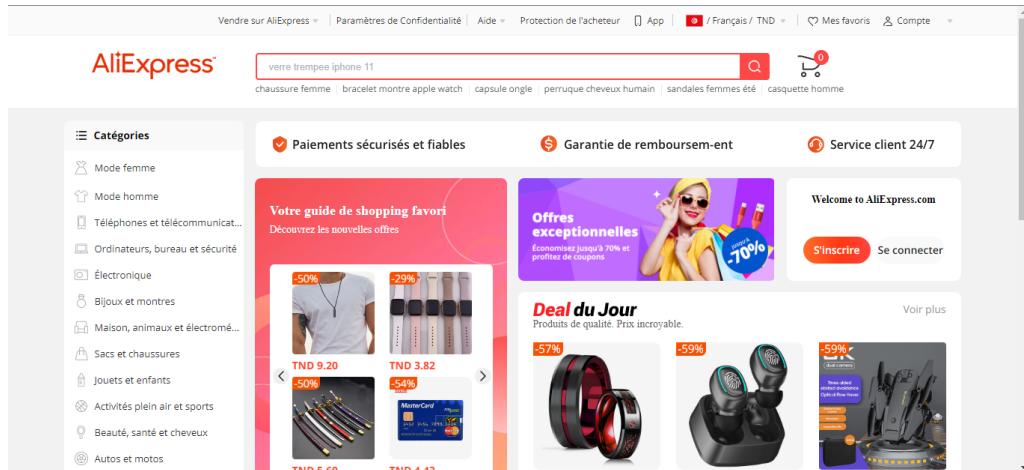


Figure 1.6: Screenshot from AliExpress

[1]AliExpress is the best and recommended option when we talk about product variety and low price at the same time. Also, do not forget a nice user-interface that allows access to everything that the store offers.

1.5.2 Review

In summary, Amazon and AliExpress are two popular online marketplaces that cater to different customer needs. Amazon is a well-established brand known for its seamless shopping experience and reliable cloud computing platform, whereas AliExpress is known for its affordability and wide range of products from various vendors.

Both platforms offer multiple payment methods, but Amazon features a more integrated billing system. Technically, Amazon's build system is more advanced compared to AliExpress. However, AliExpress boasts a comprehensive control system in its store, thanks to the ease and versatility of its tools. Despite these advantages, a significant challenge in e-commerce is the need for manual data entry, which hinders the goal of fully digitizing the process. Ultimately, the choice between Amazon and AliExpress depends on the customer's preferences and priorities, such as brand reputation, delivery time, product quality, and price.

1.5.3 Solution

In response to these challenges, my project aims to optimize the e-commerce process, which has been attempted multiple times before but without success. One of the main reasons for this failure is the significant amount of manual data entry required.

Table 1.1: Comparison table of different applications

Criteria	Amazon	AliExpress	MJ-store
C1-Web application	✓	✓	✓
C2-No adds in the webSite	✗	✗	✓
C3-The WebSite is easy to use	✓	✗	✓
C4-Security	✓	✓	✓

Our platform will handle all aspects of online store management, automating manual processes and giving visibility into warehouses and inventory. It will include tools for managing everything from products to shipping, leading to increased sales, lower costs, and better customer satisfaction.

1.6 Language and working methodologies

In our field of work, Choosing the appropriate methodology affects the In our work, choosing the right method is very important for the quality of the product we create. There are many methods to choose from, and each one is good for different types of projects and ways of working. Below, we will list some of the most popular methods and pick the one that best suits our project.

1.6.1 Working methodologies

[Popular Methodologies]

Design Thinking

[5]Design thinking is a problem-solving methodology that involves a human-centered approach to innovation. It involves empathizing with the user, defining the problem, ideating solutions, prototyping and testing them.Design thinking can be applied to a wide range of problems, from product design and service design to business strategy and social innovation. It is a highly collaborative and iterative process that encourages experimentation and learning from failure.

Waterfall (Cascade)

[3]The Cascade methodology is a widely recognized and traditional project management strategy that utilizes a linear and sequential design approach, where progress flows seamlessly downward along a single path, resembling a cascade. Initially developed for use in the construction and manufacturing sectors, this method thrives in environments with rigid structures and where alterations or experimentation can be expensive. Underlying the Waterfall methodology is the principle that the project team can only advance to the subsequent stage once the current phase has been satisfactorily completed. The key phases involved in this approach include system and software requirements, analysis, design, coding, testing, and operations.

[Adopted methodology]

We will rely on the Cascade methodology to complete working on our project, because it is the best one that suits the way we work.

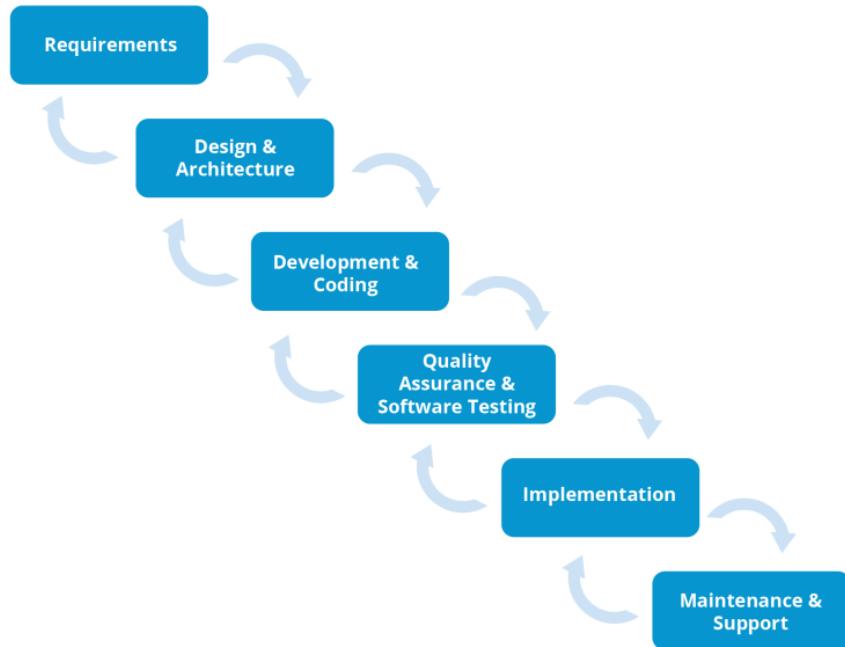


Figure 1.7: Cascade cycle

1.6.2 Modeling language:UML-Unified Modeling Language



Figure 1.8: UML Logo

[15]UML is a standardized, general-purpose visual modeling language used primarily in software engineering. It provides a set of graphical notation

techniques to create visual models of object-oriented software systems. It is a valuable tool for designing, specifying, visualizing, constructing, and documenting software systems.

UML diagrams help in understanding the system architecture, components, and relationships between various parts of the system, as well as the system's behavior and interactions. Using UML can make the software development process more efficient.

1.7 Conclusion

In this first chapter, we focused on situating the project within the broader context. We began by addressing several aspects, starting with an introduction to the host organization, LINQUBIT. Next, we described the context of our internship by identifying the problem at hand and suggesting a solution to tackle the current situation. Finally, we outlined our choice of methodology and the adopted modeling language, before moving on to discuss planning and architecture.

CHAPTER 2

Architecture Conception

Contents

2.1	Introduction	22
2.2	Requirements specification	22
2.2.1	Identification of actors	22
2.2.2	Identification of functional needs	22
2.2.3	Identification of non-functional needs	23
2.3	Presentation of use case diagrams	24
2.3.1	Overall use case diagram	24
2.3.2	”sign in” use case diagram	26
2.3.3	”Manage product” use case diagram	28
2.3.4	”Manage client” use case diagram	30
2.3.5	”Manage complaints” use case diagram	32
2.3.6	”Manage categories” use case diagram	33
2.3.7	”Manage cart” use case diagram	35
2.3.8	”Send complaint” use case diagram	37
2.4	Sequence diagrams	39
2.4.1	Sign In sequence Diagram	39
2.4.2	Add product	40
2.4.3	Delete product	41
2.4.4	Delete category	42
2.4.5	Manage complaint	43
2.5	Overall class diagram	44

2.1 Introduction

the architecture of a web application is like a blueprint that shows how different parts work together. It helps manage the complexity of the application by making it easier for the pieces to communicate and coordinate with each other

2.2 Requirements specification

At this stage, our main focus should be on planning for the project. This means we need to clearly define our goals to understand what we want to achieve.

2.2.1 Identification of actors

An actor represents a role played by an external entity that interacts with the system being modeled. Actors can be human users, other systems, or external devices. In our project which is represented by an e-commerce site, our actors are as follows:

- Internet user

Our website visitor can choose to use the site as a customer or as an agent. This means they can do different things depending on whether they're shopping or managing accounts and transactions.

- Manager

The "manager" inherits all the functionalities of the "agent" and additionally manages both agents and categories.

- Client

The client is responsible for managing their orders, which includes tasks such as placing new orders, tracking existing orders, and managing their order history. Additionally, they have the ability to submit complaints or feedback regarding their experience with the website or the products/services offered.

2.2.2 Identification of functional needs

- Internet user :

- request for administration

- consult the products
- Manager :
 - Authentication: The manager must authenticate to access accounts and benefit from services through a password to access various functionalities.
 - Managing client: viewing accounts and making "CRUD" modifications.
 - Managing orders: viewing orders and making "CRUD" modifications.
 - Managing categories: viewing categories and making "CRUD" modifications.
 - Managing products: viewing products and making "CRUD" modifications.
 - Managing complaints: viewing complaints and making "CRUD" modifications.
- Client :
 - Authentication: The client must authenticate to access accounts and benefit from services through a password to access various functionalities.
 - Managing carts: involves overseeing the process of adding, removing, and updating items within a shopping cart on an e-commerce platform.
 - Send complaints

2.2.3 Identification of non-functional needs

- Security :

The system should provide robust security measures to protect the data, and sensitive business information from unauthorized access, data breaches, and cyber threats.
- Accessibility :

The application should be easily accessible to the staff, the admin and the agency with different devices, ensuring that everyone can use the application effectively.
- Usability :

The user interface should be intuitive, easy-to-navigate, and visually appealing for all types of users, including visitors, clients, suppliers, and admins.

- Performance :

The application should have fast page load times, responsive user interactions, and the ability to handle peak traffic periods without degradation in performance.

2.3 Presentation of use case diagrams

A use case defines and describes the functional requirements of a system. It represents the interactions between different actors and the system being developed, focusing on how the system responds to the actors actions. Each use case outlines a specific scenario in which the system is used to achieve a particular goal or perform a specific function.

2.3.1 Overall use case diagram

An overall use case diagram shows a big picture of how a system works. It includes who uses the system, what they do with it, and how they interact. It's like a map that helps us understand the main parts and how they connect.

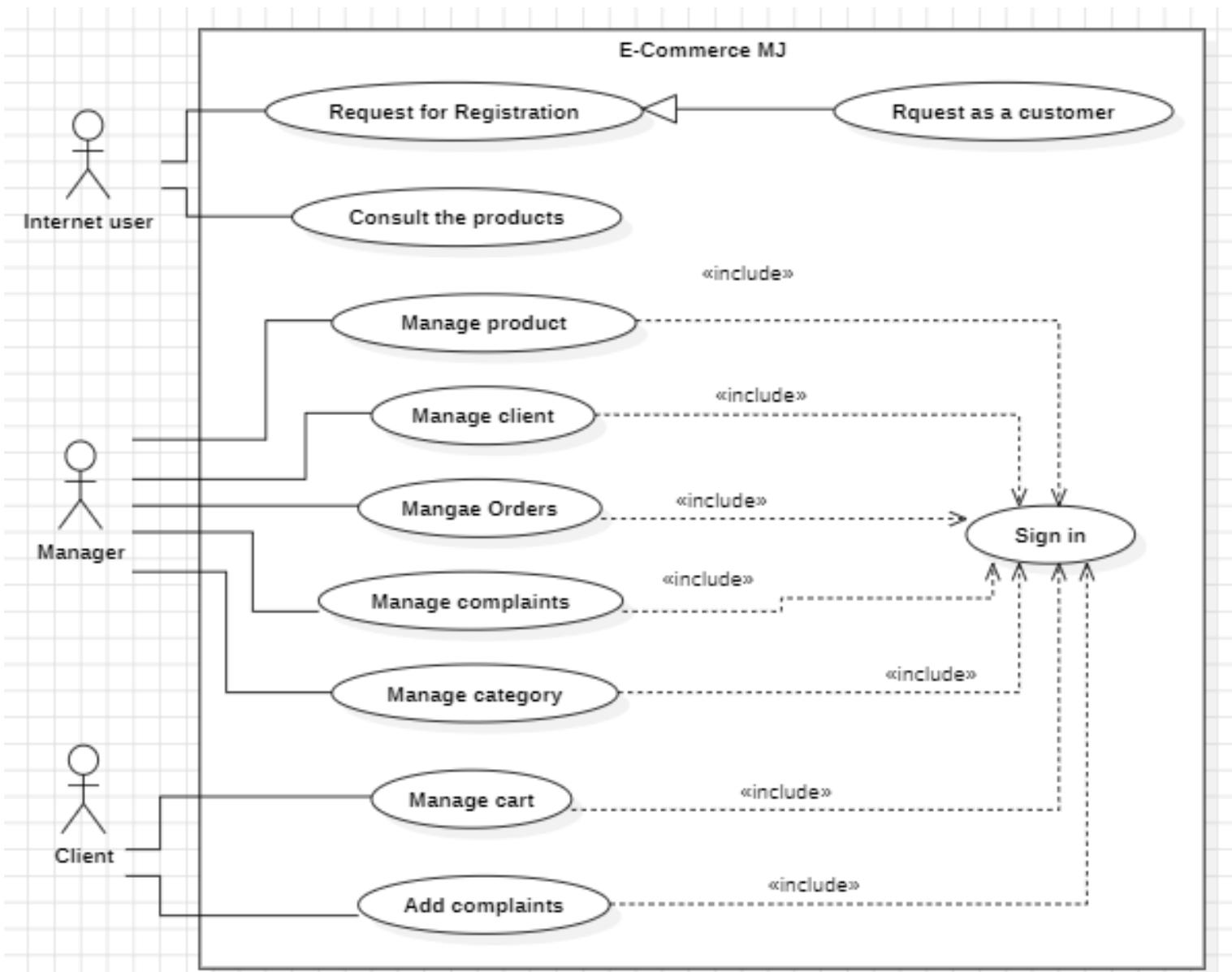


Figure 2.1: Overall use case diagram

2.3.2 "sign in" use case diagram

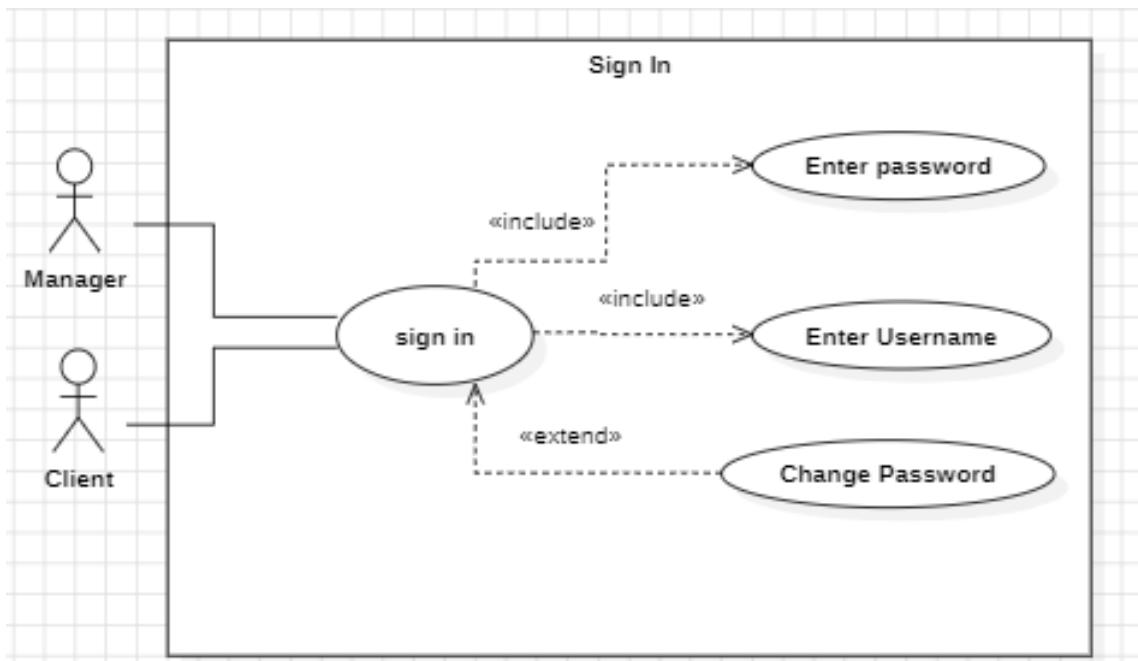


Figure 2.2: Comprehensive use case diagram "sign in"

The following table presents a textual description for this use case diagram.

Table 2.1: Use Case 01: Authentication

Actors	Objective
Manager,Client	Allows the actor to access his personal space
Precondition	Scenario description
The user that already registered has a login and a password	<ol style="list-style-type: none"> 1. The user signs in to the system and requests access to their account space. 2. The system displays a form to enter the authentication information. 3. The user enters his username and password. 4. The system checks the information entered. 5. The system opens the workspace corresponding to the profile.
Incorrect login or password:	Alternative scenario
this scenario starts at point 03 of the nominal scenario.	<ol style="list-style-type: none"> 1. The system notifies the user that the provided information is incorrect and prompts them to input the accurate username or password.

2.3.3 "Manage product" use case diagram

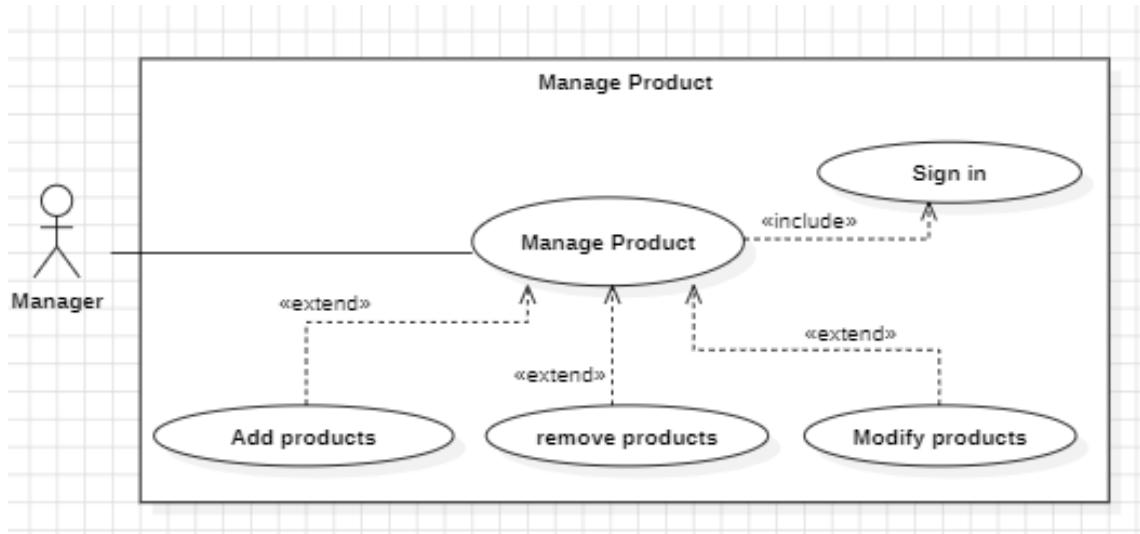


Figure 2.3: Comprehensive use case diagram "Manage product"

The following table presents a textual description for a possible scenario "Manage product" from this use case diagram.

Table 2.2: Use case 02:Add Products

Actors	Objective
Manager	Manage products
Precondition:	Scenario description
Agent authenticated	<ol style="list-style-type: none"> 1. The manager logged in and selected the products. 2. The system displays a corresponding form containing the following operations: add, delete, edit, create. 3. The manger chooses the operation "add". 4. The system executes and saves the chosen operation.
Alternative scenario:	The added account exists already .
After step 4 of the scenario description	The system displays an error message "The added products is already exist".
Error scenario:	products management is not allowed.
At any step	<ol style="list-style-type: none"> 1. The system displays an error message " products management is not allowed".

2.3.4 "Manage client" use case diagram

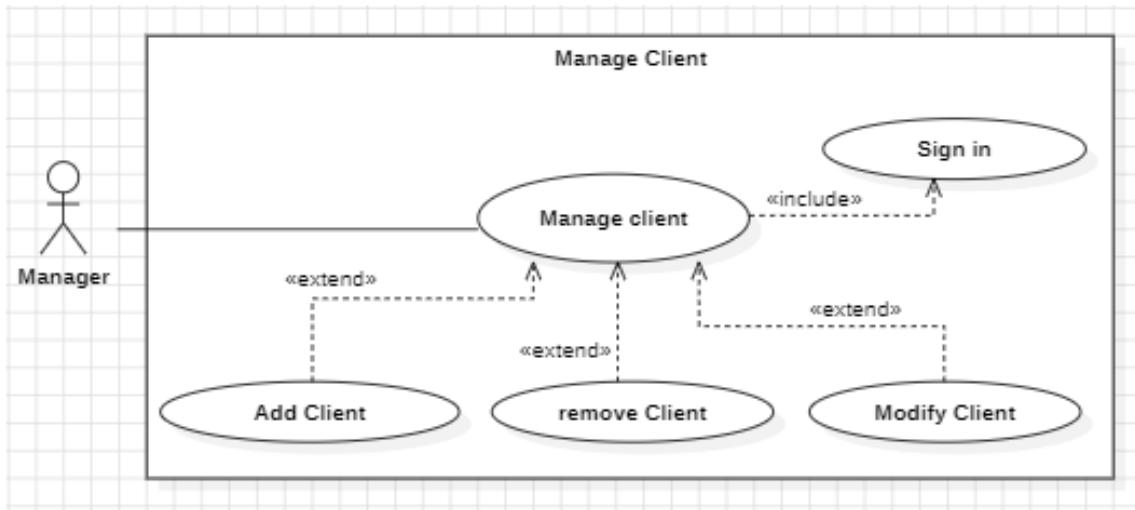


Figure 2.4: Comprehensive use case diagram "Manage client"

The following table presents a textual description for a possible scenario "manage client" from this use case diagram.

Table 2.3: Use case 03:delete client

Actors	Objective
Manager	Manage client
Precondition:	Scenario description
Agent authenticated	<ol style="list-style-type: none"> 1. The manager logged in and selected the clients. 2. The system displays a corresponding form containing the following operations: add, delete. 3. The manager chooses the operation "delete". 4. The system executes and saves the chosen operation.
Alternative scenario:	The deleted account exists already .
After step 4 of the scenario description	The system displays an error message "The deleted client doesn't exist".
Error scenario:	client management is not allowed.
At any step	<ol style="list-style-type: none"> 1. The system displays an error message " client management is not allowed".

2.3.5 "Manage complaints" use case diagram

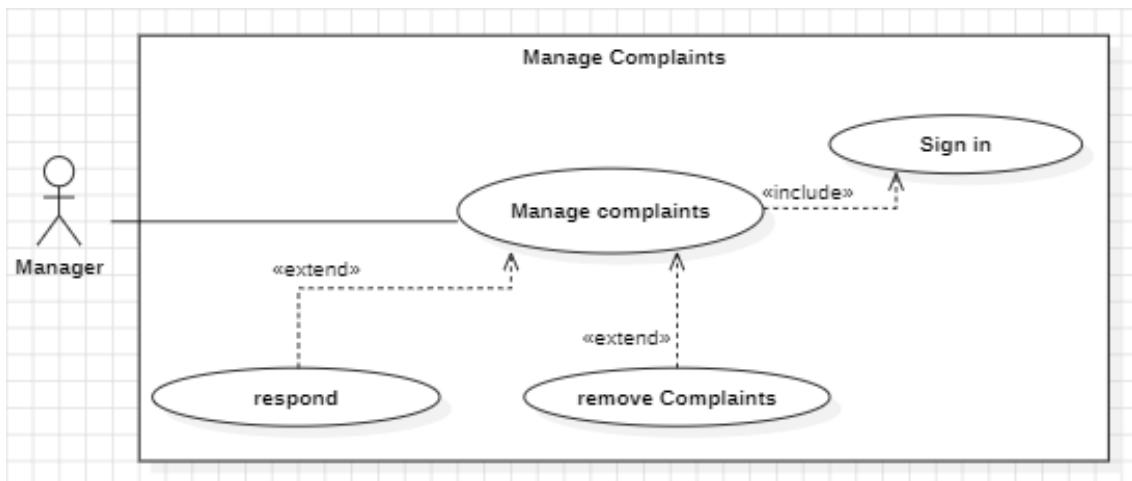


Figure 2.5: Comprehensive use case diagram "Manage complaints"

The following table presents a textual description for a possible scenario "manage complaints" from this use case diagram.

Table 2.4: Use case 04:respond complaint

Actors	Objective
Manager	Manage complaints
Precondition:	Scenario description
Agent authenticated	<ol style="list-style-type: none"> 1. The manager logged in and selected the compliants. 2. The system displays a corresponding form containing the following operations: respond or remove the complaint. 3. The manager chooses the operation "respond". 4. The system executes and saves the chosen operation.

2.3.6 "Manage categories" use case diagram

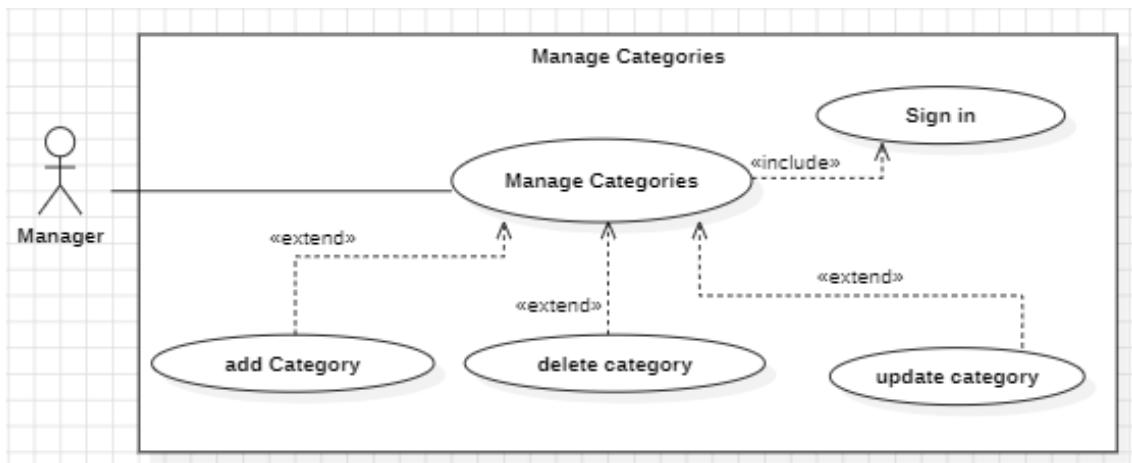


Figure 2.6: Comprehensive use case diagram "Manage categories"

The following table presents a textual description for a possible scenario "manage categories" from this use case diagram.

Table 2.5: Use case 06: Add categories

Actors	Objective
Manager	Manage categories
Precondition:	Scenario description
Manager authenticated	<p>1. The Manager logged in and selected the categories.</p> <p>2. The system displays a corresponding form containing the following operations: add, delete and update.</p> <p>3. The manager chooses the operation "add".</p> <p>4. The system executes and saves the chosen operation.</p>
Alternative scenario:	The category request exists .
After step 4 of the scenario description	The system displays an error message "The category doesn't exist".
Error scenario:	Category management is not allowed.
At any step	<p>1. The system displays an error message "category management is not allowed".</p>

2.3.7 "Manage cart" use case diagram

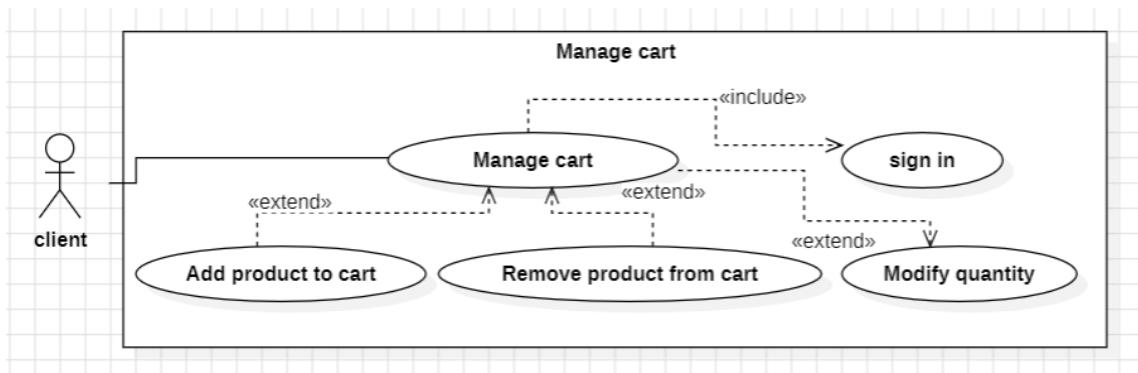


Figure 2.7: Comprehensive use case diagram "Manage cart"

The following table presents a textual description for a possible scenario "manage cart" from this use case diagram.

Table 2.6: Use case 07:Modify quantity

Actors	Objective
client	Manage carts
Precondition:	Scenario description
client authenticated	<ol style="list-style-type: none"> 1. The client logged in and selected the carts. 2. The system displays a corresponding form containing the following operations: add,modify,delete. 3. The client chooses the operation "modify quantity". 4. The system executes and saves the chosen operation.
Alternative scenario:	The product in the cart doesn't exists .
After step 4 of the scenario description	The system displays an error message "The product doesn't exist".
Error scenario:	cart management is not allowed.
At any step	<ol style="list-style-type: none"> 1. The system displays an error message " cart management is not allowed".

2.3.8 "Send complaint" use case diagram

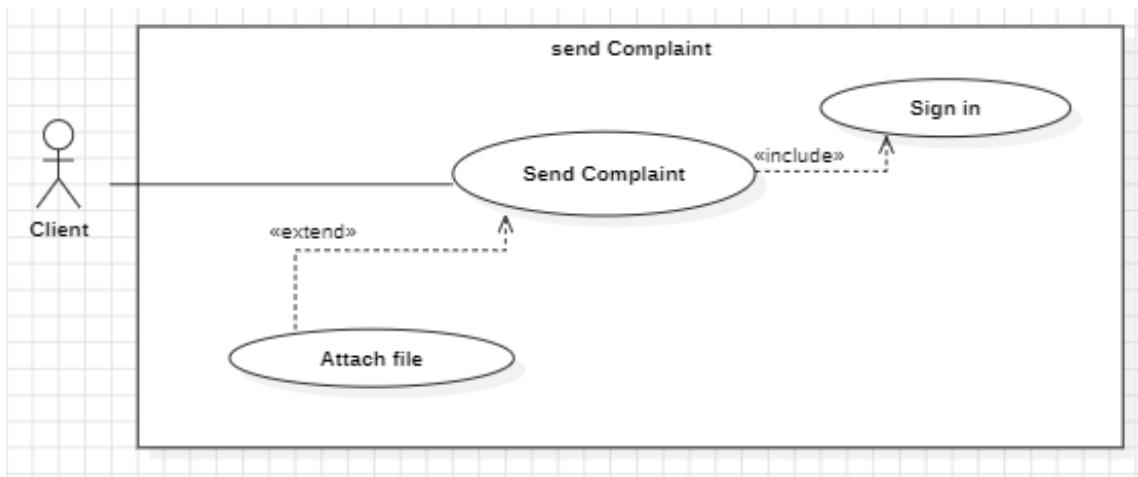


Figure 2.8: Comprehensive use case diagram "Send complaint"

The following table presents a textual description for a possible scenario "Send complaint" from this use case diagram.

Table 2.7: Use case 08:Send complaint

Actors	Objective
client	Send complaint
Precondition:	Scenario description
client authenticated	<ol style="list-style-type: none"> 1. The client logged in and selected the complaint. 2. The system displays a corresponding form containing the following operations: attach file,submit complaint. 3. The client chooses the operation "submit complaint" then attach file . 4. The system executes and saves the chosen operations.
Alternative scenario:	The attach file doesn't exists .
After step 4 of the scenario description	The system displays an error message "The complaint doesn't exist".
Error scenario:	complaint management is not allowed.
At any step	<ol style="list-style-type: none"> 1. The system displays an error message "complaint management is not allowed".

2.4 Sequence diagrams

A sequence diagram is a UML diagram that shows how objects interact in a specific order. It highlights the sequence of messages exchanged between objects to accomplish a task. Key elements include:

- Actors and Objects: Entities involved in the interaction.
- Lifelines: Represent the duration of an object's participation.
- Messages: Arrows showing communication between objects.
- Activations: Indicate when an object is performing an operation

2.4.1 Sign In sequence Diagram

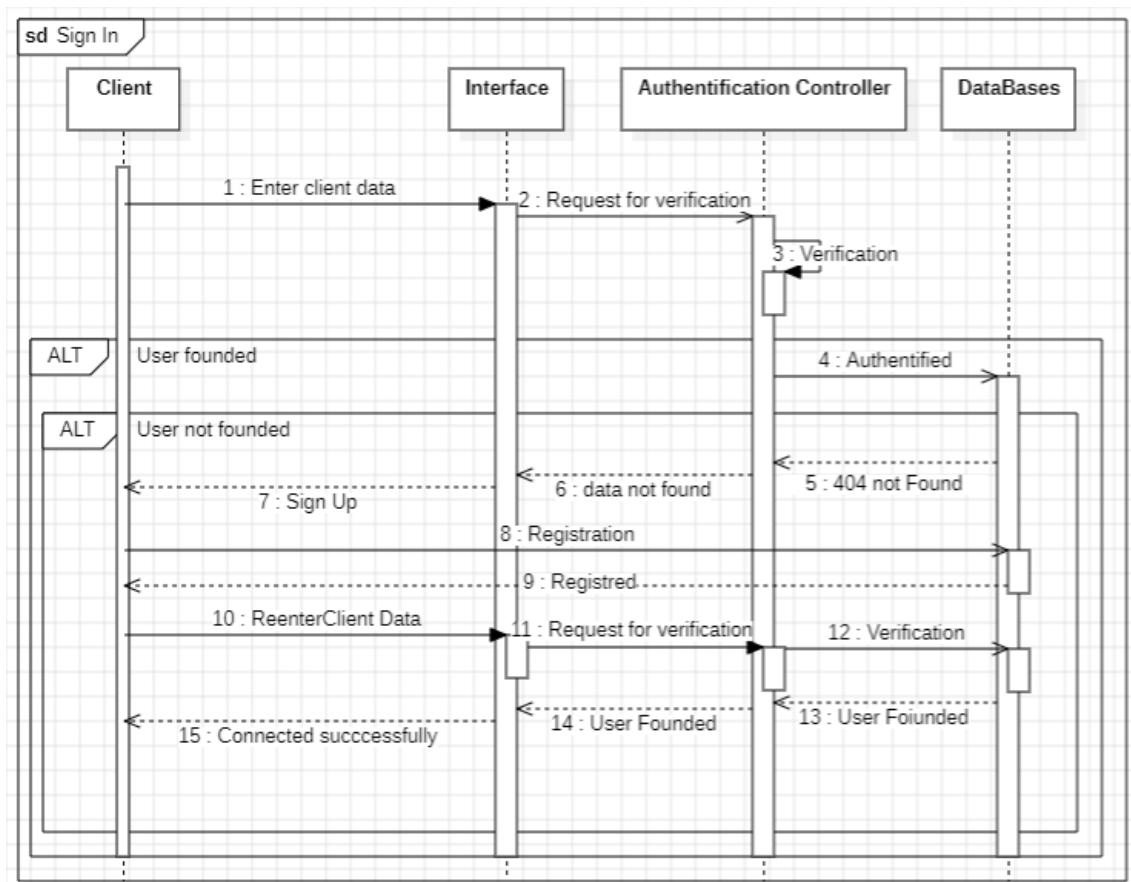


Figure 2.9: Comprehensive Sequence diagram "sign in"

2.4.2 Add product

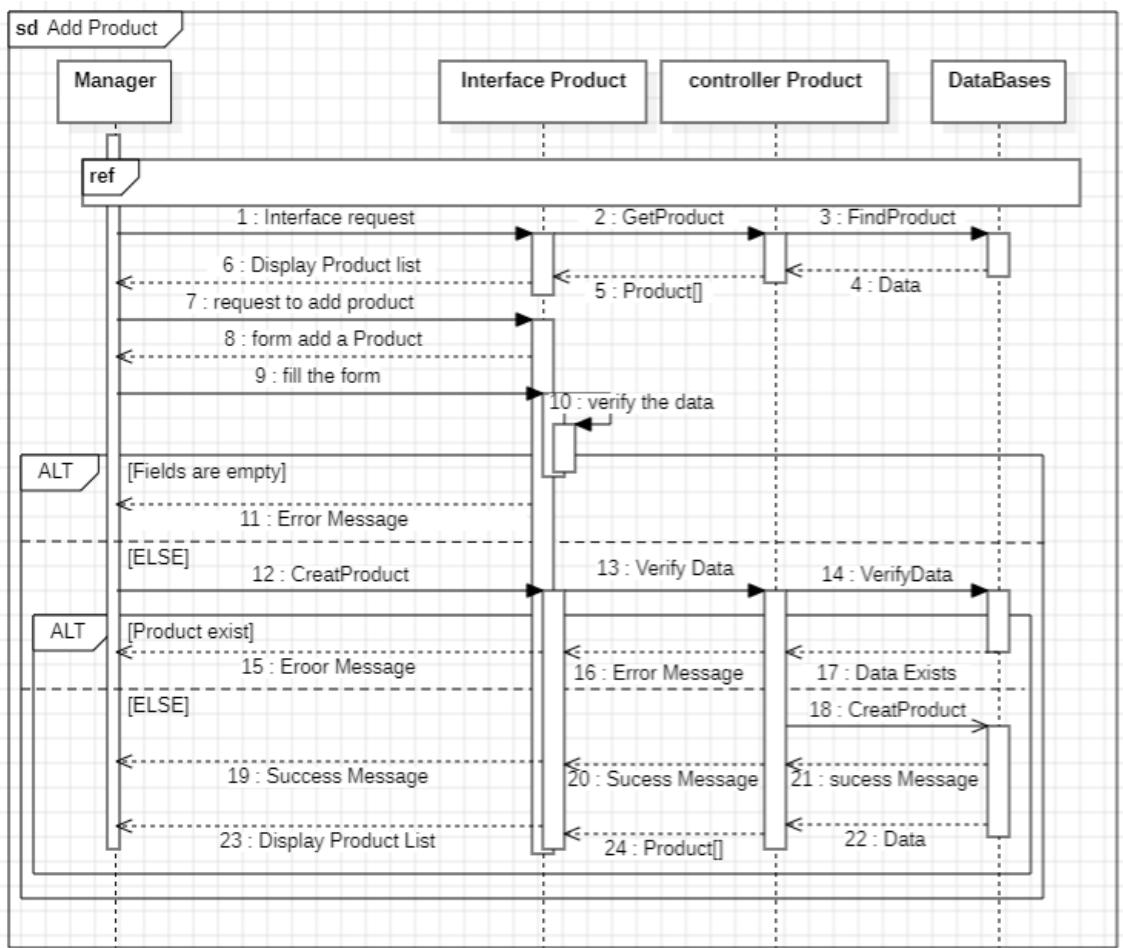


Figure 2.10: Comprehensive Sequence diagram "Add product"

2.4.3 Delete product

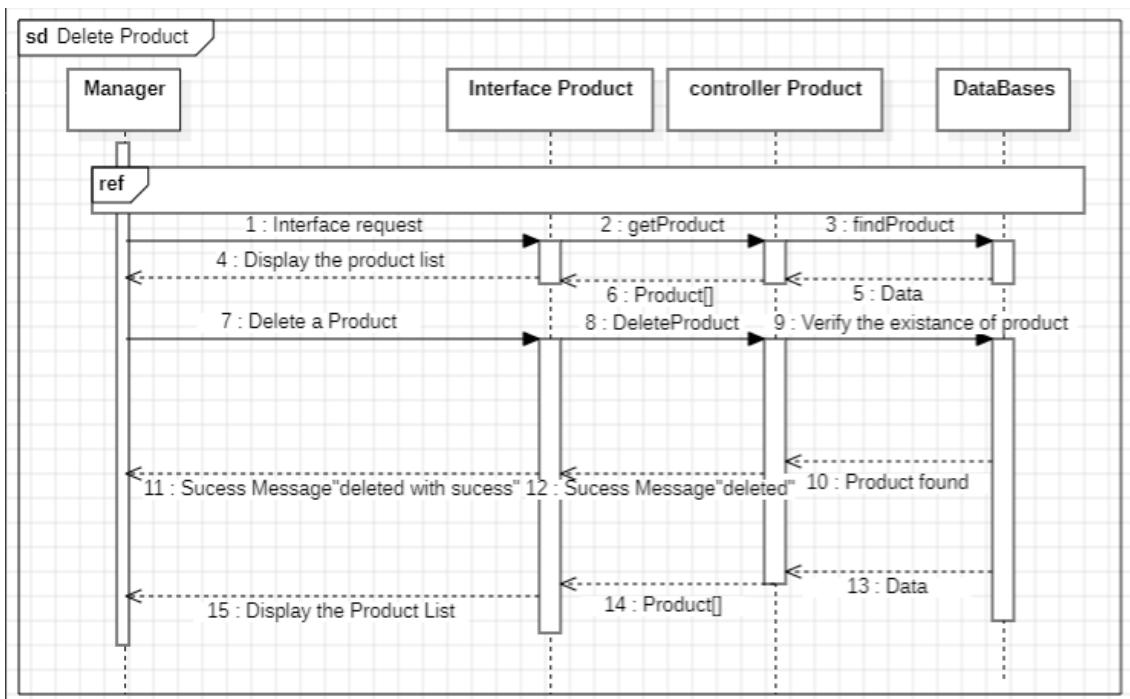


Figure 2.11: Comprehensive Sequence diagram "Delete product"

2.4.4 Delete category

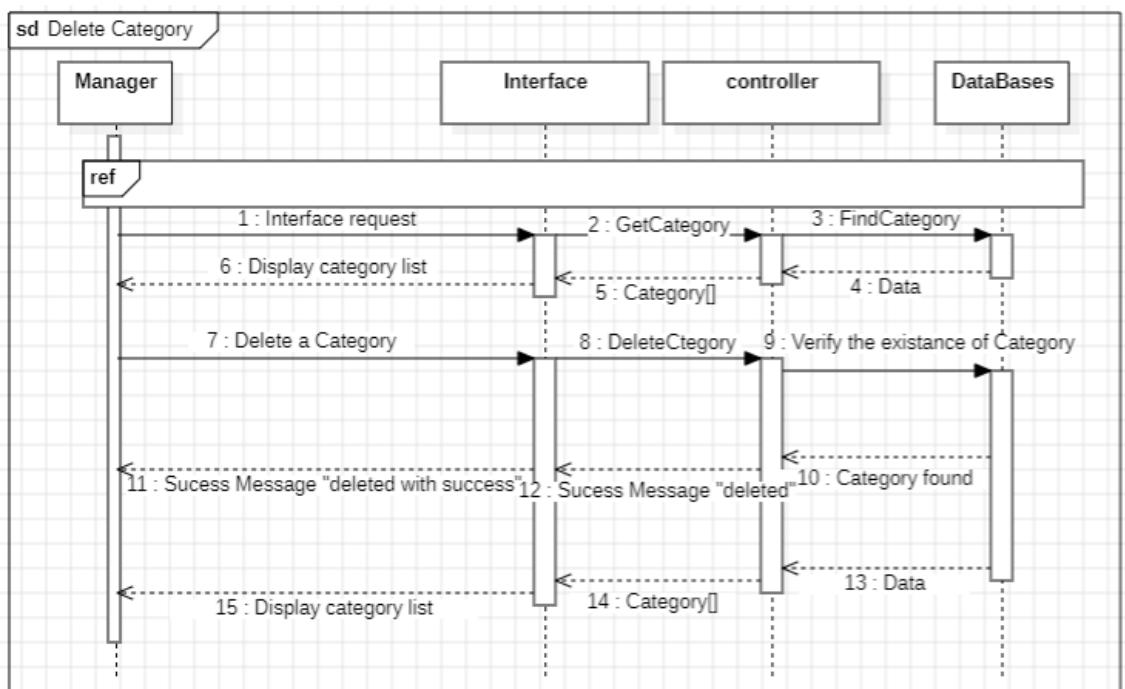


Figure 2.12: Comprehensive Sequence diagram "delete category"

2.4.5 Manage complaint

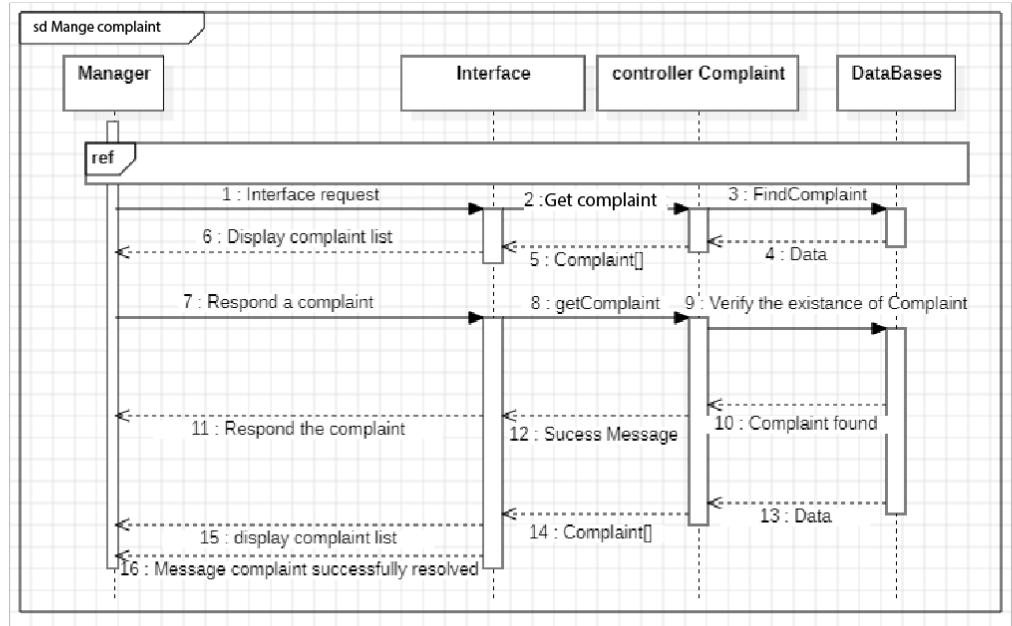


Figure 2.13: Comprehensive Sequence diagram "manage complaint"

2.5 Overall class diagram

To better understand how the system works, we need to mention the class diagram that includes all the necessary classes.

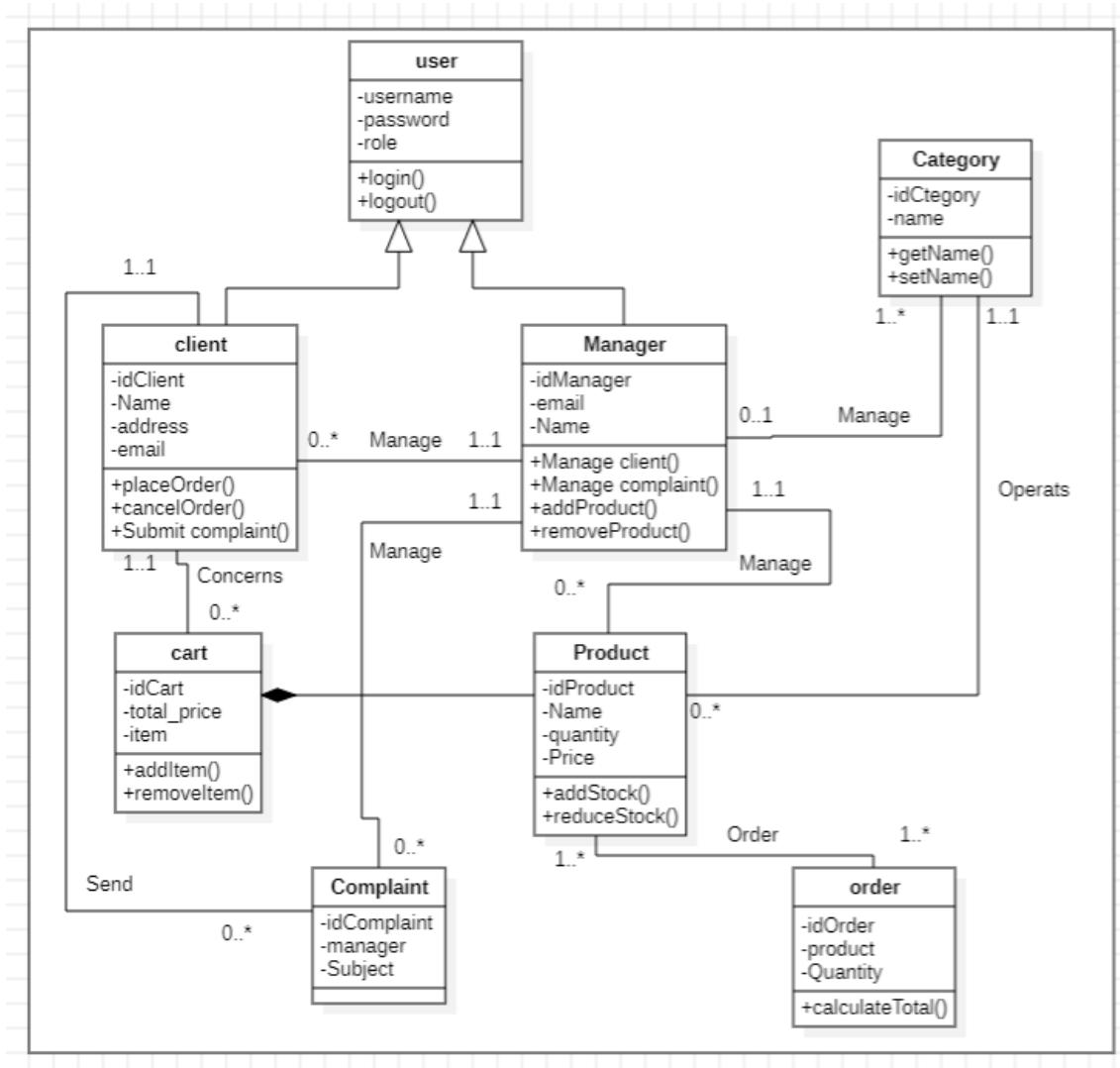


Figure 2.14: Comprehensive class diagram

A class diagram shows the structure and relationships between classes in a system. It helps to visualize classes, attributes, methods, and the connections between classes. Here are some key parts of a class diagram:

- **Class**: An abstract representation of an object or concept in the system.

- Attribute: A characteristic or property of a class.
- Method: An action or behavior associated with a class.
- Relations: The links and dependencies between classes.

2.6 Conclusion

We now have a better and more comprehensive vision of what we will achieve, due to the detailed planning process that we carried out, and we must now move to the next stage.

CHAPTER 3

Realization and implementations

Contents

3.1	Introduction	47
3.2	Technical Study:	47
3.2.1	Software Environment:	47
3.2.2	Tools for Research and Analysis	47
3.2.3	Tools of development	48
3.3	Architectural Style	53
3.4	Design Pattern	54

3.1 Introduction

In this chapter, we will explore the different tools we used, explain how they helped us build and implement the project, and show their role in making our project a reality. We will also describe the architecture we used ,the pattern and its features.

3.2 Technical Study:

3.2.1 Software Environment:

In this section, we'll introduce the software we'll be using during the project period.

3.2.2 Tools for Research and Analysis

In order to ensure the delivery of the least satisfactory product to customers. Some tools and forums were used to collect and determine what customers are waiting for

Google meet



Figure 3.1: Google meet logo

[7]Google Meet is an online video conferencing tool utilized to connect with potential users and company officials, aiming to gather as much information as possible.

3.2.3 Tools of development

Visual Studio Code

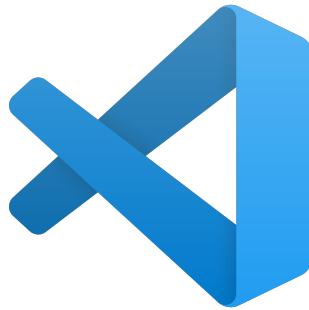


Figure 3.2: Visual Studio Code logo

[16] VS Code, short for Visual Studio Code, is an IDE (Integrated Development Environment) created by Microsoft for Linux, macOS, and Windows. It supports various programming languages and is widely regarded as one of the most powerful and popular code editors, thanks to its comprehensive set of tools and extensions.

MongoDB



Figure 3.3: MongoDB

[10] MongoDB is like a super flexible and powerful way to store and handle data for modern apps. It's popular because it's easy to use and can handle lots of different types of data. From small projects to big ones, MongoDB helps businesses move fast and do cool stuff with their data. It's all about making data work smarter for you.

Next.JS



Figure 3.4: Next.JS

[11]Next.js is a cool tool for making websites and apps. It's easy to use and makes your stuff load super fast. Whether you're building something simple or fancy, Next.js has got your back. It's the go-to choice for developers who want to build awesome web stuff without all the hassle.

CSS



Figure 3.5: CSS logo

[4]CSS CSS, or Cascading Style Sheets, is like the fashion designer for websites. It's what makes everything look pretty and stylish on the web. From choosing colors and fonts to arranging layouts and animations, CSS has got you covered. With its simple syntax and powerful features, CSS lets you customize the appearance of your web pages with ease.

Java Script



Figure 3.6: Java Script logo

[9]JavaScript is like the Swiss Army knife of web development. It's the language that makes websites interactive and cool. From simple animations to complex web applications, JavaScript does it all. With its widespread support and easy-to-learn syntax, it's the go-to language for building dynamic and engaging web experiences. Whether you're a beginner or a seasoned developer, JavaScript empowers you to bring your ideas to life on the web.

Node JS



Figure 3.7: Node JS logo

[12]Node.js is an open-source, cross-platform JavaScript runtime environment that allows us to perform actions such as writing to the standard output, opening or closing connections, and creating files using JavaScript libraries.

npm



Figure 3.8: npm logo

[13]With more than 2 million packages,npm is the largest registry around the world for all JavaScript code, providing many advantages such as downloading the tools needed in a project or adapt packages to match each user's own code....

GitHub

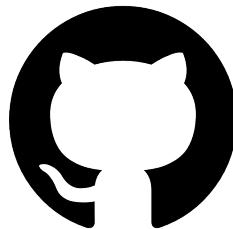


Figure 3.9: GitHub logo

[6]With more than 100 million developer users, GitHub is a website that provides a hosting service for all version control repositories, known on a global scale, which contributes to the possibility of working on the same project from different places and also provides many tools that facilitate and organize work

StarUML



Figure 3.10: StarUML logo

[14]StarUML comes in the first stage to support the UML technology, and then in other stages some other technologies. It is a program or tool for designing and diagramming all types of UML diagrams, providing for that purpose many tools and features, everything that is needed to create a complete diagram. It is also worth mentioning that It has a beautiful interface and the feature of drag and drop to facilitate work.

3.3 Architectural Style

In our work, we used a three-part architecture with three main components:

- The user interface
- The application serverHTML
- The database server.

So, our application has three layers:

- The presentation layer: This is the user interface where communication happens.
- The business layer: This handles the implementation of algorithms needed to perform the functions requested by the presentation layer.
- The data layer: This is for data access to manage the data properly.

We chose a three-tier architecture for our project because it offers great flexibility and performance by distributing tasks between servers. Additionally, this architecture is known for its high security.

3 Tiers

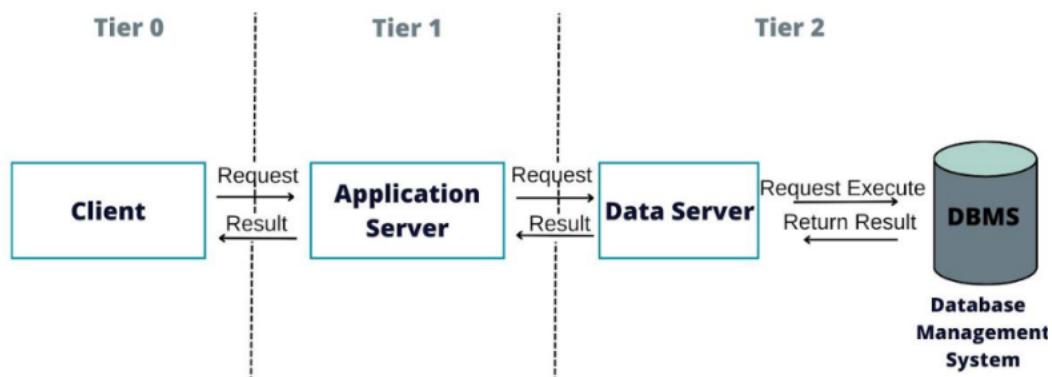


Figure 3.11: 3 Tiers

3.4 Design Pattern

In terms of design, various approaches have been developed to organize the code of a rich client system. The MVC architecture is one of the most famous; it involves dividing the code into three distinct components: the Model, the View, and the Controller. Each of these components is placed in a separate file, allowing the logic of the code to be separated into smaller, more manageable parts.

MVC

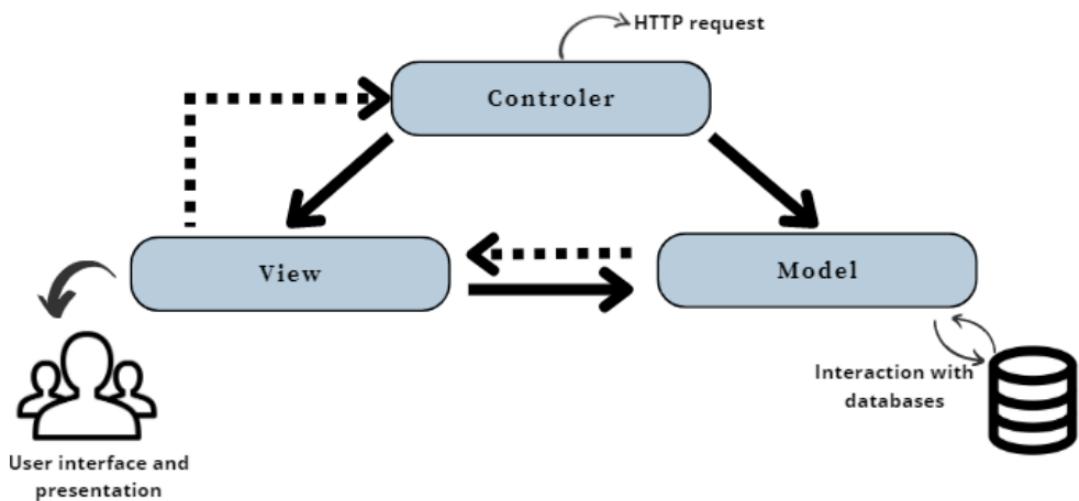


Figure 3.12: MVC pattern

CHAPTER 4

Overview and future work

Contents

4.1	Introduction	56
4.2	Design	56
4.2.1	Name	56
4.2.2	Logo	56
4.3	Application overview	56
4.3.1	Sign up UI	57
4.3.2	Sign in UI	58
4.3.3	Home page UI	59
4.3.4	Cart UI	60
4.3.5	ComplaintsUI	61
4.3.6	Admin Dashboard	62
4.3.7	Manage Product	63
4.3.8	Add Product	64
4.3.9	Manage Categories	65
4.3.10	Add Categories	66
4.3.11	Manage complaint	67
4.3.12	Manage Order	68
4.4	Futur work	69
4.5	Conclusion	69
	General Introduction	70

4.1 Introduction

In this final stage, we will review the final product we've achieved and outline our vision for future development but before that we will explain the different parts of designing.

4.2 Design

We will explain step by step the path of design because it is the critical part in completing our project because it depends mainly on the user experience, and it represents one of the biggest challenges in the success of any product.

4.2.1 Name

The name we choose is crucial as it's the first impression customers get and defines our store. So selected name is: **MJstore**

4.2.2 Logo

The logo chosen is simple and inspired by the color of the website
The following figure represent our designed logo.



Figure 4.1: Platform logo

4.3 Application overview

After we have successfully completed building our project, it is time to show some screenshots of its interfaces

4.3.1 Sign up UI

The figure displays a user interface for account creation. At the top, a navigation bar includes the store name "MJ-Store" and links for Home, Products, Categories, Complaint, Cart (0), and Account. Below this, the main area is divided into two sections: "Add Account" on the left and "Account details" on the right. The "Account details" section contains fields for Name, Email, City, Postal code, Street Address, Number(+216), and Password, each enclosed in a separate input box. A "save" button is located below these fields. At the bottom of the "Account details" section is a "Login with Google" button.

Figure 4.2: Sign up UI

The figure 4.3 represents an interface containing a form that allows registration on the site.

4.3.2 Sign in UI

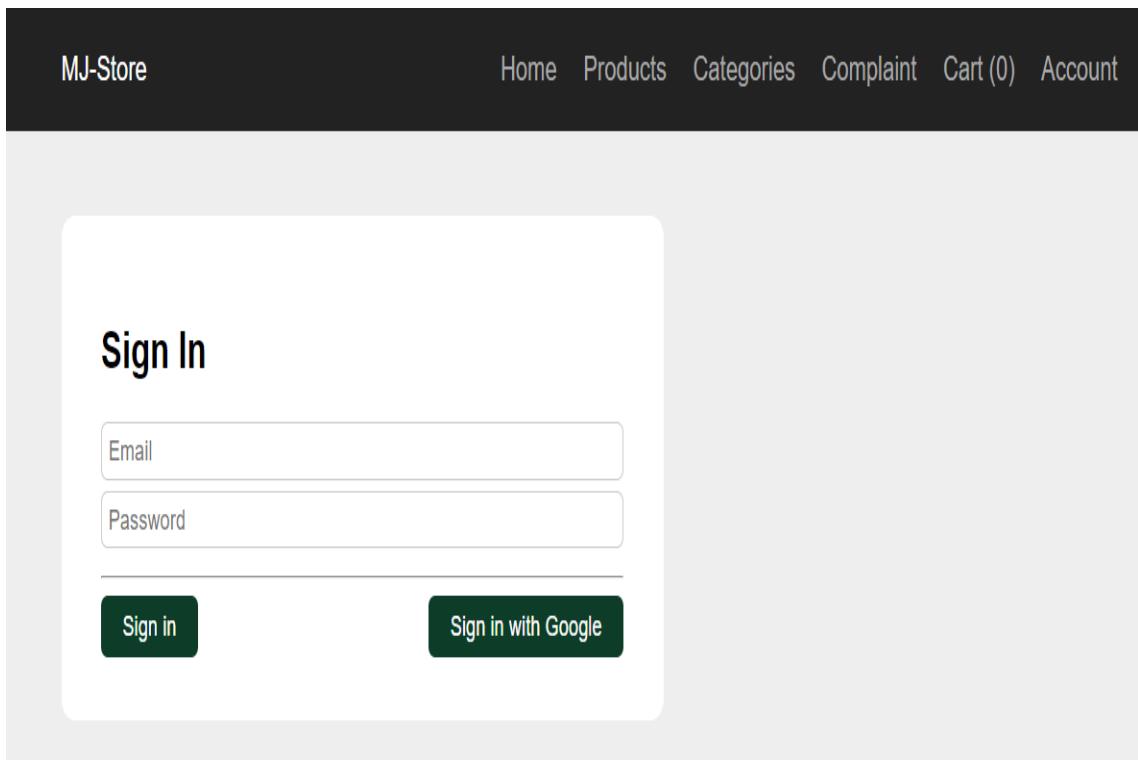


Figure 4.3: Sign in UI

The figure 4.4 represents the interface that contains a form allowing anyone to sign in to his existing account.

4.3.3 Home page UI

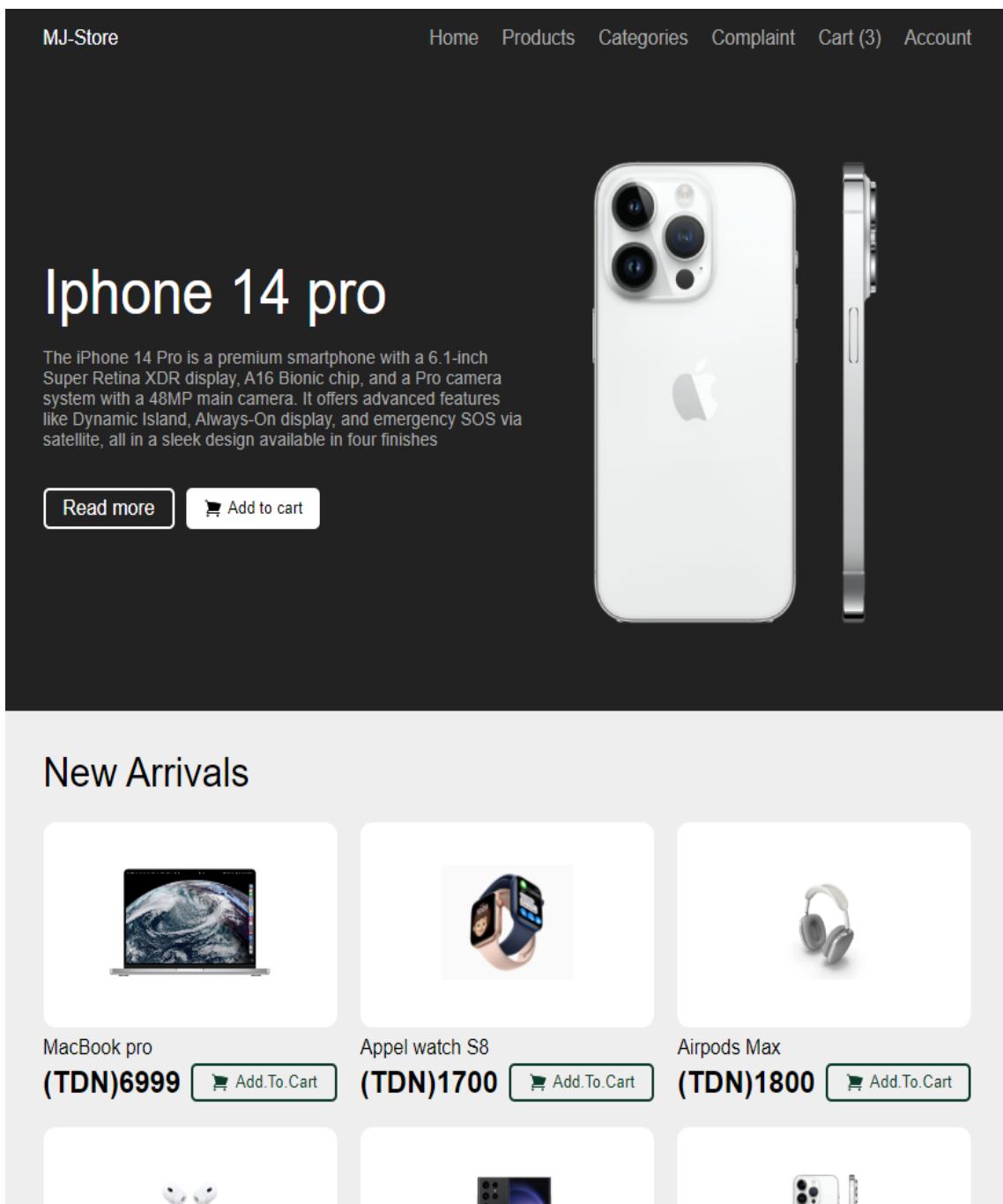


Figure 4.4: Home page UI

The figure 4.5 represents the home page of the platform, which contains most of the features.

4.3.4 Cart UI

The screenshot shows the client cart interface. At the top, there's a navigation bar with 'MJ-Store' and links for Home, Products, Categories, Complaint, Cart (3), and Account. The main area is divided into two sections: 'Cart' on the left and 'Order information' on the right.

Cart Section:

PRODUCT	QUANTITY	PRICE
Iphone 14 pro	- 2 +	(TDN)6200
MacBook 14 pro	- 1 +	(TDN)6100
		(TDN)12300

Order information Section:

Name

Email

City Postal code

Street Address

Number(+216)

Continue to payment button

Figure 4.5: Client cart UI

As shown in the figure 4.6, the cart contains the products that the customer chooses while scrolling in the store.

4.3.5 ComplaintsUI

The screenshot shows a web application interface for adding a complaint. At the top, there is a dark navigation bar with the text "MJ-Store" on the left and "Home Products Categories Complaint Cart (0) Account" on the right. Below the navigation bar, the main content area is divided into two sections. The left section, titled "Add complaint", contains the text "You can call us to complain" followed by a horizontal line and the phone number "Our number:(+216) 75 105 105". The right section, titled "Complaint details", contains four input fields: "Name", "Email", "Number(+216)", and "Your Complaint". Below these fields is a file upload input labeled "Choose File" with the placeholder "No file chosen". At the bottom of the right section is a black "save" button.

Figure 4.6: Complaints

in this figure, 4.7, we present the interface where the customer could add a complaint about the things that he didn't like so we are always connected with our clients

4.3.6 Admin Dashboard

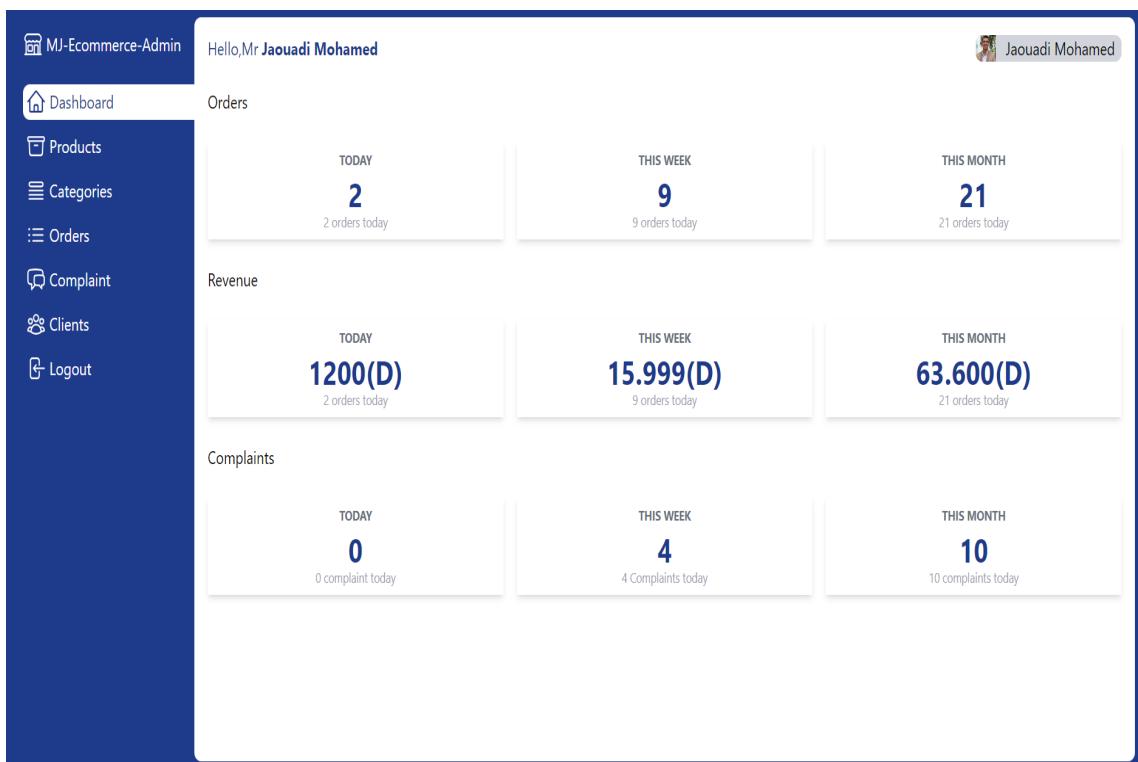


Figure 4.7: Admin Dashboard

As shown in the figure 4.8, the admin Dashboard through which the admin can control the store

4.3.7 Manage Product

The screenshot shows the 'MJ-Ecommerce-Admin' application's product management interface. On the left is a dark blue sidebar with white icons and text for 'Dashboard', 'Products' (which is selected and highlighted in blue), 'Categories', 'Orders', 'Complaint', 'Clients', and 'Logout'. The main area has a light blue header bar with the text 'Add new product'. Below this is a table with a light blue header row labeled 'PRODUCT NAME'. The table contains six rows of data, each representing a product: 'Iphone 14 pro', '\$24 ultrat', 'Airpods Pro', 'Airpods Max', 'Appel watch \$8', and 'MacBook pro'. Each row has two buttons on the right: a grey 'Edit' button with a pencil icon and a red 'Delete' button with a trash bin icon.

PRODUCT NAME	
Iphone 14 pro	Edit Delete
\$24 ultrat	Edit Delete
Airpods Pro	Edit Delete
Airpods Max	Edit Delete
Appel watch \$8	Edit Delete
MacBook pro	Edit Delete

Figure 4.8: Manage Product

In the figure 4.9, the admin could manage the product modify , delete and consult the product

4.3.8 Add Product

MJ-Ecommerce-Admin

Dashboard

Products

Categories

Orders

Complaint

Clients

Logout

New product

Product name

Apple Watch S8

Category

Watch

Color

blue

Photos

Add image

Description

The Apple Watch Series 8 is a smartwatch that's easy to use and packed with features. It's tough, with water and dust resistance, and has a bright screen you can always see. It can check your heart and blood oxygen, and even help in emergencies. It connects to the internet and your phone, lasts about a day on a charge, and is made with recycled materials. It's a great choice for staying fit, safe, and in touch

Price (in TDN)

1700

Save

Figure 4.9: Add Product

In the figure 4.9, the admin could also add the product

4.3.9 Manage Categories

The screenshot shows the 'Categories' management screen in the MJ-Ecommerce-Admin application. The left sidebar has a dark blue background with white icons and text for 'Dashboard', 'Products', 'Categories' (which is highlighted in blue), 'Orders', 'Complaint', 'Clients', and 'Logout'. The main area has a light blue header with the title 'Categories'. Below it is a form for 'Create new category' with fields for 'Category name' (containing 'Category name') and 'Parent category' (containing 'No parent category'). There's a 'Properties' section with a 'Add new property' button and a 'Save' button. The main table lists categories with columns 'CATEGORY NAME' and 'PARENT CATEGORY'. The data is as follows:

CATEGORY NAME	PARENT CATEGORY	Actions
Mobiles		Edit Delete
Iphones	Mobiles	Edit Delete
Sumsung	Mobiles	Edit Delete
Laptop		Edit Delete
MacBook	Laptop	Edit Delete
Airpods		Edit Delete

Figure 4.10: Manage Category

In the figure 4.9, the admin could manage the category modify , delete and consult the category

4.3.10 Add Categories

The screenshot shows the MJ-Ecommerce-Admin interface with a sidebar containing links for Dashboard, Products, Categories (selected), Orders, Complaint, Clients, and Logout. The main area is titled 'Categories' and shows a form to 'Create new category' with 'Apple Watch' entered. Under 'Properties', there is a 'Color' field with 'black,silver,red,green,blue' and a 'Series' field with 'S6,S7,S8,S9'. A 'Save' button is present. Below this is a table listing categories:

CATEGORY NAME	PARENT CATEGORY	Actions
Mobiles		Edit Delete
Iphones	Mobiles	Edit Delete
Samsung	Mobiles	Edit Delete
Laptop		Edit Delete
MacBook	Laptop	Edit Delete

Figure 4.11: Add Category

In the figure 4.9, the admin could also add a category "add the category"

4.3.11 Manage complaint

The screenshot shows the 'Complaints' section of the MJ-Ecommerce-Admin dashboard. On the left, there is a sidebar with the following menu items:

- MJ-Ecommerce-Admin
- Dashboard
- Products
- Categories
- Orders
- Complaint
- Clients
- Logout

The main content area is titled 'Complaints' and contains a table with two rows of data:

DATE	COMPLAINT	RECIPIENT	
5/3/2024, 1:09:08 AM	I complaint for the 14 pro : I'm having problems with the battery on my new iPhone 14 pro. It drains quickly and doesn't last as long as it should.	Jamel fal foul // jamel123@gmail.com Kairouan 3000 Kairouan centre 23955652	Noted Delete
6/4/2024, 4:28:54 AM	I complaint for the 11 pro : I'm unhappy with my recent iPhone 11 pro delivery. It arrived late and the package was damaged. Can you help fix this?	saif essid // Saif.essid12@gmail.com sousse 4100 sousse khzema gharbia 25819701	Noted Delete

Figure 4.12: Manage Complaint

In the figure 4.10, the admin could manage the complaint by consulting or responding them

4.3.12 Manage Order

Orders				
DATE	PAID	RECIPIENT	PRODUCTS	
6/4/2024, 5:30:58 PM	YES (Order Paid)	mohamed zeyri // zeiri.mohamed000@gmail.com Kairouan 3000 Kairouan centre 29 800 925	Appel watch S8 : (x2) S24 ultrat : (x1)	<button>Delivery</button> <button>Delete</button>
6/4/2024, 5:30:12 PM	NO (Paid on delivery!)	mohamedjaouadi // mjaouadi122@gmail.com sousse 4100 sousse khzema 55429621	Airpods Pro : (x1) Iphone 14 pro : (x1)	<button>Delivery</button> <button>Delete</button>
6/4/2024, 4:32:08 AM	YES (Order Paid)	Jamel falfoul // jamel123@gmail.com Kairouan 3000 Kairouan centre 23955652	Airpods Pro : (x1) MacBook 14 pro : (x1)	<button>Delivery</button> <button>Delete</button>
6/4/2024, 4:28:54 AM	NO (Paid on delivery!)	saiif essid // Saif.essid12@gmail.com sousse 4100 sousse khzema gharbia 25819701	Iphone 14 pro : (x2) MacBook 14 pro : (x1)	<button>Delivery</button> <button>Delete</button>

Figure 4.13: Manage Order

In the figure 4.9, the admin could manage the Orders by consulting or deleting the order

4.4 Futur work

Every web application must have a future vision in order to continue to develop, and one of the most important factors is that I'll use BI to estimate what type of product should i work on more. also we could use IA as it's a tendency by developing a chatbot so he could respond all complaints

4.5 Conclusion

Through this last chapter, we clarified the last stages of completing our project. We put some scenes of what we got in the end, and we also did not forget to allocate a part to contain our future vision of what this work can reach.

General Conclusion

It cannot be denied that e-commerce have become a crucial element in every business seeking to enhance its efficiency and productivity. This is especially true for companies that conduct most of their tasks using computer tools, which involve handling vast amounts of data and managing them easily and quickly.

Furthermore, working on this e-commerce project has been an enlightening journey into the ever-evolving technology of online shopping. It reflects the dynamic nature of the digital world and allows us to explore the core of digital commerce, its workings, and the immense potential it offers for the future.

This report has not only let us delve into the complexities of creating an e-commerce platform but also provided us with the chance to innovate and enhance existing models, with a keen eye on improving the user experience.

In conclusion, this work marks the beginning of our journey as technologists who can make meaningful contributions to the field. We are eager to continue refining our skills and creating innovative solutions that enrich the digital world.

Bibliography

- [1] *AliExpress*. URL: <https://fr.wikipedia.org/wiki/AliExpress>. (accessed: 17.03.2024).
- [2] *Amazon*. URL: <https://fr.wikipedia.org/wiki/Amazon>. (accessed: 17.03.2024).
- [3] *Cascade*. URL: <https://www.cascade.app/about-us>. (accessed: 02.04.2024).
- [4] *css*. URL: <https://www.w3schools.com/css/>. (accessed: 30.03.2024).
- [5] *Design thinking*. URL: <https://www.klap.io/definition-design-thinking/>. (accessed: 01.04.2024).
- [6] *GitHub*. URL: <https://github.com/git-guides>. (accessed: 10.04.2024).
- [7] *Google meet*. URL: <https://workspace.google.fr/intl/fr/products/meet/>. (accessed: 12.05.2024).
- [8] *History of e-commerce*. URL: <https://www.ecwid.com/blog/ecommerce.html>. (accessed: 10.03.2024).
- [9] *Java Script*. URL: <https://developer.mozilla.org/fr/docs/Web/JavaScript>. (accessed: 01.05.2024).
- [10] *MongoDB*. URL: <https://www.w3schools.com/mongodb/>. (accessed: 14.03.2024).
- [11] *Next.Js*. URL: <https://nextjs.org/>. (accessed: 20.03.2024).
- [12] *Node JS*. URL: <https://nodejs.org/en>. (accessed: 02.05.2024).
- [13] *npm*. URL: <https://docs.npmjs.com/about-npm>. (accessed: 02.05.2024).
- [14] *StarUML*. URL: <https://docs.staruml.io/>. (accessed: 20.04.2024).
- [15] *UML*. URL: [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)). (accessed: 15.04.2024).
- [16] *Visual Studio Code*. URL: <https://code.visualstudio.com/docs>. (accessed: 13.05.2024).

Abstract:

This internship is part of my end-of-studies project, carried out at LINQUIBIT, with the aim of obtaining the national bachelor's diploma in Computer Science, Software Engineering, and Information Systems from Ecole Polytechnique de Sousse. The project focused on developing an e-commerce application, MJ Store, with an admin dashboard, employing a waterfall (cascade) and leveraging modern technologies such as NextJs for the frontend, JavaScript for the backend, MongoDB and for data processing.

Résumé :

Ce stage s'inscrit dans le cadre de mon projet de fin d'études, réalisé chez LINQUIBIT, en vue de l'obtention du diplôme national de licence en Informatique, Génie Logiciel et Systèmes d'Information de l'École Polytechnique de Sousse. Le projet a porté sur le développement d'une application e-commerce, MJ Store, avec un tableau de bord administratif, en utilisant une méthodologie cascade et en exploitant des technologies modernes telles que NextJs pour le frontend, JavaScript pour le backend, et MongoDB pour le traitement des données.

DES COMPÉTENCES D'AVANCE



- 📍 Campus universitaire
Route ceinture Sahloul, Sousse
- 📞 +216 99 277 877
- 🌐 polytecsousse.tn

