

University of Manouba

Higher School Of Digital Economy



Graduation Project Report

Topic:

**Conception and development of a  
Content Management System based  
Web Applications**

Elaborated by:

**Yasmine Aridhi**

Presented in the aim of obtaining the diploma  
**Professional master in modeling, databases, and systems integration**

Host organization:



Framed by:

ESEN: Mr. Yassin Ben Youssef

UIB: Mr. Mohamed Aymen Brahim



# *Acknowledgements*

This journey would not have been possible without the support of my family, professors and mentors, and friends.

I would like to express my deep sense of gratitude and convey thanks to my family, thank you for encouraging me in all of my pursuits and inspiring me.

I am especially grateful to my parents, who supported me emotionally and financially. I always knew that you believed in me and wanted the best for me.

I would like to thank my project supervisor Mr. Yassin Ben Youssef, for his guidance and support throughout the project.

I would like to thank Mr. Mohamed Aymen Brahim, for helping me develop the skills necessary to design this application as part of my master's research.

# TABLE OF CONTENTS

<b>FIRST CHAPTER: REQUIREMENTS ANALYSIS</b>	<b>2</b>
INTRODUCTION:	3
1. <i>Presentation of the host organization</i>	3
2. <i>Analysis of existing systems / study of the existing</i>	3
2.1. Existing Solutions	3
2.1.1. Web developers	3
2.1.2. Most famous Content Management System (CMS)	4
2.2. Criticism of existing system	5
3. <i>Project outline</i>	5
3.1. Project Presentation	5
3.2. What is Content Management System CMS?	5
3.3. What is Front-end	6
3.4. What is Back-end	6
3.5. Project objective	6
3.6. Project Description / Featured Modules	6
4. <i>Adopted methodology</i>	6
4.1. What is V-Model?	6
4.2. V-Model phases	7
4.3. Why adopting V-Model?	7
CONCLUSION:	7
<b>SECOND CHAPTER: GENERAL DESIGN SPECIFICATION</b>	<b>8</b>
INTRODUCTION	9
1. <i>Design language</i>	9
1.1. What is Design language?	9
1.2. What is UML?	9
2. <i>Adopted architecture</i>	10
2.1. MVC architecture	10
3. <i>Requirements definition</i>	10
3.1. Actors identification	10
3.1.1. What is an actor?	10
3.1.2. Actors identification	10
3.2. Functional requirements identification	11
3.2.1. Super Web Master Functionalities:	11
3.2.2. Web Master Functionalities:	11
3.2.3. Customer Functionalities:	11
3.2.4. Summary of use cases	11
3.3. Non-functional requirements identification	12
4. <i>Global use case diagram</i>	12
4.1. What is an Use Case diagram?	13
4.2. The Global use case diagram	13
5. <i>Use case by actor</i>	14
5.1. Webmaster's Use cases	14
5.2. Super Webmaster's Use cases	14
5.3. Customer's Use Case	16
6. <i>Textual description of the use cases</i>	16
6.1. Textual description of the Webmaster's use cases	16
6.2. Textual description of the Super Webmaster's use cases	20
6.3. Textual description of the Customer's use cases	25
6.4. Textual description of Common Use cases	30
CONCLUSION:	31
<b>THIRD CHAPTER: DETAILED DESIGN SPECIFICATION</b>	<b>32</b>
INTRODUCTION:	33
1. <i>Sequence Diagram</i>	33

1.1.	What is a Sequence Diagram?	33
1.2.	System Sequence Diagrams	33
1.2.1.	System sequence diagram of the use case "Authentication"	33
1.2.2.	System sequence diagram of the use case "Registration"	34
1.2.3.	System sequence diagram of the use case "Explore Catalog"	34
1.2.4.	System sequence diagram of the use case "Add Product to Cart (Panier)"	35
1.2.5.	System sequence diagram of the use case "Display shopping cart (Panier) details"	36
1.2.6.	System sequence diagram of the use case "Delete Product from Cart (Panier)"	36
1.2.7.	System sequence diagram of the use case "Buy Product"	36
1.2.8.	System sequence diagram of the use case "Empty Shopping Cart (Panier)"	38
1.2.9.	System sequence diagram of the use case "Request a Help"	38
1.2.10.	System sequence diagram of the use case "Display Webmaster's management page"	39
1.2.11.	System sequence diagram of the use case "Add Webmaster"	39
1.2.12.	System sequence diagram of the use case "Delete Webmaster"	40
1.2.13.	System sequence diagram of the use case "Search Webmaster"	40
1.2.14.	System sequence diagram of the use case "Display Product's management page"	41
1.2.15.	System sequence diagram of the use case "Add Product"	41
1.2.16.	System sequence diagram of the use case "Delete Product"	42
1.2.17.	System sequence diagram of the use case "Search Product"	42
1.2.18.	System sequence diagram of the use case "Update Product"	43
1.3.	Detailed Sequence Diagrams	43
1.3.1.	Detailed sequence diagram of the use case "Authentication"	43
1.3.2.	Detailed sequence diagram of the use case "Registration"	44
1.3.3.	Detailed sequence diagram of the use case "Explore Catalog"	44
1.3.4.	Detailed sequence diagram of the use case "Add Product to Cart (Panier)"	45
1.3.5.	Detailed sequence diagram of the use case "Display shopping cart (Panier) details"	46
1.3.6.	Detailed sequence diagram of the use case "Delete Product from Cart (Panier)"	46
1.3.7.	Detailed sequence diagram of the use case "Buy Product"	46
1.3.8.	Detailed sequence diagram of the use case "Empty Shopping Cart (Panier)"	48
1.3.9.	Detailed sequence diagram of the use case "Request a Help"	48
1.3.10.	Detailed sequence diagram of the use case "Display Webmaster's management page"	49
1.3.11.	Detailed sequence diagram of the use case "Add Webmaster"	49
1.3.12.	Detailed sequence diagram of the use case "Delete Webmaster"	51
1.3.13.	Detailed sequence diagram of the use case "Search Webmaster"	51
1.3.14.	Detailed sequence diagram of the use case "Display Product's management page"	52
1.3.15.	Detailed sequence diagram of the use case "Add Product"	52
1.3.16.	Detailed sequence diagram of the use case "Delete Product"	53
1.3.17.	Detailed sequence diagram of the use case "Search Product"	53
1.3.18.	Detailed sequence diagram of the use case "Update Product"	54
2.	<i>Class diagram</i>	54
2.1.	What is Class diagram?	54
2.2.	Class diagram schema	54
3.	<i>Relational Schema</i>	55
3.1.	What is Relational Schema?	55
3.2.	Relational Schema	56
4.	<i>Activity diagram</i>	56
4.1.	What is Activity diagram?	56
4.2.	Activity diagram	56
CONCLUSION:		57
<b>FORTH CHAPTER: SYSTEM IMPLEMENTATION</b>		58
INTRODUCTION:		59
1.	<i>Project architecture</i>	59
2.	<i>Design pattern</i>	60
3.	<i>Adopted programming languages</i>	60
3.1.	What is programming languages?	60
3.2.	Java EE	60
3.3.	HTML	60
3.4.	CSS	60
3.5.	JavaScript	60
4.	<i>Development environment</i>	61
4.1.	SAP PowerDesigner	61
4.2.	XAMPP	61
4.3.	Eclipse IDE	61

4.4.	Spring Framework _____	61
4.5.	JPA (Hibernate) _____	62
4.6.	JSF 2.0 (PrimeFaces)_____	62
4.7.	Maven _____	62
5.	<i>Deployment</i> _____	62
6.	<i>Navigation diagram</i> _____	63
6.1.	Super webmaster's navigation diagram _____	63
6.2.	Webmaster's navigation diagram _____	64
6.3.	Customer navigation diagram _____	64
7.	<i>Application interfaces</i> _____	64
7.1.	Login Interface _____	65
7.2.	Super Webmaster Interfaces _____	65
7.2.1.	Maintain Layout Interface _____	65
7.2.2.	Change website colors Interface _____	66
7.2.3.	Add new webmaster Interface _____	66
7.2.4.	Configure Database Interface _____	67
7.3.	Customer Interfaces _____	67
7.3.1.	Explore Catalog Interface _____	67
7.3.2.	View Product Details Interface _____	68
7.3.3.	Customer's Cart Interface _____	68
	CONCLUSION: _____	68

# TABLE OF FIGURES

FIGURE 1: LOGO OF UIB BANK.	3
FIGURE 2: MORE INFORMATION ABOUT UIB BANK.	3
FIGURE 3: LOGO OF WORDPRESS	4
FIGURE 4: LOGO OF JOOMLA!	4
FIGURE 5: LOGO OF DRUPAL	4
FIGURE 6: THE V-MODEL OF THE SYSTEMS ENGINEERING PROCESS.	7
FIGURE 7: THE DIFFERENT UML DIAGRAM TYPES	9
FIGURE 8: MVC INFRASTRUCTURE	10
FIGURE 9: GLOBAL USE CASE DIAGRAM	13
FIGURE 10: WEB MASTER'S USE CASE DIAGRAM	14
FIGURE 11: SUPER WEBMASTER'S USE CASE DIAGRAM	15
FIGURE 12: CUSTOMER'S USE CASE DIAGRAM	16
FIGURE 13: PRODUCTS MANAGEMENT USE CASE DIAGRAM	17
FIGURE 14: CUSTOMERS MANAGEMENT USE CASE DIAGRAM	18
FIGURE 15: PROCESSING A HELP REQUESTS USE CASE DIAGRAM	19
FIGURE 16: PROCESSING A RETURN REQUEST USE CASE DIAGRAM	19
FIGURE 17: WEBMASTERS MANAGEMENT USE CASE DIAGRAM	20
FIGURE 18: WEBSITE MAINTAINING USE CASE DIAGRAM	21
FIGURE 19: TEXTUAL DESCRIPTION OF THE USE CASE "MAINTAIN WEBSITE"	22
FIGURE 20: MAINTAIN ACCOUNT USE CASE	26
FIGURE 21: EXPLORE CATALOG USE CASE	27
FIGURE 22: MAINTAIN SHOPPING CART USE CASE	28
FIGURE 23: REQUESTING HELP USE CASE DIAGRAM	29
FIGURE 24: REQUESTING RETURN USE CASE DIAGRAM	30
FIGURE 25: "AUTHENTICATION" SYSTEM SEQUENCE DIAGRAM	33
FIGURE 26: "REGISTRATION" SYSTEM SEQUENCE DIAGRAM	34
FIGURE 27: "EXPLORE CATALOG" SYSTEM SEQUENCE DIAGRAM	35
FIGURE 28: "ADD PRODUCT TO CART (PANIER)" SYSTEM SEQUENCE DIAGRAM	35
FIGURE 29: "DISPLAY SHOPPING CART (PANIER) DETAILS" SYSTEM SEQUENCE DIAGRAM	36
FIGURE 30: "DELETE PRODUCT FROM CART (PANIER)" SYSTEM SEQUENCE DIAGRAM	36
FIGURE 31: "BUY PRODUCT(S)" SYSTEM SEQUENCE DIAGRAM	37
FIGURE 32: "EMPTY SHOPPING CART (PANIER)" SYSTEM SEQUENCE DIAGRAM	38
FIGURE 33: "REQUEST A HELP" SYSTEM SEQUENCE DIAGRAM	38
FIGURE 34: "DISPLAY WEBMASTER'S MANAGEMENT PAGE" SYSTEM SEQUENCE DIAGRAM	39
FIGURE 35: "ADD WEBMASTER" SYSTEM SEQUENCE DIAGRAM	39
FIGURE 36: "DELETE WEBMASTER" SYSTEM SEQUENCE DIAGRAM	40
FIGURE 37: "SEARCH WEBMASTER" SYSTEM SEQUENCE DIAGRAM	40
FIGURE 38: "DISPLAY PRODUCT'S MANAGEMENT PAGE" SYSTEM SEQUENCE DIAGRAM	41
FIGURE 39: "ADD PRODUCT" SYSTEM SEQUENCE DIAGRAM	41
FIGURE 40: "DELETE PRODUCT" SYSTEM SEQUENCE DIAGRAM	42
FIGURE 41: "SEARCH PRODUCT" SYSTEM SEQUENCE DIAGRAM	42
FIGURE 42: "UPDATE PRODUCT" SYSTEM SEQUENCE DIAGRAM	43
FIGURE 43: "AUTHENTICATION" DETAILED SEQUENCE DIAGRAM	43
FIGURE 44: "REGISTRATION" DETAILED SEQUENCE DIAGRAM	44
FIGURE 45: "EXPLORE CATALOG" DETAILED SEQUENCE DIAGRAM	45
FIGURE 46: "ADD PRODUCT TO CART (PANIER)" DETAILED SEQUENCE DIAGRAM	45
FIGURE 47: "DISPLAY SHOPPING CART (PANIER) DETAILS" DETAILED SEQUENCE DIAGRAM	46
FIGURE 48: "DELETE PRODUCT FROM CART (PANIER)" DETAILED SEQUENCE DIAGRAM	46
FIGURE 49: "BUY PRODUCT(S)" DETAILED SEQUENCE DIAGRAM	47
FIGURE 50: "EMPTY SHOPPING CART (PANIER)" DETAILED SEQUENCE DIAGRAM	48
FIGURE 51: "REQUEST A HELP" DETAILED SEQUENCE DIAGRAM	48
FIGURE 52: "DISPLAY WEBMASTER'S MANAGEMENT PAGE" DETAILED SEQUENCE DIAGRAM	49
FIGURE 53: "ADD WEBMASTER" DETAILED SEQUENCE DIAGRAM	50

FIGURE 54: "DELETE WEBMASTER" DETAILED SEQUENCE DIAGRAM	51
FIGURE 55: "SEARCH WEBMASTER" DETAILED SEQUENCE DIAGRAM	51
FIGURE 56: "DISPLAY PRODUCT'S MANAGEMENT PAGE" DETAILED SEQUENCE DIAGRAM	52
FIGURE 57: "ADD PRODUCT" DETAILED SEQUENCE DIAGRAM	52
FIGURE 58: "DELETE PRODUCT" DETAILED SEQUENCE DIAGRAM	53
FIGURE 59: "SEARCH PRODUCT" DETAILED SEQUENCE DIAGRAM	53
FIGURE 60: "UPDATE PRODUCT" DETAILED SEQUENCE DIAGRAM	54
FIGURE 61: CLASS DIAGRAM	55
FIGURE 62: RELATIONAL SCHEMA	56
FIGURE 63: ACTIVITY DIAGRAM	57
FIGURE 64: THREE TIER ARCHITECTURE DIAGRAM	59
FIGURE 65: 3.1. SAP POWERDESIGNER LOGO	61
FIGURE 66: XAMPP LOGO	61
FIGURE 67: ECLIPSE IDE LOGO	61
FIGURE 68: SPRING LOGO	61
FIGURE 69: HIBERNATE LOGO	62
FIGURE 70: PRIMEFACES LOGO	62
FIGURE 71: MAVEN LOGO	62
FIGURE 72: DEPLOYMENT DIAGRAM	63
FIGURE 73: SUPER WEBMASTER'S NAVIGATION DIAGRAM	64
FIGURE 74: WEBMASTER'S NAVIGATION DIAGRAM	64
FIGURE 75: CUSTOMER NAVIGATION DIAGRAM	64
FIGURE 76: LOGIN INTERFACE	65
FIGURE 77: MAINTAIN LAYOUT INTERFACE	65
FIGURE 78: CHANGE WEBSITE COLORS INTERFACE	66
FIGURE 79: ADD NEW WEBMASTER INTERFACE	66
FIGURE 80: CONFIGURE DATABASE INTERFACE	67
FIGURE 81: EXPLORE CATALOG INTERFACE	67
FIGURE 82: VIEW PRODUCT DETAILS INTERFACE	68
FIGURE 83: CUSTOMER'S CART INTERFACE	68

# TABLE OF TABLES

TABLE 1: SUMMARY OF USE CASES	12
TABLE 2: TEXTUAL DESCRIPTION OF THE USE CASE "ADD NEW PRODUCT"	17
TABLE 3: TEXTUAL DESCRIPTION OF THE USE CASE "UPDATE PRODUCT"	17
TABLE 4: TEXTUAL DESCRIPTION OF THE USE CASE "DELETE PRODUCT"	18
TABLE 5: TEXTUAL DESCRIPTION OF THE USE CASE "SEARCH PRODUCT"	18
TABLE 6: TEXTUAL DESCRIPTION OF THE USE CASE "DELETE CUSTOMER"	19
TABLE 7: TEXTUAL DESCRIPTION OF THE USE CASE "SEARCH CUSTOMER"	19
TABLE 8: TEXTUAL DESCRIPTION OF THE USE CASE "PROCESS HELP REQUEST"	19
TABLE 9: TEXTUAL DESCRIPTION OF THE USE CASE "PROCESS RETURN REQUEST"	20
TABLE 10: TEXTUAL DESCRIPTION OF THE USE CASE "ADD NEW WEBMASTER"	20
TABLE 11: TEXTUAL DESCRIPTION OF THE USE CASE "DELETE WEBMASTER"	21
TABLE 12: TEXTUAL DESCRIPTION OF THE USE CASE "ADD NEW PAGE"	22
TABLE 13: TEXTUAL DESCRIPTION OF THE USE CASE "DELETE PAGE"	23
TABLE 14: TEXTUAL DESCRIPTION OF THE USE CASE "UPDATE PAGE"	23
TABLE 15: TEXTUAL DESCRIPTION OF THE USE CASE "CONFIGURE DATABASE"	23
TABLE 16: TEXTUAL DESCRIPTION OF THE USE CASE "CHANGE WEBSITE LAYOUT"	24
TABLE 17: TEXTUAL DESCRIPTION OF THE USE CASE "ADD NEW CATEGORY"	24
TABLE 18: TEXTUAL DESCRIPTION OF THE USE CASE "DELETE CATEGORY"	24
TABLE 19: TEXTUAL DESCRIPTION OF THE USE CASE "UPDATE LOGO"	25
TABLE 20: TEXTUAL DESCRIPTION OF THE USE CASE "CHANGE COLORS"	25
TABLE 21: TEXTUAL DESCRIPTION OF THE USE CASE "CHANGE FONT"	25
TABLE 22: TEXTUAL DESCRIPTION OF THE USE CASE "CREATE ACCOUNT"	26
TABLE 23: TEXTUAL DESCRIPTION OF THE USE CASE "UPDATE ACCOUNT"	26
TABLE 24: TEXTUAL DESCRIPTION OF THE USE CASE "DELETE ACCOUNT"	27
TABLE 25: TEXTUAL DESCRIPTION OF THE USE CASE "EXPLORE CATALOG"	27
TABLE 26: TEXTUAL DESCRIPTION OF THE USE CASE "ADD PRODUCT TO CART"	28
TABLE 27: TEXTUAL DESCRIPTION OF THE USE CASE "DELETE PRODUCT FROM CART"	28
TABLE 28: TEXTUAL DESCRIPTION OF THE USE CASE "EMPTY CART"	29
TABLE 29: TEXTUAL DESCRIPTION OF THE USE CASE "BUY PRODUCTS"	29
TABLE 30: TEXTUAL DESCRIPTION OF THE USE CASE "REQUEST HELP"	30
TABLE 31: TEXTUAL DESCRIPTION OF THE USE CASE "REQUEST RETURN"	30
TABLE 32: TEXTUAL DESCRIPTION OF THE USE CASE "AUTHENTICATION"	31

# General Introduction

Nowadays, the Internet became so popular, everyday more and more people are consulting the web for various purposes. Indeed, it affects and facilitates every field of the modern life. The same goes to the Electronic Commerce (E-Commerce), it has been affected positively and became far easier. So nowadays, we can see enterprises with big and small sizes entering national as well as international markets, without any problem caused by the limited geographic frontier. With the help of Electronic Commerce, even small sized enterprises has gained the chance and acceded to the international markets to buy and sell different products and services. Without time restriction, Electronic Commerce allows to execute commercial transactions, 24 hours per day and 7days per week, it does not matter whether it is a working or non-working days.

An IT (Information technology) enterprise can easily create its own website, but the other enterprises or even individuals might find this bothersome. It would be highly desirable if there was a system that could enable enterprise members to update the desired information on their website on their own and whenever they want. This way, they can use the internet to communicate with their clients much more effectively, maintain up to date their information of any product/service, upcoming events like discounts...

Within my internship period at the Bank “Union Internationale de Banques” (UIB), we were entrusted with the task of Conception, Development of a web application, which contain very important phase, which is the “Content Management System” (CMS). In other word, it's web application with a “Backend” administration dashboard, where every non-technical person with a little knowledge on computers can update and maintain a shopping web site.

The thesis report describes development life cycle of back-ended e-commerce web application (Analysis, Design, Implementation and Deployment phases) with Use Case Sequences, activities, class and database design diagrams. Further, it details the system overview, detailed system description with its architecture.

# First Chapter: Requirements Analysis

## Introduction:

This chapter is divided into four sections. The first section presents the host organization ‘Union Internationale de Banques (UIB)’, second section study of the existing systems, the third section 3 presents the project’s outline and the last section will present the adopted methodology.

### 1. Presentation of the host organization



Figure 1: Logo of UIB Bank.

The Union Internationale de Banques (UIB): was incorporated in 1963 and privatized in November 2002 with the repurchase of part of its capital by Societe Generale group. UIB is one of the biggest private banks in Tunisia.

UIB is a universal bank that offers a complete range of products and services to all customer segments. A benchmark in its sector, the bank is able to draw upon the support and resources of Societe Generale group to constantly reinforce its strengths and reach.

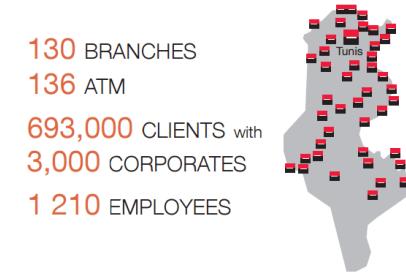


Figure 2: More information about UIB Bank.

UIB currently has a network of 130 points of sale and dedicated sales resources for corporate and individual clients.

### 2. Analysis of existing systems / study of the existing

#### 2.1. Existing Solutions

There's are various ways to build a web site, the most common options are, hiring a web site developer or using a content management system (CMS).

##### 2.1.1. Web developers

In an agency or as a freelancer, a web developer's job is to create web sites or web applications that meet clients' unique needs. The role of building a web site involve multi-step process, which include innovating a new design, writing the code, adding animation (if needed), checking for bugs and fixing them, presenting the client with a test site and moving the site to the client's server.

## 2.1.2. Most famous Content Management System (CMS)

- ~ **WordPress**



WordPress is a free and open-source, a well-architected content management system (CMS) built on PHP and MySQL and licensed under the GPLv2 (or later). Its founders Matt Mulling and Mike Little released it, on May 27, 2003. Since then, WordPress became more popular and one of the most used content management system (CMS) on the internet.

Some Notable Big Name Brands that are Using WordPress: NASA, CNN, NEW YORK TIMES...

- ~ **Joomla!**



*Figure 4: Logo of Joomla!*

Joomla! is a content management system, developed with PHP by Open Source Matters and it was released on 17 August 2005, its database management system is MySQL. Joomla! is free, open source.

Some Notable Big Name Brands that are Using Joomla! : Linux.com, Harvard University - The Graduate School of Arts and Sciences...

- ~ **Drupal**



*Figure 5: Logo of Drupal*

Drupal is free, open source, content management system (CMS), which provides excellent flexibility to share, distribute, and modify content. Drupal developed with PHP as well as MySQL as a database management system. It was released on 18 May 2000.

Some Notable Big Name Brands that are Using Drupal: The White House, University of Oxford

### ~ Other Content Management System (CMS)

- Magento
- DotNetNuke
- ExpressionEngine
- TextPattern
- Xoops

### 2.2. Criticism of existing system

Hiring web developer or using content management system (CMS), both, have some disadvantages:

Hiring web developer is actually quite costly. In addition, the product might not meet with requested needs ...

Moreover, users might not be able to maintain their own websites. In other word, they will be calling the developer every time they need to update their website, which add more cost.

The biggest problem during using existing content management system (CMS) like (WordPress or Joomla) , is that the users will be limited with what to do with their web sites, unless using a built-in plugin to add some features to customize it, while adding plugins require web developer to build it!

## 3. Project outline

In this section, we are going to take look at the project's presentation, its objectives, and its description.

### 3.1. Project Presentation

Within the realization of our graduation project to obtain a professional master's degree in "Modeling, Databases and Systems Integration" (MBDS) from the Higher School of Digital Economics (ESEN). Within my internship period at the Bank "Union Internationale de Banques" (UIB), we were entrusted with the task of Conception, Development of a web application, which contain very important phase, this phase is the "Content Management System" (CMS). In other word, it's website with a "Backend" administration dashboard, where every person with a little knowledge on computers can update and maintain a shopping web site.

This project aims to improve the process of creating a website without the need of second party.

The web application can be divided into two big parties; the first part is the Back-end administration platform (Back-Office) which belongs to Super-Webmaster and the Webmaster. The second part is the Front-end (Front-Office) which belong the commercial services offered to the customer.

This project will be developed with JAVA Enterprise Edition (JEE) environment and MySQL databases.

### 3.2. What is Content Management System CMS?

A content management system (CMS) supports the creation, management, distribution, publishing, and discovery of corporate information. It covers the complete lifecycle of the pages on your site, from providing simple tools to create the content, through to publishing, and finally to archiving. It also provides the ability to manage the structure of the site, the appearance of the published pages, and the navigation provided to the users.

### 3.3. What is Front-end

A website's front-end is the part that users see on the browsers and interact with, also known as the "client-side".

### 3.4. What is Back-end

The back-end is the side that makes the front-end possible, it contains three parts: server, application and the database. Also known as the "server-side".

### 3.5. Project objective

- ~ To make an opportunity to the clients of Bank "UIB" to create their own shopping sites, with no programming knowledge required.
- ~ Secure online payment.
- ~ Website with fully customizable design option.
- ~ To make a website that easy to manage, safe and secure.

### 3.6. Project Description / Featured Modules

This project can be broken down into different but interrelated sections:

- ~ **The backend administration platform:**

The content management system part, the dashboard that allows the website's owner to customize and update the website: add page, change the Logo, and change the colors ...

- ~ **Purchase management module:**

This module deals with online purchasing, shopping cart management, and payment for the registered customer on the supplier's website.

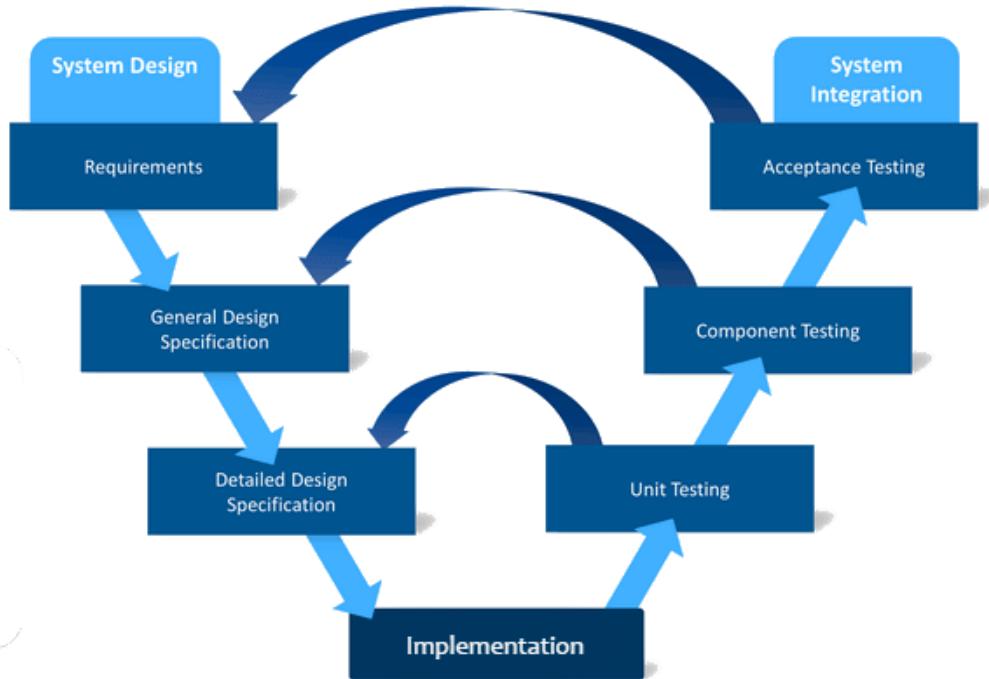
## 4. Adopted methodology

In hope that the aimed goals to be properly achieved and that the development process to be easier, we ought to adopt a "Software development process" method and to follow its every single step.

The adopted methodology during the development of this project is shown in Figure 5. To accomplish this project, the V-model will be followed, for the reason that, it is the most suitable methodology for medium sized projects.

### 4.1. What is V-Model?

Also known as Verification and Validation model, V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.



*Figure 6: The V-model of the systems engineering process.*

#### 4.2. V-Model phases

Those are the different phases of V-Models:

- ~ System Design Phases: requirements, general design specification, detailed design specification.
- ~ Implementation Phase.
- ~ System Integration Phases: Unit testing, Component Testing, Acceptance Testing.

#### 4.3. Why adopting V-Model?

This methodology has been chosen because:

- ~ The requirements are clear, simple, and well understood for the project.
- ~ Can be developed gradually.
- ~ It is suitable for medium sized projects.

## Conclusion:

We have studied during this chapter to give a global view of our project, through analyzing the host organization needs, criticizing the existing solution, proposing as solutions ... which made us ready to take the next step, the General design specification, which is the subject of the next chapter.

# Second chapter: General design specification

## Introduction:

In the previous chapter, we have introduced our project context. This chapter, which **represents the second phase of the V-Model called “general design specification”** will turn our attention to the planning of this project. Which consist of specifying the fundamental planning to follow while realization the project.

Began by specifying System’s different actors, then functional and nonfunctional requirements, then the system’s functionalities, to the Use cases diagrams and their textual descriptions.

### 1. Design language

To start designing our web application, we need to initiate a design language (also known as modeling language).

#### 1.1. What is Design language?

A modeling language is any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. The rules are used for interpretation of the meaning of components in the structure.

During this work, we have adopted UML as a Design language.

#### 1.2. What is UML?

UML stands for Unified Modeling Language which is used in object oriented software engineering. Although typically used in software engineering it is a rich language that can be used to model an application structures, behavior, and even business processes. The following figure shows the different types of diagrams under the UML:

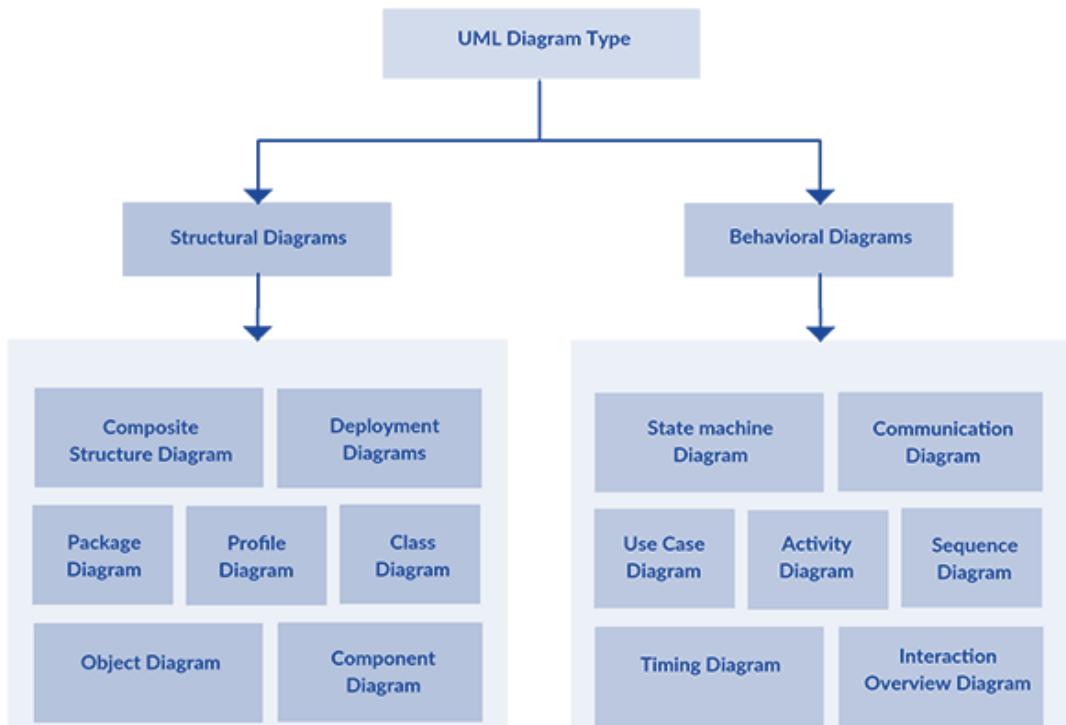


Figure 7: The different UML diagram types

During the designing of this work, we have used the use case diagram, sequence diagram, class diagram, activity diagram, and deployment diagram.

## 2. Adopted architecture

Our web application is based on the 3-tier architecture, which is a special type of client/server architecture consisting of three well-defined and separated processes, each running on a different platform. Therefore, we have adopted the MVC pattern.

### 2.1. MVC architecture

In object-oriented programming development, Model-View-Controller (MVC) is the name of a methodology or design pattern for successfully and efficiently relating the user interface to underlying data models.

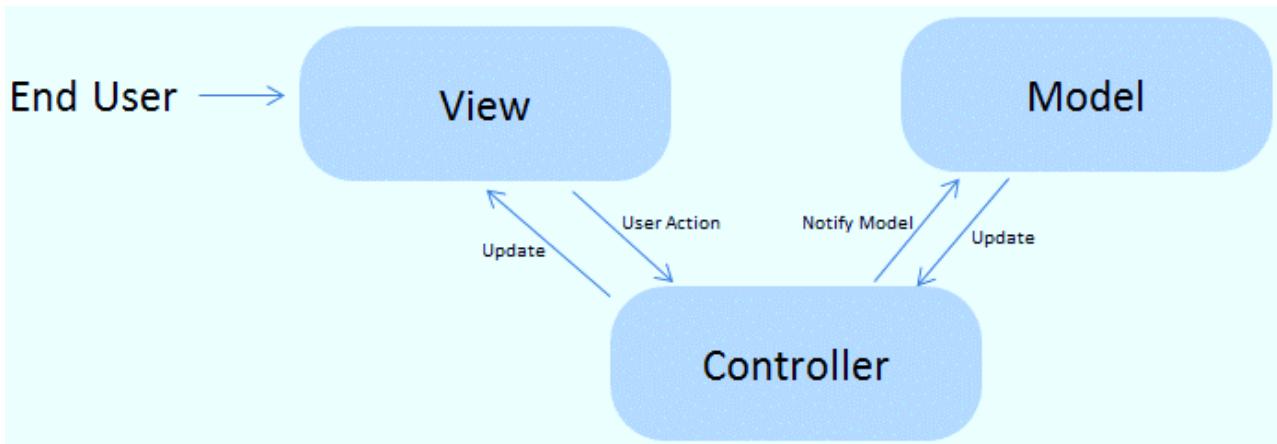


Figure 8: MVC infrastructure

The Model-View-Controller pattern proposes three main components or objects to be used in software development:

- ~ **Model:** Structures your data in a reliable form and prepares it based on controller's instructions
- ~ **View:** Displays data to user in easy-to-understand format, based on the user's actions
- ~ **Controller:** Takes in user requests, sends requests to the model for data updates, sends instructions to view to update interface.

## 3. Requirements definition

### 3.1. Actors identification

In this section, we are going to pay attention to our project's different actors:

#### 3.1.1. What is an actor?

An actor, in UML, specifies a role-played by an external entity as a human person, organization or an external system interacting with the system.

#### 3.1.2. Actors identification

In our project's case, we have three principal actor (Super Web Master, Web Master and customer), and only one single secondary actor, it is the payment system (UIB Bank).

The actors that will interact with our future system are:

- ~ **Super Web Master:** corresponds to the environment administrators. This actor has the highest privilege level that makes him able to handle all the system features.

- ~ **Web Master:** although he corresponds to the environment administrators, he has the lower privilege level compared to the Super Web Master.
- ~ **Customer:** is the one who is going to benefit from the services to make purchases after authenticating.

### 3.2. Functional requirements identification

In this part, we are going to identify functional requirements categorized by actors:

#### 3.2.1. Super Web Master Functionalities:

- ~ Authentication.
- ~ Manage Web Master: add a new webmaster, remove existing webmaster.
- ~ Maintain Web site: manage pages, manage themes, configure website...
- ~ Manage Customer: delete customer, search customer...
- ~ Manage Product: add, update, search, delete product...
- ~ Process Help Request: reply customer help's request...
- ~ Process Return Request: start processing the product returning if user not satisfied.

#### 3.2.2. Web Master Functionalities:

- ~ Authentication.
- ~ Manage Customer: delete customer, search customer...
- ~ Manage Product: add, update, search, delete product...
- ~ Process Help Request: reply customer help's request...
- ~ Process Return Request: start processing the product returning if user not satisfied.

#### 3.2.3. Customer Functionalities:

- ~ Register: create a new account.
- ~ Log in.
- ~ Maintain Account: update, cancel his account...
- ~ Explore Catalog.
- ~ Maintain Shopping Cart: add, delete item to shopping cart...
- ~ Buy Product.
- ~ Request Help: send help request in case of facing problem.
- ~ Request Return: send return request if the product did not reach the customer's satisfaction.

#### 3.2.4. Summary of use cases

<u>Actor</u>	<u>Use case</u>
<b>Super Web Master</b>	Authentication. Manage Web Masters. Maintain Web site. Manage Customers. Manage Products. Process Help Requests. Process Return Requests.

<b>Web Master</b>	Authentication. Manage Customers. Manage Products. Process Help Requests. Process Return Requests.
<b>Customer</b>	Register. Authentication. Maintain Account. Explore Catalog. Maintain Shopping Cart. Buy Product. Request Help. Request Return.

Table 1: Summary of use cases

### 3.3. Non-functional requirements identification

A non-functional requirement are some characteristics or aspects that the system must fulfill, which used to judge the system quality, but do not affect directly his functioning. Performance related issues, security issues and availability issues ... could be considered as non-functional attributes.

The non-functional requirements of our system are:

- ~ **Availability:** offering the advantage and the ability to accede to the website in 24/24hour and 7/7day.
- ~ **Security:** protecting the privacy and integrity of the website data, considered as one of the highest priorities
- ~ **Performance:** loading of the website and the response time remain a priority.
- ~ **Conviviality:** in order to facilitate the use of our system, user's interfaces should be clear and simple to interact with.
- ~ **Documentation:** without documentation, our system cannot be considered as finished. Hence, our system documentation must be well done.
- ~ **Error management:** in attempt to improve the system performance, the users will be given the chance to inform about the system's malfunctions.

## 4. Global use case diagram

In the previous section, we have defined our project requirements, such as actors and functionalities... therefore, now we are able to display our main system use case diagram, using Unified Modeling Language (UML).

## 4.1. What is an Use Case diagram?

A use case is a list of actions or event steps, defining the different interactions between an actor and a system that enables the user to achieve a goal.

## 4.2. The Global use case diagram

Our system's global use case diagram could be illustrated by the flowing image:

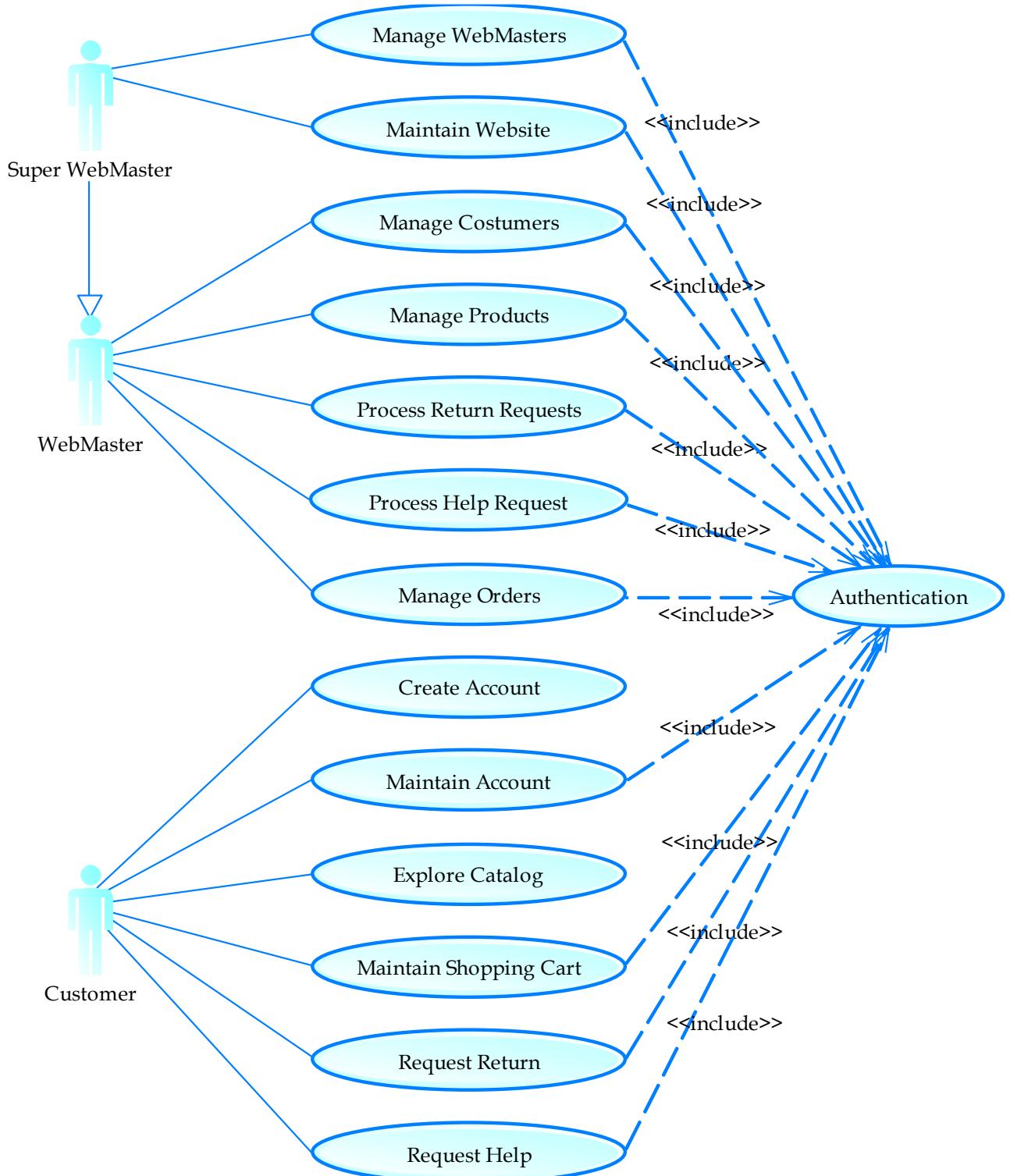


Figure 9: Global use case diagram

## 5. Use case by actor

### 5.1. Webmaster's Use cases

The following diagram present the web master's use cases:

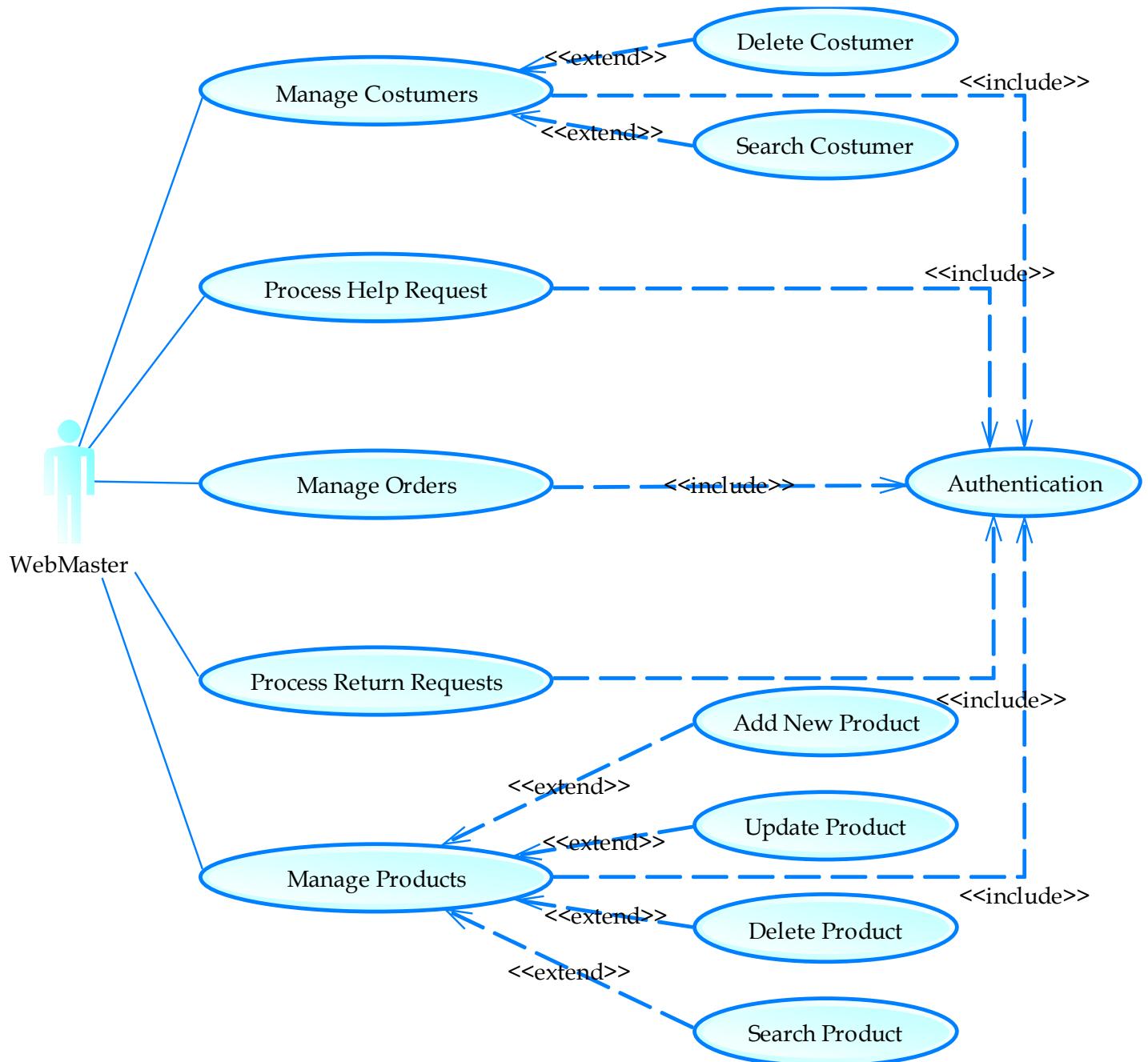


Figure 10: Web master's Use Case Diagram

### 5.2. Super Webmaster's Use cases

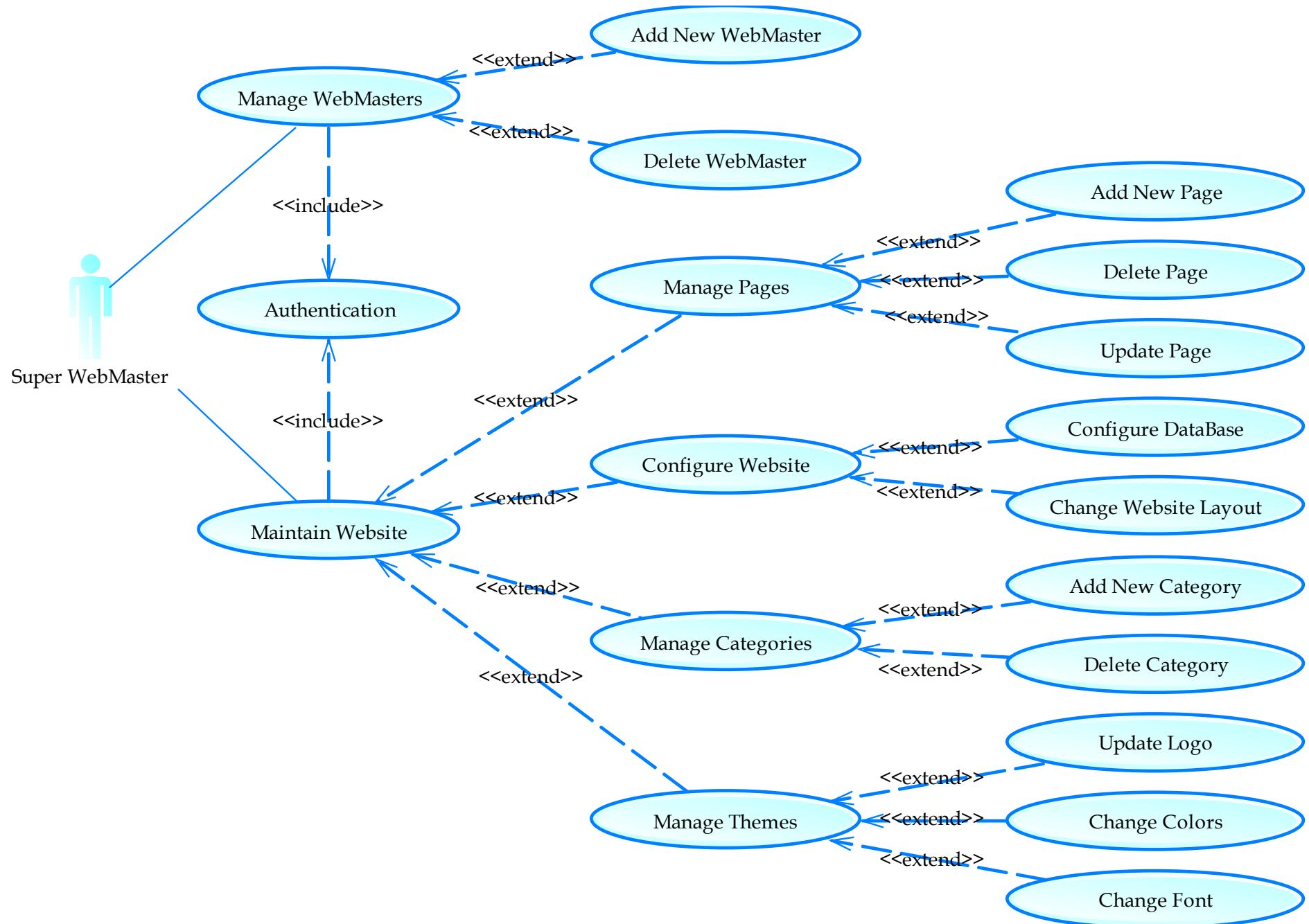


Figure 11: Super Webmaster's Use Case Diagram

### 5.3. Customer's Use Case

The following diagram presents the customer's use case

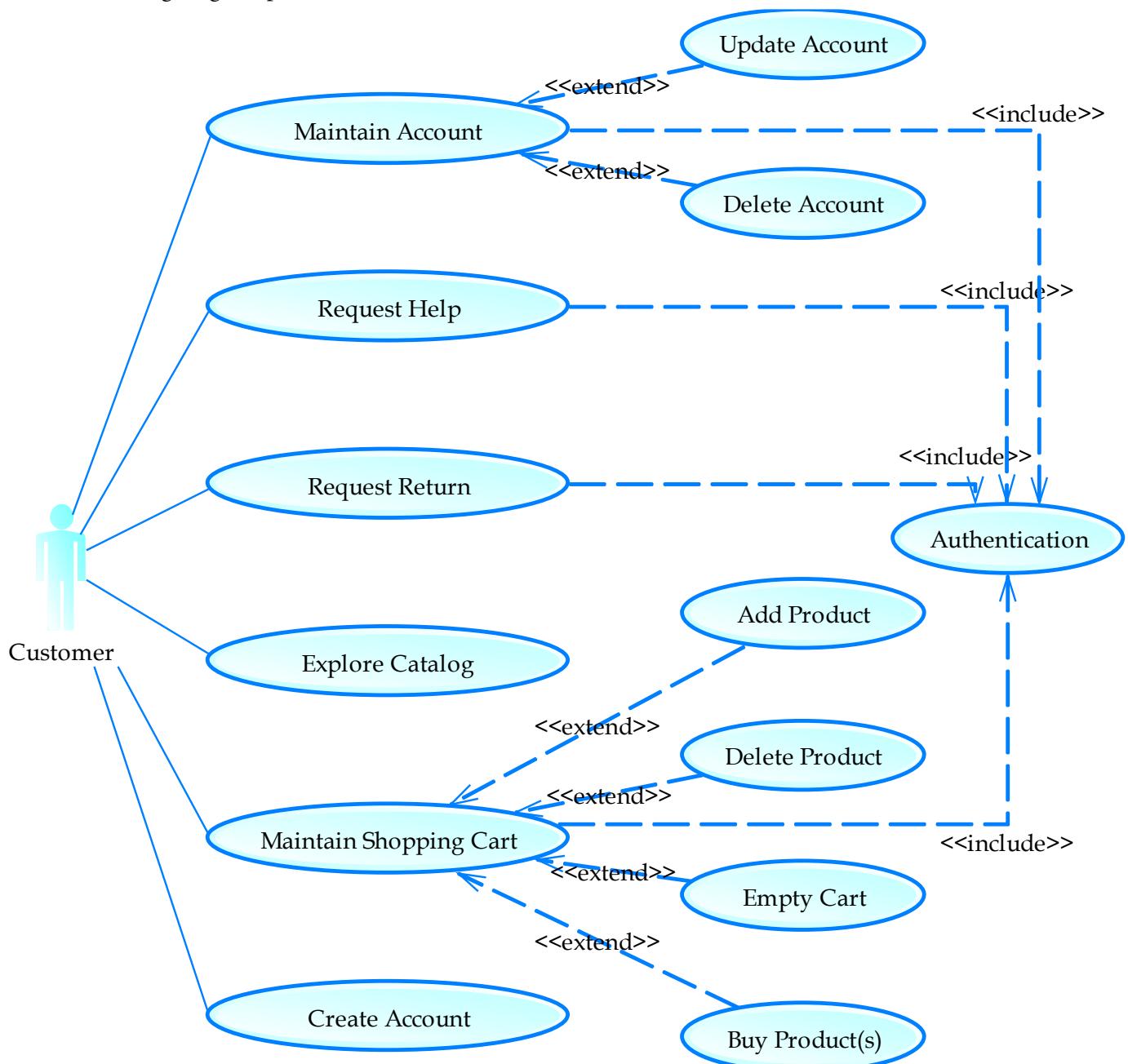


Figure 12: Customer's Use Case Diagram

## 6. Textual description of the use cases

Use cases are not just about diagrams. So in order to make our use cases easier and more understandable, we have adopted the “textual description” technique. Customer.

### 6.1. Textual description of the Webmaster's use cases

The table below describes all the interactions between the system and the Webmaster while realizing the use case “Products Management”:

~ Products Management Use Case

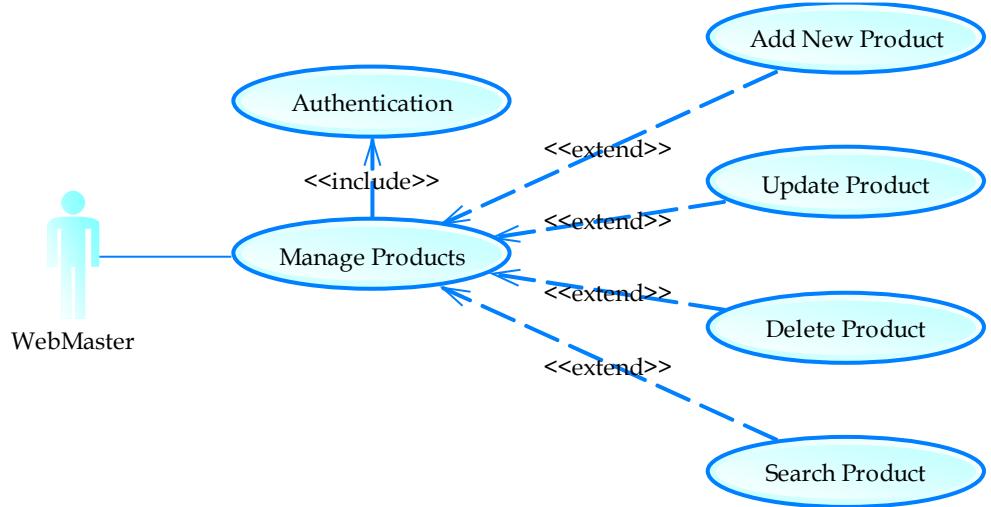


Figure 13: Products Management Use Case Diagram

Use Case	Add new Product
<b>Actor</b>	Webmaster, Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Product added with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Webmaster click on “add new product”.</li> <li>2. The system displays the product adding form.</li> <li>3. Webmaster fills the form then click on “add”.</li> <li>4. The system updates the new product details in the database (added with success).</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>4.a. If the webmaster fails to enter any of the mandatory product information, then the system displays an appropriate error message.</li> </ol>

Table 2: Textual description of the use case “Add New Product”

Use Case	Update Product
<b>Actor</b>	Webmaster, Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Product updated with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Webmaster search for the product to update and click on “Update product”.</li> <li>2. The system displays the product modifying form.</li> <li>3. Webmaster fills the form then click on “Update”.</li> <li>4. The system updates the product details in the database (updated with success).</li> </ol>
<b>Alternative scenario</b>	None.

Table 3: Textual description of the use case “Update Product”

Use Case	Delete Product
Actor	Webmaster, Super Webmaster.
Pre-condition	Being authenticated.
Post-condition	Product deleted with success.
Basic scenario	<ol style="list-style-type: none"> <li>1. Webmaster search for the product to delete and click on “Delete product”.</li> <li>2. The system display Confirmation dialog “Are you sure?”</li> <li>3. Webmaster click on “Yes”.</li> <li>4. The system removes the product from the database (deleted with success).</li> </ol>
Alternative scenario	<ol style="list-style-type: none"> <li>3.a. Webmaster click on “No”.</li> <li>3.b. The system cancels the operation.</li> </ol>

Table 4: Textual description of the use case “Delete Product”

Use Case	Search Product
Actor	Webmaster, Super Webmaster.
Pre-condition	Being authenticated.
Post-condition	Result list displayed.
Basic scenario	<ol style="list-style-type: none"> <li>1. Webmaster input searching criteria and click on “Search”.</li> <li>2. The system displays the search result.</li> </ol>
Alternative scenario	2.a. The system show “No result found” message.

Table 5: Textual description of the use case “Search Product”

### ~ Customers Management Use Case

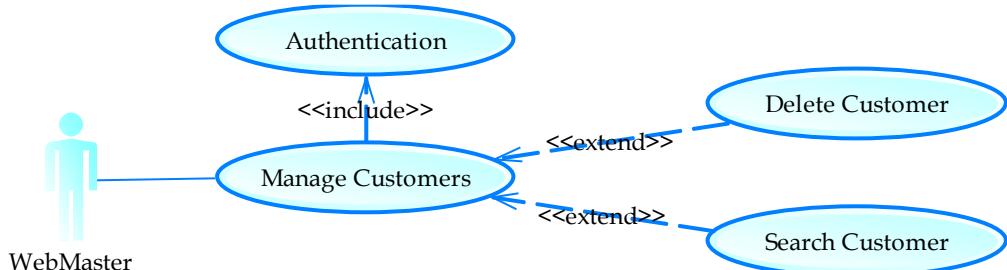


Figure 14: Customers Management Use Case Diagram

Use Case	Delete Customer
Actor	Webmaster, Super Webmaster.
Pre-condition	Being authenticated.
Post-condition	Customer deleted with success.
Basic scenario	<ol style="list-style-type: none"> <li>1. Webmaster search for the customer to delete and click on “Delete Customer”.</li> <li>2. The system display Confirmation dialog “Are you sure?”</li> <li>3. Webmaster click on “Yes”.</li> </ol>

	4. The system removes the customer from the database (deleted with success).
<b>Alternative scenario</b>	3.a. Webmaster click on “No”. 3.b. The system cancels the operation.

Table 6: Textual description of the use case “Delete Customer”

Use Case	Search Customer
<b>Actor</b>	Webmaster, Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Result list displayed.
<b>Basic scenario</b>	1. Webmaster input searching criteria and click on “Search”. 2. The system displays the search result.
<b>Alternative scenario</b>	2.a. The system show “No result found” message.

Table 7: Textual description of the use case “Search Customer”

### ~ Processing a Help Requests Use Case

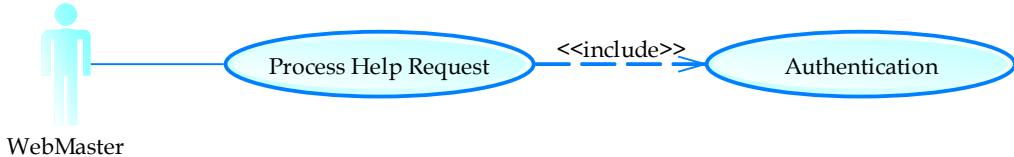


Figure 15: Processing a Help Requests Use Case Diagram

Use Case	Process Help Request
<b>Actor</b>	Webmaster, Super Webmaster.
<b>Pre-condition</b>	Being authenticated. An existing Help request placed by the customer.
<b>Post-condition</b>	The customer has received the desired help/support.
<b>Basic scenario</b>	1. Webmaster click on “Help Requests” button. 2. The system displays the list of received help requests. 3. Webmaster click on the desired request. 4. The system displays the request details. 5. Webmaster fills the response form and click on “send”.
<b>Alternative scenario</b>	2.a. The system show “No help requests received” message.

Table 8: Textual description of the use case “Process Help Request”

### ~ Processing a Return Requests Use Case

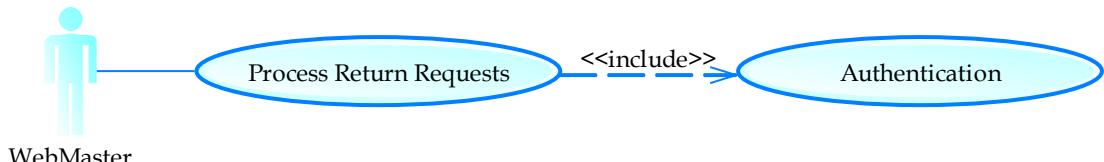


Figure 16: Processing a Return Request Use Case diagram

Use Case	Process Return Request
<b>Actor</b>	Webmaster, Super Webmaster.
<b>Pre-condition</b>	Being authenticated. The customer has purchased an item and wishes to return it.
<b>Post-condition</b>	The customer has received either approval to return the product, or a denial of the return request.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Webmaster click on “Return Requests” button.</li> <li>2. The system displays the list of received return requests.</li> <li>3. Webmaster click on the desired request.</li> <li>4. The system displays the request details.</li> <li>5. Webmaster chooses the response (“Yes” or “No”) and click on “send”.</li> </ol>
<b>Alternative scenario</b>	2.a. The system show “No return requests received” message.

Table 9: Textual description of the use case “Process Return Request”

## 6.2. Textual description of the Super Webmaster’s use cases

### ~ Webmasters Management Use Case

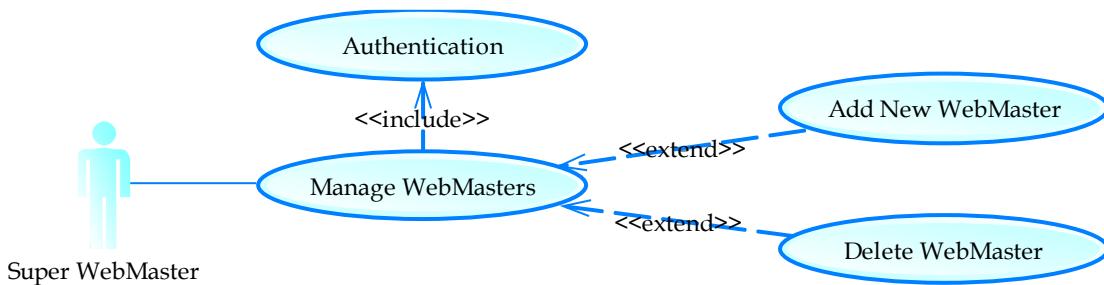


Figure 17: Webmasters Management Use Case Diagram

Use Case	Add new Webmaster.
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Webmaster added with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “add new Webmaster”.</li> <li>2. The system displays the Webmaster adding form.</li> <li>3. Super Webmaster fill the form then click on “add”.</li> <li>4. The system updates the new Webmaster details in the database (added with success).</li> </ol>
<b>Alternative scenario</b>	4.a. If the Super webmaster fails to enter any of the mandatory Webmaster information, then the system displays an appropriate error message.

Table 10: Textual description of the use case “Add new Webmaster”

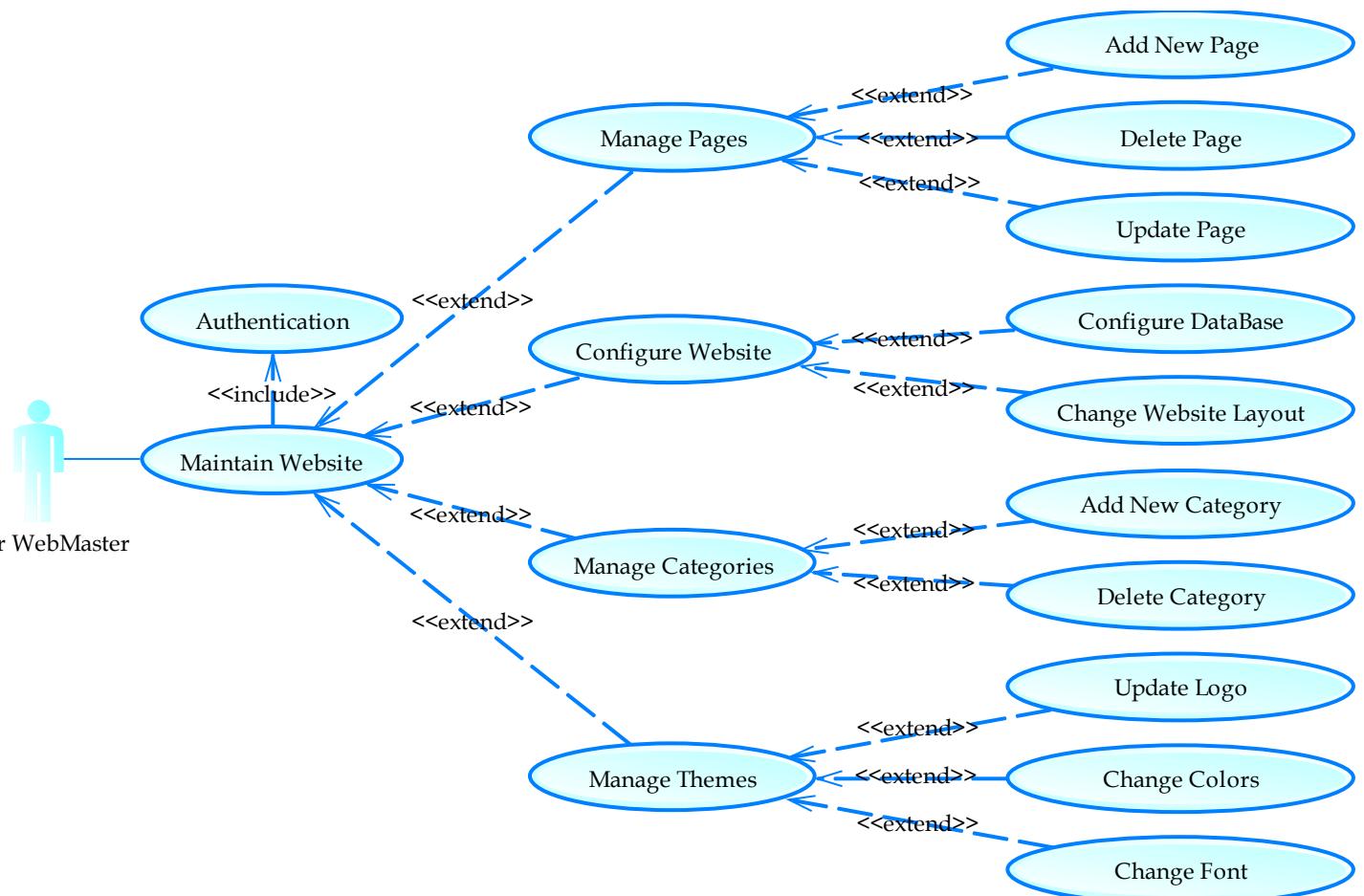


Figure 18: Website Maintaining Use Case Diagram

Use Case	Delete Webmaster
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Webmaster deleted with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster search for the Webmaster to delete and click on “Delete Webmaster”.</li> <li>2. The system display Confirmation dialog “Are you sure?”</li> <li>3. Super Webmaster click on “Yes”.</li> <li>4. The system removes the Webmaster from the database (deleted with success).</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>3.a. Super Webmaster click on “No”.</li> <li>3.b. The system cancels the operation.</li> </ol>

Table 11: Textual description of the use case “Delete Webmaster”

### ~ Website Maintaining Use Case

Use Case	Maintain Website
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.

<b>Post-condition</b>	Website maintained with success
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “Maintain Website”.</li> <li>2. The system prompts the Super Webmaster to select one of the following three options: Manage Pages, Configure website and manage Themes.</li> <li>3. Super Webmaster choose one of the following three options: <ul style="list-style-type: none"> <li><b>3.A.</b> Manage Pages,</li> <li><b>3.B.</b> Configure Website,</li> <li><b>3.C.</b> Manage Categories,</li> <li><b>3.D.</b> Manage Themes.</li> </ul> </li> </ol>
<b>Alternative scenario</b>	None

Figure 19: Textual description of the use case “Maintain Website”

**3.A.** If the Super Webmaster selects the “Manage Pages” option, the system prompts the Super Webmaster to select one of the following options: add new page, update Page, and delete Page.

Use Case	Add new Page.
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Page added with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “add new Page”.</li> <li>2. The system displays the Page adding form.</li> <li>3. Super Webmaster fill the form then click on “add”.</li> <li>4. The system updates the new Page details in the database (added with success).</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>4.a. If the Super webmaster fails to enter any of the mandatory Page information, then the system displays an appropriate error message.</li> </ol>

Table 12: Textual description of the use case “Add new Page”

Use Case	Delete Page
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Page deleted with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster search for the Page to delete and click on “Delete Page”.</li> <li>2. The system display Confirmation dialog “Are you sure?”</li> <li>3. Super Webmaster click on “Yes”.</li> <li>4. The system removes the Page from the database (deleted with success).</li> </ol>

<b>Alternative scenario</b>	3.c. Super Webmaster click on “No”. 3.d. The system cancels the operation.
-----------------------------	---

Table 13: Textual description of the use case “Delete Page”

Use Case	Update Page
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Page updated with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster search for the Page to update and click on “Update Page”.</li> <li>2. The system displays the Page modifying form.</li> <li>3. Super Webmaster fill the form then click on “Update”.</li> <li>4. The system updates the Page details in the database (updated with success).</li> </ol>
<b>Alternative scenario</b>	None.

Table 14: Textual description of the use case “Update Page”

3.B. If the Super Webmaster selects the “Configure Website” option, the system prompts the Super Webmaster to select one of the following options: Configure database, Change Website Layout.

Use Case	Configure database
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Database Configured with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “Configure database”.</li> <li>2. The system displays the database configuration form.</li> <li>3. Super Webmaster fill the form then click on “Submit”.</li> <li>4. The system updates the database details.</li> </ol>
<b>Alternative scenario</b>	None.

Table 15: Textual description of the use case “Configure database”

Use Case	Change Website Layout
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Website Layout Changed with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “Change Website Layout”.</li> <li>2. The system displays the website’s layout options.</li> <li>3. Super Webmaster pick the desired layout option then click on “Submit”.</li> <li>4. The system updates the website’s layout.</li> </ol>

<b>Alternative scenario</b>	None.
-----------------------------	-------

Table 16: Textual description of the use case “Change Website Layout”

3.C. If the Super Webmaster selects the “Manage Categories” option, the system prompts the Super Webmaster to select one of the following options: add new Category and delete Category.

Use Case	Add new Category.
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Category added with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “add new Category”.</li> <li>2. The system displays the Category adding form.</li> <li>3. Super Webmaster fill the form then click on “add”.</li> <li>4. The system updates the new Category details in the database (added with success).</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>4.a. If the Super webmaster fails to enter any of the mandatory Page information, then the system displays an appropriate error message.</li> </ol>

Table 17: Textual description of the use case “Add new Category”

Use Case	Delete Category
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Category deleted with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster search for the Category to delete and click on “Delete Category”.</li> <li>2. The system display Confirmation dialog “Are you sure?”</li> <li>3. Super Webmaster click on “Yes”.</li> <li>4. The system removes the Category from the database (deleted with success).</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>3.a. Super Webmaster click on “No”.</li> <li>3.b. The system cancels the operation.</li> </ol>

Table 18: Textual description of the use case “Delete Category”

3.D. If the Super Webmaster selects the “Manage Themes” option, the system prompts the Super Webmaster to select one of the following options: Update Logo, Change colors and Change Font.

Use Case	Update Logo
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.

<b>Post-condition</b>	Logo updated with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “Update Logo”.</li> <li>2. The system displays the Logo updating form.</li> <li>3. Super Webmaster picks the desired Logo file (.png) to upload it then click on “Upload Logo”.</li> <li>4. The system uploads the new Logo and update the website.</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>4.a. If the Super webmaster fails to pick a file with a “png” extension, then the system displays an appropriate error message.</li> </ol>

Table 19: Textual description of the use case “Update Logo”

Use Case	Change Colors
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Colors changed with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “Change Colors”</li> <li>2. The system displays the color picker.</li> <li>3. Super Webmaster pick the desired Color then click on “Change Color”.</li> <li>4. The system updates the website colors.</li> </ol>
<b>Alternative scenario</b>	None

Table 20: Textual description of the use case “Change Colors”

Use Case	Change Font
<b>Actor</b>	Super Webmaster.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Font Changed with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Super Webmaster click on “Change Font”.</li> <li>2. The system displays the font changing form.</li> <li>3. Super Webmaster pick an existing font from the default list, or choose a desired personal font file (.ttf) to upload it then click on “Change Font”.</li> <li>4. The system changes the font and updates the website.</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>3.a. If the Super webmaster fails to pick a file with a “ttf” extension, then the system displays an appropriate error message.</li> </ol>

Table 21: Textual description of the use case “Change Font”

### 6.3. Textual description of the Customer’s use cases

#### ~ Account Maintaining Use Case

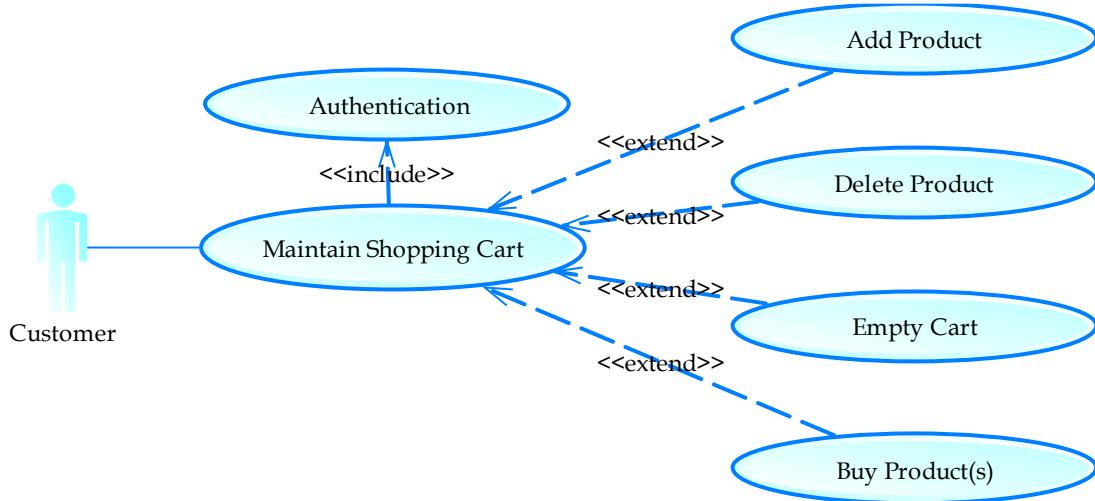


Figure 20: Maintain Account Use Case

Use Case	Create Account.
Actor	Customer.
Pre-condition	None.
Post-condition	Account created with success.
Basic scenario	<ol style="list-style-type: none"> <li>1. Customer clicks on “Register”.</li> <li>2. The system displays the registration form.</li> <li>3. Customer fills the form then click on “Submit”.</li> <li>4. The system updates the new customer details in the database (added with success).</li> </ol>
Alternative scenario	4.a. If the customer fails to enter any of the mandatory personal information, then the system displays an appropriate error message.

Table 22: Textual description of the use case “Create Account”

Use Case	Update Account
Actor	Customer.
Pre-condition	Being authenticated.
Post-condition	Account updated with success.
Basic scenario	<ol style="list-style-type: none"> <li>1. Customer clicks on “Update Account”.</li> <li>2. The system displays the Account modifying form.</li> <li>3. Customer fills the form then click on “Update”.</li> <li>4. The system updates the account details in the database (updated with success).</li> </ol>
Alternative scenario	None.

Table 23: Textual description of the use case “Update Account”

Use Case	Delete Account
<b>Actor</b>	Customer.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Account deleted with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Customer clicks on “Delete Customer”.</li> <li>2. The system display Confirmation dialog “Are you sure?”</li> <li>3. Customer clicks on “Yes”.</li> <li>4. The system removes the customer from the database (deleted with success).</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>3.a. Customer clicks on “No”.</li> <li>3.b. The system cancels the operation.</li> </ol>

Table 24: Textual description of the use case “Delete Account”

### ~ Explore Catalog Use Case

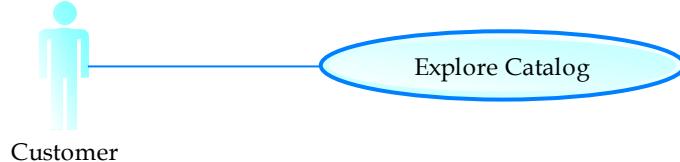


Figure 21: Explore Catalog Use Case

Use Case	Explore Catalog.
<b>Actor</b>	Customer.
<b>Pre-condition</b>	None.
<b>Post-condition</b>	Account created with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Customer clicks the “Home” page of the website.</li> <li>2. The system displays the catalog with different categories.</li> <li>3. Customer Choose a category.</li> <li>4. The system display list of the products depending on the chosen category.</li> <li>5. Customer clicks on the desired product’s link</li> <li>6. The system displays the product’s details.</li> </ol>
<b>Alternative scenario</b>	None.

Table 25: Textual description of the use case “Explore Catalog”

### ~ Shopping Cart Maintaining Use Case

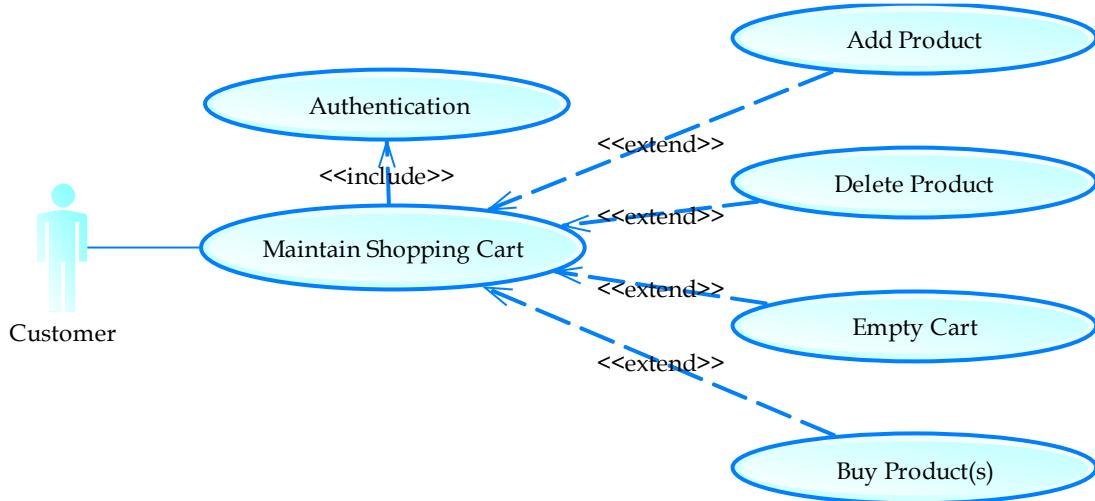


Figure 22: Maintain Shopping Cart Use Case

Use Case	Add Product to Cart
<b>Actor</b>	Customer.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Product added to cart with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Customer navigates to the product he wishes to add to the cart and click on “Add to Cart”.</li> <li>2. The system updates the cart details in the database (added with success).</li> </ol>
<b>Alternative scenario</b>	None.

Table 26: Textual description of the use case “Add Product to Cart”

Use Case	Delete Product from Cart
<b>Actor</b>	Customer.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Product deleted from the cart with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Customer chooses the unwanted product and click on “Remove”.</li> <li>2. The system display Confirmation dialog “Are you sure?”</li> <li>3. Customer clicks on “Yes”.</li> <li>4. The system removes the product from the cart database (deleted with success).</li> </ol>
<b>Alternative scenario</b>	3.a. Customer clicks on “No”. 3.b. The system cancels the operation.

Table 27: Textual description of the use case “Delete Product from Cart”

Use Case	Empty Cart
<b>Actor</b>	Customer.
<b>Pre-condition</b>	Being authenticated.

<b>Post-condition</b>	Cart emptied with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>Customer navigates to shopping cart details and click on “Empty Cart”.</li> <li>The system display Confirmation dialog “Are you sure?”</li> <li>Customer clicks on “Yes”.</li> <li>The system removes the entire product from the cart database (deleted with success).</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>Customer clicks on “No”.</li> <li>The system cancels the operation.</li> </ol>

Table 28: Textual description of the use case “Empty Cart”

Use Case	Buy Product(s)
<b>Actor</b>	Customer.
<b>Pre-condition</b>	Being authenticated.
<b>Post-condition</b>	Product bought with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>Customer navigates to shopping cart details and select the products he wish to buy (or click on “Select All”) then click on “Buy Products”.</li> <li>The system displays the payment form.</li> <li>Customer enters the requested information (Credit card number, Credit expiry date, Card type ...) then click on “Submit”.</li> <li>The System displays ‘Confirm the payment information again’ to the Customer.</li> <li>Customer clicks on 'Confirm'.</li> <li>System sends a confirmation message.</li> </ol>
<b>Alternative scenario</b>	<ol style="list-style-type: none"> <li>Customer clicks on “No”.</li> <li>The system cancels the operation.</li> </ol>

Table 29: Textual description of the use case “Buy Products”

### ~ Requesting Help Use Case

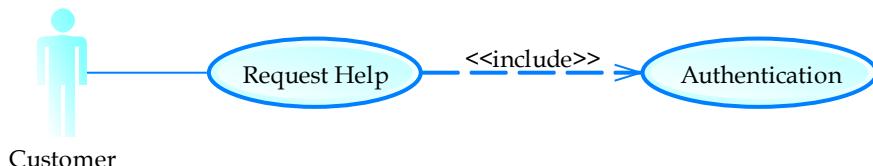


Figure 23: Requesting Help Use Case Diagram

Use Case	Request Help
<b>Actor</b>	Customer.
<b>Pre-condition</b>	Being authenticated.

<b>Post-condition</b>	Help request sent with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Customer navigates to the help requesting page.</li> <li>2. The system display the request form</li> <li>3. Customer fills the form then click on “Send Request”.</li> <li>4. The system send the request to the Webmasters</li> </ol>
<b>Alternative scenario</b>	None.

Table 30: Textual description of the use case “Request Help”

### ~ Requesting Return Use Case

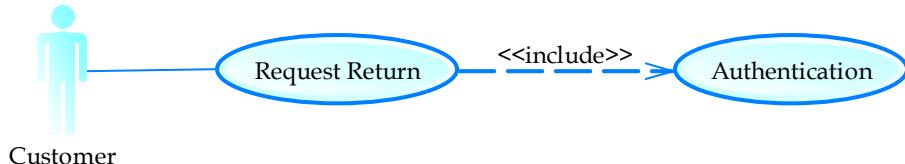


Figure 24: Requesting Return Use Case Diagram

Use Case	Request Help
<b>Actor</b>	Customer.
<b>Pre-condition</b>	<p>Being authenticated.</p> <p>Customer received the product, but it didn't reach its satisfaction</p>
<b>Post-condition</b>	Return request sent with success.
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. Customer navigates to the return requesting page then click on “Send Return Request”.</li> <li>2. The system send the request to the Webmasters</li> </ol>
<b>Alternative scenario</b>	None.

Table 31: Textual description of the use case “Request Return”

### 6.4. Textual description of Common Use cases

The only common use case shared between the system's different actors is the “Authentication” Use case. Every user in our application is demanded to authenticate.

Use Case	Authentication
<b>Actor</b>	Super Webmaster, Webmaster, Customer.
<b>Pre-condition</b>	None
<b>Post-condition</b>	User Authenticated with Success
<b>Basic scenario</b>	<ol style="list-style-type: none"> <li>1. User navigates to the authentication page.</li> <li>2. The system displays the authentication form.</li> <li>3. User fills the required data and click on “Sign In”.</li> </ol>

	4. The system checks the inputted data with the stored data and open the user's session depending on his Role (Super Webmaster / Webmaster / Customer).
<b>Alternative scenario</b>	4.a. If the User fails to enter any of the mandatory personal information, or the entered information is not compatible with the stored data, then the system displays an appropriate error message.

*Table 32: Textual description of the use case “Authentication”*

## Conclusion:

Specifying our application actors, functional, and non-functional requirements cleared our vision and made it possible to understand it better. Now we are ready to make another step in our project, to the Detailed Design specification, where we are going to go deeper in our application design.

# Third chapter: Detailed Design specification

## Introduction:

In the previous chapter, we have introduced our project requirements and architecture. Now, this chapter will turn our attention to the detailed design of this project. We are going to design the sequence diagrams, detailed sequence diagrams, navigation diagram, and class diagram.

This chapter represents the Third phase of the V-Model called “Detailed Design specification”.

### 1. Sequence Diagram

#### 1.1. What is a Sequence Diagram?

Sequence diagram is the most common kind of interaction diagram, which represents the sequence of messages between instances of classes, components, subsystems, or actors. With their interactions over time represented as messages drawn as arrows from the source to the target. Sequence diagrams are good at showing, which objects communicate with which other objects and what messages trigger those communications.

#### 1.2. System Sequence Diagrams

##### 1.2.1. System sequence diagram of the use case “Authentication”

The interactions between our system and the users (Super Webmaster, Webmaster and Customer) while authenticating is illustrated in the following figure

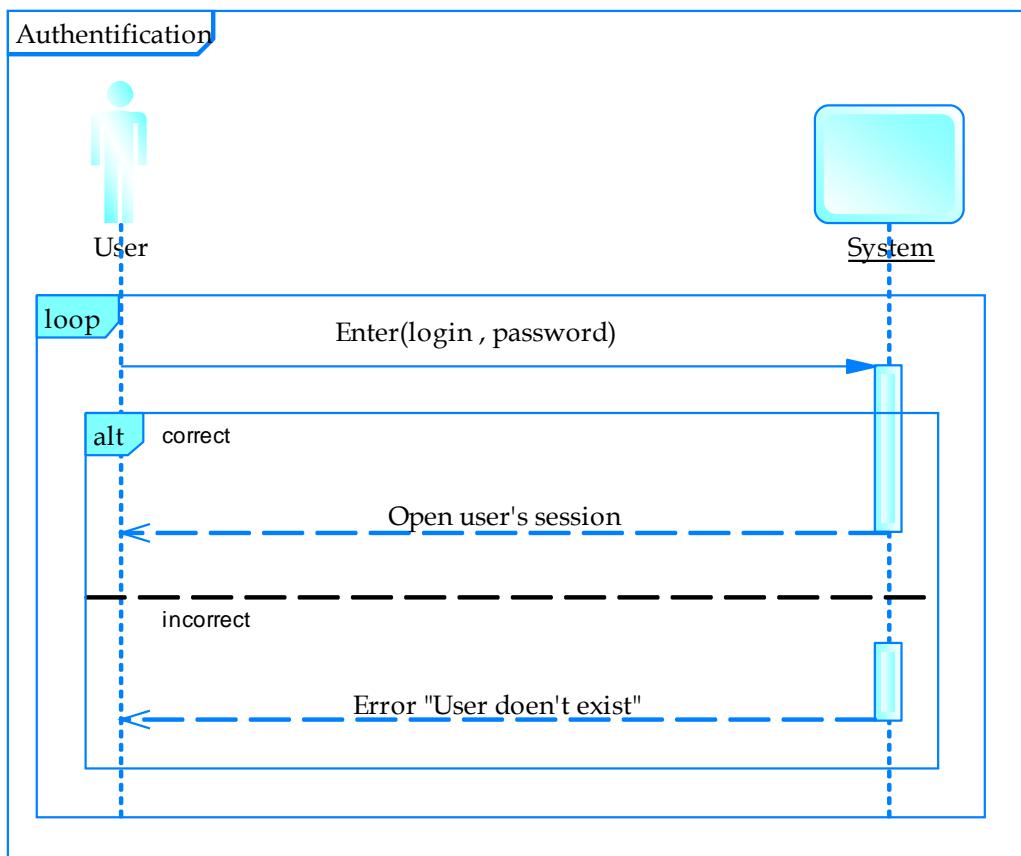


Figure 25: "Authentication" System Sequence Diagram

### 1.2.2. System sequence diagram of the use case “Registration”

The interactions between our system and the Customer while creating an account is illustrated in the following figure:

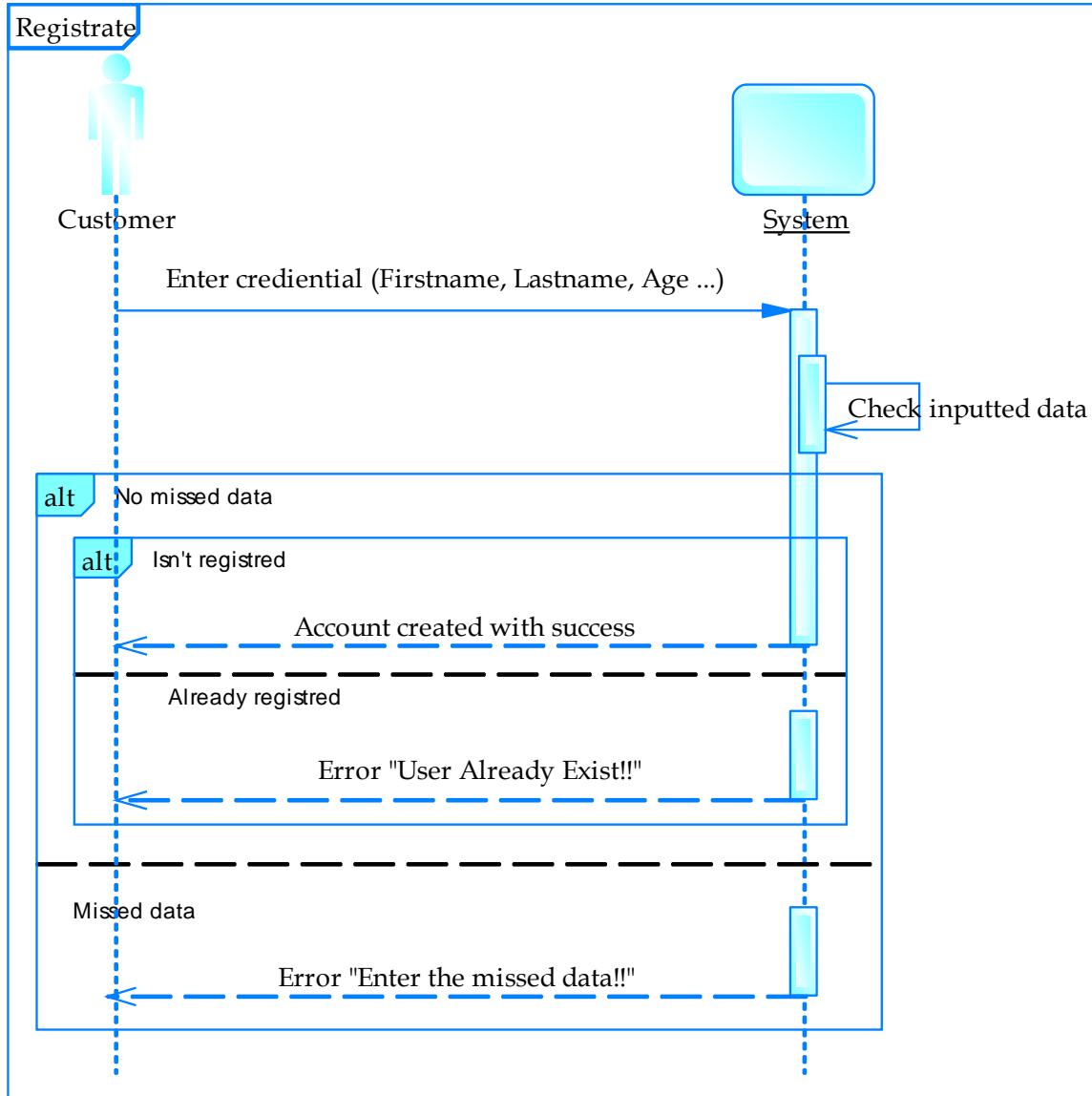


Figure 26: “Registration” System Sequence Diagram

### 1.2.3. System sequence diagram of the use case “Explore Catalog”

The interactions between our system and the Customer while creating an account is illustrated in the following figure:

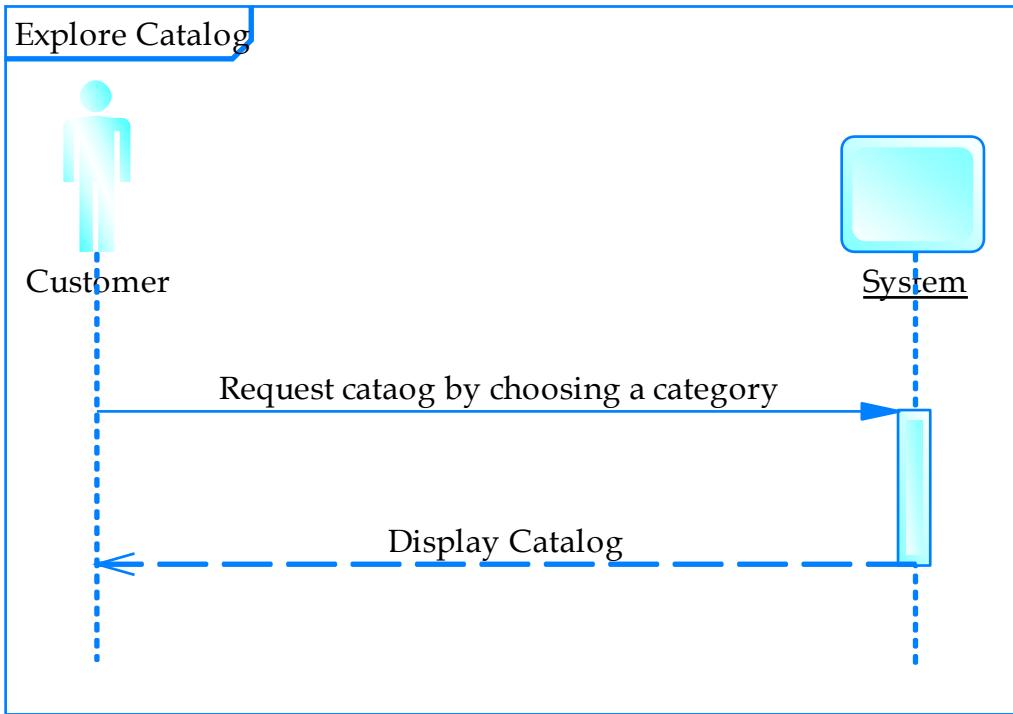


Figure 27: "Explore Catalog" System Sequence Diagram

#### 1.2.4. System sequence diagram of the use case “Add Product to Cart (Panier)”

The interactions between our system and the Customer while adding a product to his shopping cart (panier) is illustrated in the following figure:

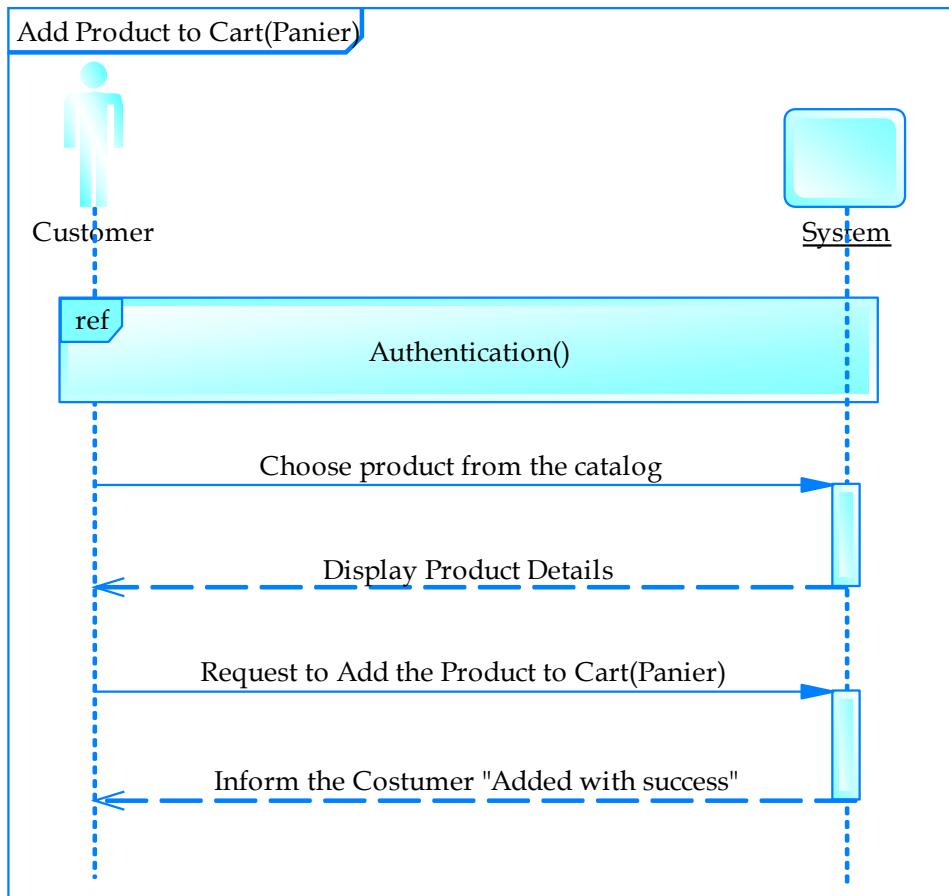


Figure 28: "Add Product to Cart (Panier)" System Sequence Diagram

### 1.2.5. System sequence diagram of the use case “Display shopping cart (Panier) details”

The interactions between our system and the Customer while displaying the details (existing products) of the shopping cart (Panier) is illustrated in the following figure:



Figure 29: “Display shopping cart (Panier) details” System Sequence Diagram

### 1.2.6. System sequence diagram of the use case “Delete Product from Cart (Panier)”

The interactions between our system and the Customer while deleting a product from his shopping cart (panier) is illustrated in the following figure:

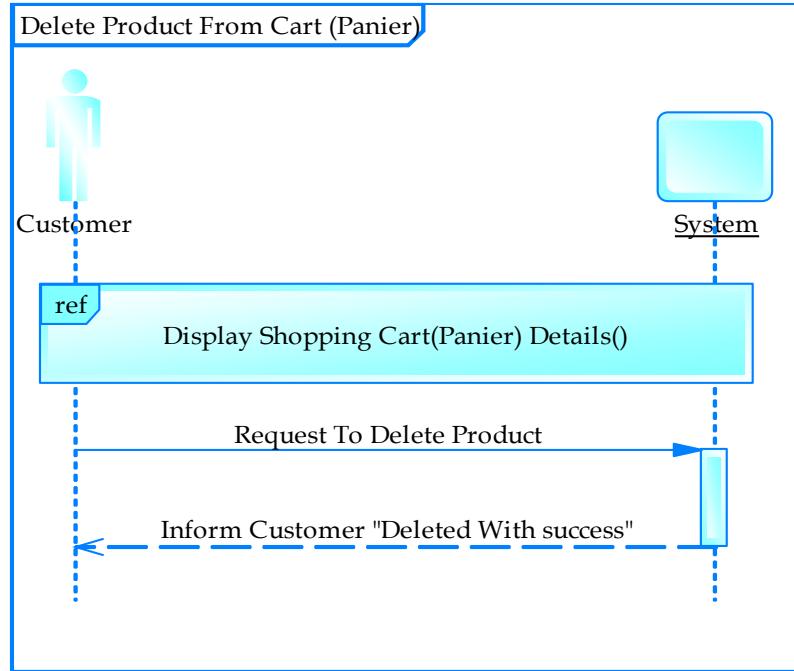


Figure 30: “Delete Product from Cart (Panier)” System Sequence Diagram

### 1.2.7. System sequence diagram of the use case “Buy Product”

The interactions between our system and the Customer while buying a product is illustrated in the following figure:

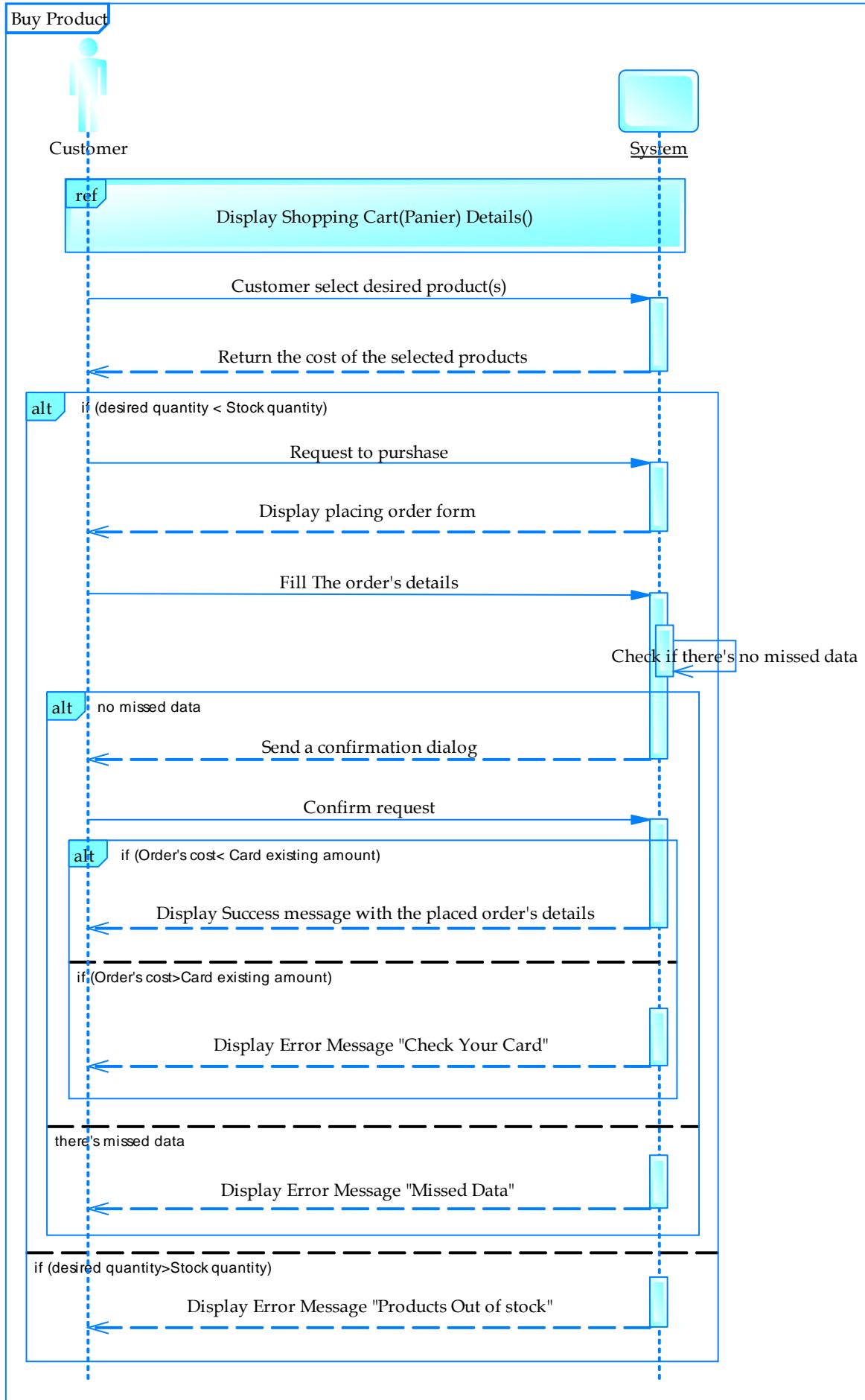


Figure 31: "Buy Product(s)" System Sequence Diagram

### 1.2.8. System sequence diagram of the use case “Empty Shopping Cart (Panier)”

The interactions between our system and the Customer while deleting all products from his shopping cart (panier) is illustrated in the following figure:

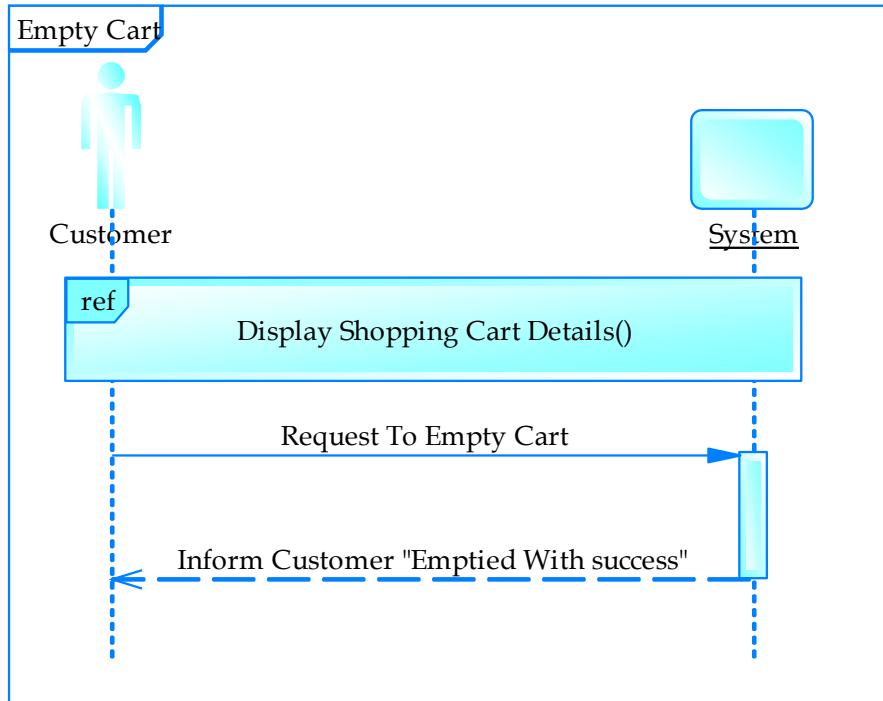


Figure 32: “Empty Shopping Cart (Panier)” System Sequence Diagram

### 1.2.9. System sequence diagram of the use case “Request a Help”

The interactions between our system and the Customer while sending a help request is illustrated in the following figure:

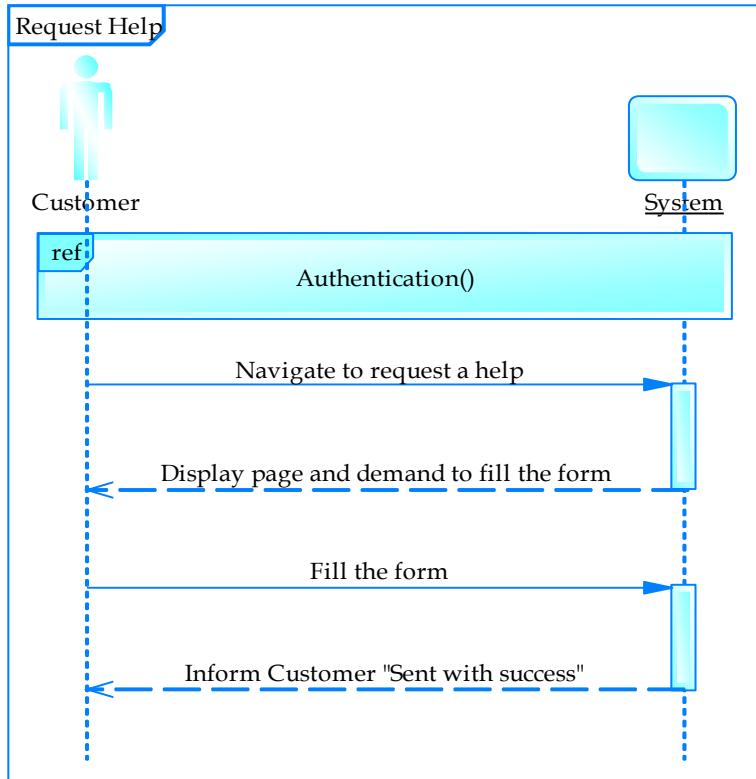


Figure 33: “Request a Help” System Sequence Diagram

### 1.2.10. System sequence diagram of the use case “Display Webmaster’s management page”

The interactions between our system and the Super Webmaster while displaying the webmaster’s management page is illustrated in the following figure:

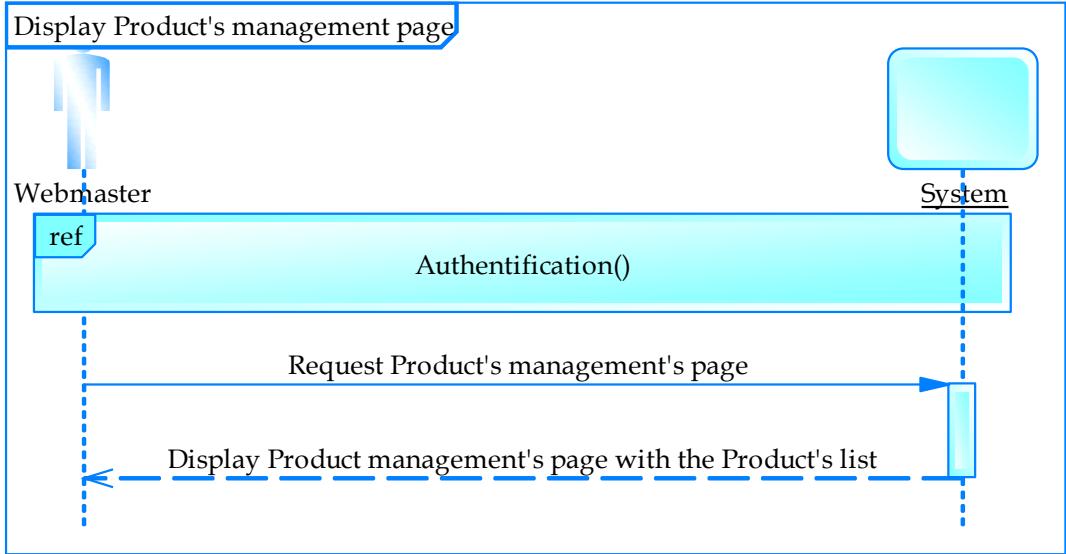


Figure 34: “Display Webmaster’s management page” System Sequence Diagram

### 1.2.11. System sequence diagram of the use case “Add Webmaster”

The interactions between our system and the Super Webmaster while adding new webmaster is illustrated in the following figure:

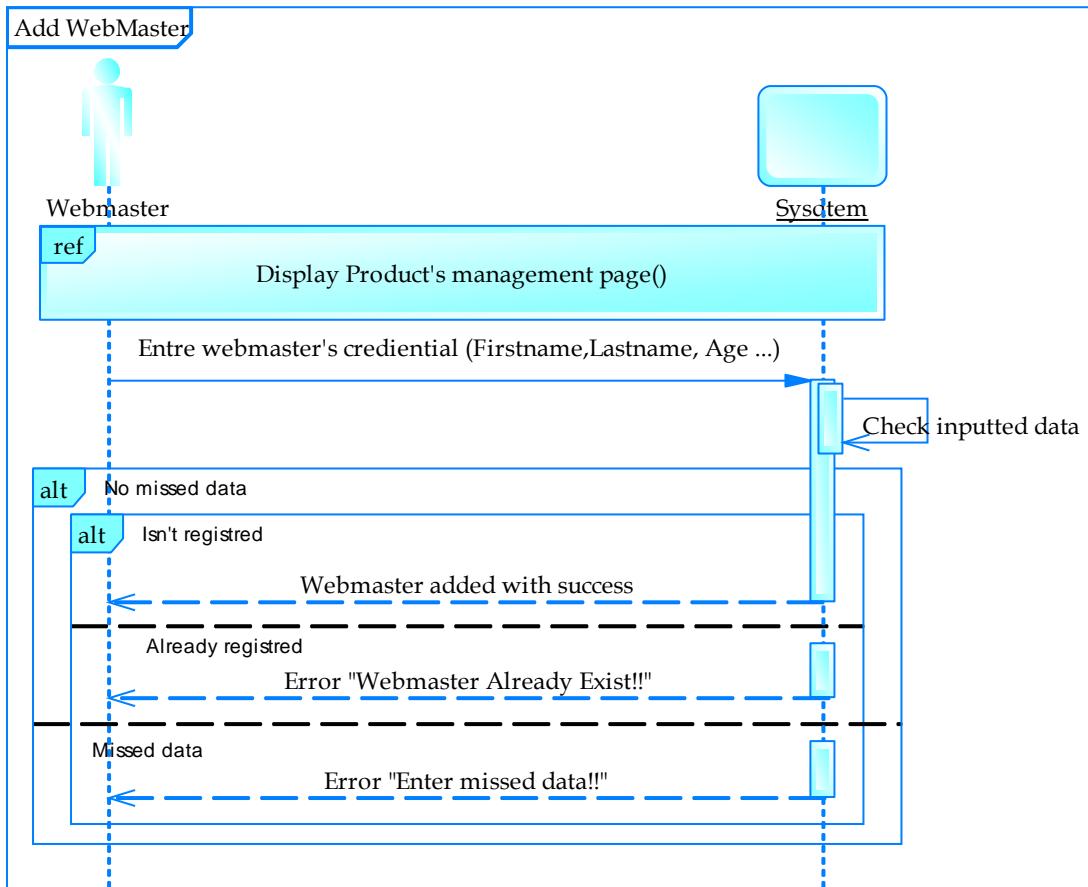


Figure 35: “Add Webmaster” System Sequence Diagram

#### 1.2.12. System sequence diagram of the use case “Delete Webmaster”

The interactions between our system and the Super Webmaster while deleting webmaster is illustrated in the following figure:

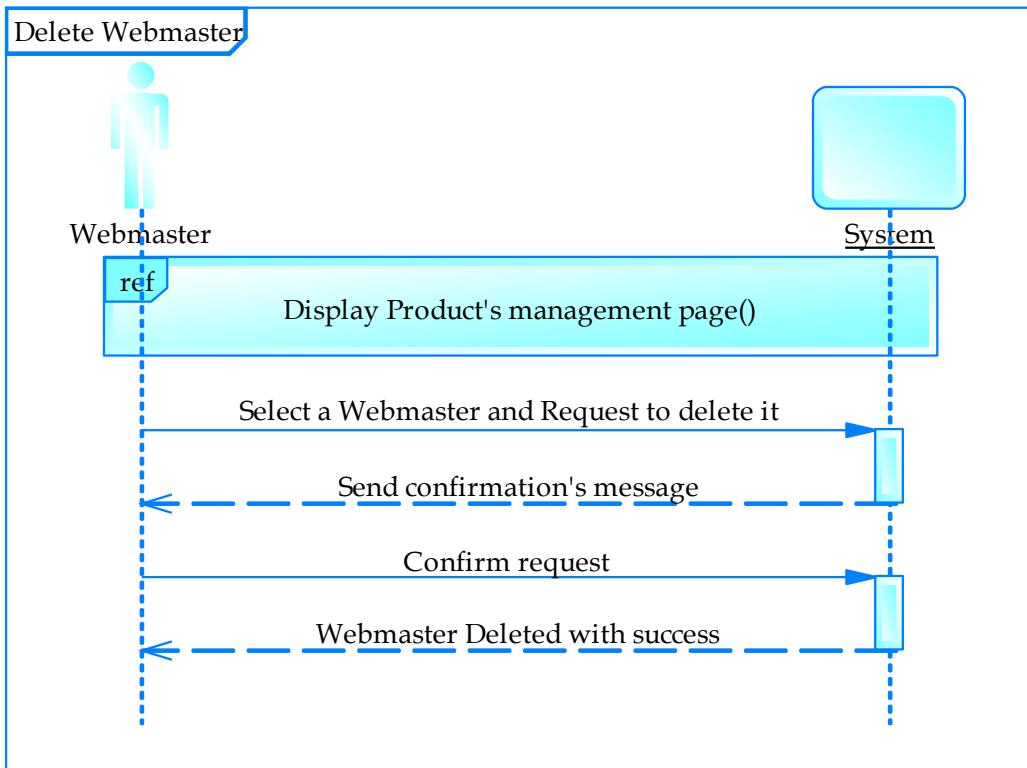


Figure 36: “Delete Webmaster” System Sequence Diagram

#### 1.2.13. System sequence diagram of the use case “Search Webmaster”

The interactions between our system and the Super Webmaster while searching for a webmaster is illustrated in the following figure:

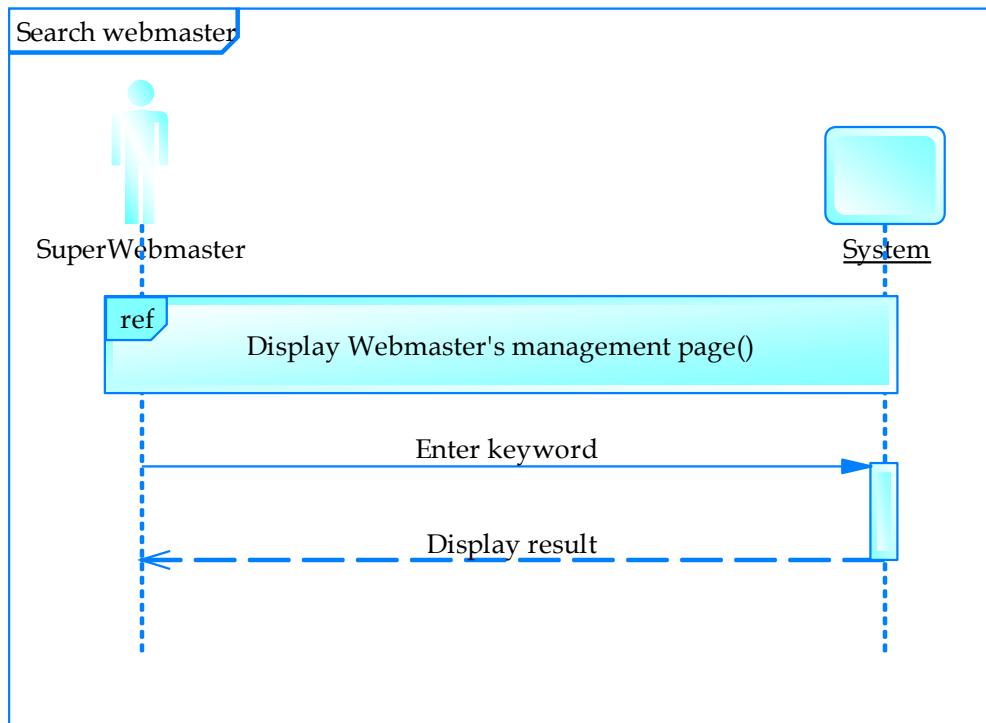


Figure 37: “Search Webmaster” System Sequence Diagram

#### 1.2.14. System sequence diagram of the use case “Display Product's management page”

The interactions between our system and the Webmaster while displaying the Product's management page is illustrated in the following figure:

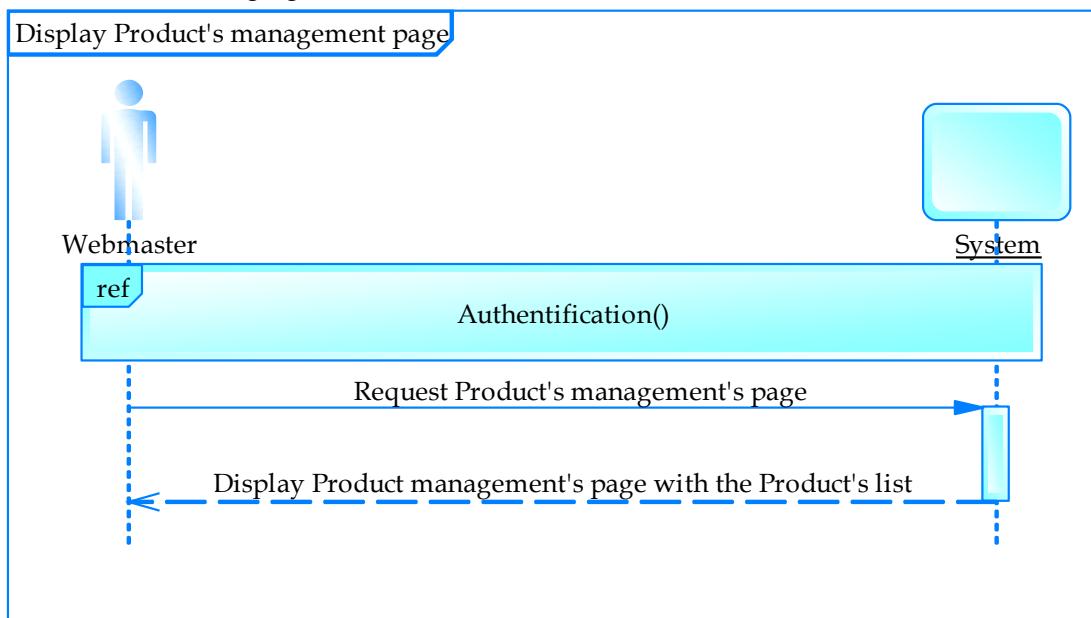


Figure 38: “Display Product's management page” System Sequence Diagram

#### 1.2.15. System sequence diagram of the use case “Add Product”

The interactions between our system and the Webmaster while adding new Product is illustrated in the following figure:

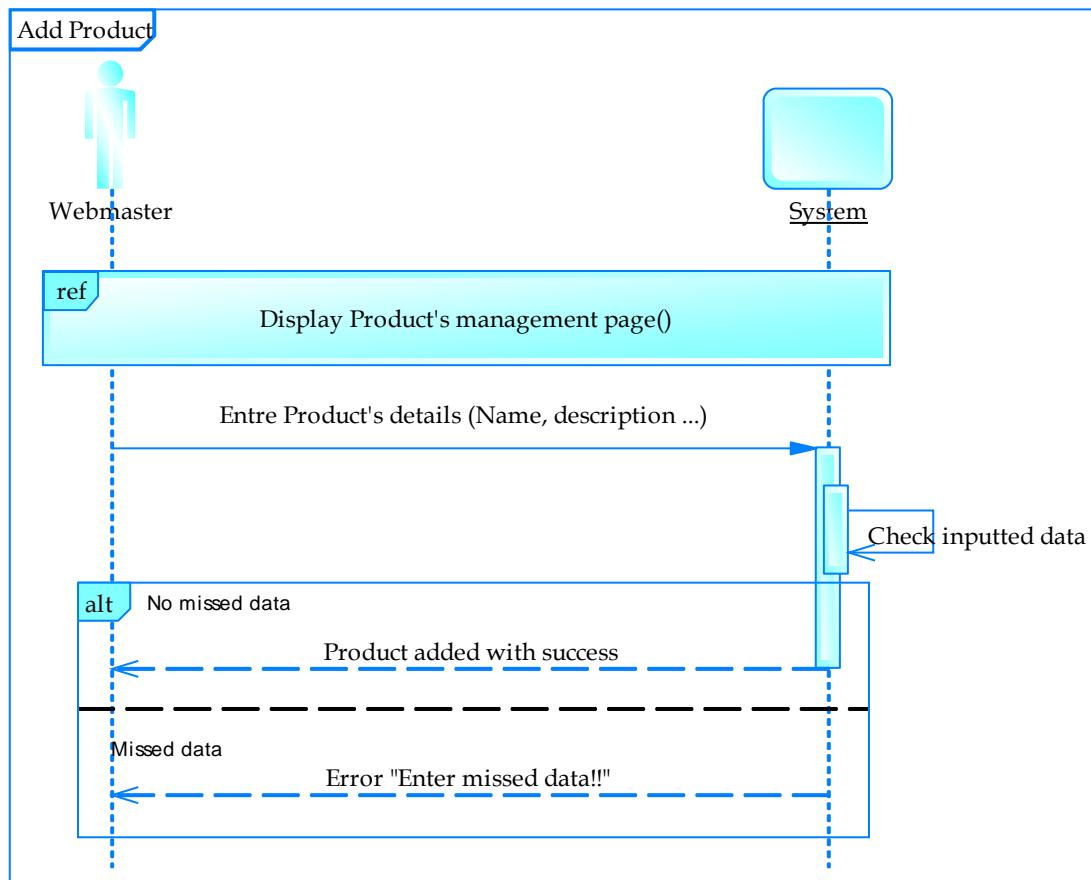


Figure 39: “Add Product” System Sequence Diagram

#### 1.2.16. System sequence diagram of the use case “Delete Product”

The interactions between our system and the Webmaster while deleting Product is illustrated in the following figure:

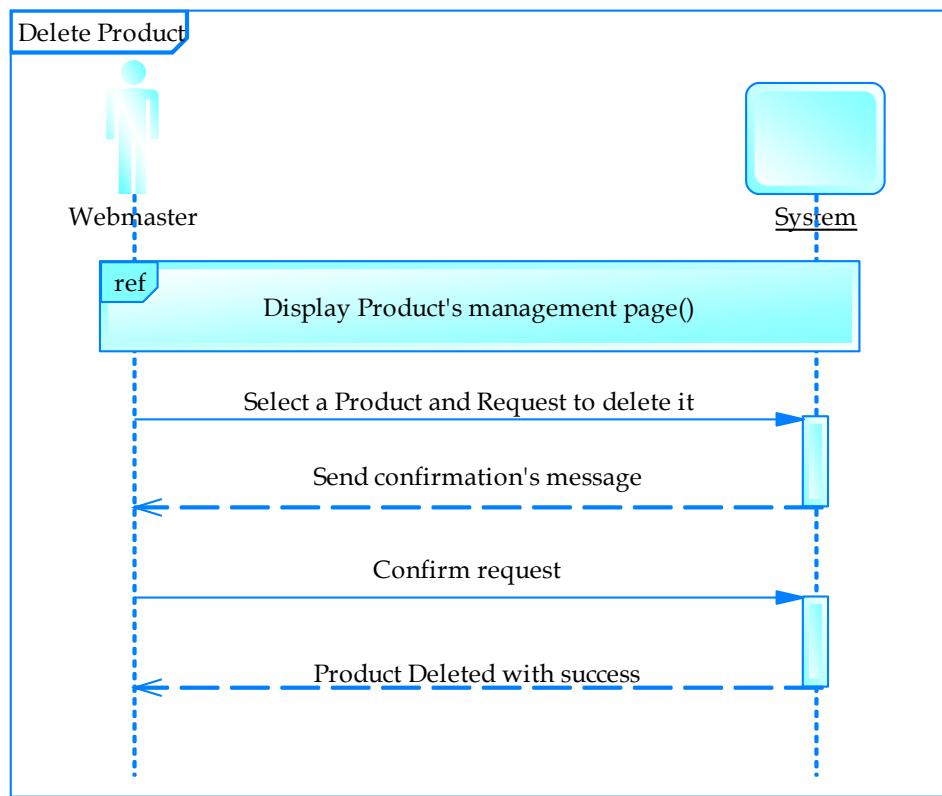


Figure 40: “Delete Product” System Sequence Diagram

#### 1.2.17. System sequence diagram of the use case “Search Product”

The interactions between our system and the Webmaster while searching for a Product is illustrated in the following figure:

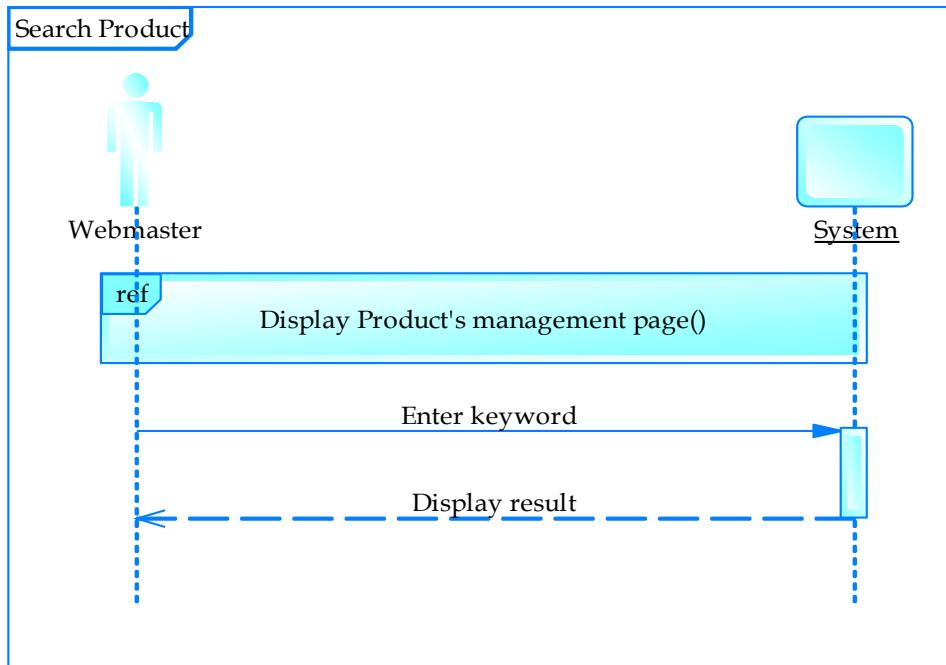


Figure 41: “Search Product” System Sequence Diagram

### 1.2.18. System sequence diagram of the use case “Update Product”

The interactions between our system and the Webmaster while updating for a Product is illustrated in the following figure:

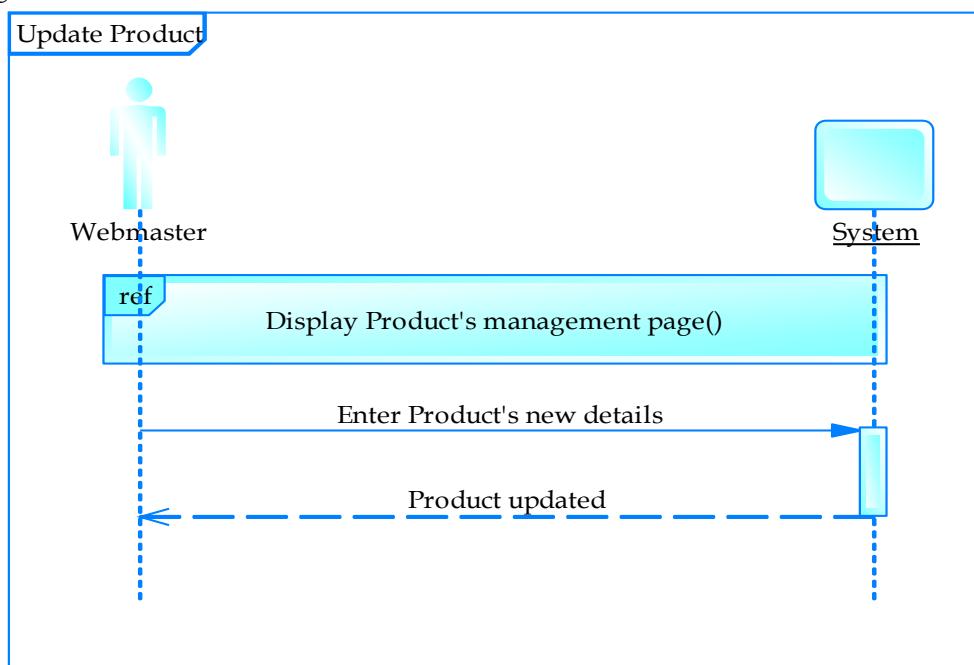


Figure 42: “Update Product” System Sequence Diagram

## 1.3. Detailed Sequence Diagrams

### 1.3.1. Detailed sequence diagram of the use case “Authentication”

The interactions between our system and the users (Super Webmaster, Webmaster and Customer) while authenticating is Illustrated in the following figure

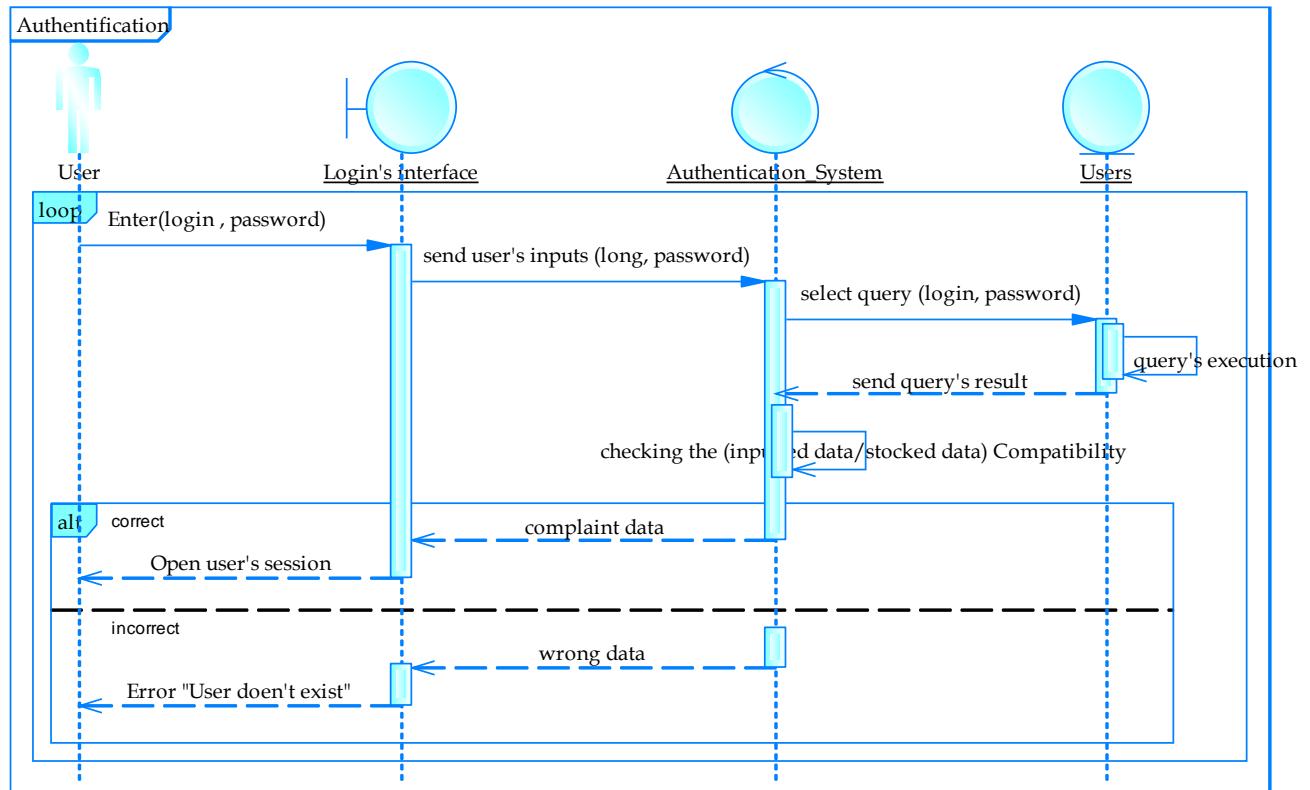


Figure 43: "Authentication" Detailed Sequence Diagram

### 1.3.2. Detailed sequence diagram of the use case “Registration”

The interactions between our system and the Customer while creating an account is illustrated in the following figure:

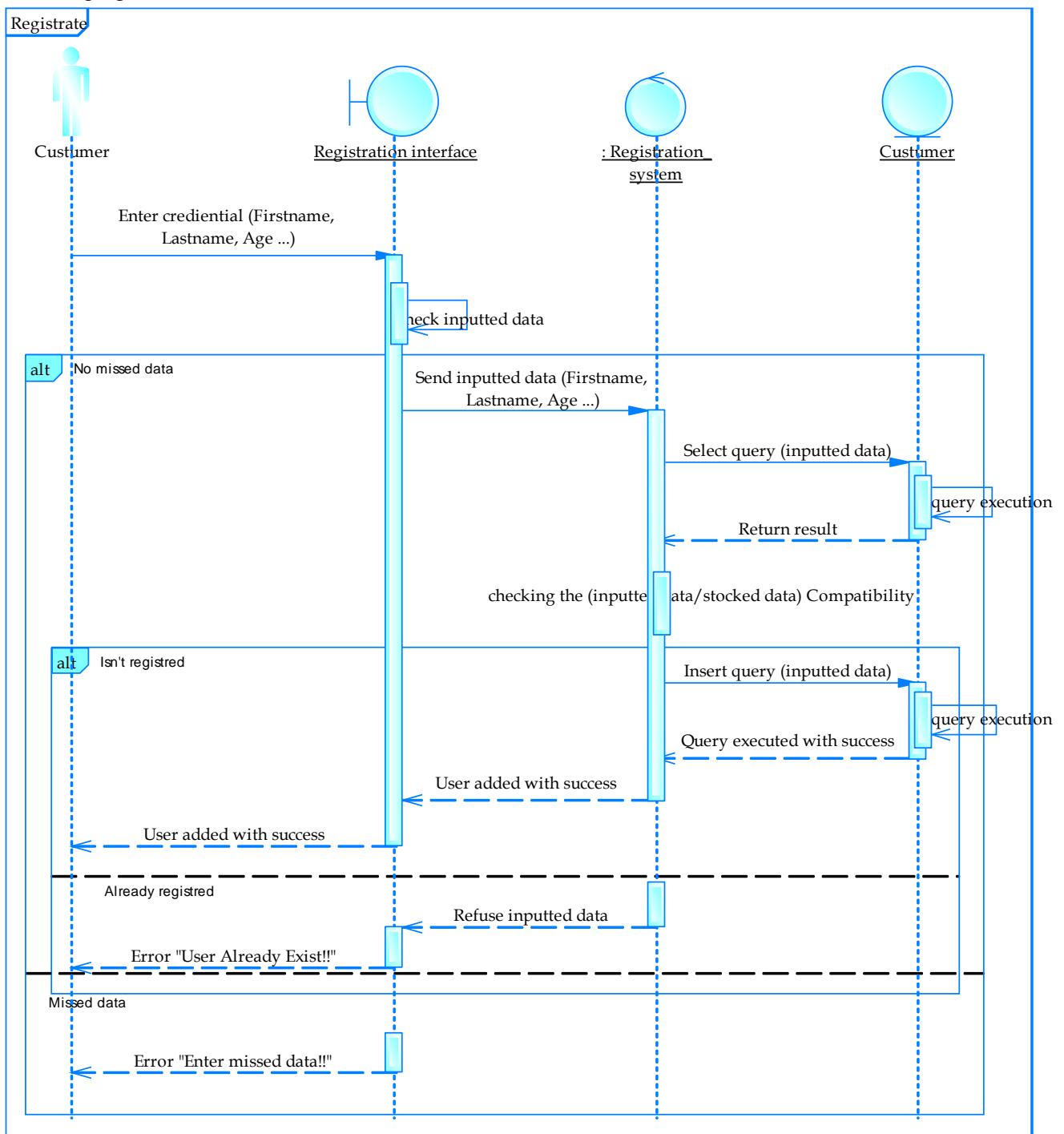


Figure 44: “Registration” Detailed Sequence Diagram

### 1.3.3. Detailed sequence diagram of the use case “Explore Catalog”

The interactions between our system and the Customer while creating an account is illustrated in the following figure:

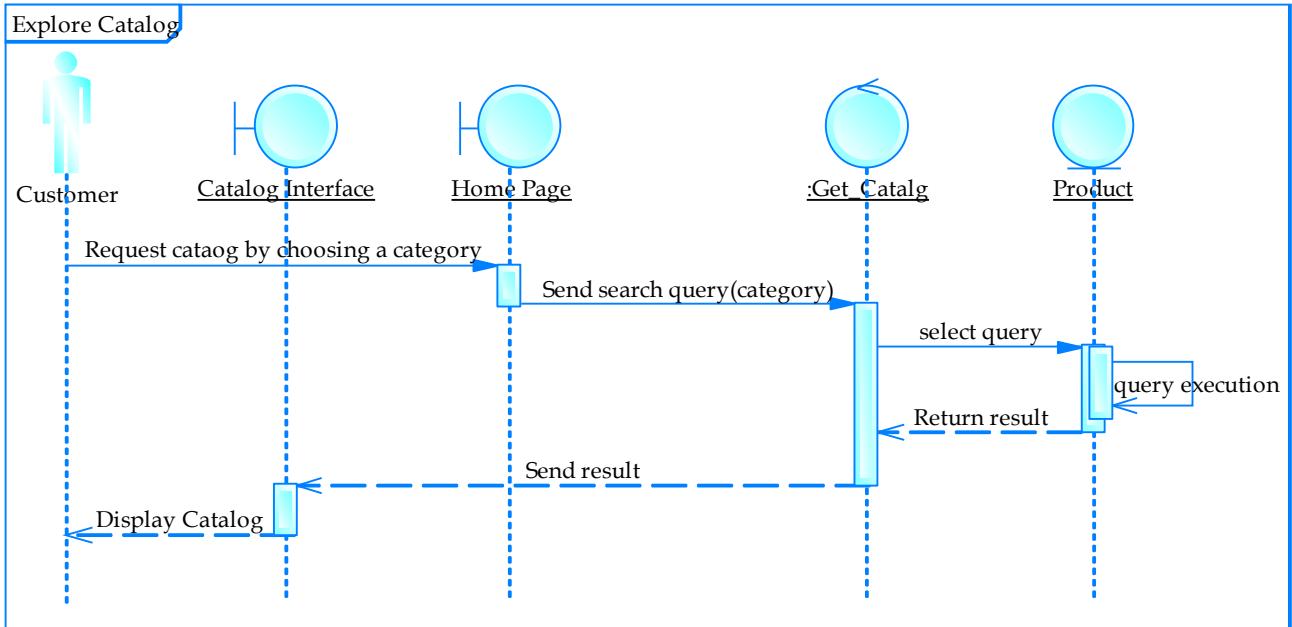


Figure 45: "Explore Catalog" Detailed Sequence Diagram

#### 1.3.4. Detailed sequence diagram of the use case "Add Product to Cart (Panier)"

The interactions between our system and the Customer while adding a product to his shopping cart (panier) is illustrated in the following figure:

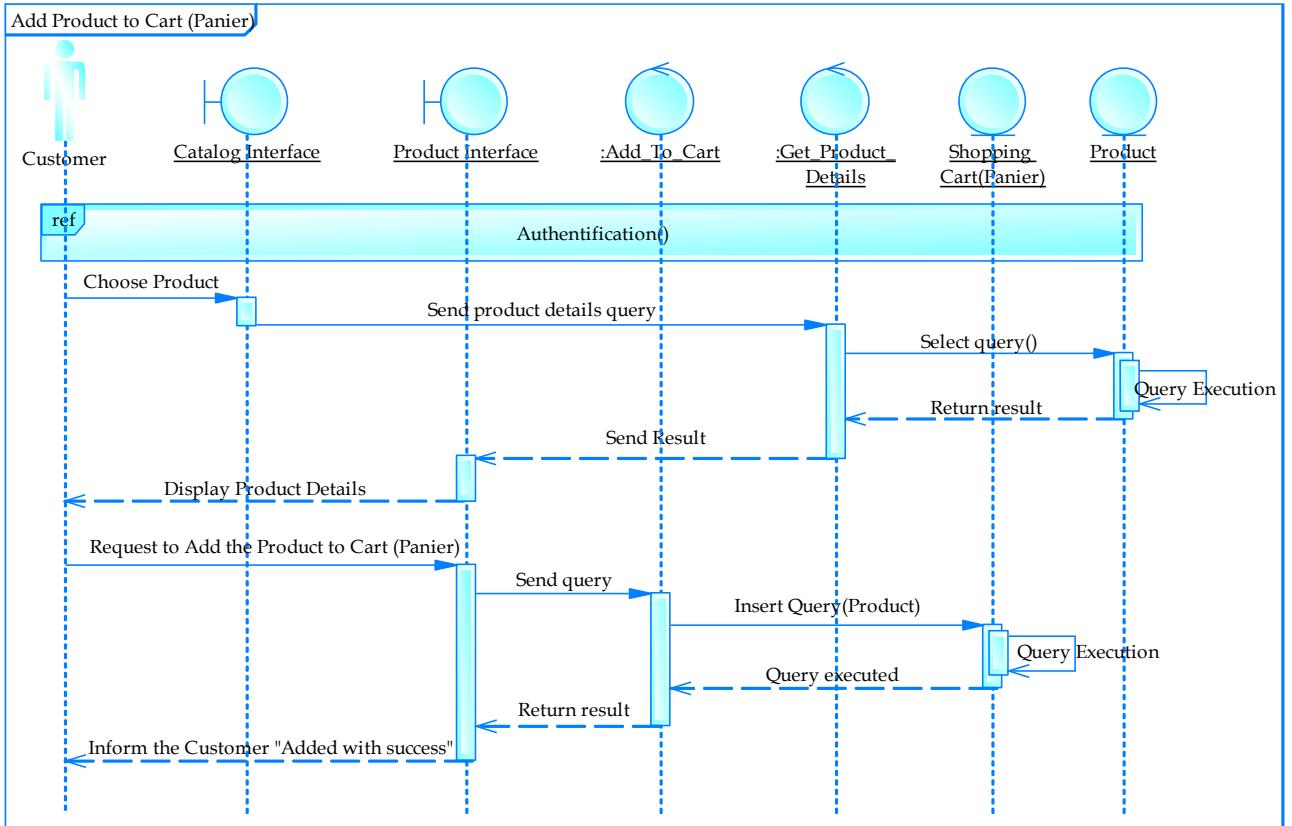


Figure 46: "Add Product to Cart (Panier)" Detailed Sequence Diagram

### 1.3.5. Detailed sequence diagram of the use case “Display shopping cart (Panier) details”

The interactions between our system and the Customer while displaying the details (existing products) of the shopping cart (Panier) is illustrated in the following figure:

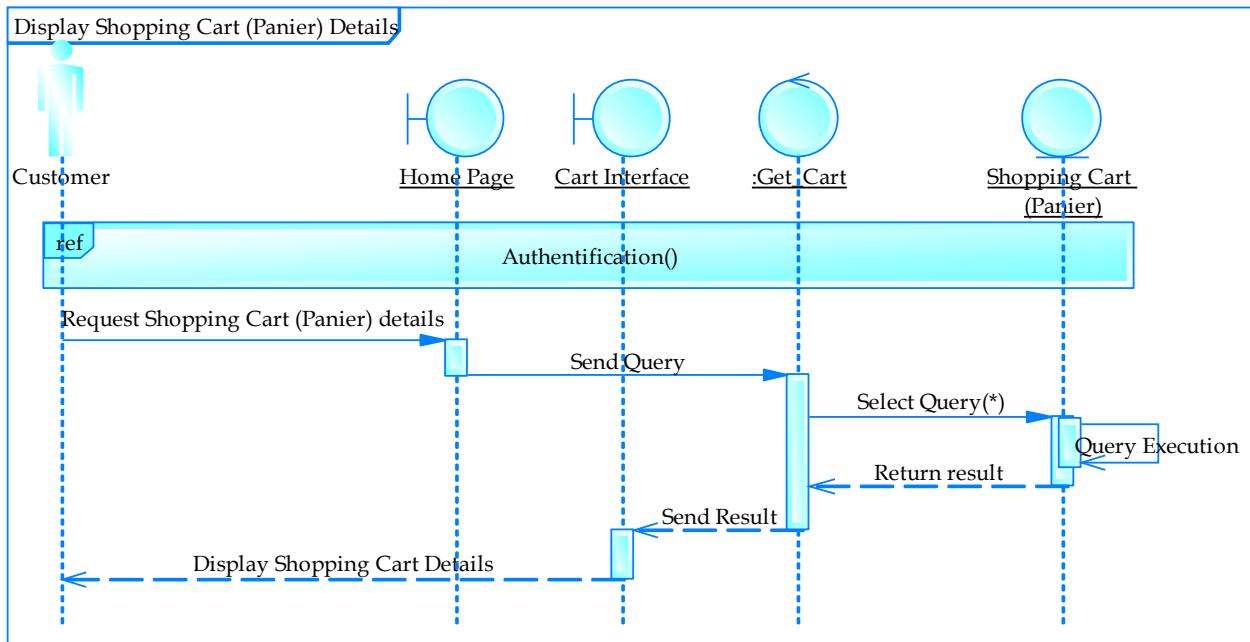


Figure 47: “Display shopping cart (Panier) details” Detailed Sequence Diagram

### 1.3.6. Detailed sequence diagram of the use case “Delete Product from Cart (Panier)”

The interactions between our system and the Customer while deleting a product from his shopping cart (panier) is illustrated in the following figure:

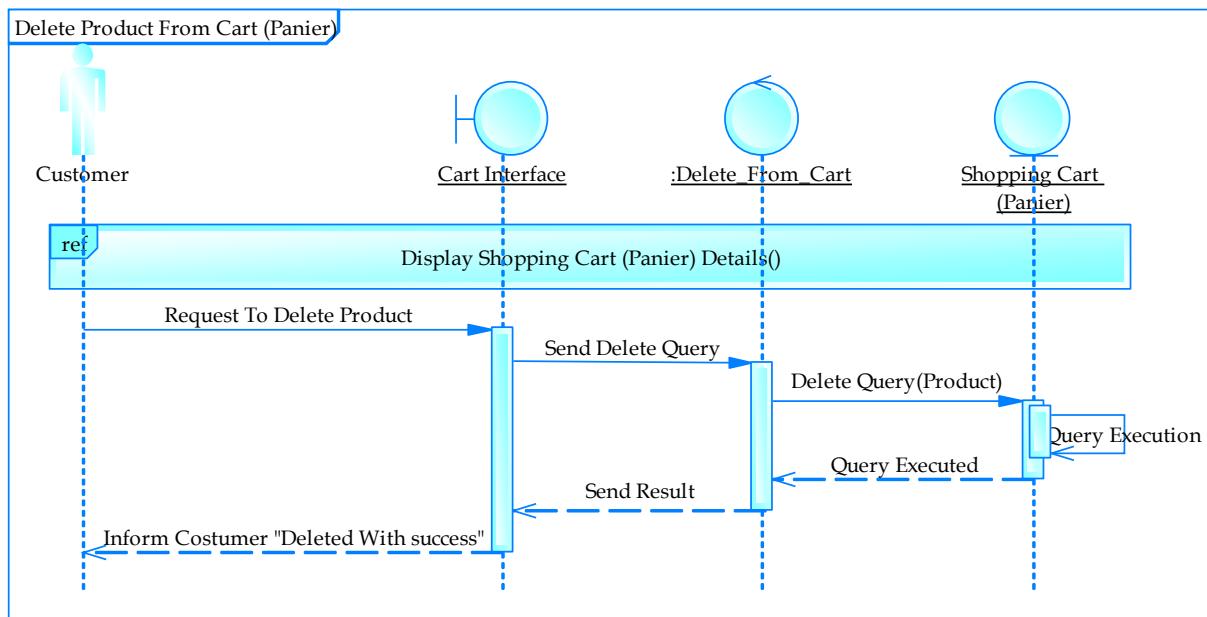


Figure 48: “Delete Product from Cart (Panier)” Detailed Sequence Diagram

### 1.3.7. Detailed sequence diagram of the use case “Buy Product”

The interactions between our system and the Customer while buying a product is illustrated in the following figure:

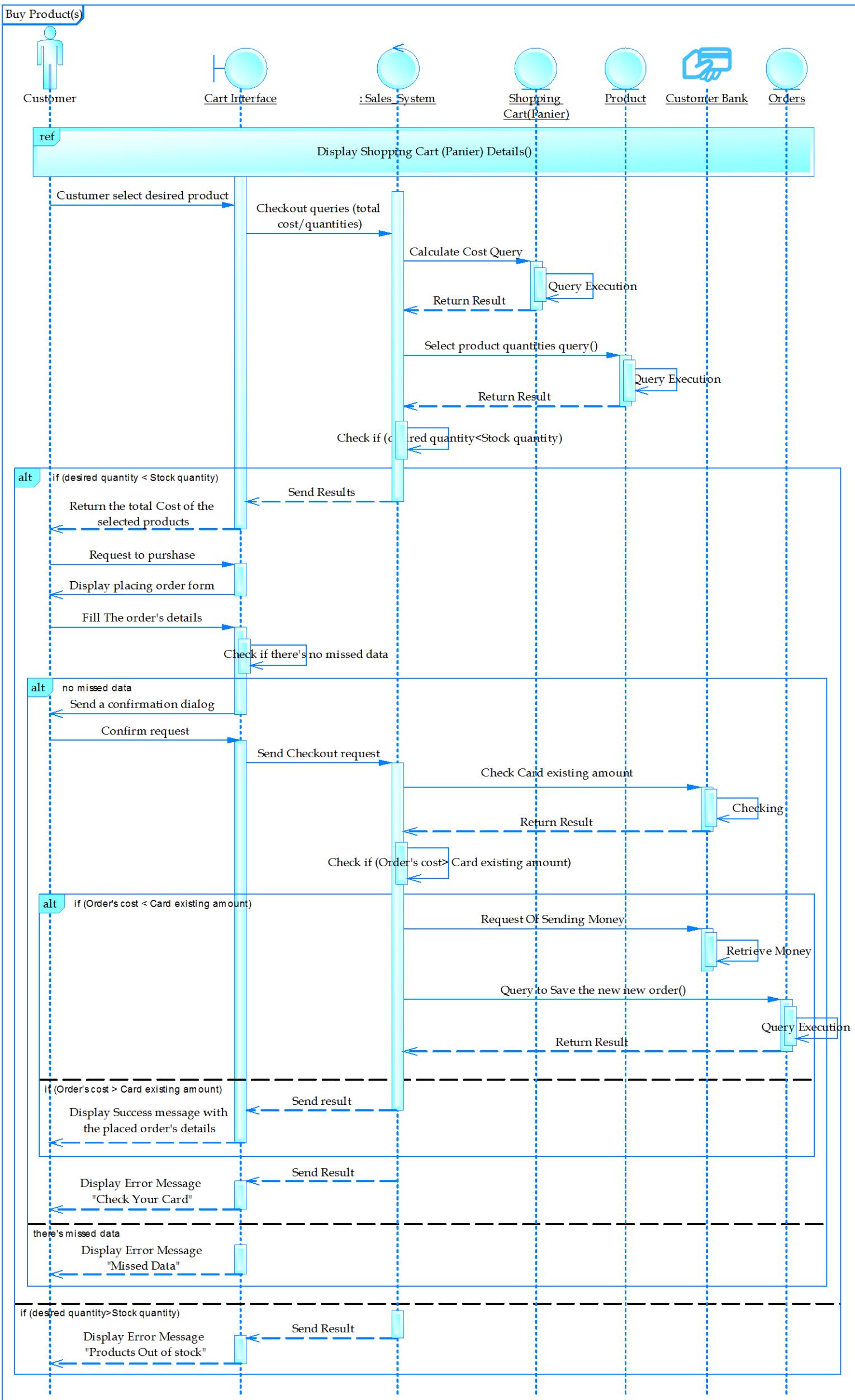


Figure 49: "Buy Product(s)" Detailed Sequence Diagram

### 1.3.8. Detailed sequence diagram of the use case “Empty Shopping Cart (Panier)”

The interactions between our system and the Customer while deleting all products from his shopping cart (panier) is illustrated in the following figure:

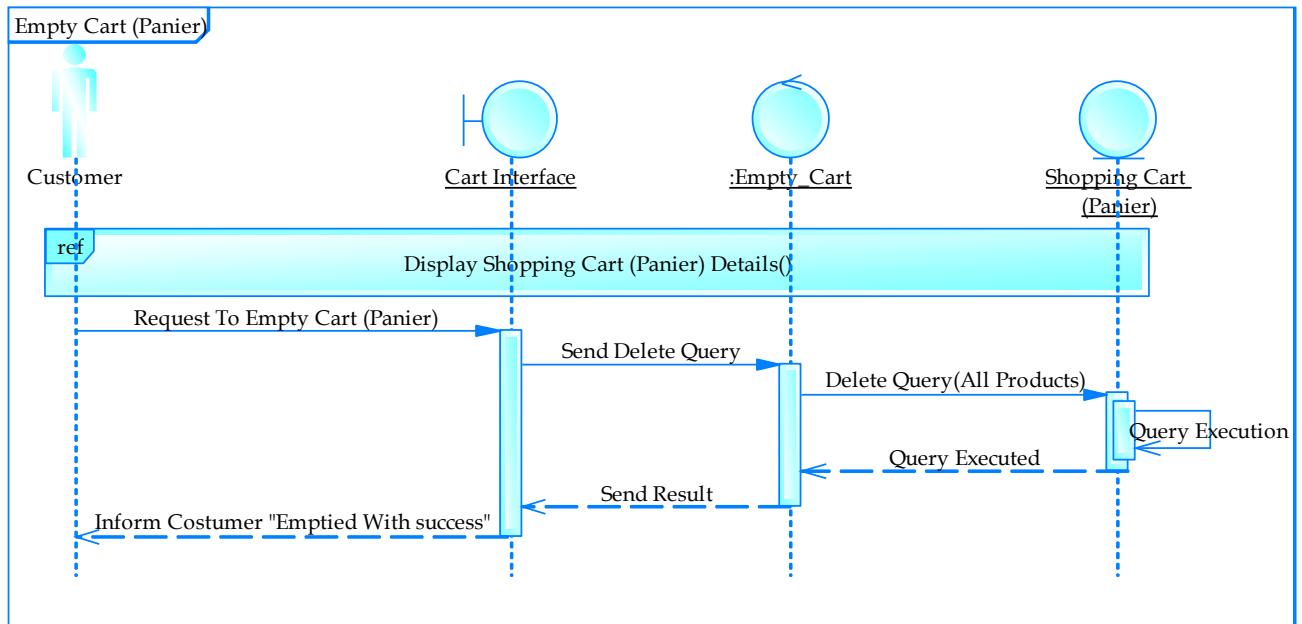


Figure 50: “Empty Shopping Cart (Panier)” Detailed Sequence Diagram

### 1.3.9. Detailed sequence diagram of the use case “Request a Help”

The interactions between our system and the Customer while sending a help request is illustrated in the following figure:

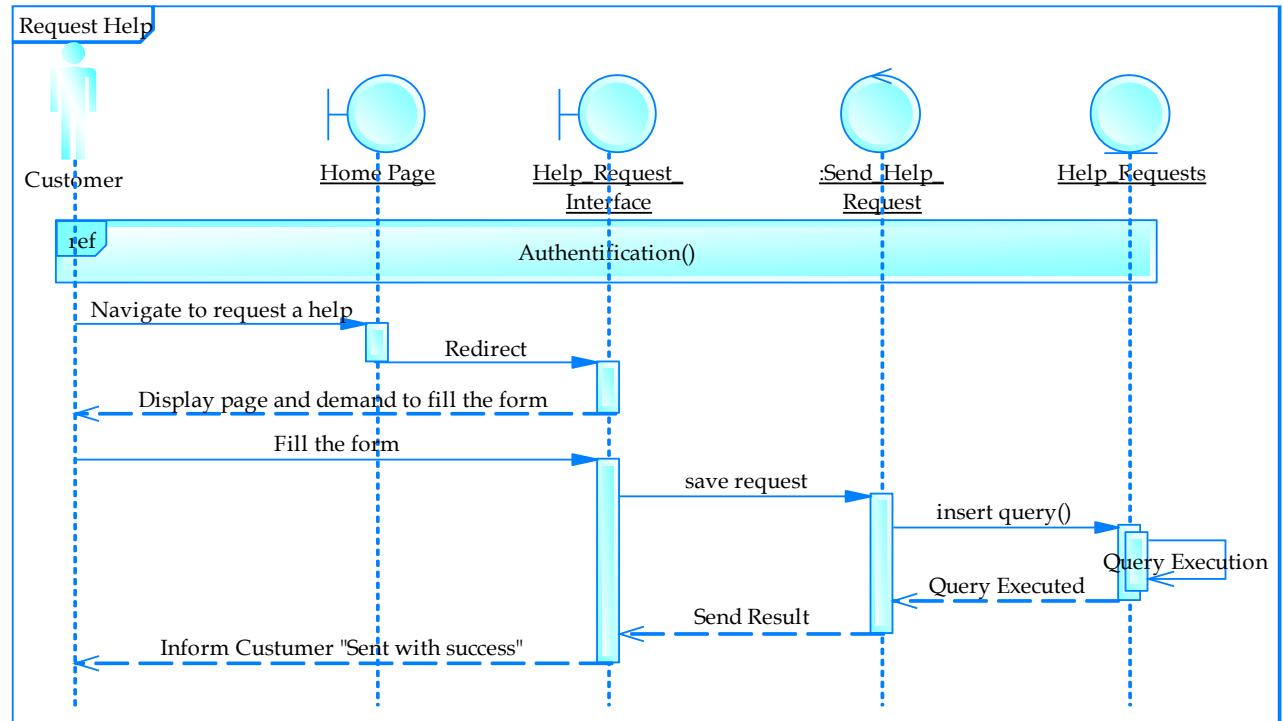


Figure 51: “Request a Help” Detailed Sequence Diagram

### 1.3.10. Detailed sequence diagram of the use case “Display Webmaster's management page”

The interactions between our system and the Super Webmaster while displaying the webmaster's management page is illustrated in the following figure:

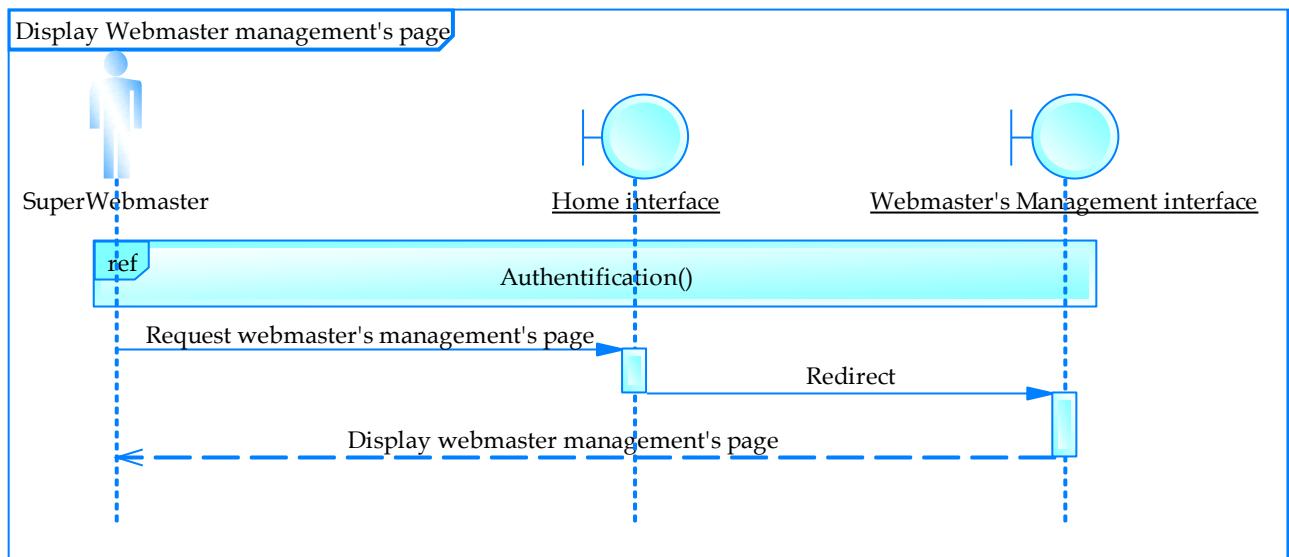


Figure 52: “Display Webmaster's management page” Detailed Sequence Diagram

### 1.3.11. Detailed sequence diagram of the use case “Add Webmaster”

The interactions between our system and the Super Webmaster while adding new webmaster is illustrated in the following figure:

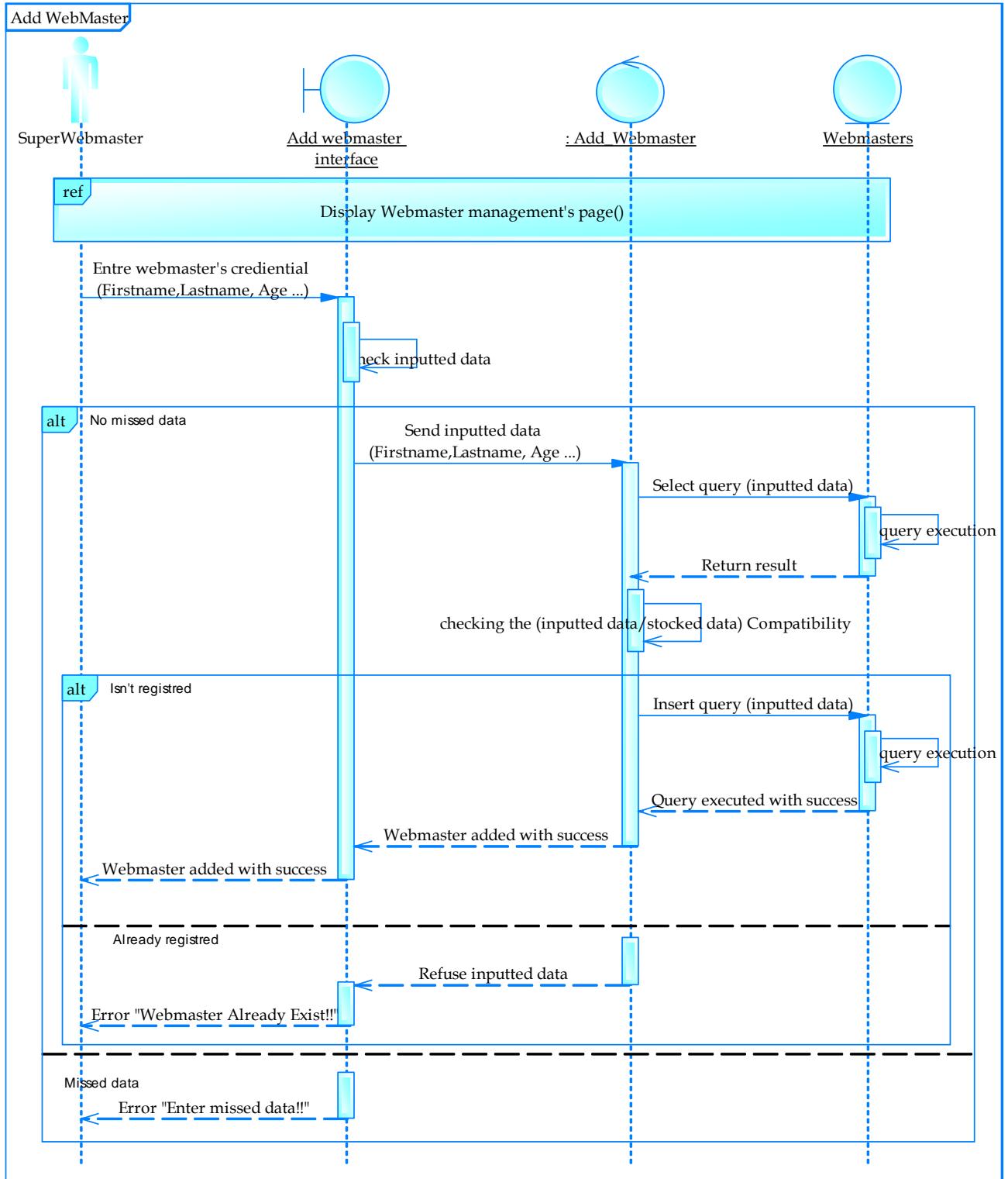


Figure 53: “Add Webmaster” Detailed Sequence Diagram

### 1.3.12. Detailed sequence diagram of the use case “Delete Webmaster”

The interactions between our system and the Super Webmaster while deleting webmaster is illustrated in the following figure:

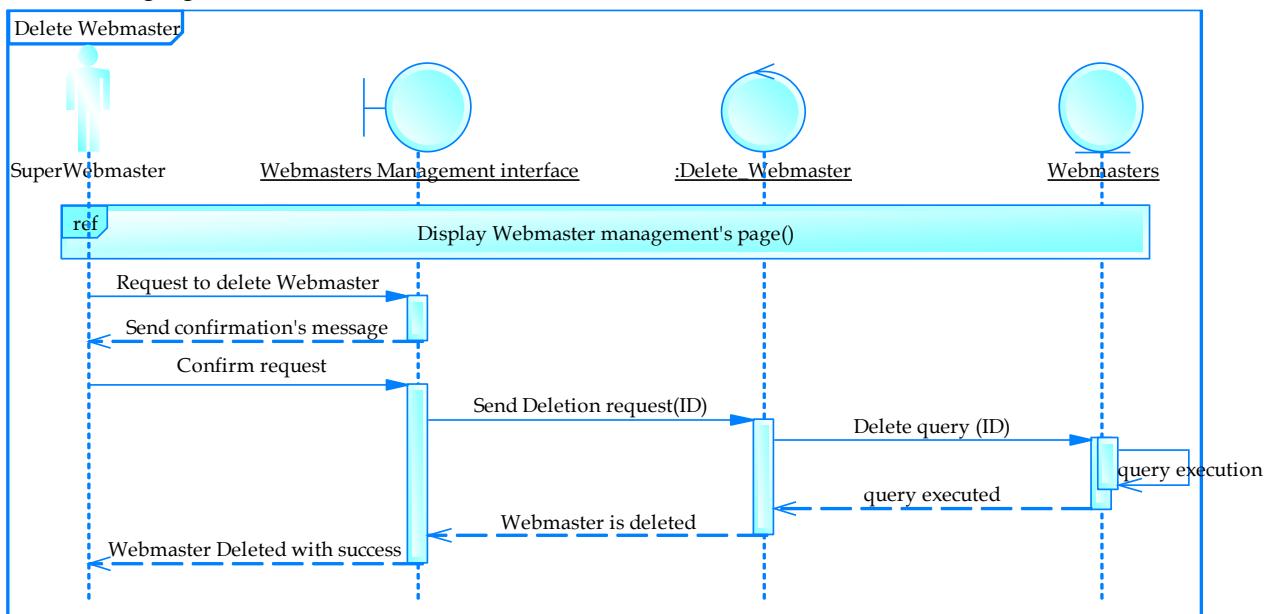


Figure 54: “Delete Webmaster” Detailed Sequence Diagram

### 1.3.13. Detailed sequence diagram of the use case “Search Webmaster”

The interactions between our system and the Super Webmaster while searching for a webmaster is illustrated in the following figure:

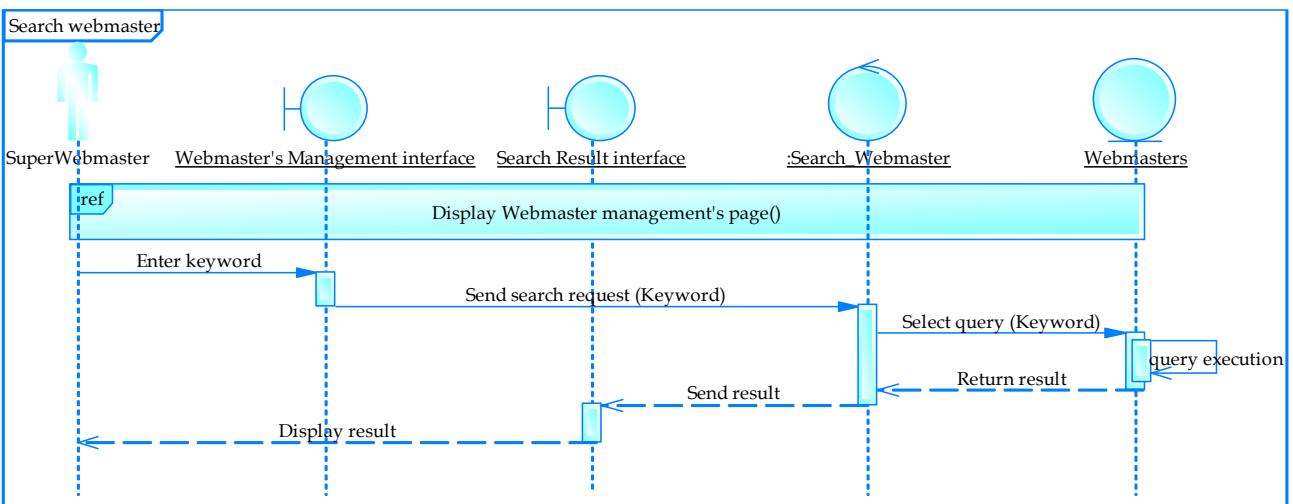


Figure 55: “Search Webmaster” Detailed Sequence Diagram

### 1.3.14. Detailed sequence diagram of the use case “Display Product's management page”

The interactions between our system and the Webmaster while displaying the Product's management page is illustrated in the following figure:

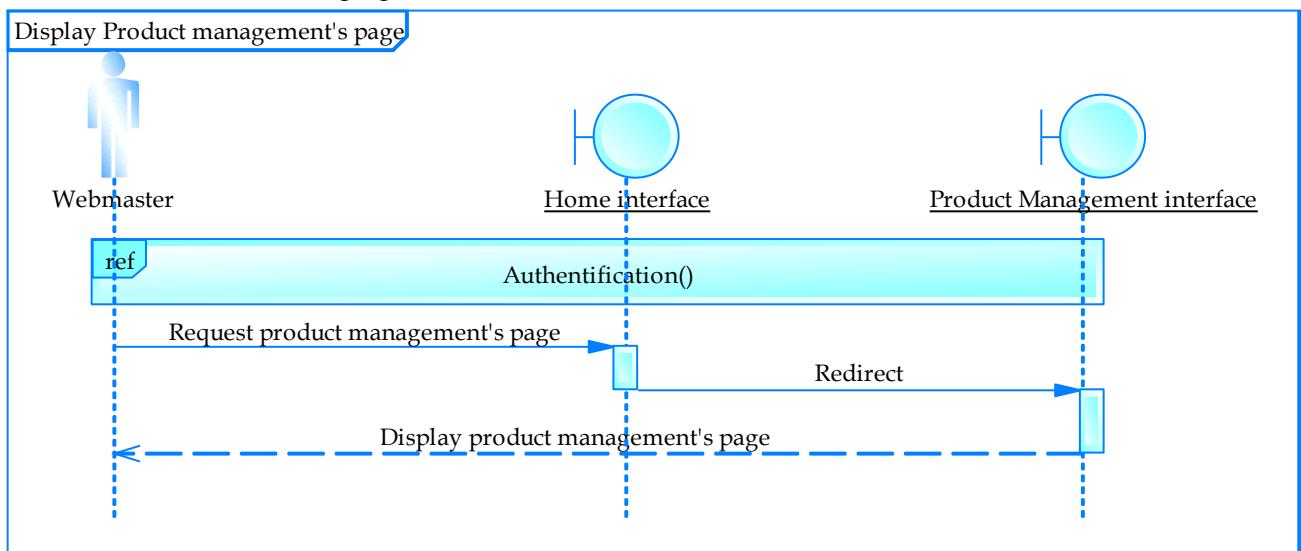


Figure 56: “Display Product’s management page “Detailed Sequence Diagram

### 1.3.15. Detailed sequence diagram of the use case “Add Product”

The interactions between our system and the Webmaster while adding new Product is illustrated in the following figure:

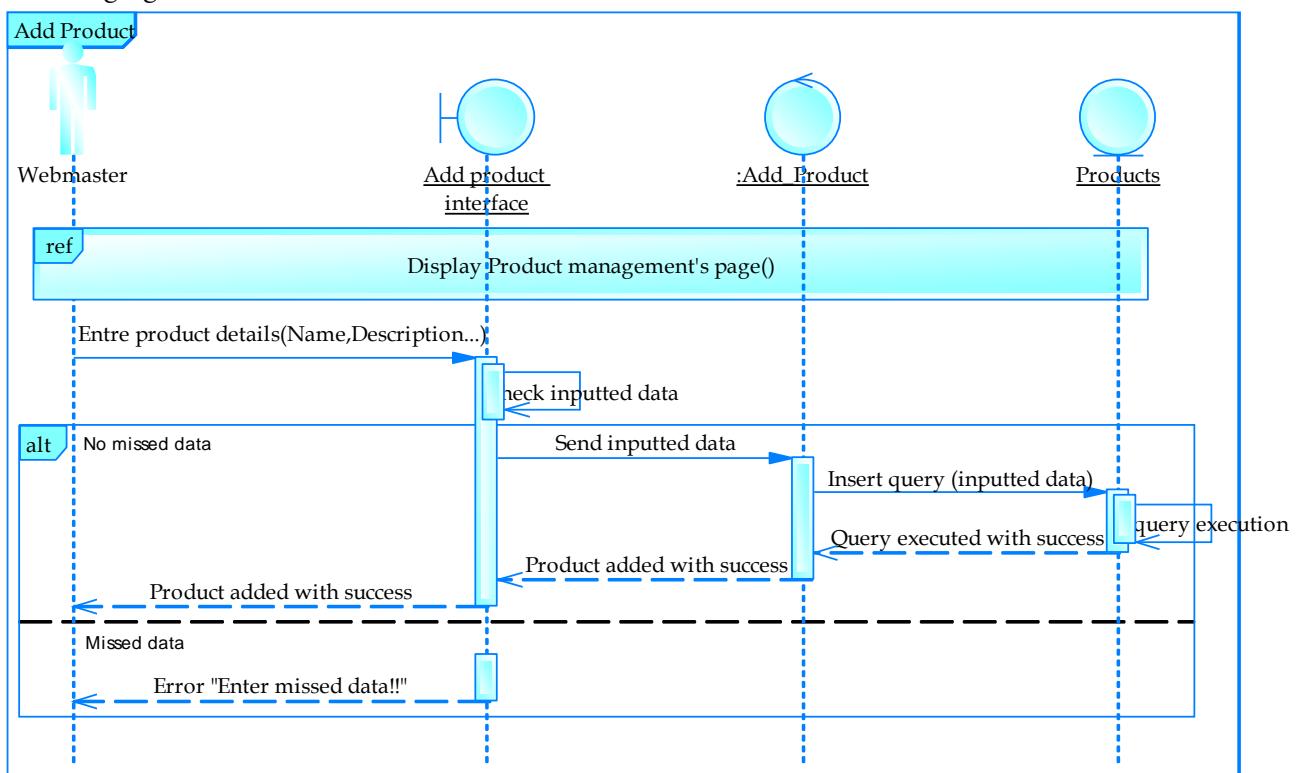


Figure 57: “Add Product” Detailed Sequence Diagram

### 1.3.16. Detailed sequence diagram of the use case “Delete Product”

The interactions between our system and the Webmaster while deleting Product is illustrated in the following figure:

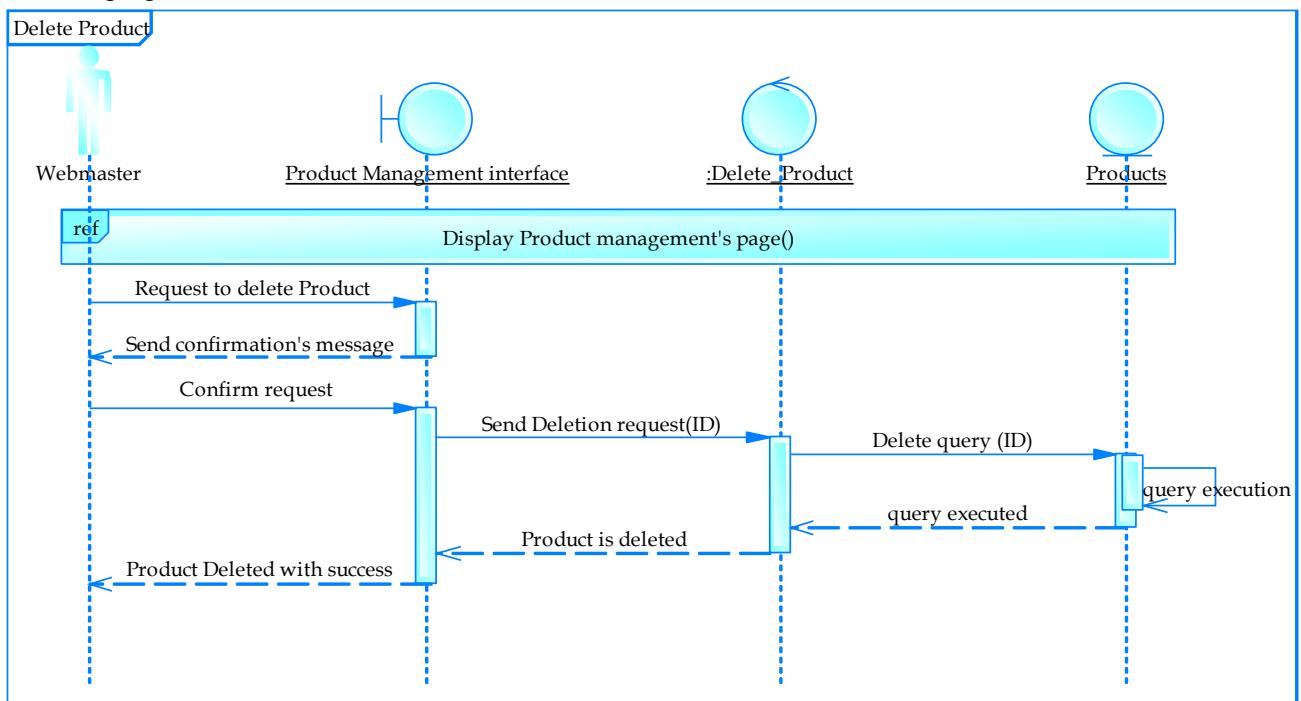


Figure 58: “Delete Product” Detailed Sequence Diagram

### 1.3.17. Detailed sequence diagram of the use case “Search Product”

The interactions between our system and the Webmaster while searching for a Product is illustrated in the following figure:

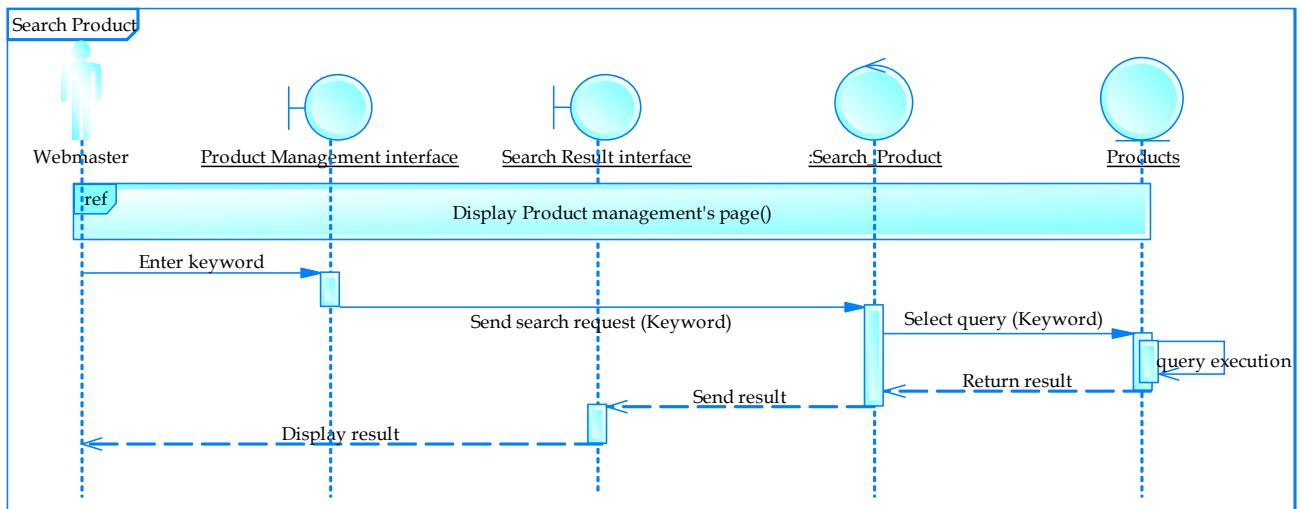


Figure 59: “Search Product” Detailed Sequence Diagram

### 1.3.18. Detailed sequence diagram of the use case “Update Product”

The interactions between our system and the Webmaster while updating for a Product is illustrated in the following figure:

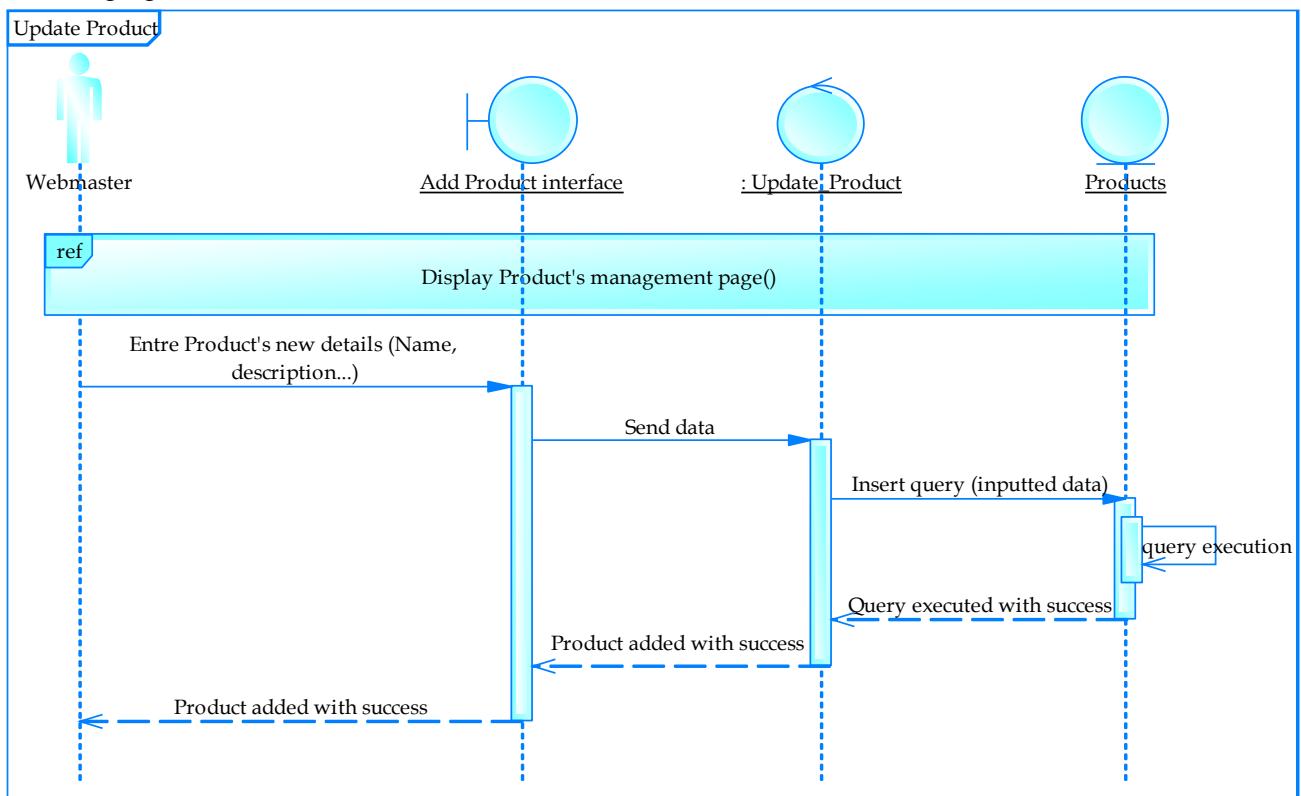


Figure 60: “Update Product” Detailed Sequence Diagram

## 2. Class diagram

### 2.1. What is Class diagram?

a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

### 2.2. Class diagram schema

The following figure represents our application’s class diagram:

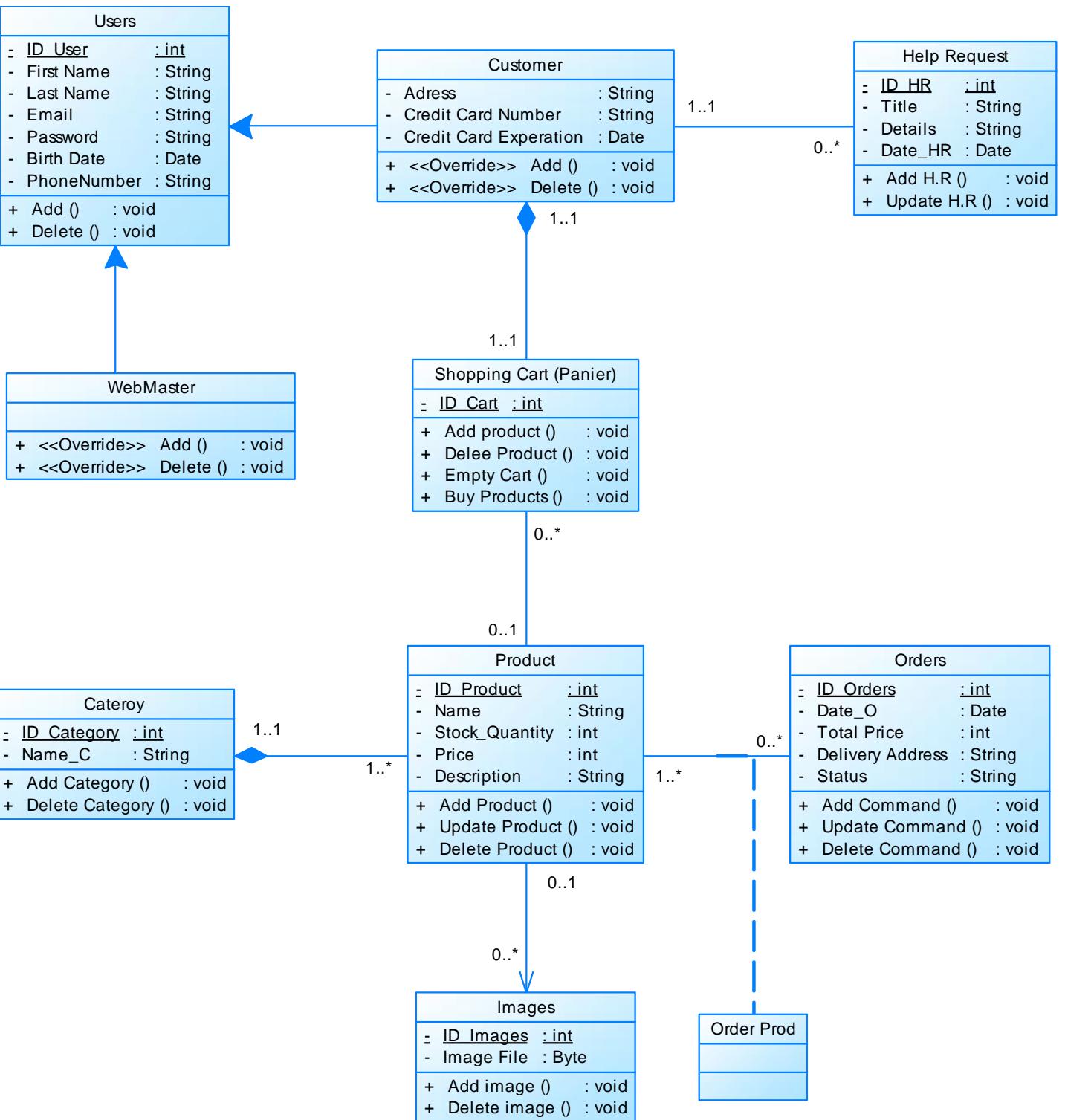


Figure 61: Class diagram

### 3. Relational Schema

#### 3.1. What is Relational Schema?

Relational Schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases).

### 3.2. Relational Schema

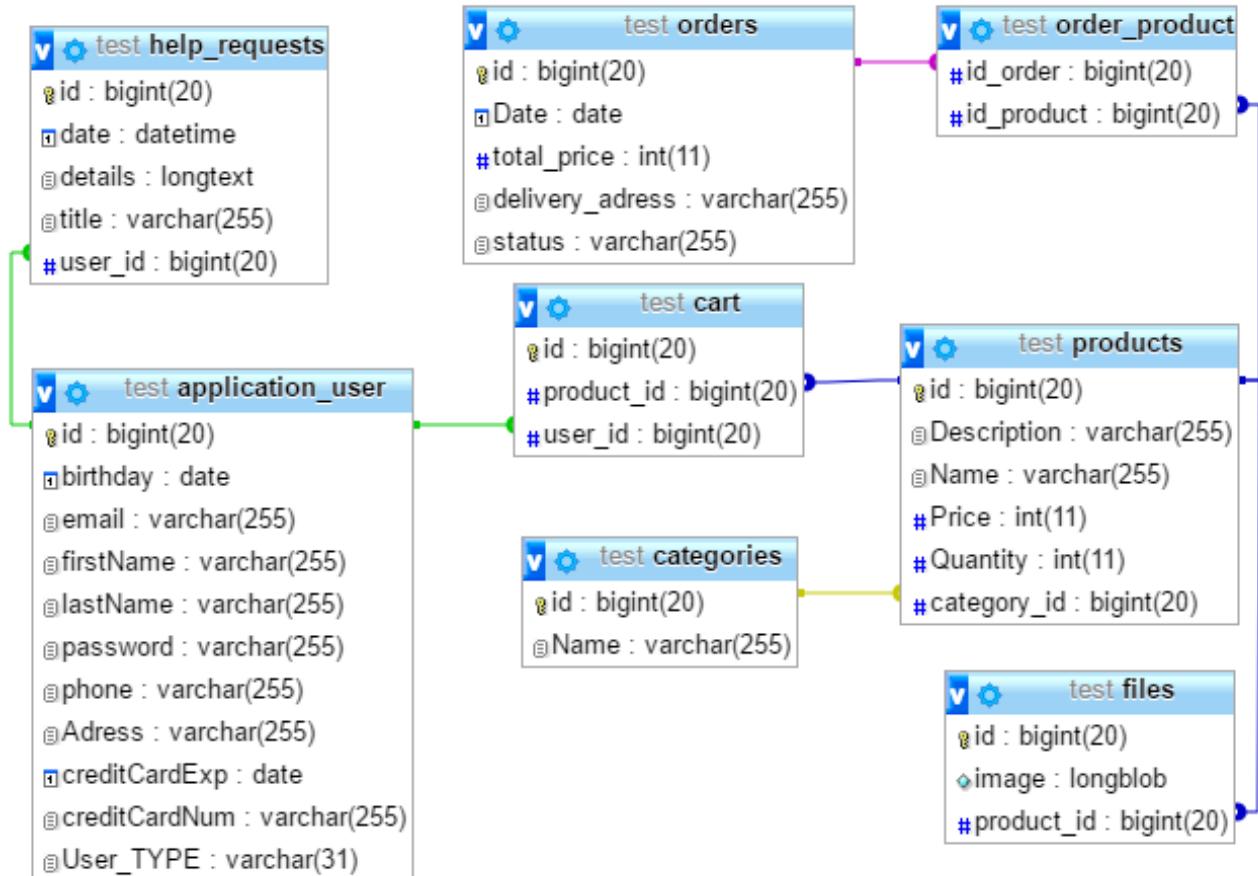


Figure 62: Relational Schema

## 4. Activity diagram

### 4.1. What is Activity diagram?

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

### 4.2. Activity diagram

This section presents the activity diagram and describes the flow of main activities that will be done by the costumer. The following figure provides the overview of the activity of exploring the catalog, then adding product to the cart and finally the placing an order and the payment process.

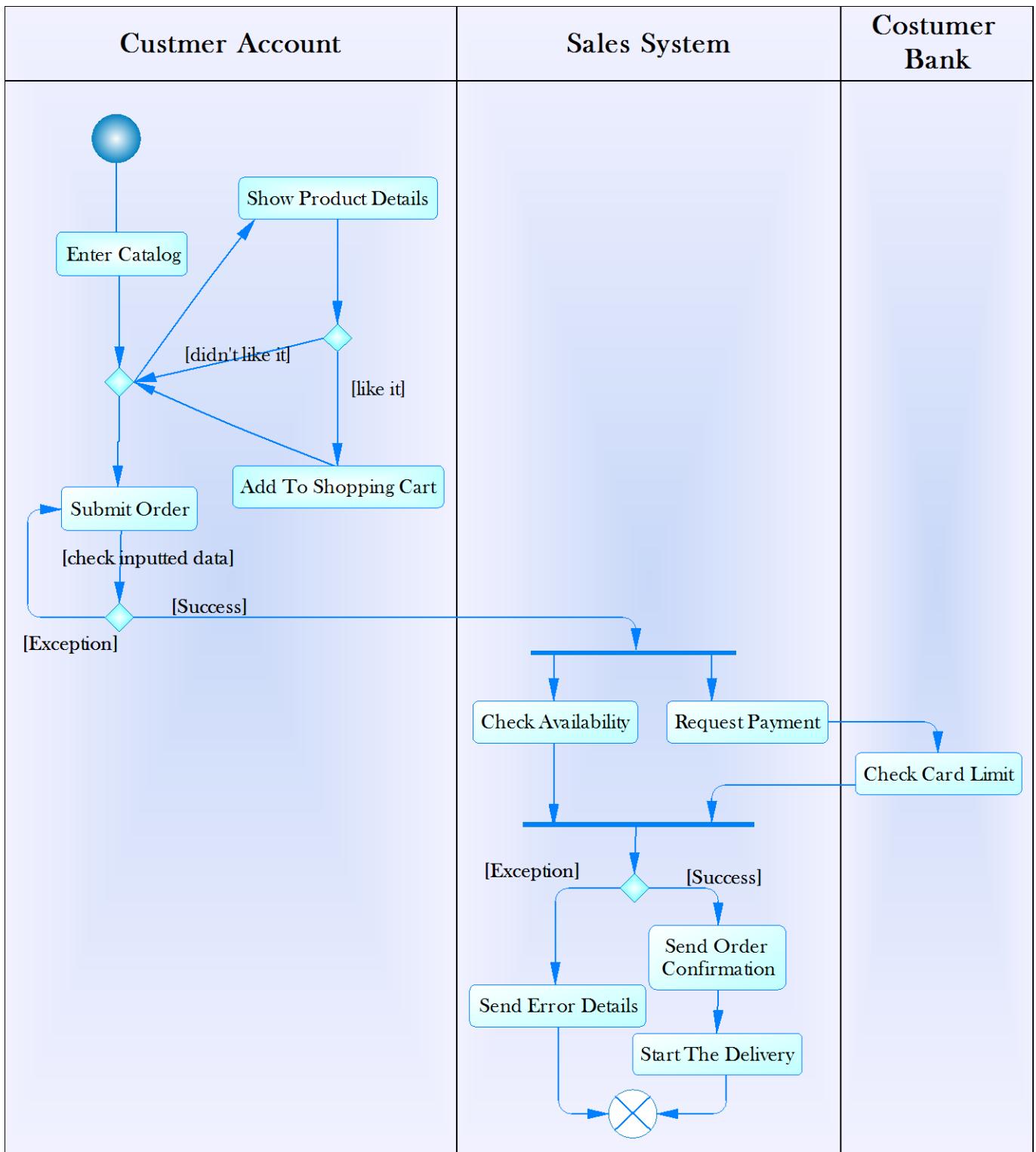


Figure 63: Activity diagram

## Conclusion:

We have finished the detailed design specification, which mainly consist of systems sequence diagrams, detailed sequence diagrams, and class diagram ...Therefore, now we have reached the stage where we can start implementing our project, it a stage where we are ready to code, and deploy our web application, so more other details in the next chapter.

# Forth chapter: System Implementation

## Introduction:

This chapter represents the fourth phase of the V-Model called “Implementation”, details the implementation process of our web application, and describes how the requirements and the targeted state have been reached as well as used methods and practices. First of all, we are going to start by describing the Project architecture, then the design pattern, development environment, deployment and finally by presenting project deliverables through screen-shots (some interfaces).

### 1. Project architecture

Our application is based on the 3-tier architecture, which is hierarchical software architecture with three distinct, independent tiers or layers. It is comprised of the following tiers: presentation, business, and data access. Each tier has a distinct job to perform. The main job of the architecture is to enable software applications to efficiently and quickly respond to user requests. The image below shows a simplified representation of three-tier architecture:

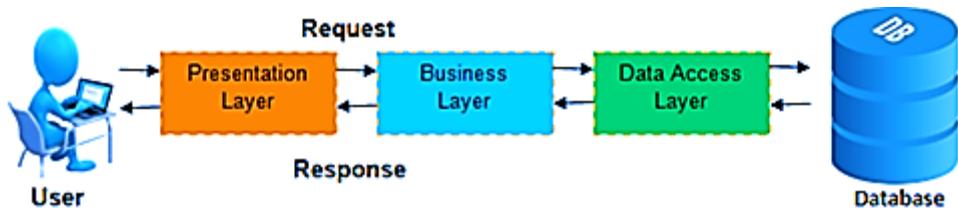


Figure 64: Three tier architecture diagram

Those three tiers are:

- ~ Presentation Tier: is implemented as client machine. A web browser is used to render the graphical user interface that allows the user to interact with a web application by communicating with other tiers by sending requests received from user to the business layer, and displaying the results received from the business layer.
- ~ Business Tier: Also known as the logic tier, middle tier or application tier. Business layer performs database operation after handle the request based on its logic rule, then encapsulates the data, which are returned from the database as result-sets and saved into class format and present as result for the presentation layer.
- ~ Data Tier: is dedicated to store and retrieve the application data independently of application servers or business logic. This framework is especially designed to support any third party relational data base management systems, which give all data manipulation facility to users.

The main advantages of the 3-tier architecture are:

- ~ Since the sizing of each layer can be adapted according to performance requirements, it became easier to handle large volumes of complex and varied operations.
- ~ It is more scalable since it allows the distribution of application components across multiple servers.
- ~ Easier to reuse components.

- ~ Strict separation of the three different tiers, making each tier independent of the others in terms of technology, platform, implementation, and language. In this case, any tier can be modified, maintained, or removed without affecting the others.

## 2. Design pattern

As we motioned earlier, our web application will be based on the MVC pattern (in the Second chapter).

## 3. Adopted programming languages

In this section, we will specify the choice of our application's programming languages

### 3.1. What is programming languages?

A programming language is a formal language that specifies a set of instructions that can be used to produce various kinds of output. Programming languages generally consist of instructions for a computer.

### 3.2. Java EE

Java Platform, Enterprise Edition, or Java EE is a widely used computing platform for development and deployment of enterprise software (network and web services). Java EE was formerly known as Java 2 Platform, Enterprise Edition, or J2EE. The platform uses the object-oriented Java programming language, and provides an API and runtime environment for developing and running large-scale, multi-tiered, scalable, reliable, and secure network applications.

### 3.3. HTML

Short for Hypertext Markup Language, the authoring language used to create documents on the World Wide Web. HTML defines the structure and layout of a Web document by using a variety of tags and attributes.

HTML5 version was used in our application, which is a W3C specification that defines the fifth major revision of the HTML.

### 3.4. CSS

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files.

### 3.5. JavaScript

JavaScript (JS) is a scripting languages, primarily used on the Web. It is commonly found embedded in HTML code. JavaScript is an interpreted language. Thus, it doesn't need to be compiled. JavaScript renders web pages in an interactive and dynamic fashion. This allowing the pages to react to events, exhibit special effects, accept variable text, validate data, create cookies, detect a user's browser, etc.

## 4. Development environment

### 4.1. SAP PowerDesigner



Figure 65: 3.1. SAP PowerDesigner logo

PowerDesigner is the industry-leading data-modeling tool; it provides a wide range of models and diagrams. Such as Merise, UML, Data Warehouse, and business processes. In order to optimize individual and collective productivity, PowerDesigner offers an intuitive, simple to use and customizable interfaces.

### 4.2. XAMPP



Figure 66: XAMPP logo

XAMPP is easy development environment for web development. It is open source and a free download that can install the various programming tools such as Apache, phpMyAdmin, FileZilla, MariaDB... Among those tools, we used **phpMyAdmin**, which is a free and open source administration tool for MySQL.

### 4.3. Eclipse IDE



Figure 67: Eclipse IDE logo

Eclipse is a Java-based open source platform that allows a software developer to create a customized development environment (IDE) from plug-in components built by Eclipse members.

### 4.4. Spring Framework



Figure 68: Spring logo

Spring framework is an open source Java platform. Spring Framework can be used in developing any Java application, but there are extensions for building web applications on top of the Java EE platform. Spring framework targets to make J2EE development easier to use and promotes good programming practices. The spring extensions that were used in our project: Spring Webflow, Spring Security, Spring MVC.

#### 4.5. JPA (Hibernate)



Figure 69: Hibernate logo

Hibernate is an open source object relational mapping (ORM) tool that provides a framework to map object-oriented domain models to relational databases for web applications.

#### 4.6. JSF 2.0 (PrimeFaces)



Figure 70: PrimeFaces logo

PrimeFaces is a Java Server Faces (JSF) component library that ships with a large set of rich components, which utilizes jQuery and jQuery UI under the covers.

#### 4.7. Maven



Figure 71: Maven Logo

Apache Maven is a project management and comprehension tool. Based on the concept of a project object model (POM). Better than any other project build tool, Maven takes in charge the repetitive tasks, download of third party libraries and their dependencies, build all components and create a deployable Jar, War or Ear files.

### 5. Deployment

A web application deployment is a set of activities that enables an application to be ready for the market, and to be publically available and execute in the production environment. It is one of the final phases of software development process. Deployment mostly depends on the application's type, application's infrastructure and the application's design pattern. Since this project is a web application, therefore a server is required so that users can access it via network. A server is a software that manages web application pages and displays them to user via web browser. The web server can be accessed remotely or locally. There are many web servers available. In our case, we used Apache Tomcat for the deployment.

For further explanation, a deployment diagram is represented in the following figure:

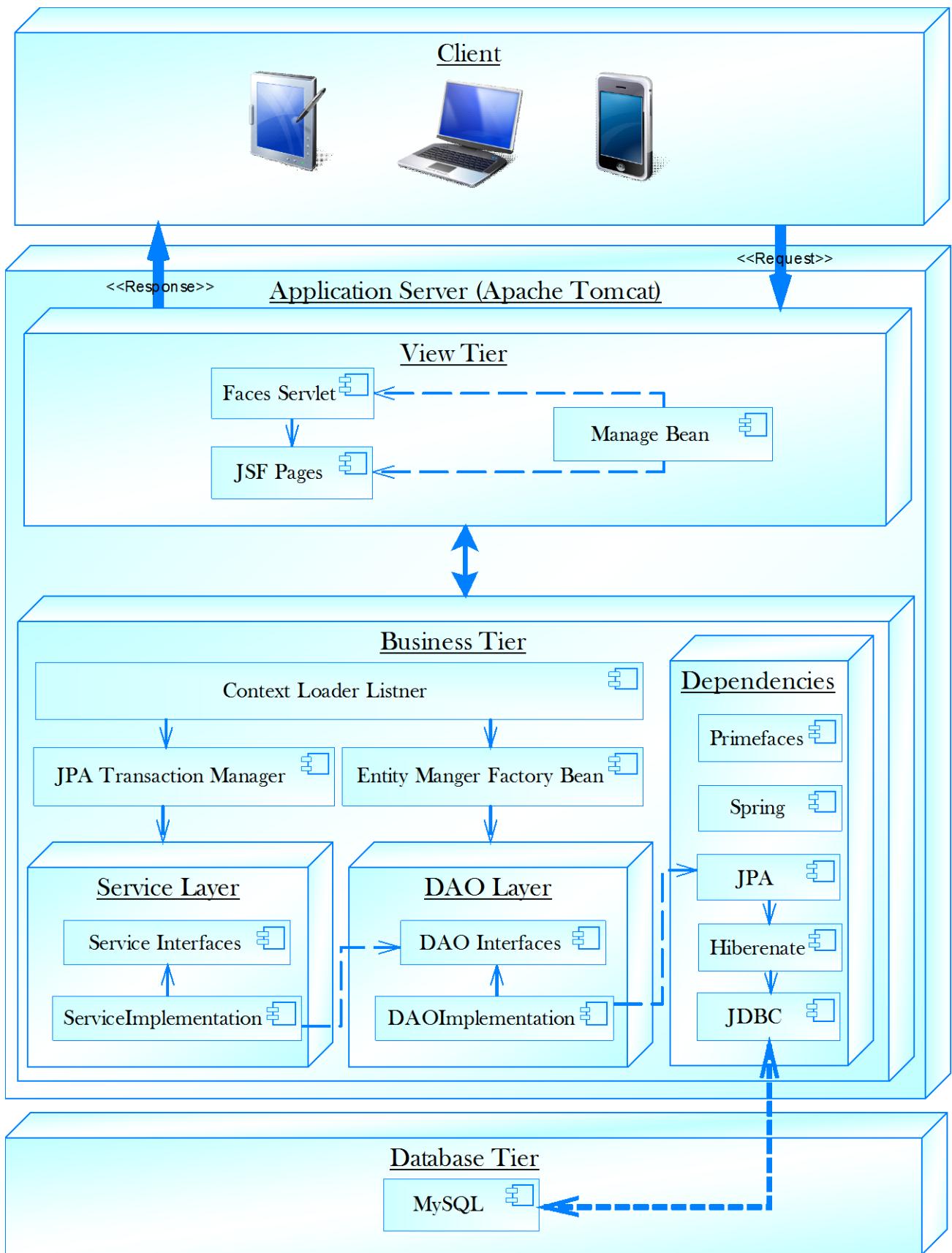


Figure 72: Deployment Diagram

## 6. Navigation diagram

### 6.1. Super webmaster's navigation diagram

The following figure represents the Super Webmaster navigation schema:

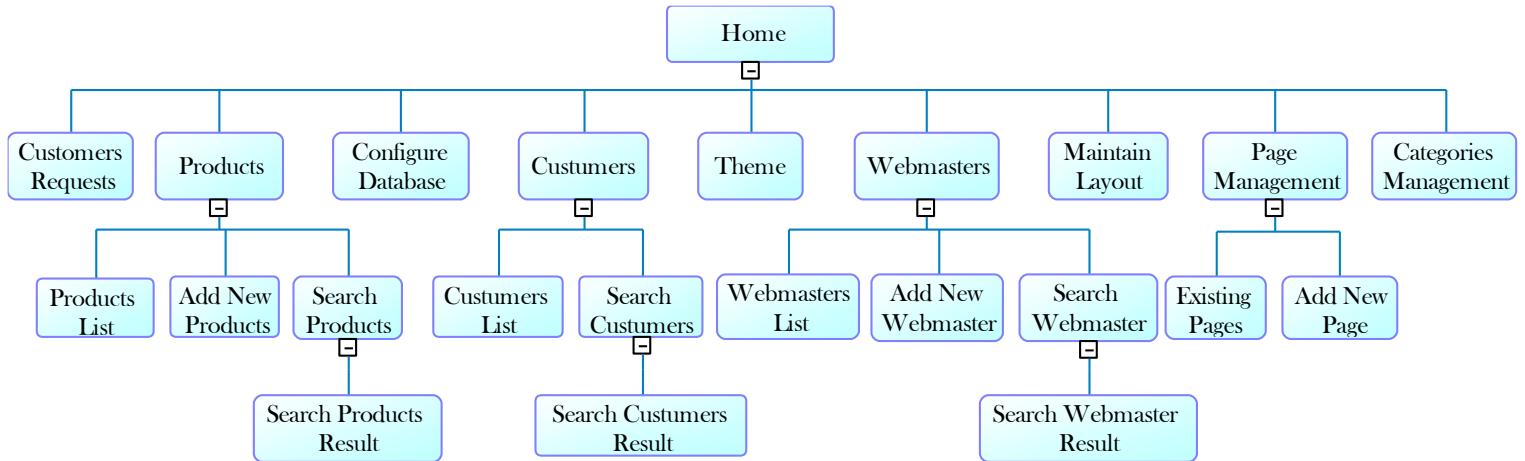


Figure 73: Super webmaster's navigation diagram

## 6.2. Webmaster's navigation diagram

The following figure represents the Webmaster navigation schema, it have the architecture as the Super master but with limited functionalities:

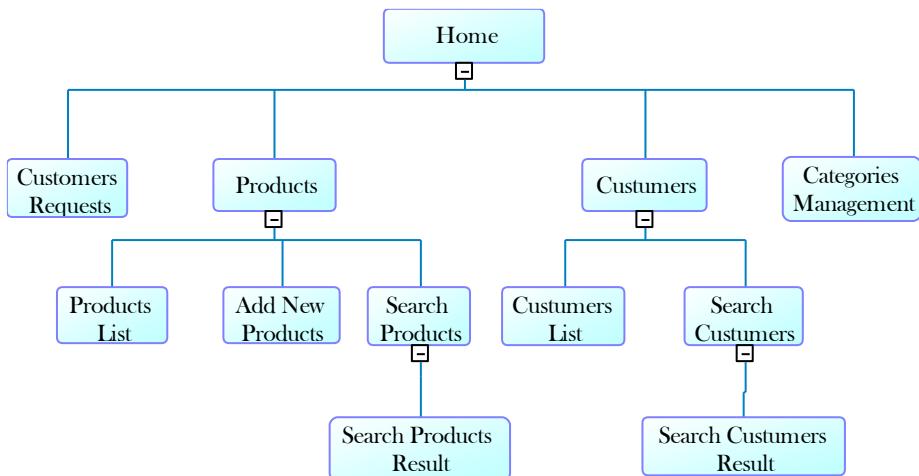


Figure 74: Webmaster's navigation diagram

## 6.3. Customer navigation diagram

The following figure represents the Customer navigation schema:

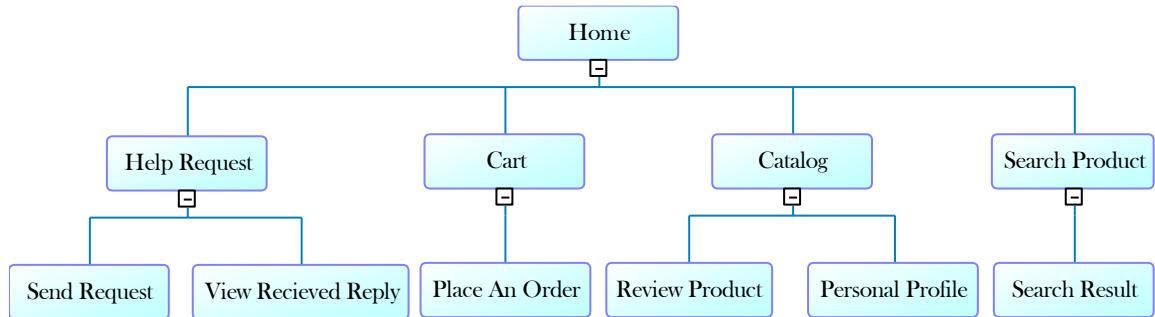


Figure 75: Customer navigation diagram

# 7. Application interfaces

Here are the main interfaces of our application:

## 7.1. Login Interface

The following figure is the login interface of our application, which will be used by every user, whether if he is Super webmaster, webmaster, or a customer then our system will redirect him to the right session.

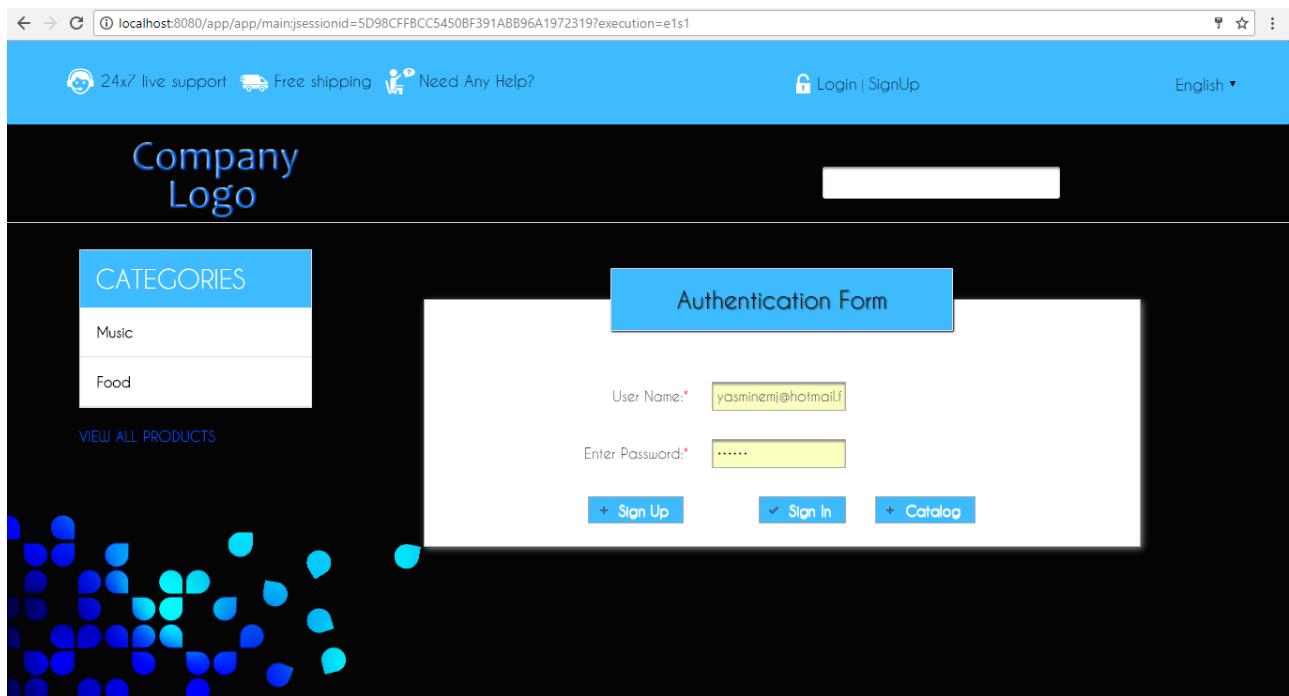


Figure 76: Login Interface

## 7.2. Super Webmaster Interfaces

The Template used for the Super webmaster and the webmaster side (the backend) is fixed as UIB color chart, Logo ... and it is unmodifiable.

### 7.2.1. Maintain Layout Interface

The following interface allows the super webmaster to change the website layout (only on the customer side), the position, and the size of the logo... every component of the website is modifiable on this interface:

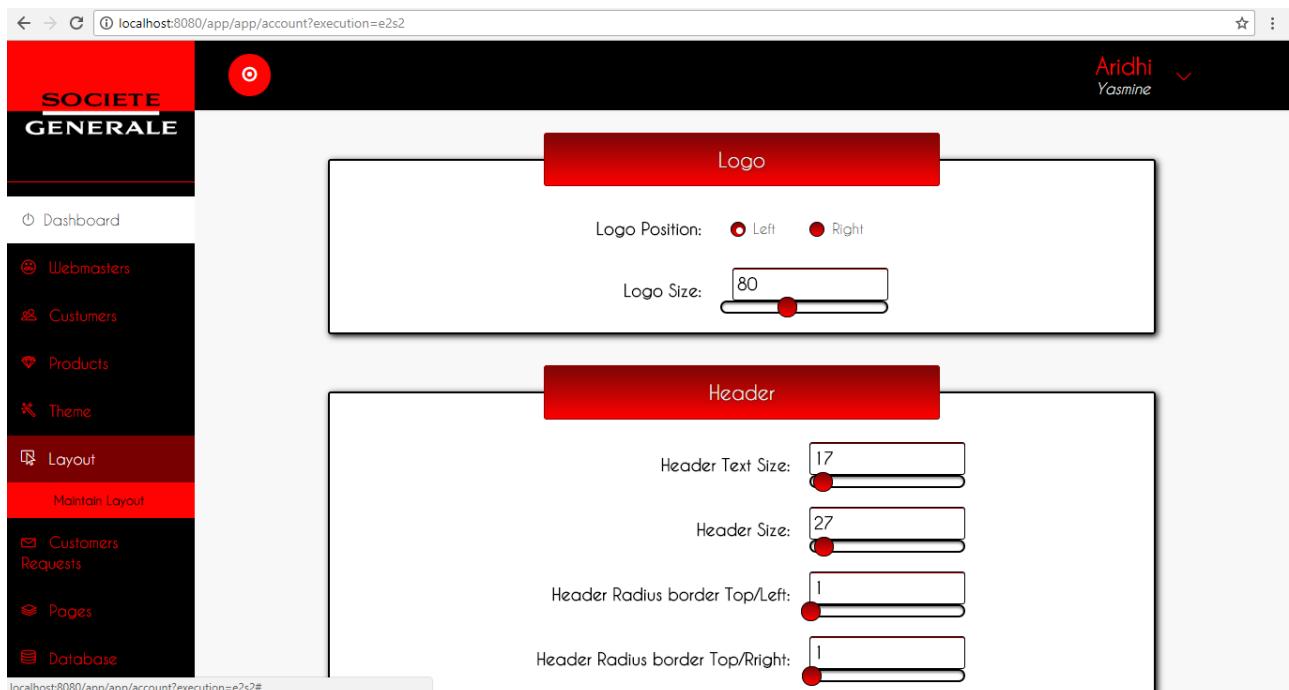


Figure 77: Maintain Layout Interface

### 7.2.2. Change website colors Interface

The following interface allows the super webmaster to change the every color existing in the website (only on the customer side):

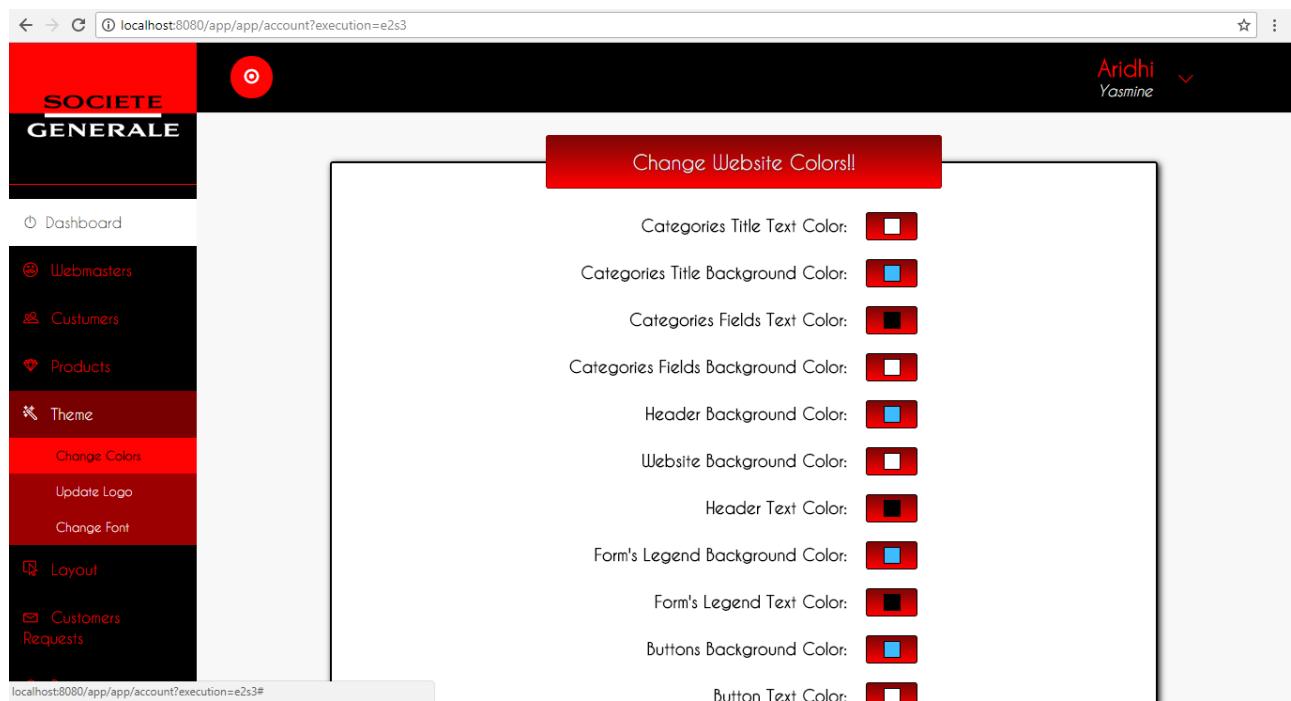


Figure 78: Change website colors Interface

### 7.2.3. Add new webmaster Interface

The following interface provides the super webmaster the form to add new webmaster to the application:

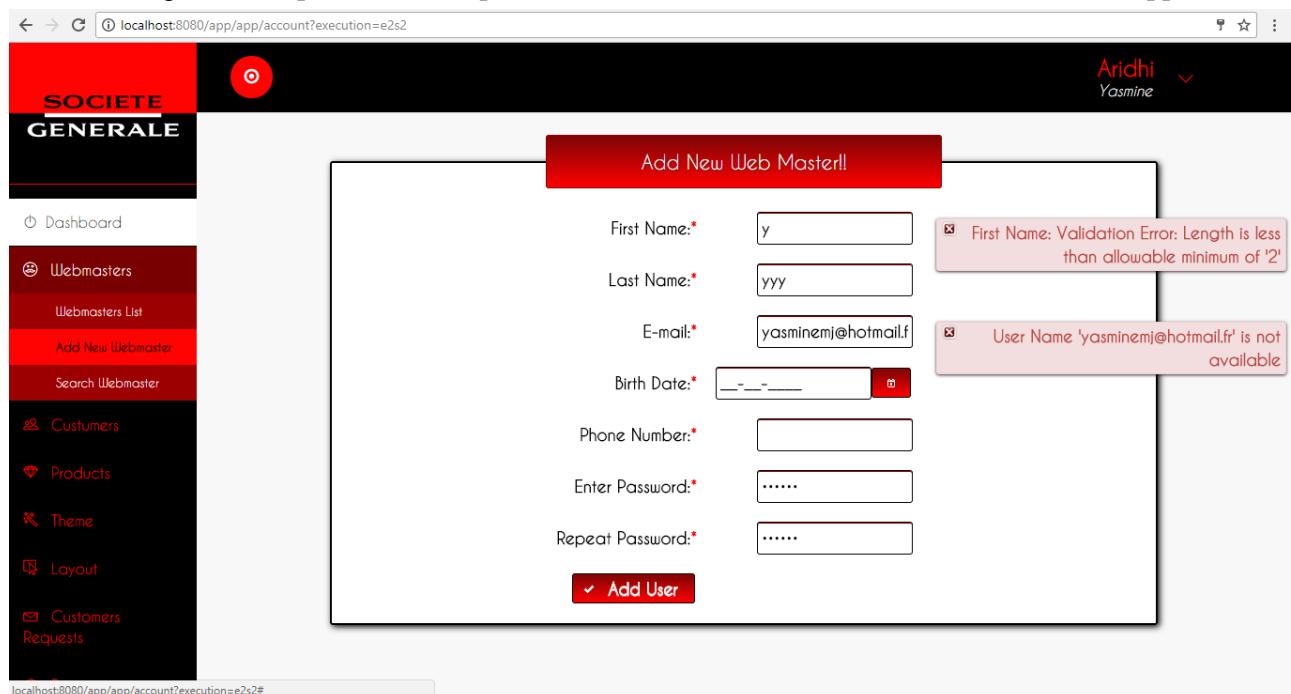


Figure 79: Add new webmaster Interface

#### 7.2.4. Configure Database Interface

If the super webmaster needs to change the database, he only has to fill the form in the following interface:

localhost:8080/app/app/account?execution=e2s3#

SOCIETE GENERALE

Aridhi  
Yasmine

Configure Database!!

Database Url:

User:

Password:

✓ Save

Figure 80: Configure Database Interface

### 7.3. Customer Interfaces

The customization that the super webmaster on the website are applied only on the front-end (customer side).

#### 7.3.1. Explore Catalog Interface

The following figure represents interface of the products catalog:

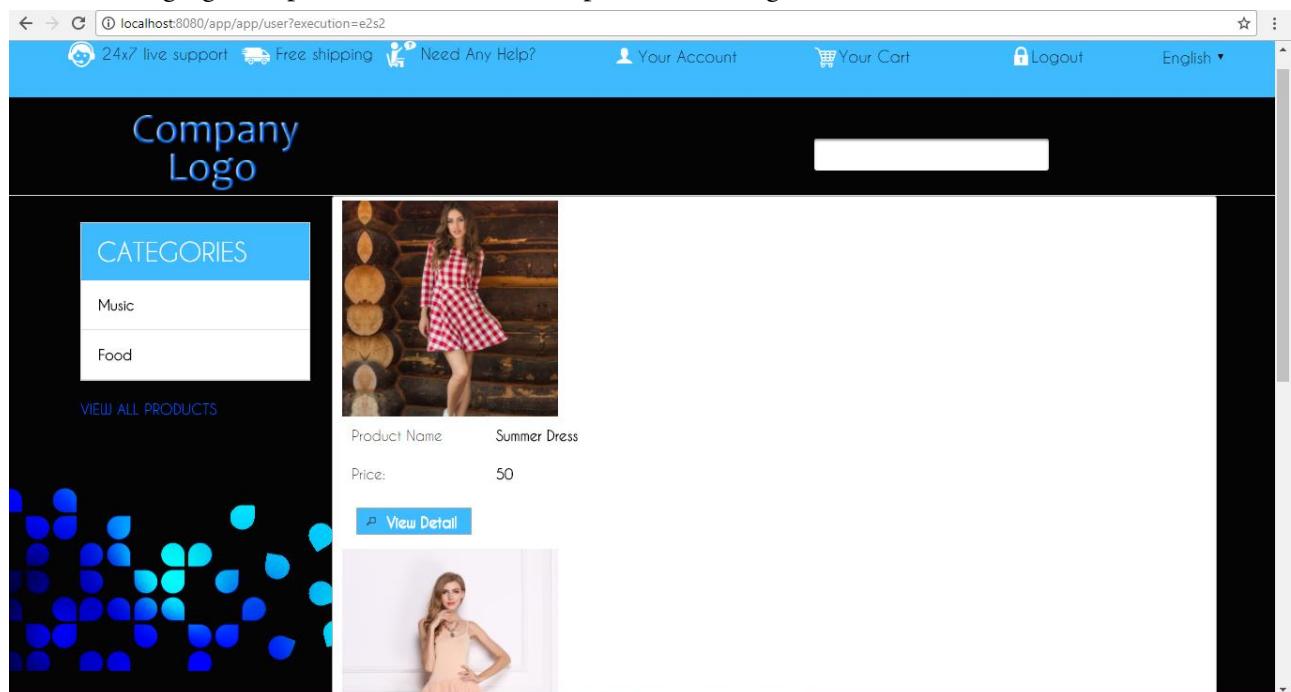


Figure 81: Explore Catalog Interface

### 7.3.2. View Product Details Interface

The following figure represents interface where the product details are displayed:

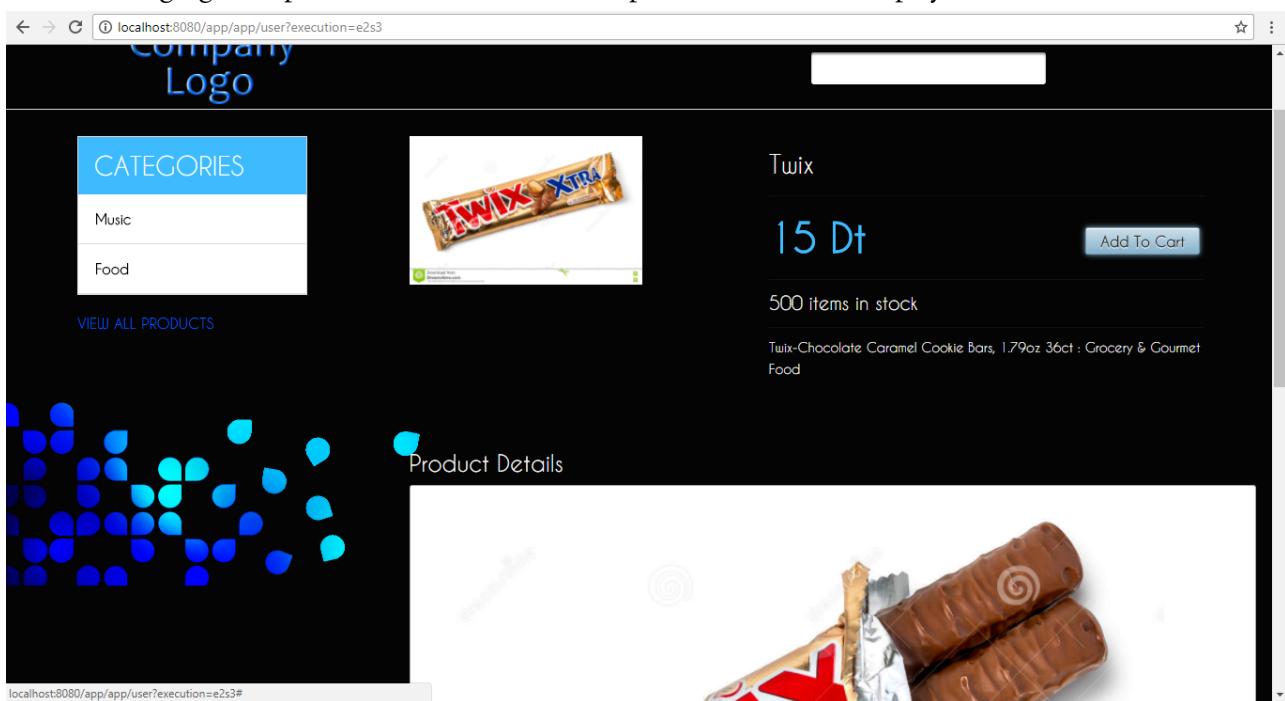


Figure 82: View Product Details Interface

### 7.3.3. Customer's Cart Interface

The following interface display where the products existing on the customer Cart (Panier):

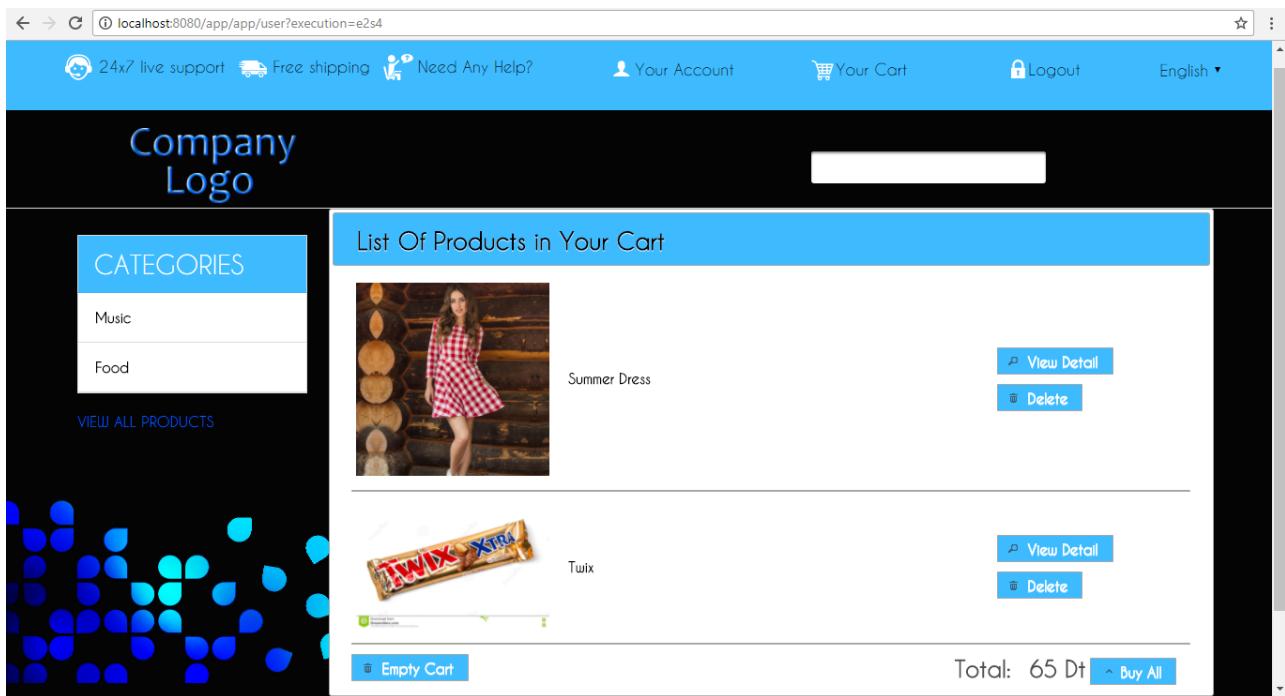


Figure 83: Customer's Cart Interface

## Conclusion:

This chapter, introduced a modeling and generating Web applications built-on the MVC pattern, based on the JSF framework, which can be deployed on the Apache Tomcat Web server. Finally, our project is done, and our application is ready to be used.

# Conclusion

The focus of this work has been the design and realization of a content management system based web application for the UIB Bank, as part of graduation project.

This report, describes the enchainment we followed to realize this project step by step, which its life cycle was divided into four axial phases as follow: Requirements Analysis, General design specification, Detailed Design specification, and System Implementation.

Throughout this work, we build our application using the V-model.

This project helped us in gaining valuable information and practical knowledge on several topics while handling a project from the start to the end, which was a great opportunity to understand the development phases of a project and software development life cycle.

We are glad that we got this opportunity which improved our technical skills. Now we are comfortable using spring Webflow, spring security, hibernate ... which we know nothing about them before.

Our system has been developed with much care, I hope our future client whose will be using it, would find it comfortable, useful and worthy, and grow big business using it.

# WEBOGRAPHY

Société Générale, 2017-03-16

[http://groupe.sgafrique.4lunesdigitales.com/fileadmin/AFRIQUE/Plaquette\\_Pays\\_Tunisie\\_VA\\_Novembre.pdf](http://groupe.sgafrique.4lunesdigitales.com/fileadmin/AFRIQUE/Plaquette_Pays_Tunisie_VA_Novembre.pdf)

What is CMS? , 2017-03-21

[http://www.steptwo.com.au/papers/kmc\\_what/](http://www.steptwo.com.au/papers/kmc_what/)

What is V-Model? , 2017-03-21

[https://www.tutorialspoint.com/sdlc/sdlc\\_v\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm)

Sequence Diagram, 2017-04-27

[http://www.sparxsystems.com/resources/uml2\\_tutorial/uml2\\_sequencediagram.html](http://www.sparxsystems.com/resources/uml2_tutorial/uml2_sequencediagram.html)

Hibernate, 2017-06-02

<http://www.theserverside.com/definition/Hibernate>

XAMPP, 2017-06-02

<http://3dspaint.com/memberblog.php?blog=60004>

Spring, 2017-06-02

[http://www.tutorialspoint.com/spring/spring\\_overview.htm](http://www.tutorialspoint.com/spring/spring_overview.htm)

Eclipse IDE, 2017-06-01

<http://searchmicroservices.techtarget.com/definition/Eclipse>

SAP PowerDesigner, 2017-06-01

[www.powerdesigner.biz/EN/powerdesigner/powerdesigner-features.html](http://www.powerdesigner.biz/EN/powerdesigner/powerdesigner-features.html)

Three Tier Architecture, 2017-06-01

<https://www.techopedia.com/2/32100/software/a-detailed-look-at-3-tier-software-architecture>

Class Diagram, 2017-05-23

<https://www.coursehero.com/file/p57knsuh/CLASS-DIAGRAM-A-class-diagram-in-the-unified-modeling-language-UML-is-a-type-of>

MVC, 2017-05-20

<http://whatis.techtarget.com/definition/model-view-controller-MVC>

What is an Actor?

<http://www.uml-diagrams.org/use-case-actor.html>

UML, 2017-06-09

<http://creately.com/blog/diagrams/uml-diagram-types-examples/>

What is design language? , 2017-06-09

[https://en.wikipedia.org/wiki/Modeling\\_language](https://en.wikipedia.org/wiki/Modeling_language)

What is Programming language? , 2017-06-09

[https://en.wikipedia.org/wiki/Programming\\_language](https://en.wikipedia.org/wiki/Programming_language)

Java EE, 2017-06-09

[https://en.wikipedia.org/wiki/Java\\_Platform,\\_Enterprise\\_Edition](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition)

<http://docs.oracle.com/javaee/6/firstcup/doc/gkhoy.html>

HTML, 2017-06-09

<http://www.webopedia.com/TERM/H/HTML.html>

CSS, 2017-06-09

[https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)

JavaScript, 2017-06-09

<https://www.techopedia.com/definition/3929/javascript-js>

Maven, 2017-06-09

<http://www.javacreed.com/why-should-we-use-maven/>

<https://maven.apache.org/>

Relational Schema, 2017-06-09

[https://en.wikipedia.org/wiki/Database\\_schema](https://en.wikipedia.org/wiki/Database_schema)

## BIBLIOGRAPHY

Design and realization of an automation solution for installing and tracking Delivery Packs Elaborated by:

Asma Ben Mahmoud

# **Conception and development of a Content Management System based Web Applications.**

---

**Yasmine Aridhi**

---

## **Abstract:**

The present project consists on realizing a Content Management System based Web Applications for UIB Bank.

The developed software helps the non-technical people with a little knowledge on computers to update and maintain a shopping web site throughout a “Backend” administration dashboard.

Our system was developed with new web technologies, citing JEE, HTML5, CSS3... by relying on architecture suitable to the MVC architectural pattern.

Furthermore, this project was developed through V-model.