

REPUBLIC OF CAMEROON

Peace – work – fatherland



AFRICAN INSTITUTE OF COMPUTER
SCIENCES PAUL BIYA TECHNOLOGICAL
CENTER OF EXCELLENCE

P.O Box: 13719

Tel: (+237) 242-729-957 | (+237) 242-729-958

Email: contact@iaicameroun.com

Website: www.iaicameroun.com

REPUBLIC OF CAMEROON

Peace – work – fatherland

AFRICAN INSTITUTE OF COMPUTER
SCIENCES PAUL BIYA TECHNOLOGICAL
CENTER OF EXCELLENCE

P.O Box: 13719

Tel: (+237) 242-729-957 | (+237) 242-729-958

Email: contact@iaicameroun.com

Website: www.iaicameroun.com

ACADEMIC INTERNSHIP REPORT

THEME:

DEVELOPMENT OF A WEB-BASED INVENTORY AND SALES
MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR

Internship carried out from 3rd July to the 30th September 2025

In view of obtaining a Higher Technician Diploma (HTD)

Option: SOFTWARE ENGINEERING

Written by:

BALEMBA JESSY NJEA MASSOMA

Third year student in software engineering at AICS Cameroon

Supervised by:

SUPERVISOR:

Mr. Bill Nelson

Lecturer at AICS-Cameroon

Academic Year 2024 - 2025

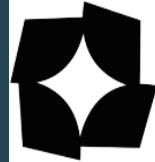


DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



DEDICATION

**THIS DOCUMENT IS DEDICATED
TO THE
BALEMBA'S**



ACKNOWLEDGEMENTS

The path to completing this dissertation has been fraught with both professional challenges and personal adversities. I am profoundly grateful for the enduring support of:

- ❖ God Almighty for His unconditional love and grace, allowing me to go through this intense period safely, successfully, and overcome my fears and challenges with growth.
- ❖ **Mr. Abanda ARMAND Claude**, the Resident Representative of AICS-Cameroon.
- ❖ **Mr. NGU Prince**, my Academic Supervisor for his sage advice, rigorous academic guidance and the confidence he instilled in me.
- ❖ **Mr. Bill Nelson** Our supervisor for his availability, advices and indispensable assistance throughout the internship period.
- ❖ **Mr. AGBOR Anderson**, the Head of the Software Engineering department and lecturer at AICS-Cameroon for his Inspirational pieces of advices, encouragements and moral support.
- ❖ **Mr. MESSIO Ulrich**, Computer Engineer and lecturer at AICS-Cameroon for his academic support. His expertise in UML analysis greatly enriched my work.
- ❖ My Lecturers for their support through out our training at AICS-Cameroon.
- ❖ The amazing family at **IAI** for their necessary support that greatly enhanced my research experience. Their dedication and engagement create an environment conducive to academic success.
- ❖ My friends, who provided both distractions when needed and encouragement when it seemed impossible to continue.
- ❖ My Lovely Family who stepped in not only with emotional reassurance, but also with critical financial support, when personal challenges arose and special thanks to my late father for his moral support and for always being there for me.



SUMMARY

DEDICATION	i
ACKNOWLEDGEMENTS	ii
SUMMARY	iii
LIST OF FIGURES	iv
LIST OF TABLES.....	vii
GLOSSARY.....	viii
ABSTRACT	ix
RESUME	x
GENERAL INTRODUCTION	1
TECHNICAL PHASE	2
CHAPTER 1: EXISTING SYSTEM	3
CHAPTER 2: SPECIFICATION BOOK	12
CHAPTER 3: ANALYSIS	31
CHAPTER 4: CONCEPTION PHASE	76
CHAPTER 5: REALIZATION PHASE	93
CHAPTER 6: FUNCTIONALITY TESTING	103
CHAPTER 7: USER GUIDE	113
GENERAL CONCLUSION	129
ANNEX.....	XIV
BIBLIOGRAPHY	XV
WEBOGRAPHY	XVI
TABLE OF CONTENTS	XVII





LIST OF FIGURES

Figure 1: Project plan	22
Figure 2: UML hierarchy diagram (source: https://images.edrawsoft.com/articles/what-is-uml/uml-diagram-types.png)	35
Figure 3: Phases of UP (source: https://ars.els-cdn.com/content/image/3-s2.0-B9780128054765000022-f02-01-9780128054765.jpg)	38
Figure 4: Graphical representation of 2TUP (source: Mr. Messio UML course AICS-Cameroon 2023-2024)	40
Figure 5: Use Case Diagram Formalism	42
Figure 6: General Use Case Diagram	45
Figure 7: Client Manage Cart Specific Use Case diagram	46
Figure 8: Supplier Overview Use Case Diagram	47
Figure 9: Communication Diagram Formalism	53
Figure 10: Communication Diagram for <<Authentication (Login)>>	55
Figure 11: Communication diagram for <<Make Order>>	56
Figure 12: Communication for <<Create Category>>	57
Figure 13: Sequence Diagram Formalism	58
Figure 14: Sequence diagram for <<Authentication (Login)>>	60
Figure 15: Sequence diagram for <<Make Order>>	61
Figure 16: Sequence diagram for <<Create Category>>	62

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR

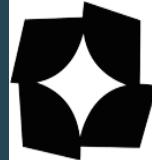


Figure 17: Activity diagram formalism	63
Figure 18: Activity diagram for <<Authentication>>	66
Figure 19: Activity diagram for <<Make Order>>.....	67
Figure 20: Activity diagram for <<Create Category>>	68
Figure 21: Activity diagram for <<Add Product>>.....	69
Figure 22: Visual representation of 3-Tier Architecture (Source: https://docs.aws.amazon.com/images/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/images/image2.png)	75
Figure 23: MVC Architecture (Source: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRx0n1oDz1SskHiZMKI5W0fAZ-KQ33yRIHruA&s).....	76
Figure 24: Class Diagram Formalism	77
Figure 25:General System Class Diagram.....	80
Figure 26: Formalism of a State machine diagram	81
Figure 27: State machine diagram for <<Order>>	83
Figure 28: Formalism of a Package diagram	84
Figure 29: System Package diagram.....	86
Figure 30: Formalism for Deployment Diagram	91
Figure 31: System's Deployment diagram	93
Figure 32: Formalism for Component Diagram	94
Figure 33: System's Component diagram	96
Figure 34:POST Request to /api/order/ to create an order.....	103
Figure 35: GET Request to /api/order to retrieve a list of all orders	104

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroonian Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



Figure 36: Get Request to s/api/order?latest=true with parameter (latest = true) to get the most recent orders	104
Figure 37: API response displaying product details for particular product Id.....	105
Figure 38:Test to request payment from a client on behalf of a supplier	106
Figure 39: Creating MongoDB Atlas Account	112
Figure 40: Creating cluster.....	113
Figure 41: Creating Database (Bigger red circle) and collection (smaller red circle)	114
Figure 42: Allow access(IP Adress) from anywhere (Application Server)	115
Figure 43: Getting connection string to connect backend to cluster in Mongodb Atlas.....	116
Figure 44: Inserting Connection URI to env.local file into Next js project.....	117
Figure 45: Nodejs website for download and installation	117
Figure 46: Install npm	117
Figure 47: Downloading project to target machine	118
Figure 48:Installing Project or Application Dependences	118
Figure 49: Starting the development server	118
Figure 50: Home page with featured and New arrival products	119
Figure 51: All product listing page	120
Figure 52: Cart page.....	120
Figure 53: Payment/Checkout page	121
Figure 54: Shop manager Dashboard.....	121
Figure 55: Delivery Person Dashboard.....	122
Figure 56: Order page	122

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroo Paul Biya Technological centre of Excellence, Academic year 2024-2025

LIST OF TABLES

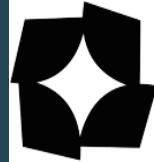
Table 3: Limitations of the existing system and proposed solutions	10
Table 4:Project Plan	21
Table 5: Software Resources (source: mercurial 2024)	23
Table 6: Hardware Resources (source: mercurial 2024).....	25
Table 7: Human Resources	25
Table 8: Total Estimated Cost for Project.....	26
Table 9 : Project Participants	27
Table 10: Comparative Study between UML and MERISE.....	36
Table 11: Comparative study between 2TUP and UP	41
Table 12: Components of a Use Case Diagram	43
Table 13: Components of Use Case Diagram (Continued)	44
Table 14: Formalism of Textual Description of a Use Case.....	48
Table 15: <<Authenticate>> Textual Description	49
Table 16: <<Make Order>> Textual Description	51
Table 17: <<Submit Driver License Application>> Textual DescriptionError! Bookmark not defined.	Error! Bookmark not defined.
Table 18: <<Submit Business License Application>> Textual DescriptionError! Bookmark not defined.	Error! Bookmark not defined.
Table 19: Components of Communication Diagram	54
Table 20: Components of a Sequence Diagram.....	59
Table 21: Components of an Activity Diagram	64
Table 22: Components of a Class Diagram	78
Table 23: Components of a State Diagram	82
Table 24: Components of a Package Diagram.....	85
Table 25: Components of a Deployment Diagram	92
Table 26: Components of a Component Diagram	95



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence,
Academic year 2024-2025



GLOSSARY

2TUP: Two track unified process

API: Application Programming Interface

AICS: African Institute of Computer Sciences

CD-ROM: Compact Disk- Read Only Memory

CRUD: Create Read Update Delete

CSS: Cascading Style Sheet

DB: Database

DBMS: Database Management System

JS: JavaScript

JSX: JavaScript XML

MOMO: Mobile Money

Mr: Mister

MVC: Model View Controller

SQL: Structured Query Language

UML: Unified Modelling Language

UP: Unified Process



ABSTRACT

In today's digital age, small electronics vendors in Cameroon face increasing pressure to modernize operations and meet evolving customer expectations. Traditional methods of managing inventory and recording sales often result in inefficiencies, such as stock mismanagement, delayed order processing, and poor customer service. To address these challenges, our project focuses on the **DEVELOPMENT OF A WEB-BASED INVENTORY AND SALES MANAGEMENT SYSTEM TAILORED FOR A LOCAL ELECTRONICS VENDOR.**

The application, named **Locavend**, allows the shop manager to easily manage product listings, track stock levels, and monitor daily sales, while enabling clients to browse and place orders online. Our approach involved conducting feasibility studies, system design, and full-stack web development using modern technologies. The solution enhances transparency, reduces manual errors, and streamlines operations, ultimately improving the vendor's business efficiency and customer satisfaction.

Keywords:

- ❖ Inventory Management
- ❖ Sales Tracking
- ❖ Order Management
- ❖ Web-based system
- ❖ Electronics Shop



RESUME

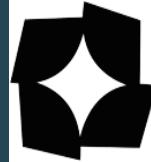
À l'ère du numérique, les petits vendeurs d'électronique au Cameroun sont de plus en plus confrontés à la nécessité de moderniser leurs opérations pour répondre aux attentes croissantes des clients. Les méthodes traditionnelles de gestion de stock et d'enregistrement des ventes entraînent souvent des inefficacités telles que la mauvaise gestion des produits, les retards dans le traitement des commandes et un service client peu satisfaisant.

Pour remédier à ces problèmes, notre projet porte sur le **DEVELOPPEMENT D'UNE APPLICATION WEB DE GESTION DES STOCKS ET DES VENTES** destinée à un **vendeur local d'équipements électroniques**. L'application, nommée **Locavend**, permet au gérant du magasin de gérer facilement les produits, suivre les niveaux de stock et consulter les ventes quotidiennes, tandis que les clients peuvent naviguer et passer leurs commandes en ligne.

Notre démarche a consisté à réaliser des études de faisabilité, l'analyse, la conception et le développement complet du système web à l'aide de technologies modernes. Cette solution permet d'améliorer la transparence, de réduire les erreurs manuelles et de faciliter les opérations, tout en augmentant l'efficacité et la satisfaction client du vendeur.

Mots-clés:

- ❖ Gestion de stock
- ❖ Système Web
- ❖ Suivi des ventes
- ❖ Commandes
- ❖ Vendeur local



GENERAL INTRODUCTION

In recent years, the need for digital transformation has become increasingly apparent among small businesses in Cameroon, especially for local electronics vendors. These businesses often rely on manual methods to manage inventory and sales, leading to inefficiencies such as stock mismanagement, delayed customer service, and lack of accurate sales tracking. As consumer expectations rise and competition intensifies, there is a growing need for local vendors to modernize their operations. This project focuses on addressing these challenges by designing and implementing a web-based application to manage inventory and sales for a local electronics shop. The system provides tools for the shop manager to track stock levels, monitor sales activities, and handle customer orders with greater accuracy and efficiency. Clients can also browse available products and place orders through the platform, enhancing convenience and customer experience. Based on our theme, "**DEVELOPMENT OF A WEB-BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR**", we conducted thorough research to provide a viable solution. This report is structured around a single phase: the **Technical Phase**, which is composed of seven key sections as outlined below:

Technical Phase

- ❖ Existing System
- ❖ Specification Book
- ❖ Analysis
- ❖ Conception
- ❖ Implementation/Deployment
- ❖ Functionality Testing

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



TECHNICAL PHASE

Preamble

This phase focuses on the various characteristics, specificities, and expectations of the subject under our study.

Content:

INTRODUCTION

- I. EXISTING SYSTEM**
- II. SPECIFICATION BOOK**
- III. ANALYSIS PHASE**
- IV. CONCEPTION PHASE**
- V. REALIZATION PHASE**
- VI. FUNCTIONALITY TESTING**
- VII. REALIZATION PHASE**

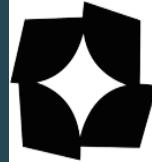
CONCLUSION

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



CHAPTER 1: EXISTING SYSTEM

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Preamble

This document involves gathering information about the current system, including its purpose, functions, processes, and data. The goal of this document is to identify the strengths and weaknesses of the existing system in order to design a new system that better meets the needs of the users or the organization.

Content:

INTRODUCTION

- ❖ PRESENTATION OF THEME
- ❖ STUDY OF THE EXISTING SYSTEM
- ❖ CRITICISMS OF THE EXISTING
- ❖ PROBLEMATIQUE
- ❖ LIMITATIONS OF EXISTING SYSTEM AND PROBLEMATIQUE

CONCLUSION

CONCLUSION

PROBLEMATIQUE

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



INTRODUCTION

We initiate our exploration by delving deep into the existing system. This serves a dual purpose: recognizing its merits and equally importantly, identifying areas where it falls short. By casting a critical eye, we don't just identify problems; we open doors to potential improvements and innovative solutions. This section will articulate challenges, discrepancies, and the gaps we aim to bridge.



I. PRESENTATION THEME

The electronics market in Cameroon, especially for small local vendors, faces several challenges such as manual stock management and the lack of tools to effectively monitor sales. To address these issues, this project aims to develop a tailored digital solution through a web-based application.

THEME: DEVELOPMENT OF A WEB-BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR

DEFINITION OF TERMS:

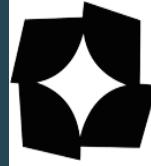
- a.) **Web-based Application:** A software application that runs on a web server and is accessed through a web browser over the internet or an intranet.
- b.) **Inventory:** The collection of products available in the vendor's stock. The application enables the vendor to add, edit, or delete products and monitor their quantities.
- c.) **Sales:** The transactions through which clients purchase products from the vendor. The application helps record completed sales.
- d.) **Management:** The process of organizing and overseeing business operations such as tracking product entries and exits, monitoring orders, and ensuring product availability.

II. STUDY OF THE EXISTING SYSTEM

i.) DELIMITATION OF THE FIELD OF STUDY

Our field of study is focused on the existing processes involved in the purchase, supply, and delivery of cooking gas bottles in Cameroon and Yaoundé in particular. It examines the manual methods of ordering gas, the interactions between clients, suppliers, and delivery personnel, as well as challenges such as fraud, pricing inconsistencies, and lack of communication.

Based on the analysis of this sector, our primary aim is to automate the process of purchase, supply, and delivery by creating a web application accessible to all actors in the sector (clients, suppliers, and delivery personnel) on any platform, as long as it has a browser and internet connection.

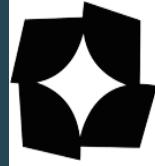


ii.) DESCRIPTION OF THE EXISTING SYSTEM

The current system used by local electronics vendors for managing sales and inventory is mostly manual and inefficient. Based on our observations, the following describes how product purchase and stock management typically occur:

- ❖ When a client wants to buy an item (e.g., phone, laptop, or accessory), they either visit the shop physically or call the vendor to check availability.
- ❖ The vendor manually checks the availability of the product, either from memory, physical stock, or a paper-based record.
- ❖ If the product is available, the vendor communicates the price and holds the product until the client arrives.
- ❖ Sales are noted in notebooks or simple registers without any centralized tracking system.
- ❖ Stock updates are rarely done in real time, leading to difficulties in knowing the exact quantity of items remaining.
- ❖ Tracking of which products sell the most, daily or weekly sales totals, and low-stock alerts is done manually, if at all.
- ❖ If the product is not available, the vendor either informs the client or tries to restock later without a proper notification system.

This manual process often leads to errors, delays, overstocking or stockouts, and missed sales opportunities.



III. CRITICISMS OF THE EXISTING

Following our study of the current operations of a local electronics vendor, several limitations were identified that affect the efficiency of sales and inventory management:

- ❖ **Manual inventory tracking:** Products are tracked using notebooks or memory, increasing the risk of stock miscounts and errors.
- ❖ **No real-time sales monitoring:** The shop manager cannot easily monitor what items sell the most or track daily and monthly revenue without calculations.
- ❖ **No alert for low stock:** Restocking decisions are reactive, based on memory or when a product is completely out, instead of proactive alerts.
- ❖ **Customer dependency on physical visits:** Clients must visit the shop or call to inquire about product availability, which is inconvenient and time-consuming.
- ❖ **Lack of structured sales records:** Without a system to track and store sales history, it's difficult to make data-driven decisions or retrieve past transaction details.

IV. PROBLEMATIC

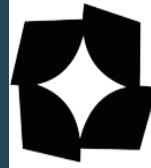
"How can we digitize and streamline the management of sales and inventory for a local electronics vendor to ensure accurate stock tracking, real-time reporting, and improved customer experience?"

V. LIMITATIONS OF EXISTING SYSTEM AND PROPOSED SOLUTIONS

In this section, we will explore the limitations(problems) in detail and propose solutions that can be implemented to address these challenges.

Table 1: Limitations of the existing system and proposed solutions

Problem	Consequences	Proposed Solution(s)
Shop manager tracks inventory manually using notebooks or memory	Risk of stock miscounts and errors Difficulty knowing what is in stock at any moment	Develop a web-based application to digitally manage inventory in real-time with automatic updates when products are sold
Sales records are not properly stored or tracked	No overview of total sales Hard to evaluate best-selling items Difficulty retrieving past sales	Include a sales tracking module in the web application to record each transaction and generate sales reports (daily, weekly, monthly)
Clients must physically visit or call to ask for available products	Time-consuming for clients Loss of potential customers due to lack of information	Build a customer interface to allow clients to browse available products and place orders online



CONCLUSION

Studying the existing system is crucial for several reasons. It allows analysts to understand the current processes, workflows, and functionalities in place. In this section, we describe the existing system, highlight its weaknesses and challenges, and propose a solution. With this foundation, we proceed to the specification book.⁶

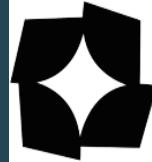


CHAPTER 2: SPECIFICATION BOOK

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



Preamble

This document frames the student's academic internship and is designed collaboratively by the project owner and project manager. This stage involves organization, planning, adherence to pedagogical standards, and work monitoring. It serves as a vital reference point throughout the project, helping to manage expectations, control costs, and maintain quality.

Content:

INTRODUCTION

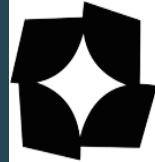
- I. CONTEXT AND JUSTIFICATION OF PROJECT
- II. PROJECT OBJECTIVES
- III. EXPRESSIONS OF NEEDS
- IV. PROJECT PLANNING
- V. ESTIMATED COST OF PROJECT
- VI. PROJECT CONSTRAINTS
- VII. DELIVERABLES

CONCLUSION

CONCLUSION

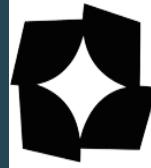
ENCLOSURE 2024-2025

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



INTRODUCTION

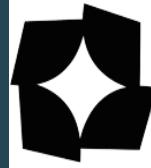
Embarking on a project without a detailed specification book can lead to numerous pitfalls. Detailed specifications are essential for avoiding these issues, ensuring that the project is completed on time, within budget, and to the highest quality standards. This document permits us to provide detailed, precise description of the requirements(clients), deliverables (products), and resources (workmanship) required to complete this project.



I. CONTEXT AND JUSTIFICATION

i.) CONTEXT

In recent years, the retail sector in Cameroon—especially among local electronics vendors—has increasingly shifted toward modern inventory and sales models. Despite this evolution, many local vendors still rely on manual methods to manage their stock and sales, such as recording product entries in notebooks or mentally tracking sales. This creates room for errors, inventory mismanagement, pricing inconsistencies, and poor visibility into daily performance. To address these limitations, we have developed a web-based application—**Locavend**—designed specifically for a local electronics vendor. This application enables digital inventory tracking, automated sales recording, and real-time access to product availability. With **Locavend**, the vendor can efficiently monitor stock levels, update prices, and generate sales reports, while clients can view available products online and place orders. This improves both management and customer experience, offering a reliable, structured, and transparent system that meets the evolving needs of small retail businesses in Cameroon.

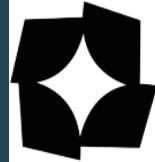


ii.) JUSTIFICATIONS

The persistent challenges faced by local electronics vendors in Cameroon, especially in urban centers like Yaoundé, highlight the urgent need for a digital solution. Traditional sales and inventory management practices—such as manual stock tracking, paper-based sales records, and lack of real-time data—lead to errors, inefficiencies, and lost revenue.

Our proposed project, “**DEVELOPMENT OF A WEB-BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR**”, directly addresses these issues. By building a dedicated web application, we aim to:

- ❖ Digitize and automate inventory management for better stock control
- ❖ Track sales activities in real-time with automated reporting
- ❖ Improve order processing through structured product listings
- ❖ Enhance accuracy and decision-making with clear data visualization
- ❖ Eliminate manual errors and inefficiencies for smoother operations



II. PROJECT OBJECTIVES

i.) GENERAL OBJECTIVE

Develop a web-based inventory and sales management system for a local electronics vendor that enables the vendor to manage stock, track sales, and process client orders more efficiently and accurately.

ii.) SPECIFIC OBJECTIVES

With this platform:

❖ **Shop Manager should be able to:**

- ✓ Add, update, and delete products from inventory
- ✓ Track product quantities and get notified of low stock
- ✓ View and manage client orders in real-time
- ✓ Update the status of each order (e.g., pending, processing, completed)
- ✓ View analytics on total sales, orders per day/week/month, and product performance

❖ **Clients should be able to:**

- ✓ Browse available electronic products (phones, laptops, etc.)
- ✓ Place orders from the shop's catalog
- ✓ View order confirmation and status updates



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



The project has the following characteristics:

- ✓ Name of Project: **LocaVend**.
- ✓ Project Target: Digitalize inventory and sales management for a local electronics vendor.
- ✓ Technical Specification: Web-Based Application using Next.js (full-stack).

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



III. EXPRESSION OF NEEDS

i.) FUNCTIONAL NEEDS

These are the requirements that the system must meet to fulfil its purpose, typically expressed in terms of the system's inputs, outputs, and behaviours. They are as follows:

- ❖ The Shop manager should be able to:

- | | |
|--|--|
| <ul style="list-style-type: none">⊕ Manage Account⊕ Manage Product Stock⊕ Add New Product (e.g., phones, laptops,)⊕ Update Product Info | <ul style="list-style-type: none">⊕ Delete Product⊕ Manage Order⊕ Management Payments⊕ Track Product quantities |
|--|--|

- ❖ The Client should be able to:

- | | |
|--|--|
| <ul style="list-style-type: none">⊕ Place Order⊕ Review Order⊕ Cancel Order⊕ Make Payment | <ul style="list-style-type: none">⊕ Successful Payment⊕ Successful Order Notification |
|--|--|
-
- | | |
|---|--|
| <ul style="list-style-type: none">⊕ Pay by Cash on Delivery⊕ Pay by Mobile Money or Orange Money | |
|---|--|



ii.) NON-FUNCTIONAL NEEDS

These specify the quality attributes of a software system. They evaluate the software system based on its performance, usability, scalability, portability, and other non-functional standards that are critical to its success. Failing to meet these non-functional needs can result in the system not fulfilling the users' needs.

- ❖ Performance and Scalability:

Performance refers to how quickly the web application responds to user actions under various workloads. It measures how long a user must wait for the system to process an action, such as placing an order, even when multiple users are using the application simultaneously. Our goal is to deliver optimal performance, as it greatly impacts user satisfaction and overall experience.

Scalability, on the other hand, refers to the system's ability to maintain its performance standards as the number of users increases. The platform should scale efficiently, ensuring clients, suppliers, and delivery personnel can all access it comfortably, even during peak demand.

- ❖ Accessibility: The platform should be accessible across multiple platforms, including mobile phones, tablets, and desktop computers. As long as a user has a browser and an internet connection, the system should work seamlessly, ensuring flexibility in use.
- ❖ The application should be intuitive, with a simple user interface that allows users to easily navigate and complete tasks;
- ❖ The application should implement robust security measures to protect user data, ensuring safe transactions and preventing unauthorized access; The code should be clean and well-organized to allow for easy updates and future improvements.

IV. PROJECT PLANNING

Project planning involves scheduling tasks and milestones within a set time frame, based on the structured phases of the project. Our personal project spanned three months, from the 03th of April to the 21st of June, and the following outlines how the work phases were organized during this period.

i.) PROJECT PLAN

Table 2:Project Plan

TASK	Duration(days)	Period
Prepare an Existing System	14 days	03 rd April - 18 th April
Production of Revised Specification Book	5 days	19 th April – 25 th April
Solution Analysis and Drafting of the Analysis Document	5 days	26 th April – 2 nd May
Conception	14 days	3 rd May – 17 th May
Realization	14 days	20 th May – 3 rd June
Functionality Test	14 days	4 th June – 18 th June
Writing User Guide	4 days	19 th June – 21 st June

ii.) GANTT DIAGRAM

The Gantt chart is a project management tool that provides a visual representation of a project's schedule. It displays the various tasks and their timelines, allowing project managers to track progress and ensure timely completion. The Gantt chart for this project is as follows:

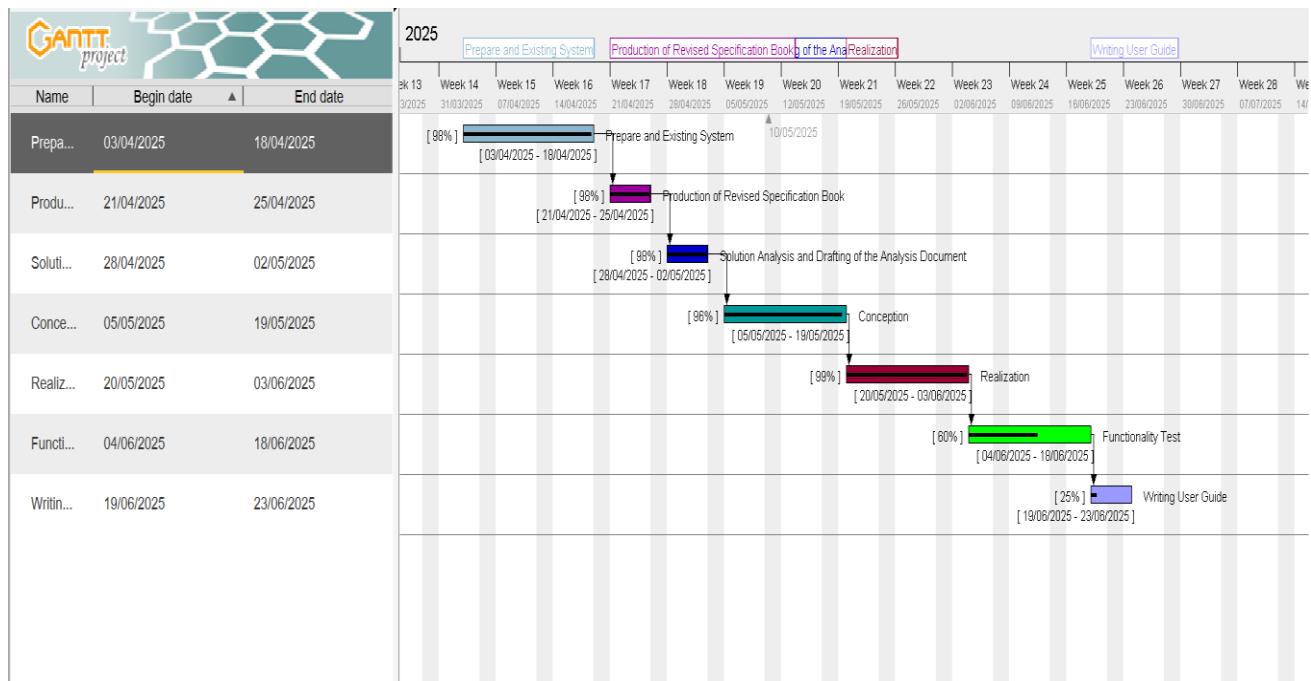


Figure 1: Project plan

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



V. ESTIMATED COST OF PROJECT

Project cost estimation involves calculating the total expenses, including human resources, hardware, and software of the project. It is critical for any project and essential for effective project management. The tables below provide a detailed breakdown of the project cost estimation.

i.) SOFTWARE RESOURCES

The following software applications or resources were necessary for the successful realization of this project:

Table 3: Software Resources (source: mercurial 2024)

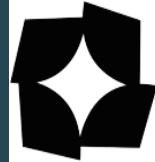
Software Resource	Description	Quantity	Price (FCFA)/ day	Total Price (FCFA)
Microsoft 365	Suite for document creation, spreadsheets, presentations, and collaboration.	1	47, 998	47, 998
Visual Studio	Integrated Development Environment (IDE) for coding and debugging.	1	Freeware	Freeware
Node.js	JavaScript runtime environment for building server-side applications.	1	Freeware	Freeware

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



MySQL Workbench Community Edition	Database design and management tool for MySQL.	1	Open Source	Open Source
XAMPP	Local server for testing and running PHP and MySQL applications.	1	Open Source	Open Source
Git	Version control system used to track changes in the project code.	1	Open Source	Open Source
GitHub	Online platform for hosting and managing code repositories.	1	Freemium	Freemium
Thunder Client	API testing tool integrated with Visual Studio Code.	1	Freeware	Freeware
Mozilla Firefox	Web browser used for testing and development.	1	Freeware	Freeware
Visual Paradigm	creating UML diagrams and system modelling.	1	Freemium	Freemium
Icogram	Software for designing illustrative diagrams and graphics.	1	Freemium	Freemium
GanttProject	Project management tool for creating Gantt charts	1	Freeware	Freeware
Total Cost of Software Resources (FCFA)				47,998

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



ii.) HARDWARE RESOURCES

This includes the material resources used to realise the project. The table below provides a summary of these materials.

Table 4: Hardware Resources (source: mercurial 2024)

Hardware Resource	Quantity	Price (FCFA) / day	Total Price (FCFA)
Laptop Acer Spin 3 10th Gen Intel Core i5- 1035G1 14, 8 GB RAM, 256 GB SSD	01	287,359.25	287,359
Modem (Camtel)	01	40,000	40,000
Internet Connection	/	20,990	20,990
Total Cost of Hardware Resources (FCFA)			348,349

iii.) HUMAN RESOURCES

Table 5: Human Resources

Human resource	Quantity	Price (FCFA)/ day	Duration (days)	Total Price (FCFA)
Project Manager	01	40,500	90	2,745,000
Analyst	01	35,000	21	735,000
UI/UX Designer	01	30,000	10	300,000
Developer	02	80,000	30	2,400,000
Tester	01	20,000	14	280,000
Total Cost of Human Resources (FCFA)				6,460,000

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



TOTAL ESTIMATED COST FOR THE PROJECT

Table 6: Total Estimated Cost for Project

SOFTWARE RESOURCE	47,998 FCFA
HARDWARE RESOURCE	348,349 FCFA
HUMAN RESOURCE	6,460,000 FCFA
Unexpected charges (10%)	680,939.4 FCFA
(Total cost) * 10%	
TOTAL COST	7,490,333.4 FCFA
Total	7,490,333.4 FCFA

SUM TOTAL: 7, 490, 333.4 FCFA**Seven Million, Four Hundred and Ninety Thousand Three****Hundred and Thirty-Three Point Four FCFA**



iv.) PARTICIPANTS

The table below presents the different individuals who took part in the accomplishment of this project. They include:

Table 7 : Project Participants

Names	Functions	Role and Tasks
Mr. BILL NELSON	Lecturer at AICS-Cameroon Software Engineer	Supervisor
Mr. BALEMBA JESSE NJEA	Second year Student at AICS - Cameroon	Project Head, Analyst, Project Design and Coding, Testing.
MASSOMA		



VI. PROJECT CONSTRAINTS

Because every project and its resources are finite, we must respect three main constraints which includes;

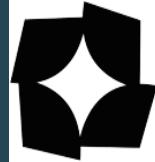
- ❖ Budget Constraint: The total budget for the project is strictly limited to 7,490,333.4 FCFA and must be adhered to without exceeding this amount;
- ❖ Deadline Constraint: The project must be completed before 21st June while meeting all specified objectives and milestones;
- ❖ Quality Constraint: The application must be flexible, web-based, and reusable, ensuring it meets high standards of user-friendliness, reliability, and security.

VII. DELIVERABLES

In this section, we will outline the elements to be delivered upon project completion.

These include:

- ❖ A CD-ROM containing the web application;
- ❖ The installation guide and user manual;
- ❖ A PowerPoint presentation of the application;
- ❖ A video demonstration of the application's functionality.



CONCLUSION

Creating a specification book is a necessary step in minimizing the risks of a project. On the one hand, it serves to fulfil the requirements listed in the specifications and to plan the implementation in the best possible way so that there are no nasty surprises at the end. On the other hand, it helps to validate the implemented solution at the end of the project and to protect both parties. Through these we are now going forth to the Analysis.



CHAPTER 3: ANALYSIS

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Preamble

The Analysis file provides a detailed explanation of the solution using diagrams. Its main purpose is to capture user's needs and define the process and methodologies used to address those needs.

Content:

INTRODUCTION

I. METHODOLOGY

- a.) COMPARATIVE STUDY BETWEEN MERISE AND UML
- b.) COMPARATIVE STUDY OF UNIFIED PROCESSES

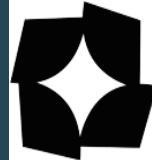
II. MODELLING

- a.) USE CASE DIAGRAM
- b.) COMMUNICATION DIAGRAM
- c.) SEQUENCE DIAGRAM
- d.) ACTIVITY DIAGRAM

CONCLUSION

CONCLUSION

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



INTRODUCTION

The analysis phase of a project is a crucial step in ensuring that the solution aligns with the needs of the users. In this phase, we delve deeper into the proposed solution, providing a comprehensive explanation supported by diagrams. This document serves to clearly capture the user's requirements while outlining the processes and methodologies that will be used to meet those needs. By offering a detailed analysis, we ensure that the project is built on a strong foundation, minimizing the risk of misunderstandings and ensuring a more successful implementation.



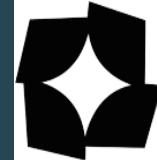
I. METHODOLOGY

Analysis is a fundamental step in the conception of software and serves as the foundation for the realization of any information system. An information system involves the organization of resources; human, software, and hardware along with data and feedback mechanisms, to achieve a specific objective. Several methods and languages, such as UML and MERISE, have been developed to facilitate the analysis and design of information systems.

MERISE (Méthode d'Etude et de Realization Informatique pour les Systèmes d'Entreprise) it is a method of analysis, conception and realization of information systems mostly used in French Speaking Companies or Enterprises. It is based on the separation of data and processing them using several conceptual and physical models. Its principal objective is to conceive an information system. Merise proposes examining the real system from two perspectives:

- ❖ The Static View (data);
- ❖ The Dynamic View (treatments).

UML (Unified Modelling System) a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. UML proposes an approach different from MERISE as its readability and reusability makes it an ideal choice for programmers.



UML is currently at its 2.5 version and presents a number of advantages which includes:

- ❖ It is a formal and standardized language.
- ❖ It provides visual representation of the structure and behaviour of the system which can help to communicate and understand the design of the system.
- ❖ It is flexible and customizable to suit different domain and technologies.
- ❖ It is readable and reusable i.e. you can easily see the relationships and interactions between classes and objects in UML.

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



UML is divided into two categories of diagrams:

- ❖ **Structural diagrams**, which illustrate the building blocks of the system, and
- ❖ **Behavioural diagrams**, which demonstrate how the system responds to changes, as shown below.

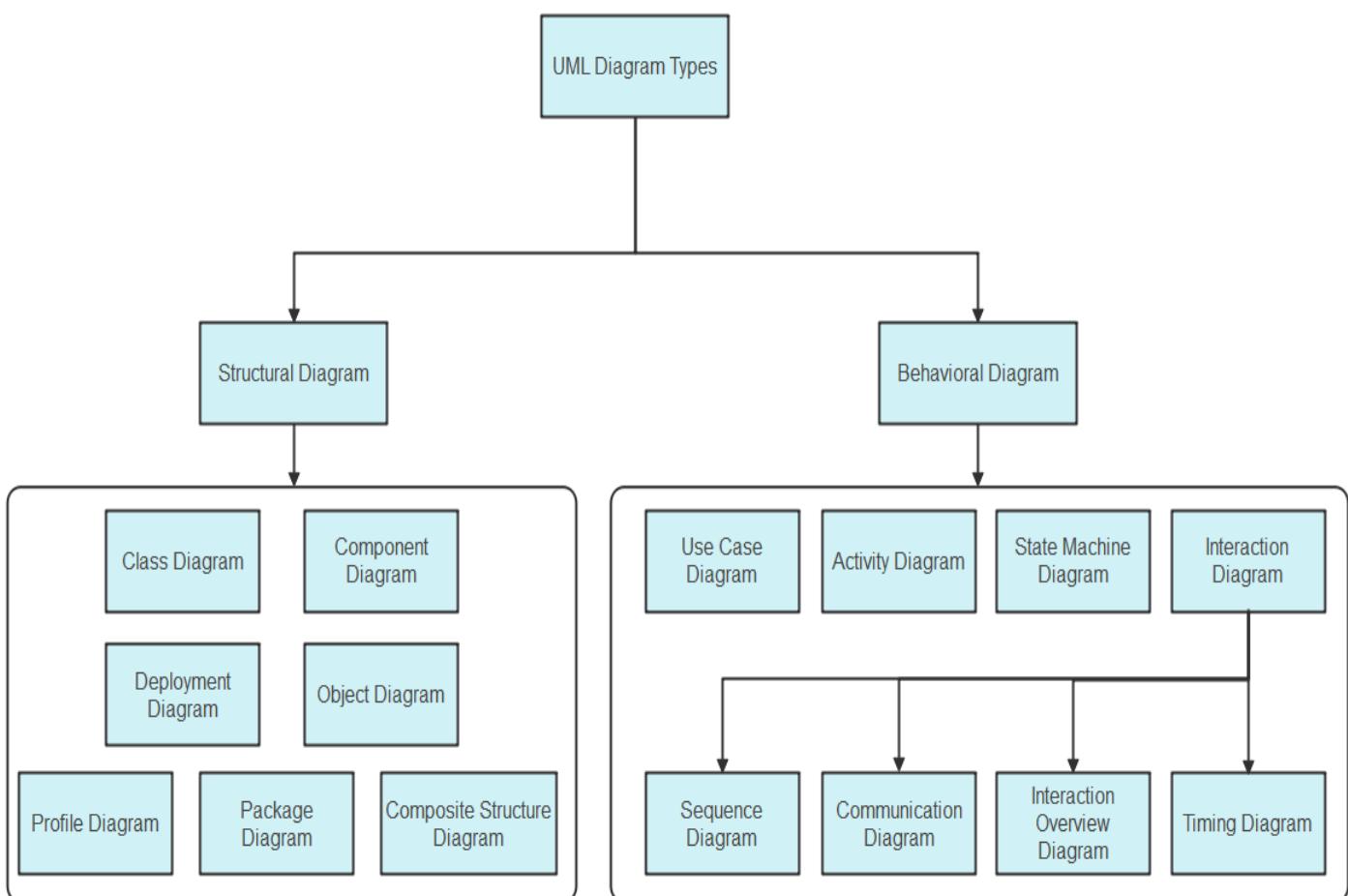
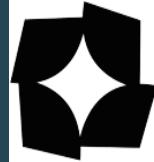


Figure 2: UML hierarchy diagram (source: <https://images.edrawsoft.com/articles/what-is-uml/uml-diagram-types.png>)

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



a.) COMPARATIVE STUDY BETWEEN MERISE AND UML

Table 8: Comparative Study between UML and MERISE

MERISE	UML
MERISE stands for Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprises.	UML stands for Unified Modelling Language.
MERISE is a method for designing processing- and data-oriented systems.	UML is a modelling language, not a method. It needs a method that uses the UML language to design the system. (e.g. UP, RUP, 2TUP and possibly agile methods).
Less widely used	Widely used
Less complex and little or no time-consuming.	Complex and time-consuming due to the many diagrams it has.
Design for Organizational Information Systems.	Designed for Object-Oriented-based Information Systems.

b.) COMPARATIVE STUDY OF UNIFIED PROCESSES

i.) PRESENTATION OF UP METHOD

UP (Unified Process) is an iterative, incremental, architecture-centric, and use-case-driven approach to software development built on UML. UP is organised into four major phases:

- ❖ **Inception:** Defines project foundation, business case, scope, and boundaries.
- ❖ **Elaboration:** Captures system requirements and plans risk management.
- ❖ **Construction:** Builds the final product iteratively, focusing on lower-risk elements.
- ❖ **Deployment:** Delivers the system to users, including data migration and training.

Each phase and its iteration consist of a set of predefined activities also known as disciplines. These disciplines include: **Business modelling, Requirements, Analysis and Design, Implementation, Test and Deployment.**



The different phases of the Unified Process (UP) are shown below:

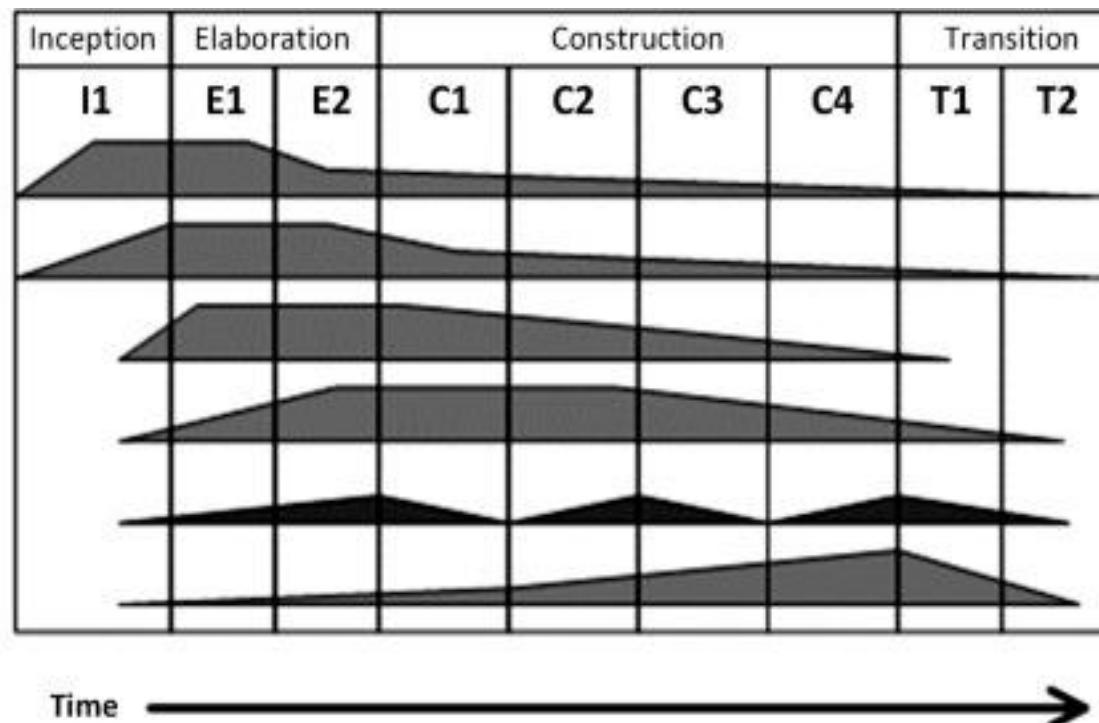


Figure 3: Phases of UP (source: <https://ars.els-cdn.com/content/image/3-s2.0-B9780128054765000022-f02-01-9780128054765.jpg>)



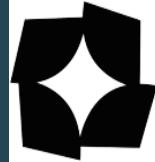
ii.) PRESENTATION OF THE 2-TUP METHOD

2TUP is a software development process that implements the unified process (i.e. iterative, incremental, based on UML). It proposes a development cycle which separates the technical aspects from the functional aspects. It begins with a preliminary study which essentially consists of identifying the actors and the system exchange. Then to produce the specifications and to model the context.

The software development process is structured around three branches:

- ❖ **Functional branch (the left-hand branch of the Y):** identifies and models the users' business needs.
- ❖ **Technical branch (the right-hand branch of the Y):** lists the technical needs and proposes a generic design validated by a prototype.
- ❖ **Realization branch (the middle branch):** combines both functional and technical branches, enabling application design and ultimately delivering a solution tailored to the needs. This branch includes:
 - + **Preliminary design:** integrates the analysis model into the technical architecture to map out the system's components;
 - + **Detailed design:** studies how to make each component;
 - + **Coding stage:** produces the components and progressively tests the code units produced;
 - + **Acceptance(recipe) stage:** consists in validating the functions of the developed system.

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



Below is a graphical representation of the Two Track Unified Process (2TUP)

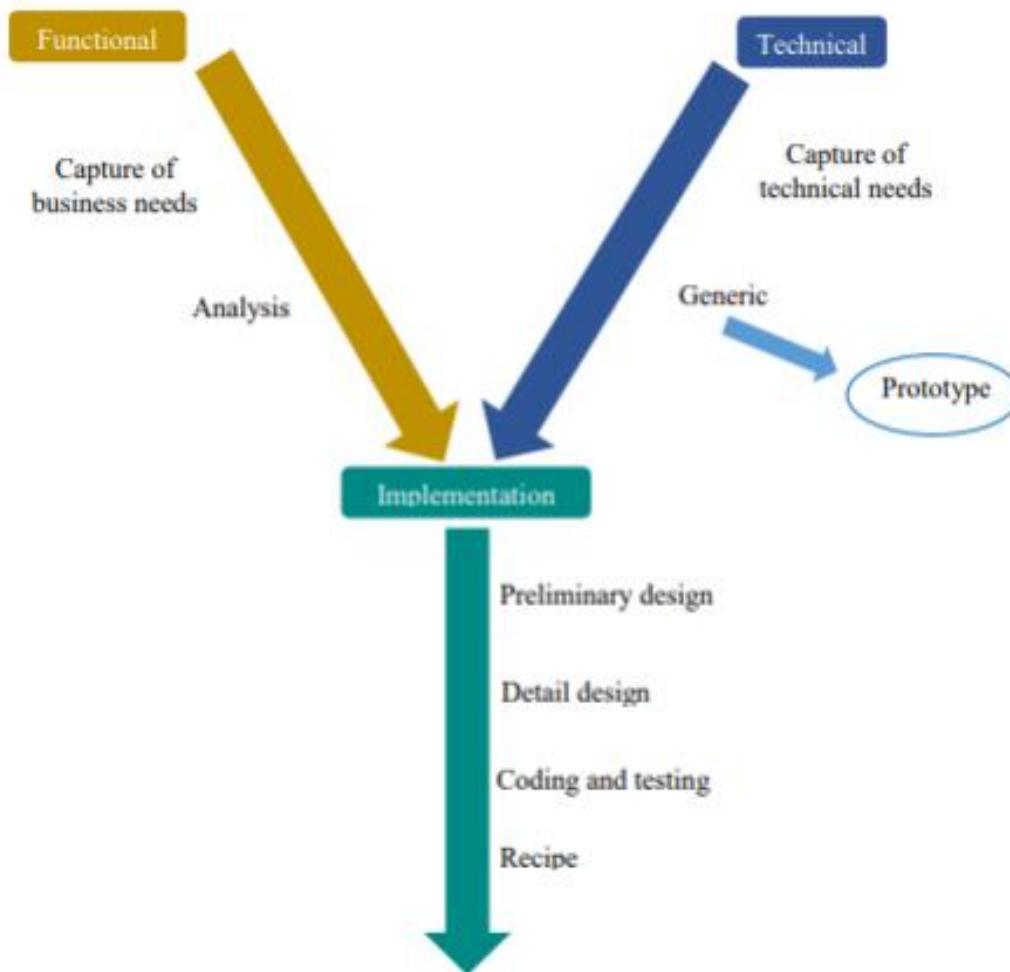


Figure 4: Graphical representation of 2TUP (source: Mr. Messio UML course AICS-Cameroon 2023-2024)

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

iii.) COMPARATIVE STUDY BETWEEN 2TUP AND UP

Table 9: Comparative study between 2TUP and UP

2TUP	UP
2TUP is a software development process that implements the unified process (i.e. iterative, incremental, based on UML).	The Unified Process is a generic name for a family of process models that meet a number of criteria, such as being iterative and incremental, driven by use cases, and focusing on addressing risks early. It defines four project phases: Inception, Elaboration, Construction, and Transition.

iv.) JUSTIFICATION ON THE CHOICE ANALYSIS

Our choice was based on the UML modelling language, combined with the 2TUP analysis method, due to the following criteria:

- ❖ UML is based on object-oriented approach;
- ❖ UML modelling also supports multiple views of the same system.
- ❖ UML enables the creation of specialized diagrams by focusing on specific elements for a particular purpose at a given time.
- ❖ 2TUP is a software development process built on the UML modelling language.
- ❖ 2TUP is user-oriented because built on their expectations (i.e. permits the development of software that responds to users' needs).



II. MODELLING

a.) USE CASE DIAGRAM

i.) Definition

A Use case diagram captures a system's high-level functions (requirements), dynamic nature and scope.

Use Case diagram serve for the following purposes;

- ❖ Specify the context or scope of a system.
- ❖ Capture the requirements of a system.

ii.) Formalism

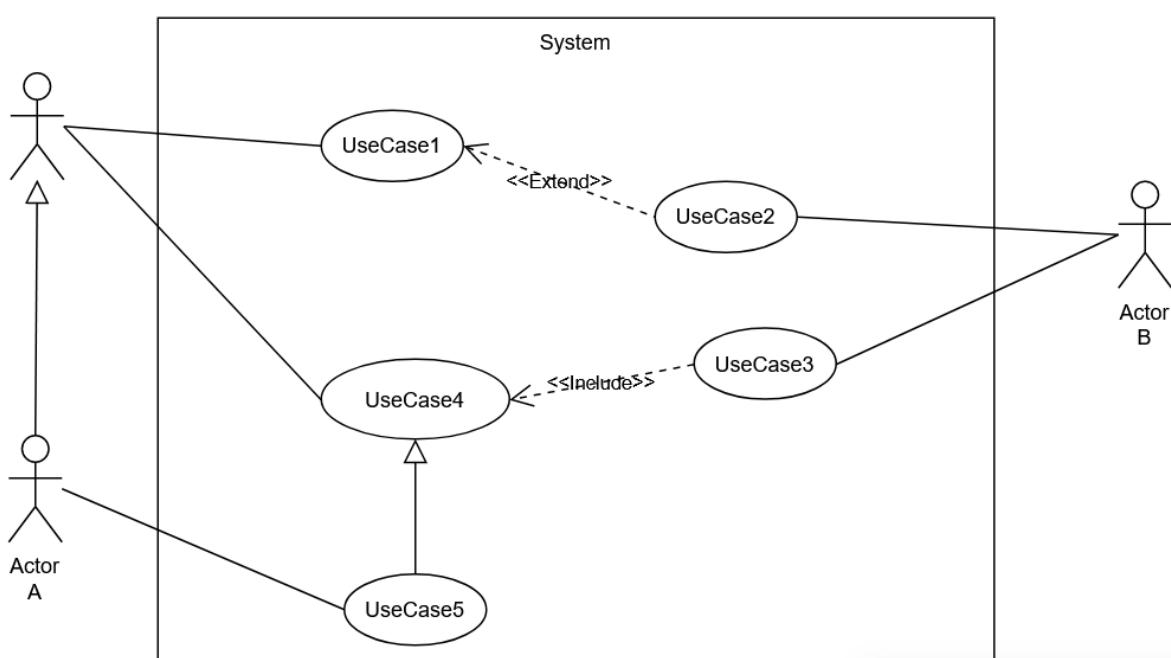


Figure 5: Use Case Diagram Formalism

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



iii.) Components of a Use Case Diagram

Table 10: Components of a Use Case Diagram

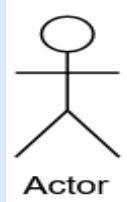
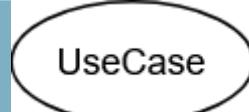
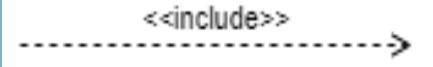
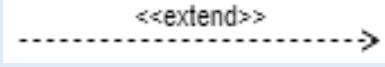
ELEMENT	DESCRIPTION	NOTATION
Actor	Any Entity external to a system interacts with the system, maybe a human user or another system, and has goals and responsibilities to satisfy in interacting with the system.	
Use Case	A use case corresponds to a system's objective motivated by the actors' needs.	
Association	It expresses the interaction between an actor and a use case.	
Include	An Include Association from one use case (called the Base use case) to another use case (called the inclusion use case) indicates that the base use case will include or call the inclusion use case.	
Extend	An Extend Association indicates that the extension use case will extend into) and augment the Base use case.	



Table 11: Components of Use Case Diagram (Continued)

ELEMENT	DESCRIPTION	NOTATION
Generalisation (Actor)	An Actor generalization from a specialized, actor to a generalized, actor indicates that instances of the more specific actor may be substituted for instances of the more general actor.	
Generalisation (Use Case)	A use case generalization indicates that a specific use case inherits the behavior and characteristics of a more general use case.	
System	It identifies what is part of the system and the actors interacting with it.	



iv.) General Use Cas Diagram

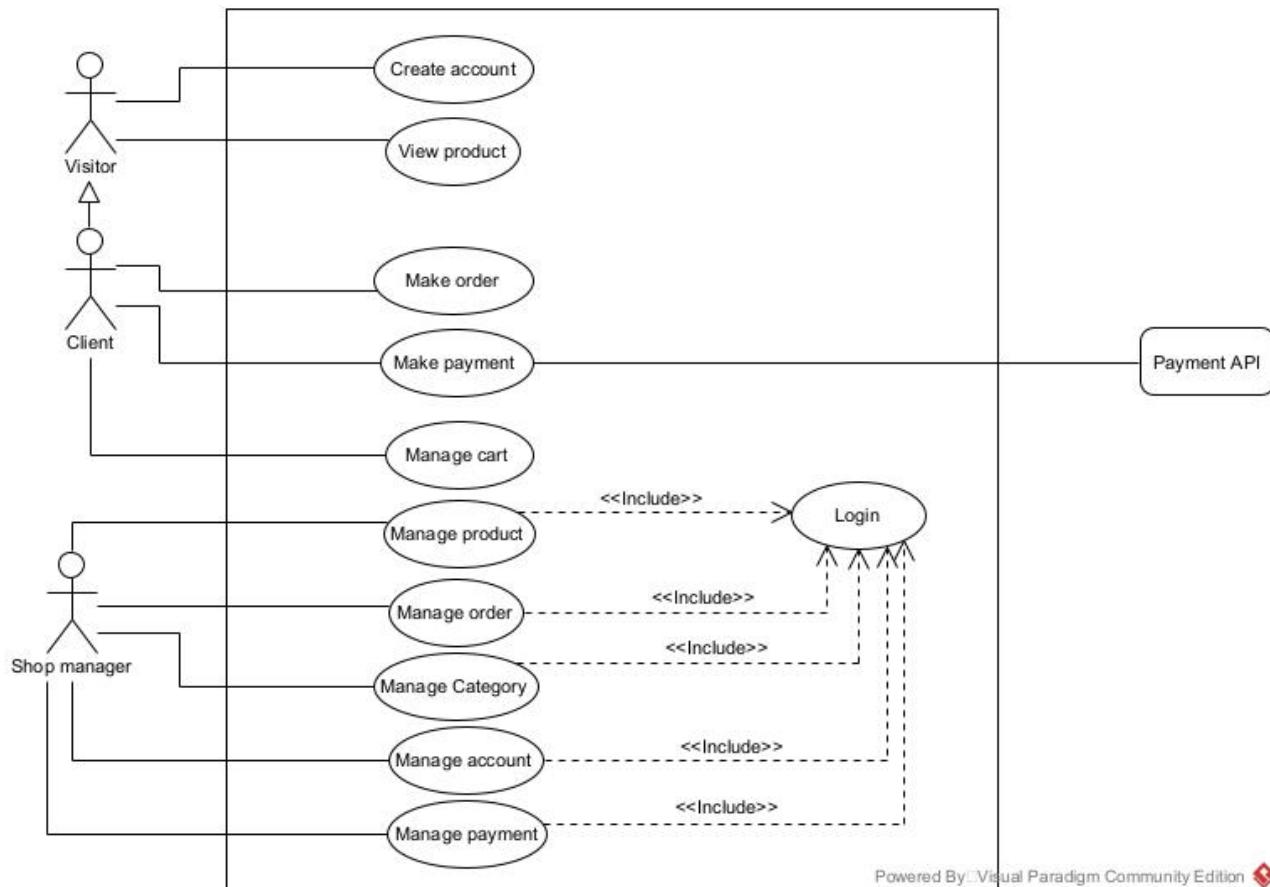


Figure 6: General Use Case Diagram

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



v.) Client Manage Cart Specific Use Case Diagram

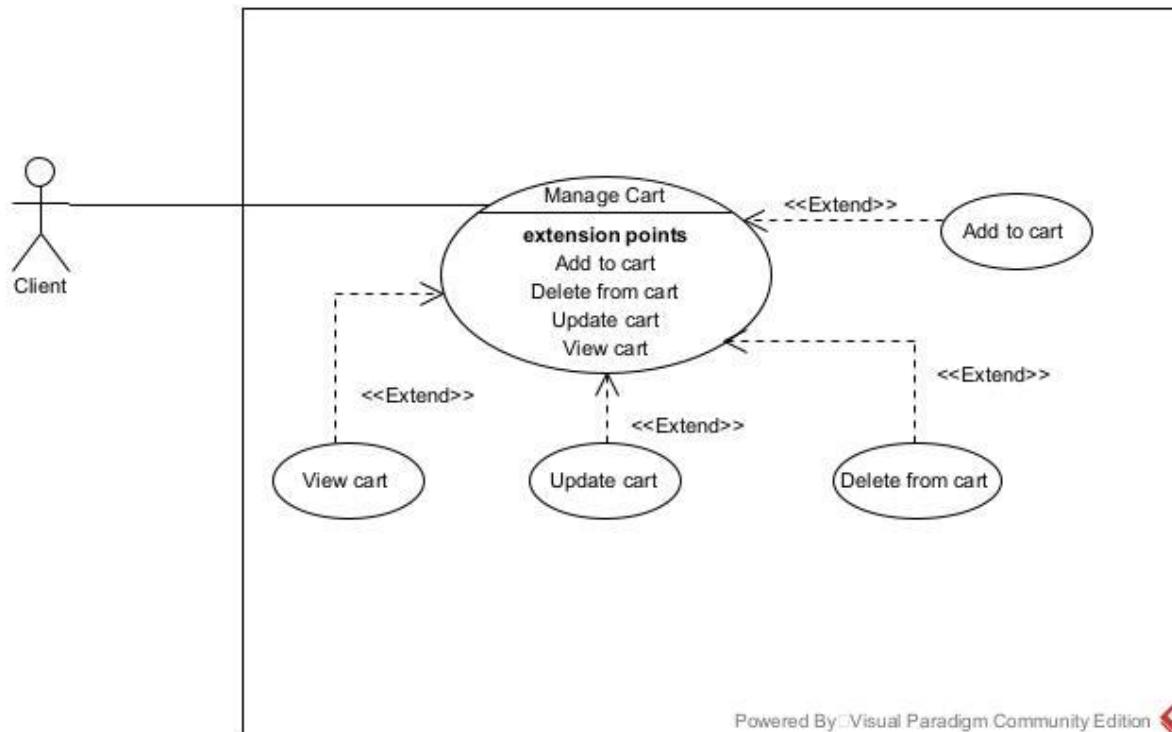


Figure 7: Client Manage Cart Specific Use Case diagram

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



vi.) Shop manager Manage Order Specific Use Case Diagram

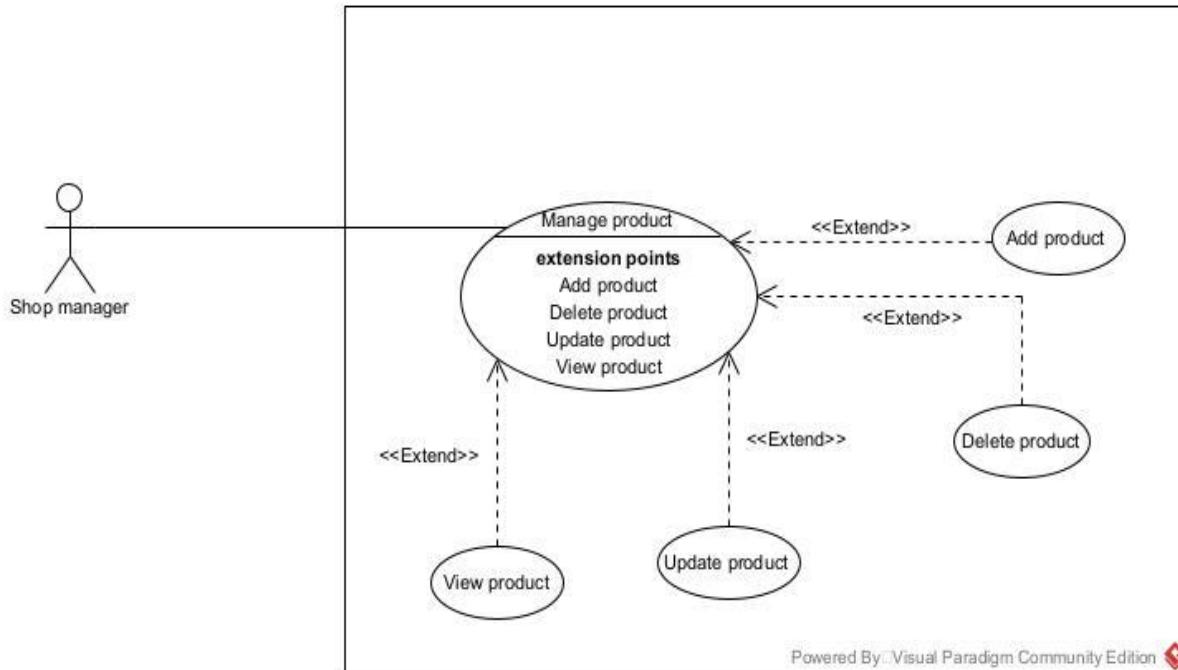


Figure 8: Supplier Overview Use Case Diagram

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



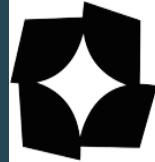
TEXTUAL DESCRIPTION OF USE CASES

Use Case can be considered as a behaviour that a system offers to the actors to help meet the actors' goals. So, we have to supply information on how the use case works — and put that information somewhere in a Textual description (Specification). This set of needed details placed inside a use case is sometimes called the **Use Case Textual Description**.

Formalism

Table 12: Formalism of Textual Description of a Use Case

Title	Represents the name or title of the textual description
Description	It explains the goals, plot, and theme of the use case.
Actors	Description of the main and the secondary actors.
Date	Creation and update date.
Stakeholder	Names of creator(s).
Version	The version number.
Precondition	Specifies the state of the world that must be held before the course can be triggered.
Trigger	Any event that causes the use case to be initiated.
Nominal Scenario (Corresponds to the normal development of a use case)	
Alternative Scenario (Corresponds to variants of the nominal scenario)	
Post Condition of Success	Describes the state of the system after the end of each scenario.
Post Condition of Failure	Describes the state of the system after the end of each Scenario.



❖ Textual Description of the Use Case <<Authenticate>>

Table 13: <<Authenticate>> Textual Description

Title	Textual description of a Use case << Authenticate >>
Description	This use case allows any service provider (Supplier, Delivery Person) with an existing account to log in to the system using their credentials.
Actors	Service Provider (Supplier, Delivery person)
Date	March 23, 2025
Stakeholder	LocaVend Directors
Version	1.0
Precondition	The actor has an account in the system
Trigger	The actor clicks on the Login button
Nominal Scenario	
1. The use case starts when the actor clicks on the "Login" button.	
2. The system displays the login page.	
3. The system prompts for the actor's credentials (Email and Password).	
4. The actor fills in the fields with their credentials.	
5. The system queries the actor's (email-related) data from the database.	
6. The database sends the query results.	
7. The system verifies the actor's credentials against the query results.	
8. The system displays the actor's dashboard.	
9. The use case ends when the service provider gains access to the dashboard.	
Alternative Scenario: Verification Failure	
10. At step 7 of the nominal scenario the system fails to verify the actor's specified credentials.	
11. The system displays an error message indicating the issue.	
12. The use case continues at step 2 of the nominal scenario.	

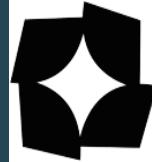


DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



Post Condition of Success	The Service provider gains access to Personal dashboard
Post Condition of Failure	Service provider unable to access dashboard

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ Textual Description of the Use Case <<Make Order>>

Table 14: <<Make Order>> Textual Description

Title	Textual description of the use case <<Make Order>>
Description	This use case allows the actor, Client , to use a web browser to obtain a confirmed order from the system.
Actor(s)	Client
Date	March 21, 2025
Stakeholder	LocaVend Directors.
Version	1.0
Precondition	Client visits welcome web page wanting to make an order.
Trigger	Client clicks on “Search here” button.
Nominal Scenario	
1. The client enters the valid website URL.	
2. The system displays the home Page with a products.	
3. The client browses and selects a stock item	
4. The client reviews the stock item and clicks a button (Add to Cart or Cancel).	
5. The system verifies the type of button clicked.	
6. If "Add to Cart" was clicked, the system redisplays the merchant's business page with a "Checkout" button.	
7. The client clicks the "Checkout" button.	
8. The system displays the checkout page with the "Place Order" button.	
9. The client confirms order information and clicks the "Place Order" button.	
10. The system sends a "Create Order" query to the database.	
11. The database executes the query and returns the results.	
12. The system verifies and processes the results.	
13. The system displays a success message to the client.	
Alternative Scenarios:	
14. At step 9 of the nominal scenario, the client clicks the "Cancel" button.	

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



15. The system redisplays the merchant's (supplier's) business page and continues from step 7 of the nominal scenario.
16. At step 9 of the nominal scenario, the database returns an error after processing or treating the query.
17. The system displays an error message and returns to step 13 of the nominal scenario for the client to attempt to place the order again.

Post Condition of Success	The client successfully places an order.
Post Condition of Failure	The client fails to place an order.

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



b.) COMMUNICATION DIAGRAM

i.) Definition

A Communication diagram focuses on the messages between a group of objects and the underlying messages of the objects. They show how objects collaborate to meet a goal. In other words, they are time- and space-oriented and emphasize the overall interaction, the elements involved, and their relationships.

ii.) Formalism

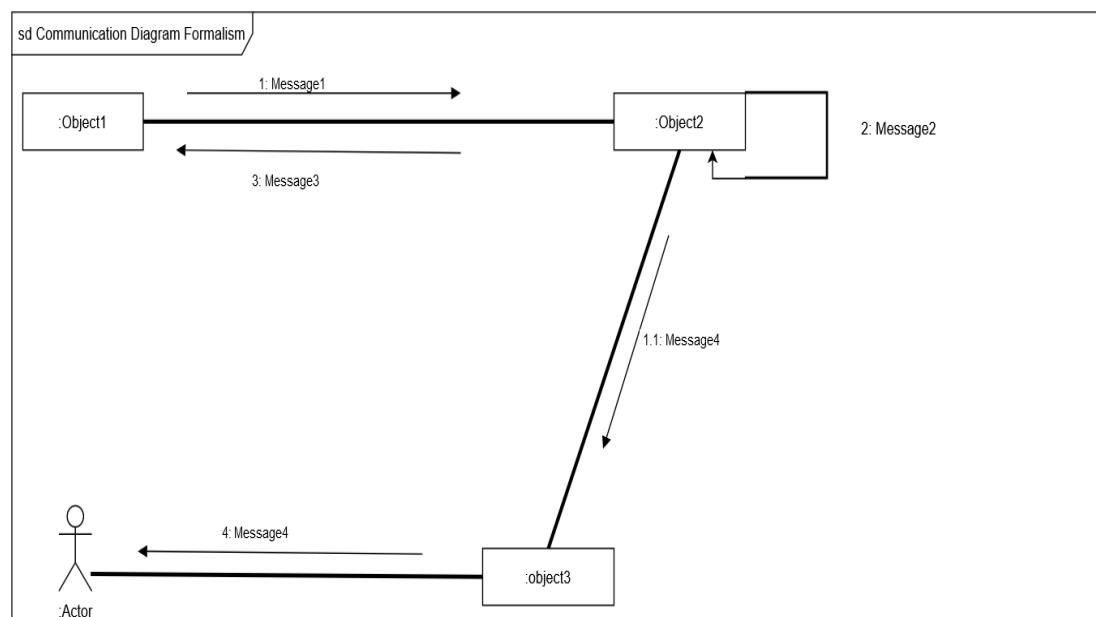
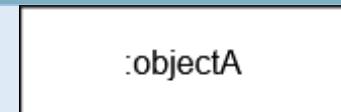
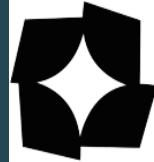


Figure 9: Communication Diagram Formalism

iii.) Components of a Communication Diagram

Table 15: Components of Communication Diagram

ELEMENT	DESCRIPTION	NOTATION
Message	It defines the communication between two objects at a given instant.	!messageA()[condition]
Link	a link is a tool in a UML diagram to indicate that two objects communicate with each other.	
Object	Objects can be any useful item that has identity, structure, and behaviour.	
Actor	A role played by an entity.	



❖ <<Authenticate (Login)>> Communication diagram

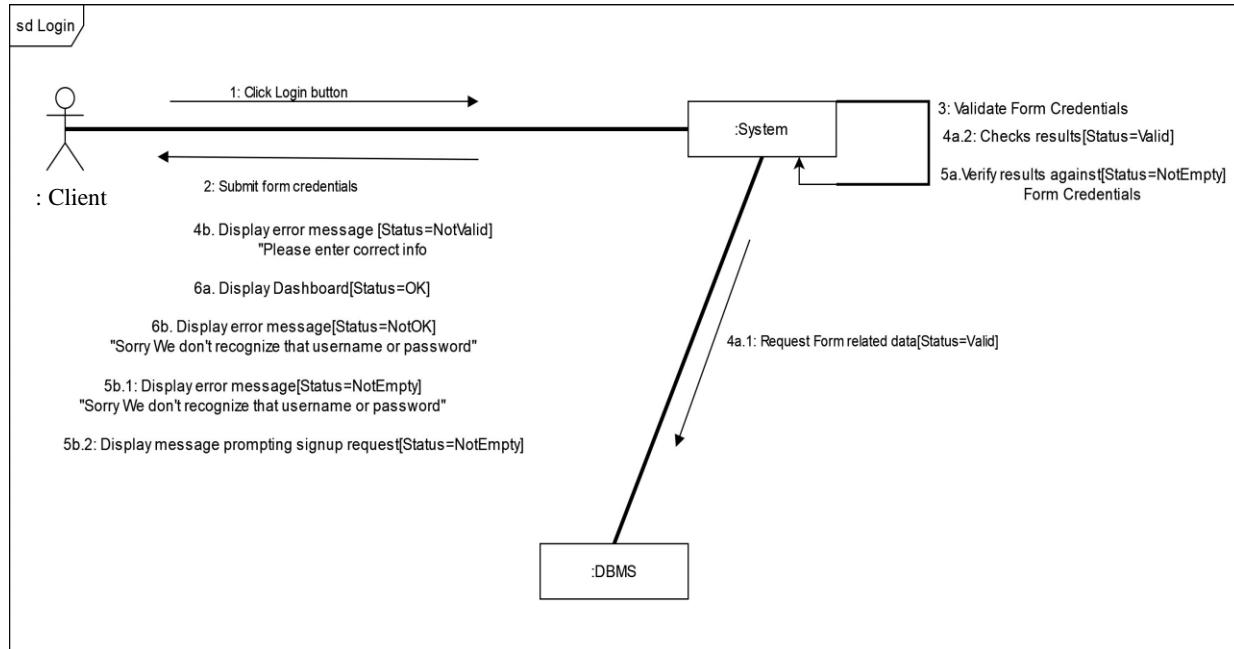


Figure 10: Communication Diagram for <<Authentication (Login)>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Make Order>> Communication diagram

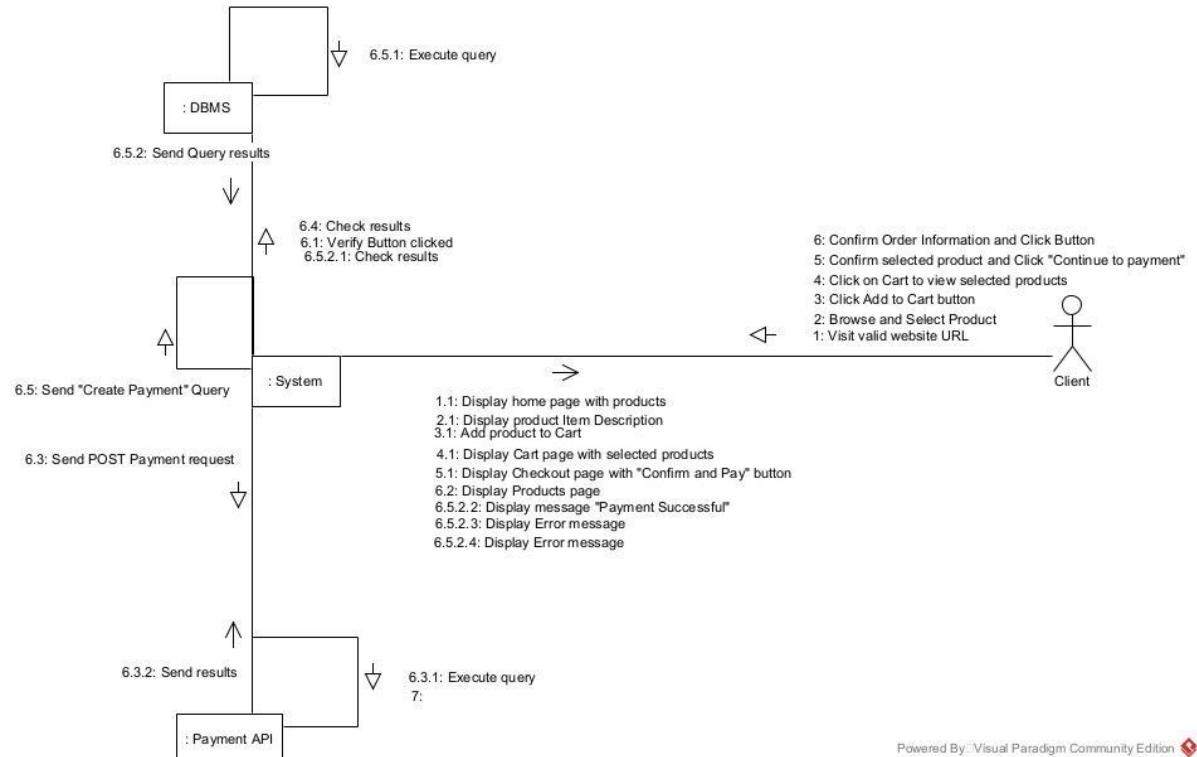
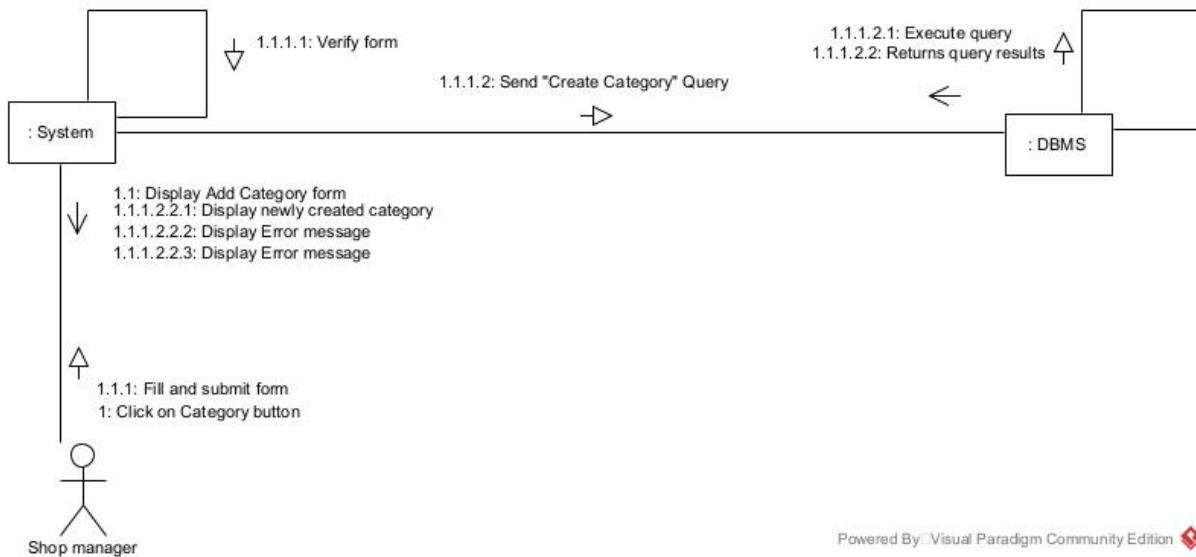


Figure 11: Communication diagram for <<Make Order>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Create Category>> Communication diagram



Powered By: Visual Paradigm Community Edition

Figure 12: Communication for <<Create Category>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

c.) SEQUENCE DIAGRAM

i.) Definition

A sequence diagram is an Interaction diagram which captures the exchange of messages between participating objects. They are time-oriented and emphasize the overall flow of an interaction.

ii.) Formalism

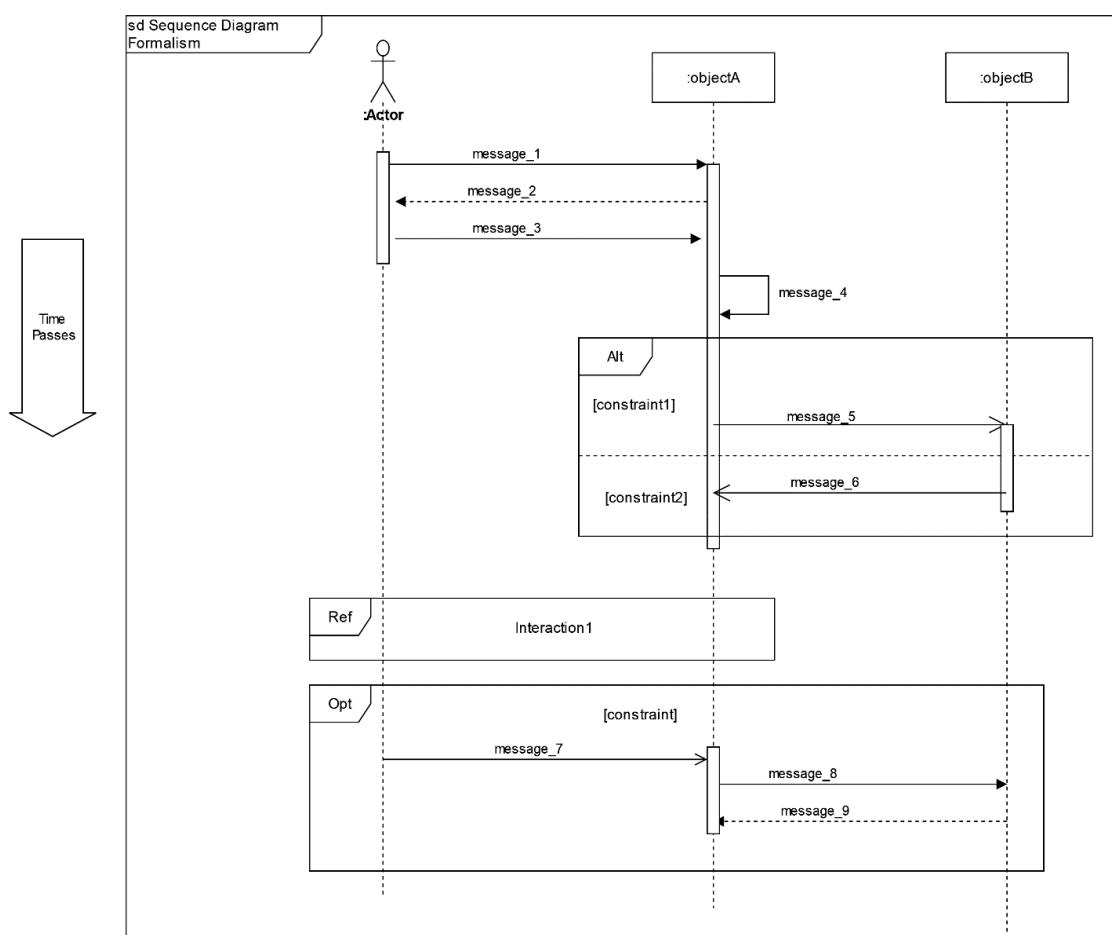


Figure 13: Sequence Diagram Formalism

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



iii.) Components of a Sequence Diagram

Table 16: Components of a Sequence Diagram

ELEMENT	DESCRIPTION	NOTATION
Lifelines	It is a vertical dashed line from an element which represents the existence of the element over time.	
Object	It's an instant of a class.	
Actor	Communicate with other objects.	
Activation	It represents the period during which an element is performing an operation.	
Message	It indicates the communication between objects. We have Synchronous messages, reply messages, Asynchronous messages and Self messages as types of messages.	<p>The diagram illustrates different types of messages in a sequence diagram:</p> <ul style="list-style-type: none"> Synchronous message: Represented by a solid arrow labeled "message_1". Reply/return message: Represented by a dashed arrow labeled "message_2". Asynchronous message: Represented by a solid arrow labeled "message_7". Self Message: Represented by a message loop on a lifeline labeled "Self Message". message_4: A message shown as a solid arrow pointing to a lifeline.

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Authentication (Login)>> Sequence Diagram

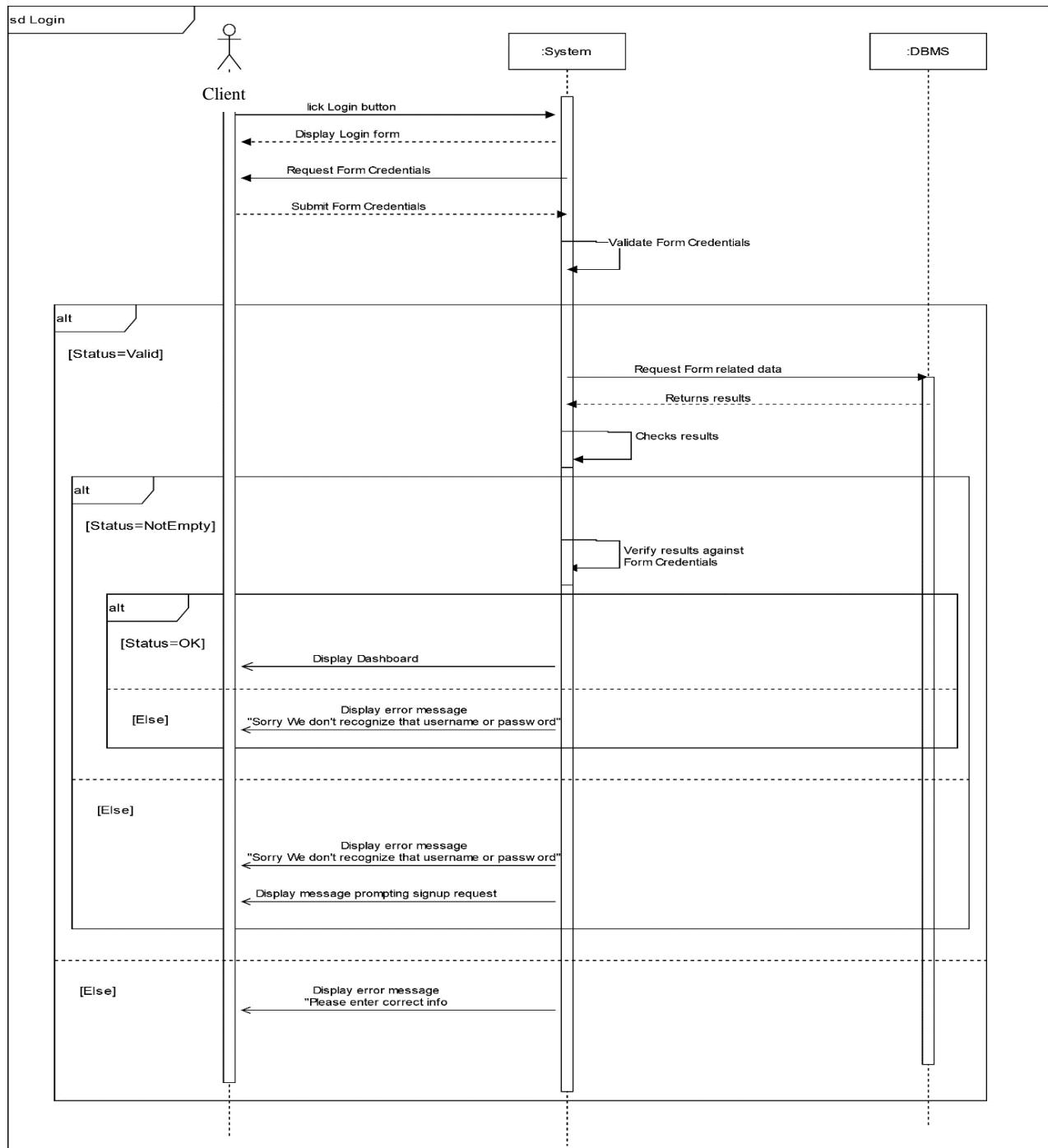


Figure 14: Sequence diagram for <<Authentication (Login)>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Make Order>> Sequence Diagram

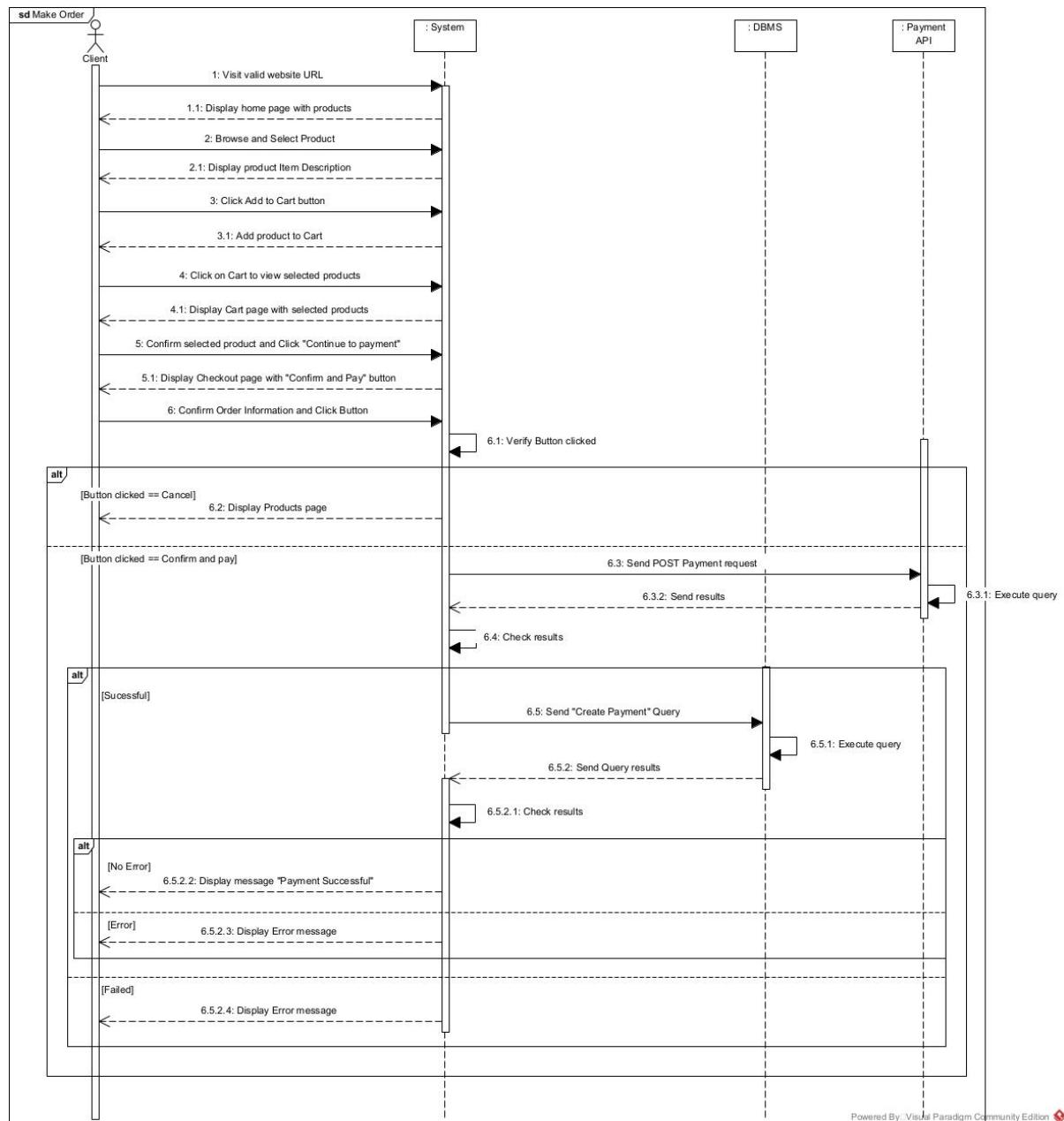


Figure 15: Sequence diagram for <<Make Order>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Create Category>> Sequence Diagram

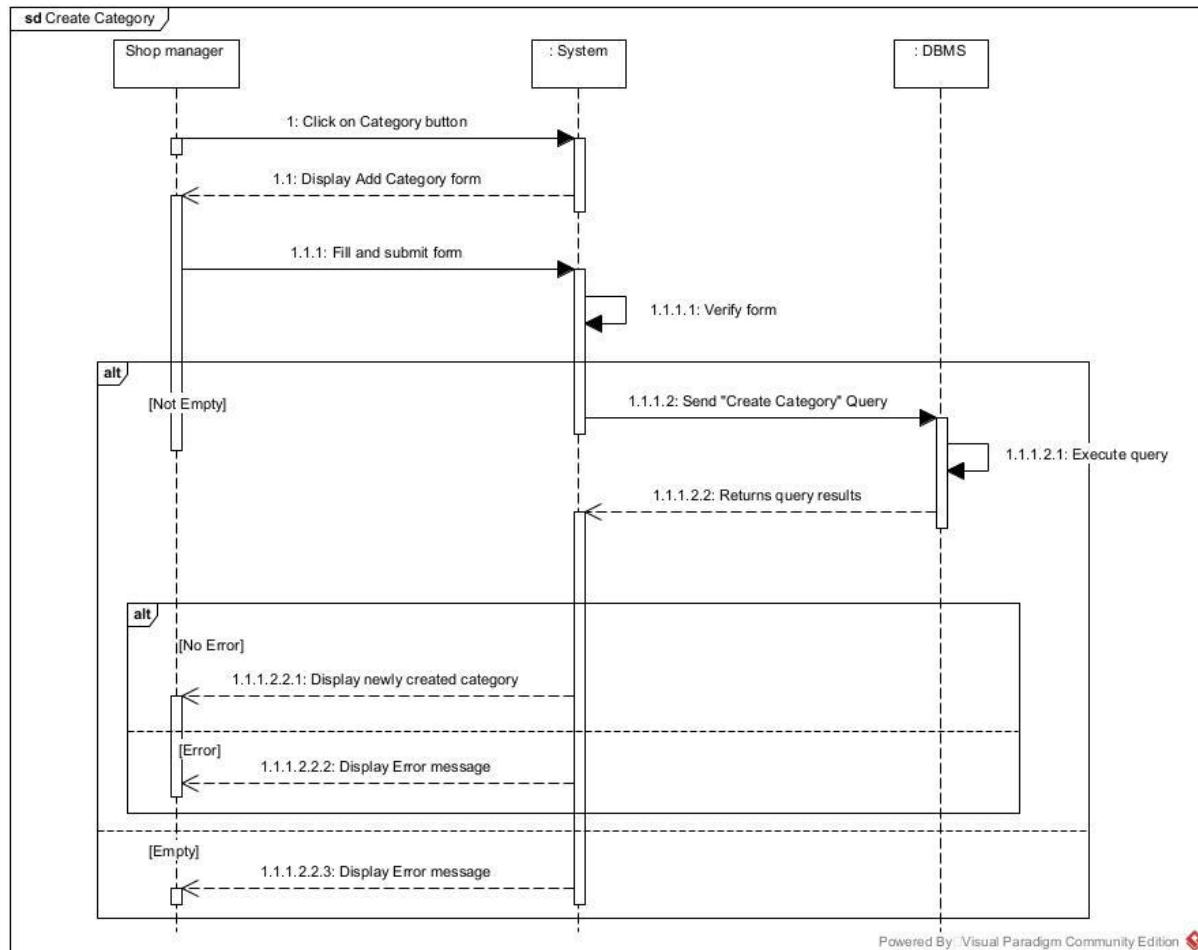


Figure 16: Sequence diagram for <<Create Category>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

d.) ACTIVITY DIAGRAM

i.) Definition

An activity diagram is a behavioural diagram that describes the sequence of actions in a process or specific activity, providing a view of the system's behaviour.

ii.) Formalism

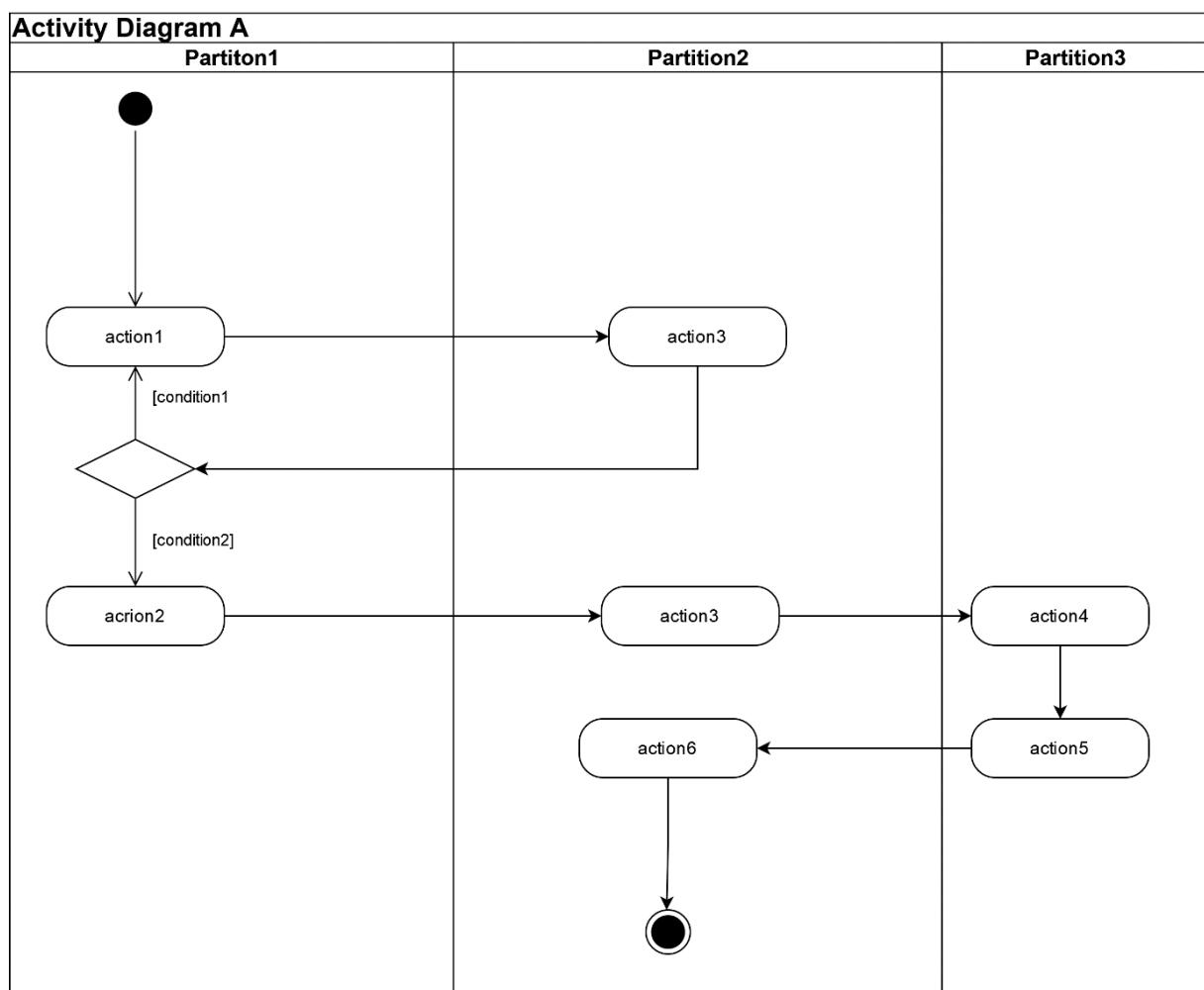
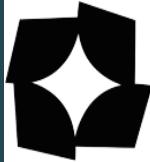


Figure 17: Activity diagram formalism

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR

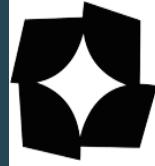


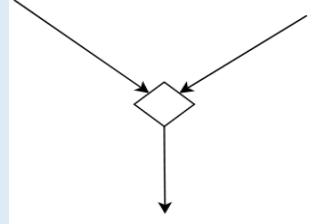
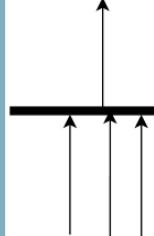
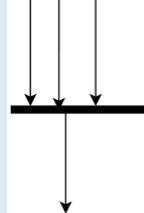
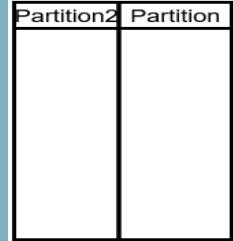
iii.) Components of an Activity Diagram

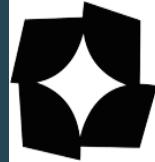
Table 17: Components of an Activity Diagram

ELEMENTS	DESCRIPTION	NOTATION
Activity	Activities contain sequences of actions and/or other activities. You use activities to group sequences of actions together.	
Action	An action is a simple piece of behaviour.	
Object node	Represents an activity node that indicates an instance of a particular classifier in the activity.	
Control flow	Connects actions and activities together; shows the sequence of execution.	
Initial node	This shows the starting point or first activity of the flow.	
Final activity node	Ends all control flows and object flows in an activity, using the final-activity node.	
Final flow node	End some — but not all — flows inside an activity.	
Decision node	A decision node uses a test to make sure that an object or control flow goes down only one path.	

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Merge	Brings separate decision paths back together.	
Fork node	Used to split behaviour into concurrent operations.	
Join node	Used to synchronize incoming concurrent flows.	
Swimlane	Swimlane is a visual region in an activity diagram that indicates the element that has responsibility for action states within the region.	



❖ <<Authentication>> Activity Diagram

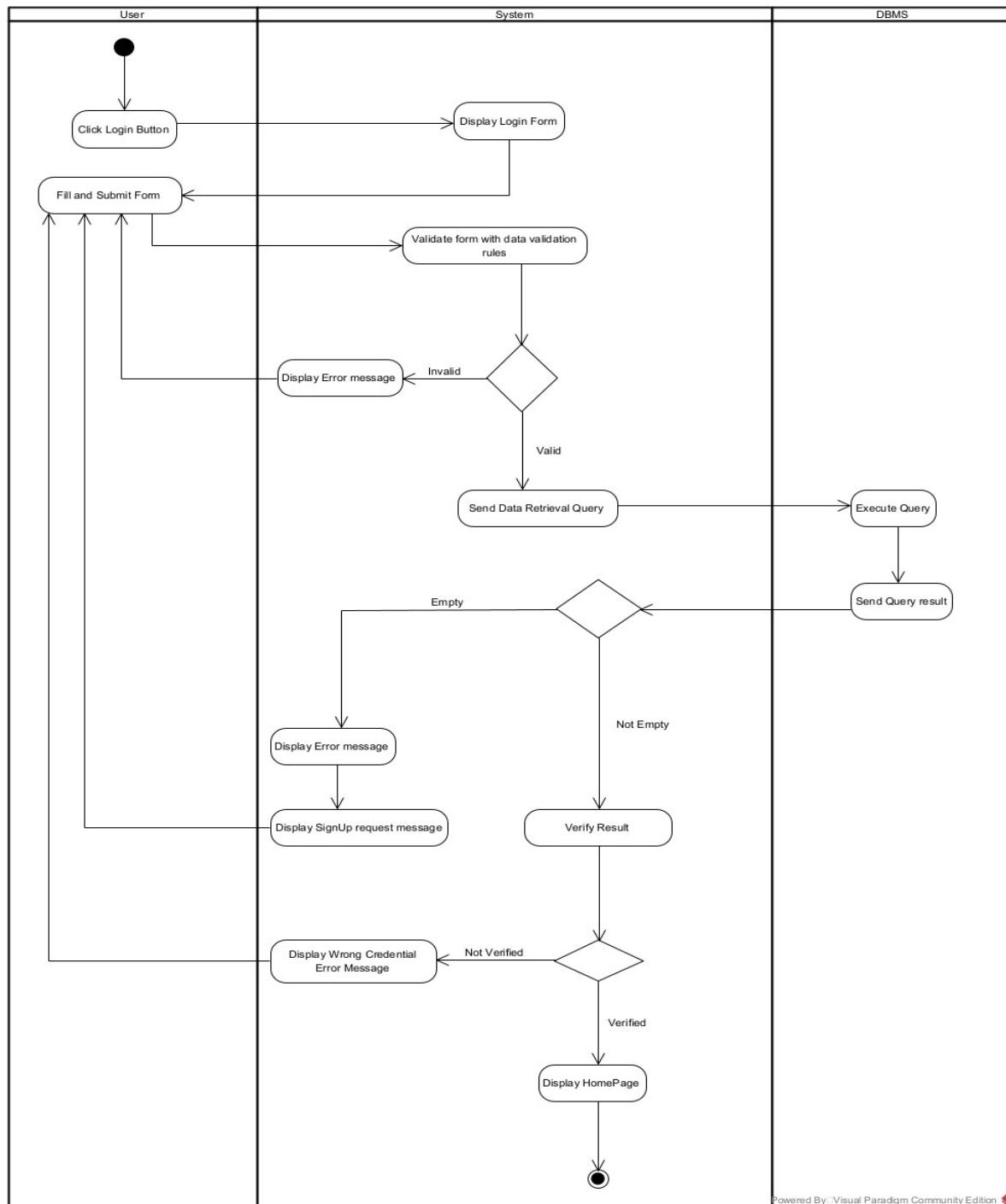
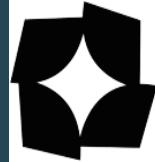


Figure 18: Activity diagram for <<Authentication>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Make Order>> Activity Diagram

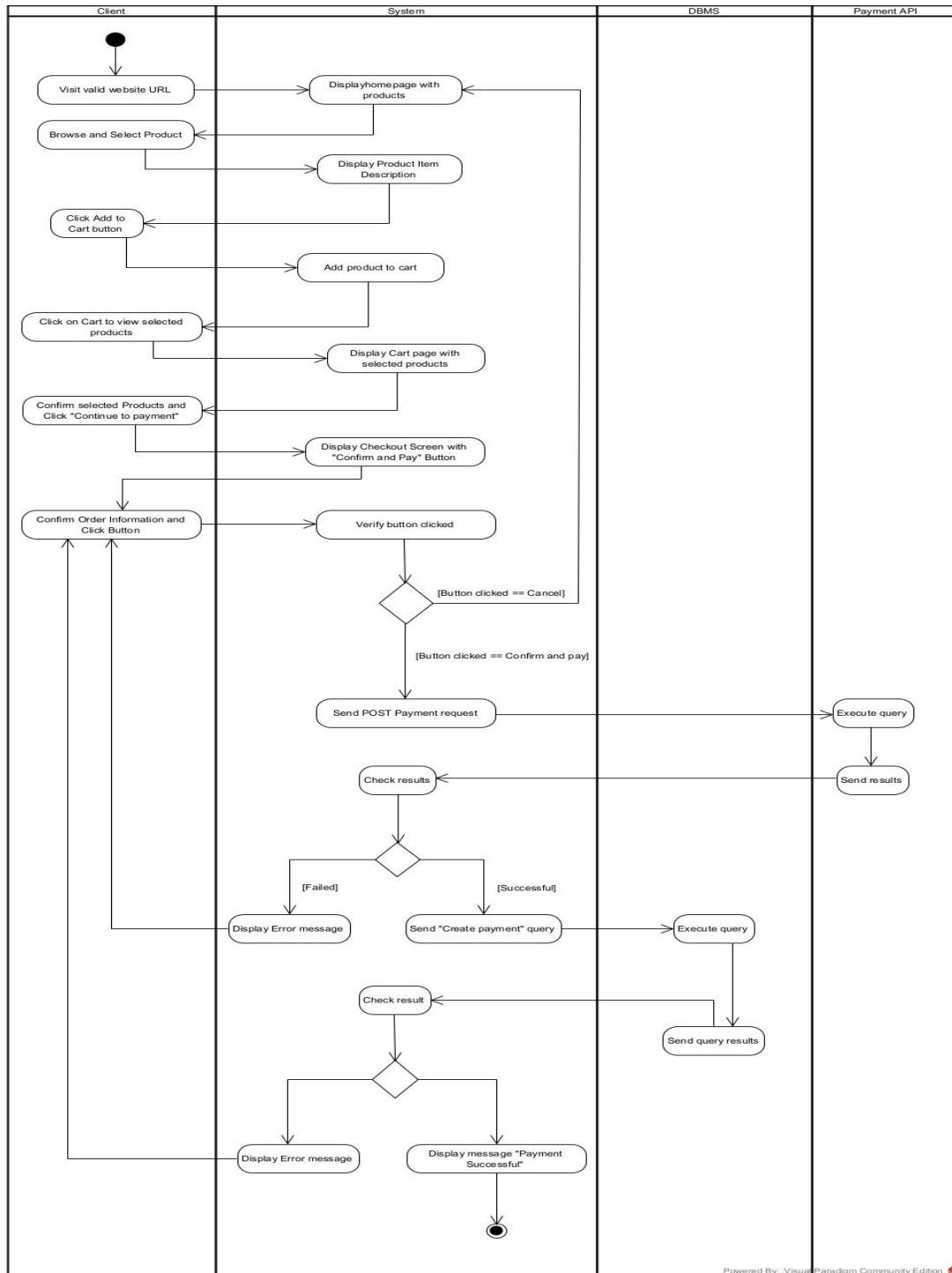


Figure 19: Activity diagram for <<Make Order>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Create Category>> Activity Diagram

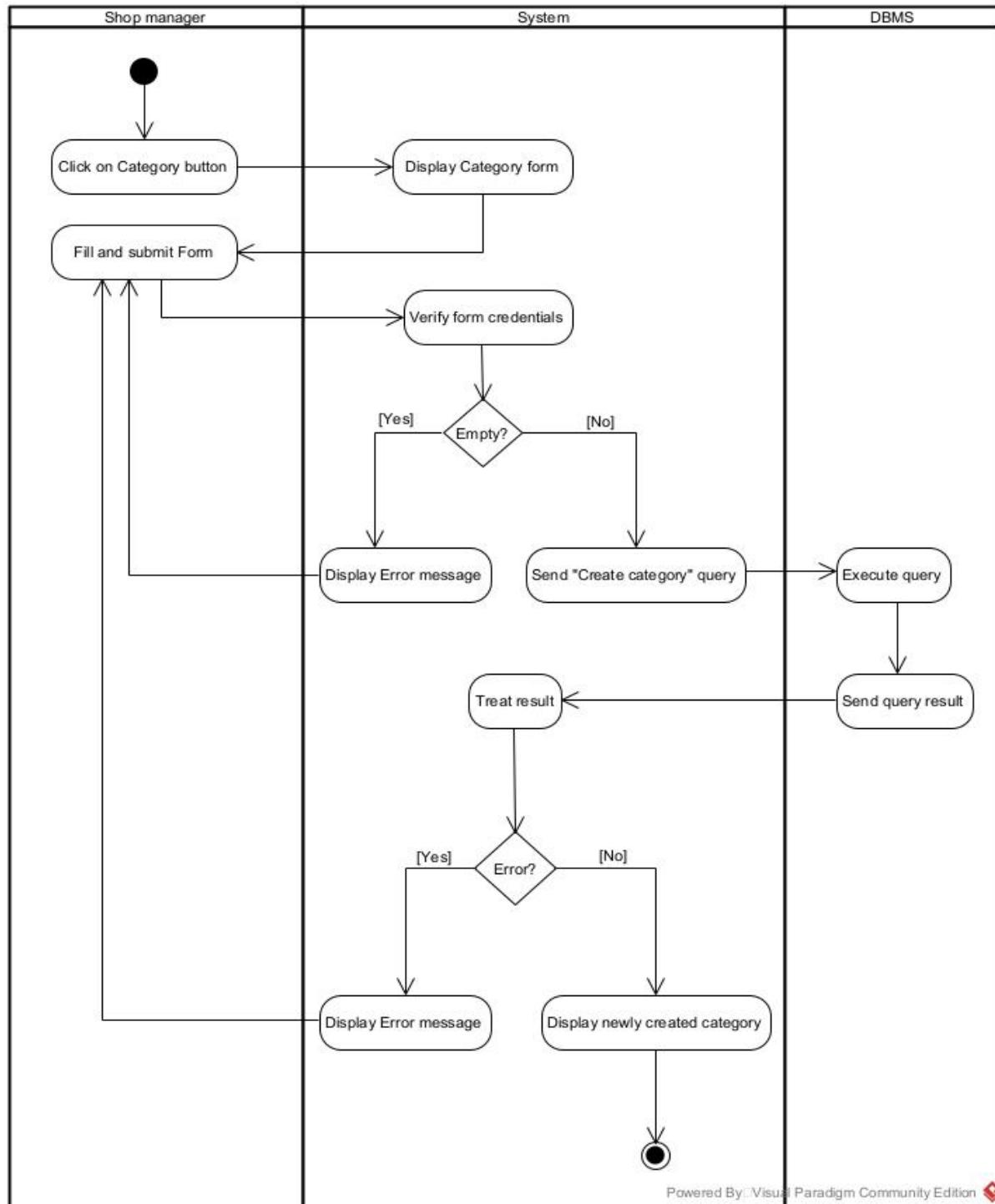


Figure 20: Activity diagram for <<Create Category>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



❖ <<Add New Product>> Activity Diagram

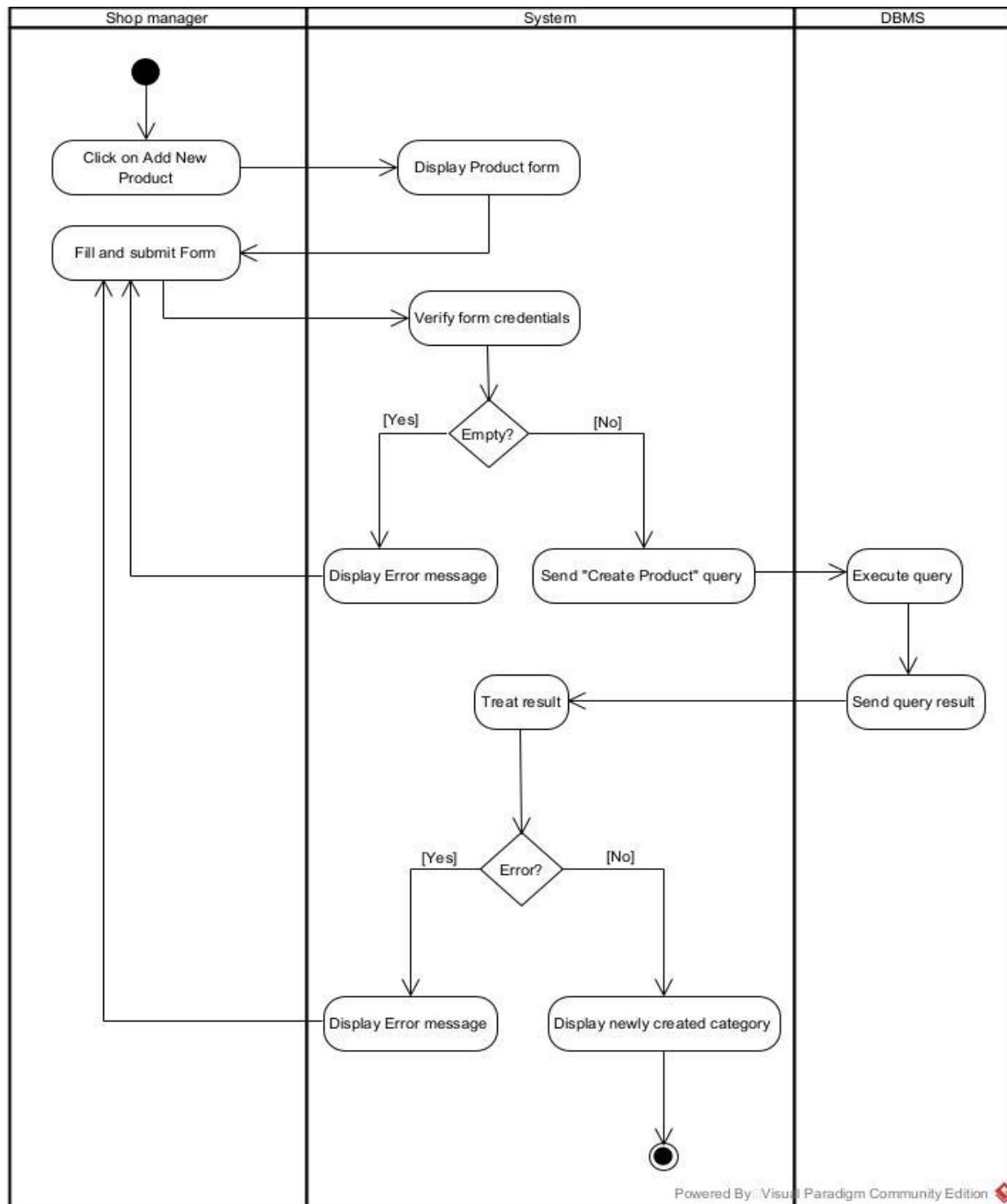


Figure 21: Activity diagram for <<Add Product>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



CONCLUSION

Having completed our analysis, we were tasked with illustrating the language and method used during this phase and presenting the various diagrams of the functional branch. This analysis has allowed us to define the functional needs of the web application and provided a detailed overview of the new system to be implemented. Without wasting time, we quickly dive into the conception phase.



CHAPTER 4: CONCEPTION PHASE

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



Preamble

This document captures the technical requirements and establishes the system's architecture, bridging the gap between the analysis and realization phases.

Content:

INTRODUCTION

- I. CLASS DIAGRAM
- II. STATE MACHINE DIAGRAM
- III. PACKAGE DIAGRAM

CONCLUSION

CONCLUSION

БКОВОЗД 2024-2025

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



INTRODUCTION

The capture of the technical needs can be carried out as soon as the required resources are identified. These resources include machines, networks, components, etc. At this stage, we will examine diagrams such as class diagrams, package diagrams, state machine diagrams, and more.



A. CAPTURE OF TECHNICAL NEEDS

i.) PHYSICAL ARCHITECTURE

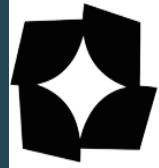
Modern web application architecture typically consists of several layers that work together to provide a robust, scalable, and maintainable system. Each layer serves a specific purpose, and together they form a cohesive and integrated solution that can handle a variety of tasks.

Choosing the right architecture is a complex decision. While scalability, maintainability, and Data Integrity are essential considerations, other factors such as budget, time-to-market, and technology stack also come into play reason why we choose Three-tier architecture. **Three-tier architecture** is a well-established software application architecture that organizes applications into three logical and physical computing tiers. The three-tier architecture is the most popular implementation of a multi-tier architecture and consists;

- ❖ **Presentation tier:** The user interface where end-users interact with the application.
- ❖ **Application tier:** The logic layer that processes data from the presentation tier using business rules.
- ❖ **Data tier:** The database management system (DBMS).



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



Below is a visual representation of the Three-Tier Architecture.

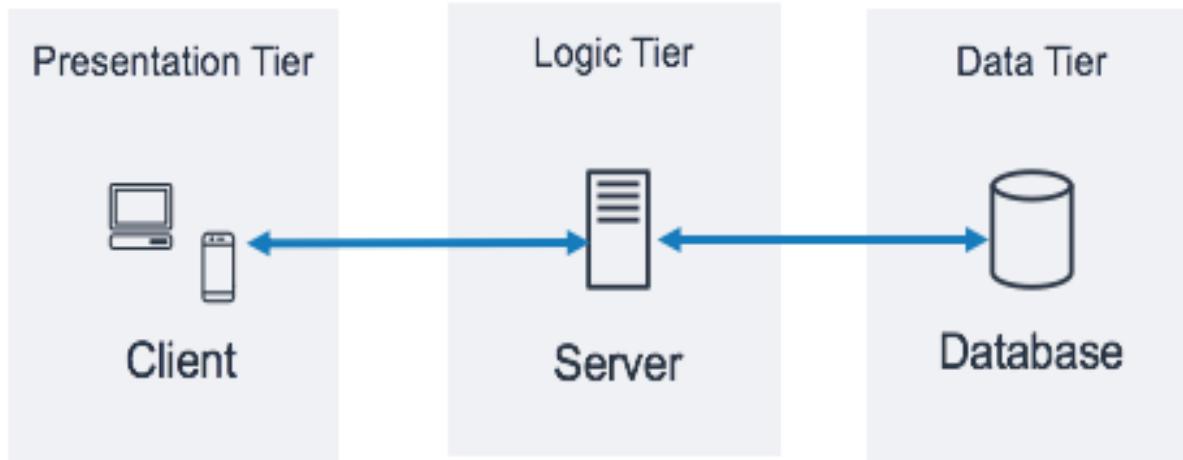


Figure 22: Visual representation of 3-Tier Architecture (Source: <https://docs.aws.amazon.com/images/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/images/image2.png>)



ii.) LOGICAL ARCHITECTURE

Model View controller or MVC as it is popularly called, is a software design pattern for developing application. A model view controller pattern is made up of the following three parts.

- ❖ **Model:** The lowest level of the pattern which is responsible for maintaining data.
- ❖ **View:** This is responsible for displaying all or a portion of data to the user.
- ❖ **Controller:** It handles software codes that controls the interactions between the model and the view.

The MVC can be represented as follows.

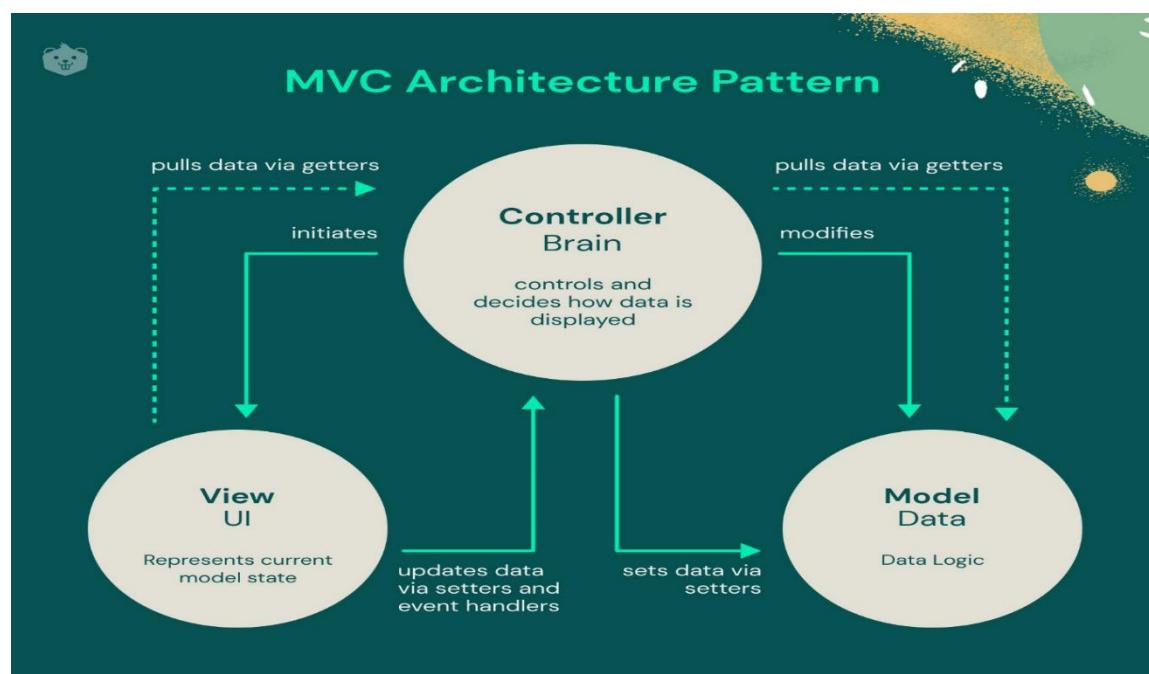


Figure 23: MVC Architecture (Source: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRx0n1oDz1SskH1ZMK15W0fAZ-KQ33yRlHruA&s>)

I.) CLASS DIAGRAM

i.) Definition

The class diagram expresses the static structure of the system in terms of classes and the relationships between those classes. The interest of the class diagram is to model the entities of the information system.

ii.) Formalism

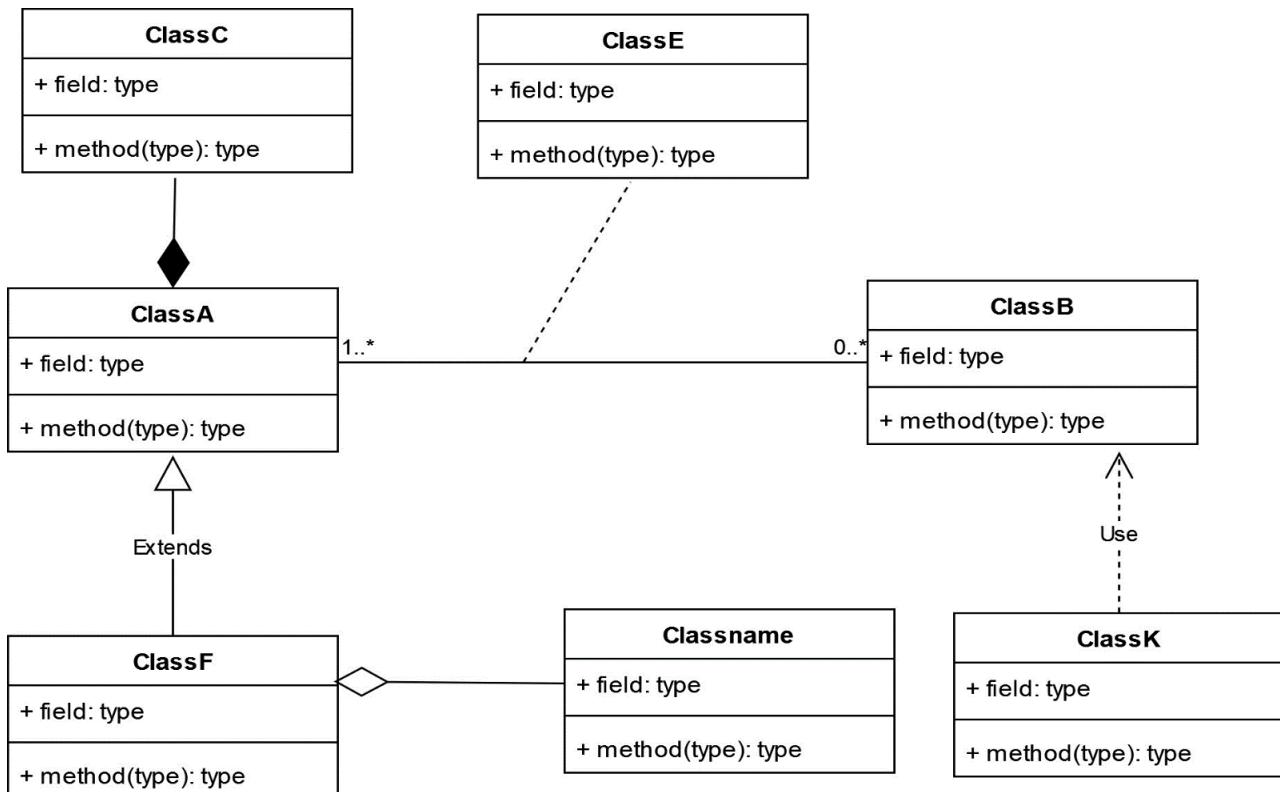


Figure 24: Class Diagram Formalism

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



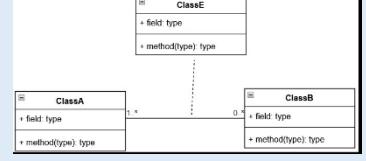
iii.) Components of a Class Diagram

Table 18: Components of a Class Diagram

Element	Description	Representation
class	It defines the structure, the behaviour and the relationship of these objects.	<pre> classDiagram class ClassA { +field: type +method(type): type } </pre>
Composition	If a parent of a composite is deleted, usually, all of its parts are deleted with it.	<pre> classDiagram class subclass1 class subclass2 subclass1 "3..>" subclass2 </pre>
Aggregation	It models the notion that one object uses another object without "owning" it and thus is not responsible for its creation or destruction.	<pre> classDiagram class ClassF { +field: type +method(type): type } class Classname { +field: type +method(type): type } ClassF "3..>" Classname </pre>
Dependency	It existed between two classes, if one changes it may cause the change in the order, but the other way around.	<pre> classDiagram class ClassB { +field: type +method(type): type } class ClassK { +field: type +method(type): type } ClassB "3..>" ClassK ClassK "3..>" ClassB </pre>
Generalization	It is a relationship between a whole thing (called superclass) and a more specific thing (called subclass)	<pre> classDiagram class ClassA { +field: type +method(type): type } class ClassF { +field: type +method(type): type } ClassA "3..>" ClassF ClassF "Extends" ClassA </pre>
Association	It is a relationship between 2 or more classes. Types of association include Binary, Nary associations.	<pre> classDiagram class ClassB { "1..>" ClassA } class ClassC { "0..>" ClassG } class ClassF { "3..>" ClassG } ClassC "3..>" ClassF </pre>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Association class	It is a class formed between 2 or more countries.	 <pre> classDiagram class ClassE { +field type +method(type) type } class ClassA { +field type +method(type) type } class ClassB { +field type +method(type) type } ClassE "1..*" -- "0..*" ClassA ClassE "1..*" -- "0..*" ClassB </pre>
-------------------	---	---

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



iv.) System Class Diagram

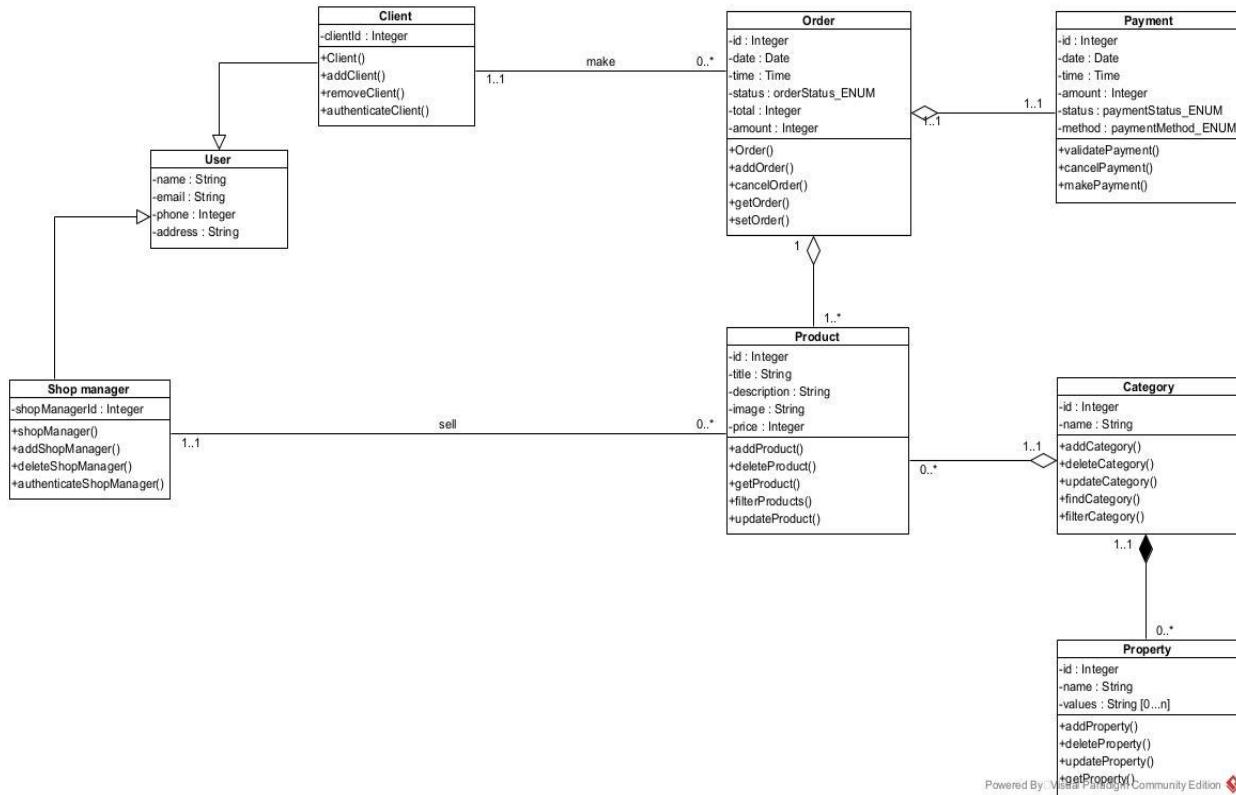


Figure 25:General System Class Diagram

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

II. STATE MACHINE DIAGRAM

i.) Definition

State diagrams are used to document the various states that a class can take, and the events that cause a state transition. They are typically applied to objects, showing how they respond to events by transitioning from one state to another.

ii.) Formalism

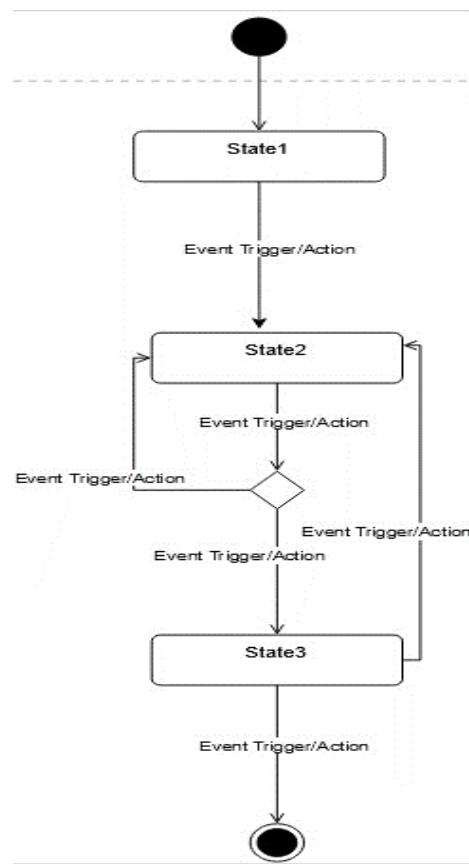


Figure 26: Formalism of a State machine diagram

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



iii.) Components of a State Machine Diagram

Table 19: Components of a State Diagram

Element	Description	Representation
State	Models a situation during which a certain invariant condition holds.	
First (Initial) State	It defines the initial state	
Final State	It represents the final state or the end of a system.	
Transition	It is a change of control from one state to another due to the occurrence of some events.	
Choice pseudo State	A diamond symbol that indicates a dynamic condition with branched potential results	
Terminate	Implies that the execution of a state by means of its context is terminated.	
Diagram Overview	A placeholder for the linked states in a state machine diagram.	

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

❖ <<Order>> State Machine Diagram

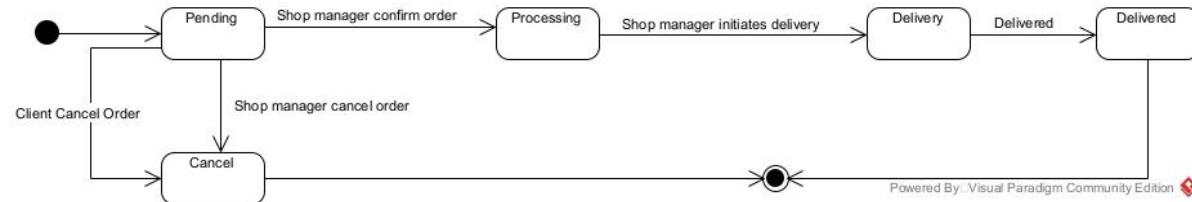


Figure 27: State machine diagram for <<Order>>

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

III. PACKAGE DIAGRAM

i.) Definition

Package diagram is a structural diagram used to show the organisation and arrangement of various model elements in the form of packages. They can show both structure and dependencies between sub-systems or modules, showing different views of a system.

ii.) Formalism

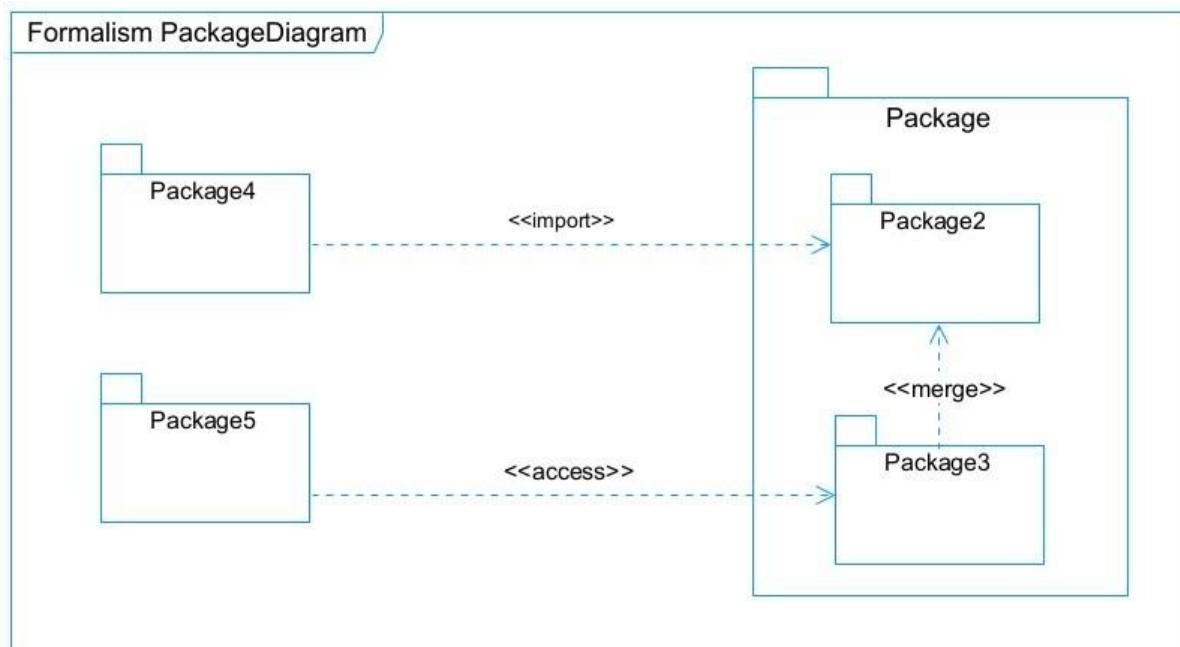
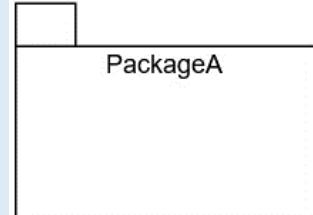
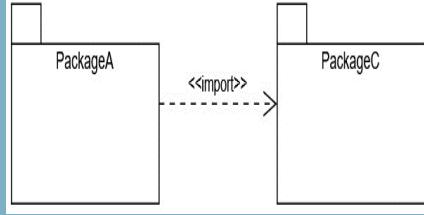
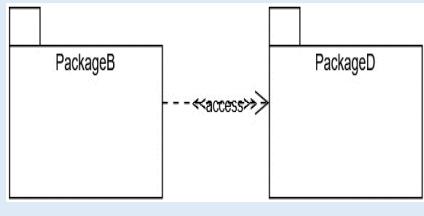
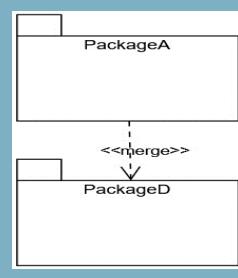


Figure 28: Formalism of a Package diagram



iii.) Components of a Package Diagram

Table 20: Components of a Package Diagram

NAME	Description	Representation
Package	A package is used to group together logically created elements within a system.	
Package import	It indicates that a functionality has been imported from one packet to another.	
Package access	A relationship Indicates that one package requires assistance from the function of another package.	
Package merge	It is a relationship which shows that, the functionality of two packages are combines to a single function.	



iv.) System Package Diagram

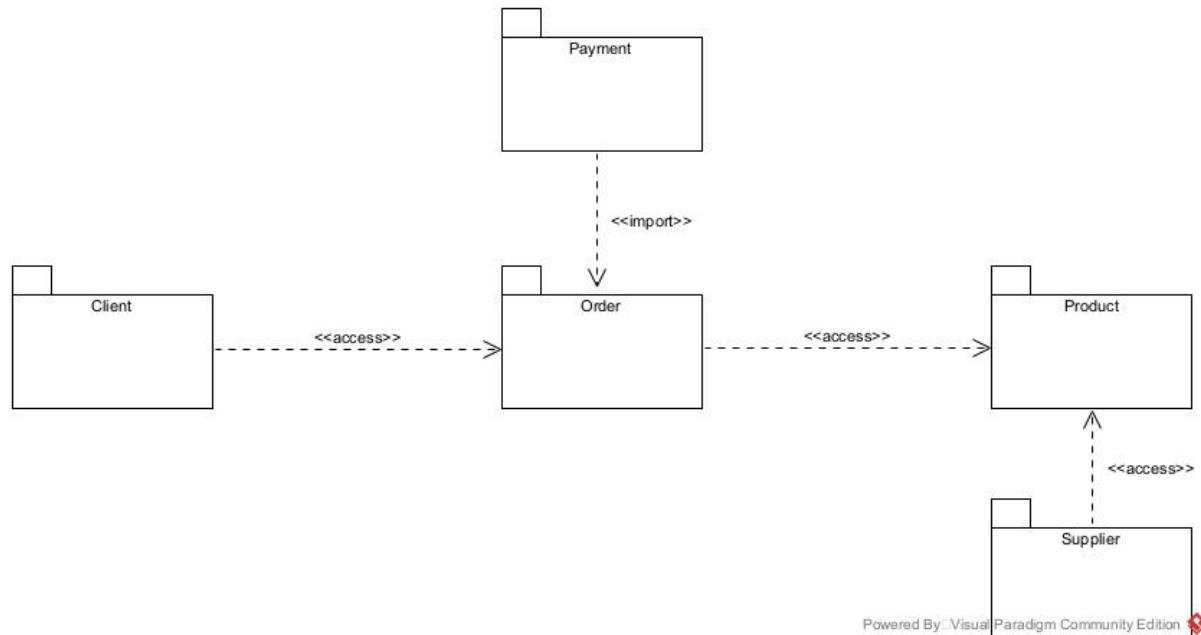
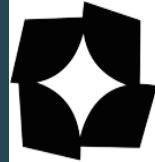


Figure 29: System Package diagram

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



CONCLUSION

The conception phase helped us identify the data needed to create our database and implement our application. The elements modelled in this phase gave us an overall view of the various modules of our application. Therefore, the next step in our project will be drafting the implementation or realization file, considering the elements modelled earlier.



CHAPTER 5: REALIZATION PHASE

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Preamble

In the realization phase, we focus on the practical aspects of implementing the system. This includes creating the deployment diagram, which shows the physical arrangement of hardware and software, and the component diagram, which outlines the structure and relationships of the system's components. These diagrams help ensure that the system is ready for deployment and functions as intended.

Content:

INTRODUCTION

I. DEPLOYMENT DIAGRAM

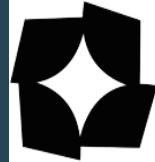
II. COMPONENT DIAGRAM

CONCLUSION

CONCLUSION

БІОЛОГІЧНІ РОДОДОНТИ

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



INTRODUCTION

The realization phase is where the system design is translated into practical implementation. In this phase, we will focus on the technical details necessary to deploy the system, including the creation of key diagrams such as the deployment and component diagrams. These will help illustrate how the system's components interact and are set up in the real world.

I. DEPLOYMENT DIAGRAM

i.) Definition

A deployment diagram is a visual representation of the physical architecture of a computer. It represents the physical disposition of material resources that constitute the system and shows the component repartition (software elements) that are executed inside these materials(nodes)

ii.) Formalism

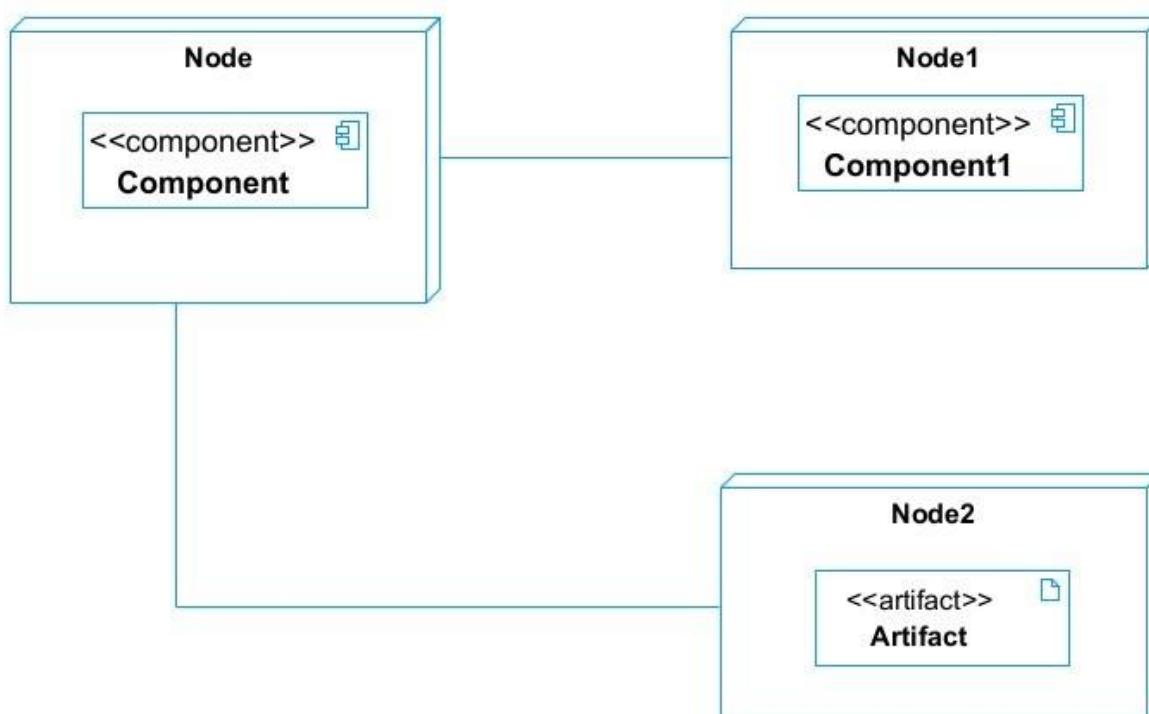


Figure 30: Formalism for Deployment Diagram

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



iii.) Components of Deployment Diagram

Table 21: Components of a Deployment Diagram

Element	Description	Notation
Node	It is a hardware used to deploy the application	
Artifact	An artefact is a major product, which is produced or used during the development of a software. E.g diagrams, data models, setup scripts	
Component	It represents a modular part of a system that encapsulates its content and whose manifestation is replaceable within its environment.	
Association	An association helps to connect two nodes together which permits them to communicate together	

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



iv.) System Deployment Diagram

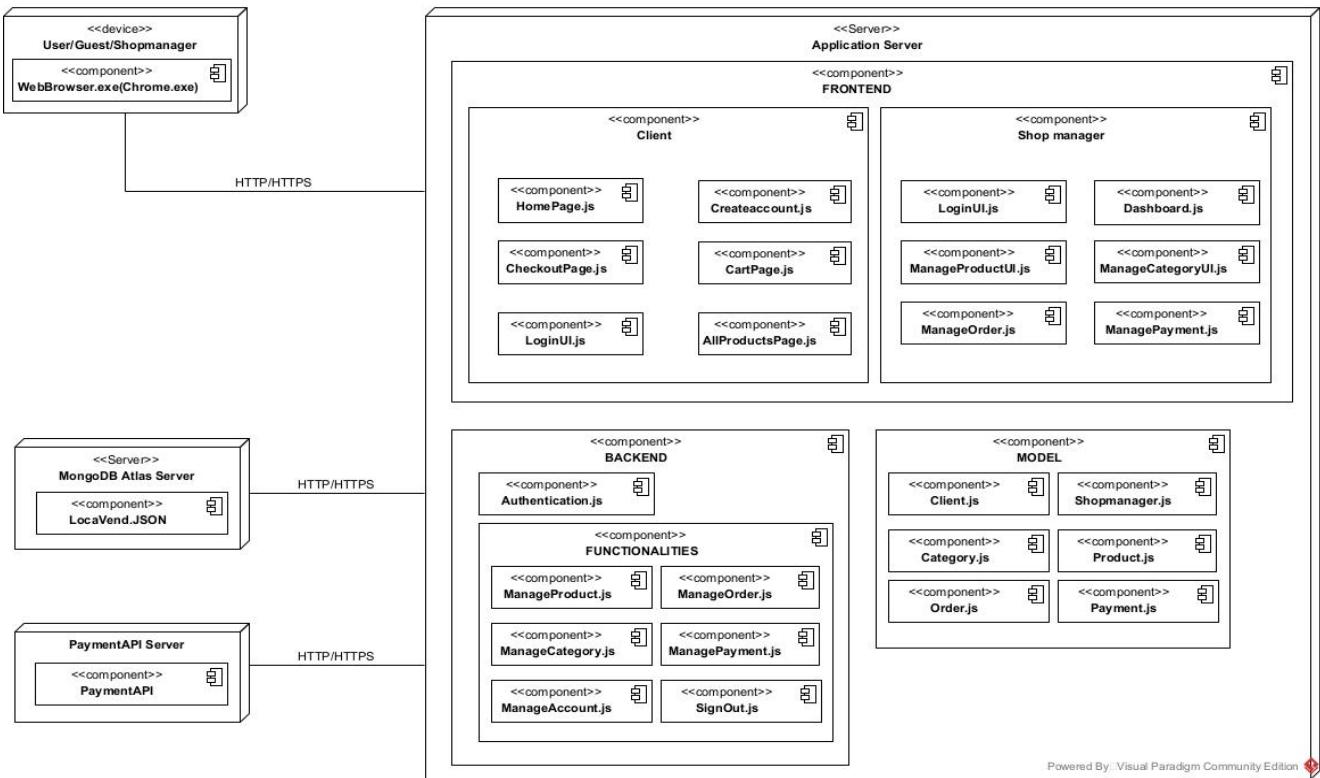


Figure 31: System's Deployment diagram

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

II. COMPONENT DIAGRAM

i.) Definition

A Component diagram describes the dependencies between various software component such as the dependency between executable file and source files. They represent the internal structure of our software (Software component).

ii.) Formalism

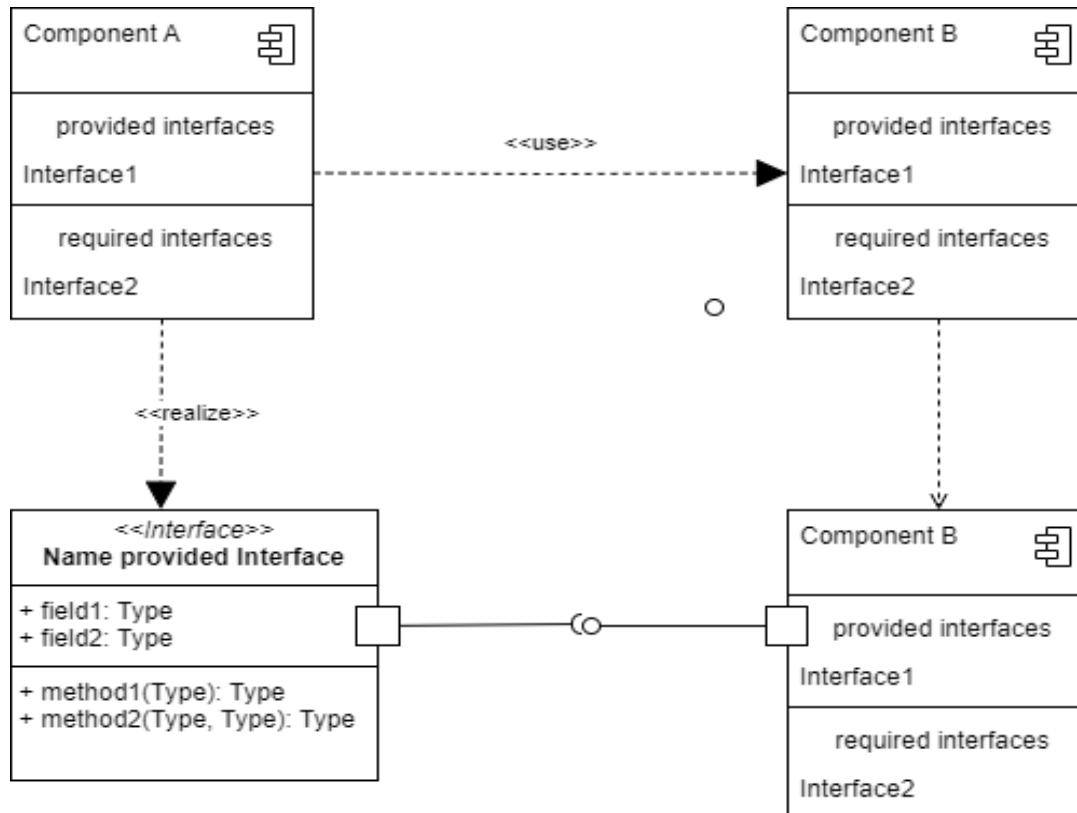
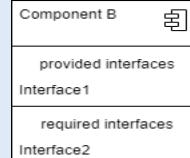
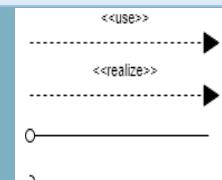
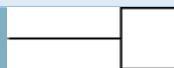
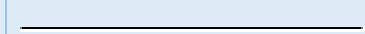
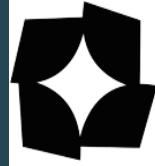


Figure 32: Formalism for Component Diagram

iii.) Components of a Component Diagram

Table 22: Components of a Component Diagram

Elements	Description	Notation
Component	A rectangle with the component's name inside, often with two smaller rectangles on the left. It represents a modular part of a system, encapsulating specific functionality or a group of functions.	
Interface	Depicted as a circle or semi-circle attached to a component, it defines a set of operations specifying the component's responsibility.	
Dependency	Illustrated as a dashed arrow, it indicates that one component relies on other components to function correctly.	
Port	Represented as a small square on a component's edge, it defines an entry or exit point for data or control flow.	
Connector	Shown as a solid line between two components or ports, it signifies the communication path between them.	



iv.) System Component Diagram

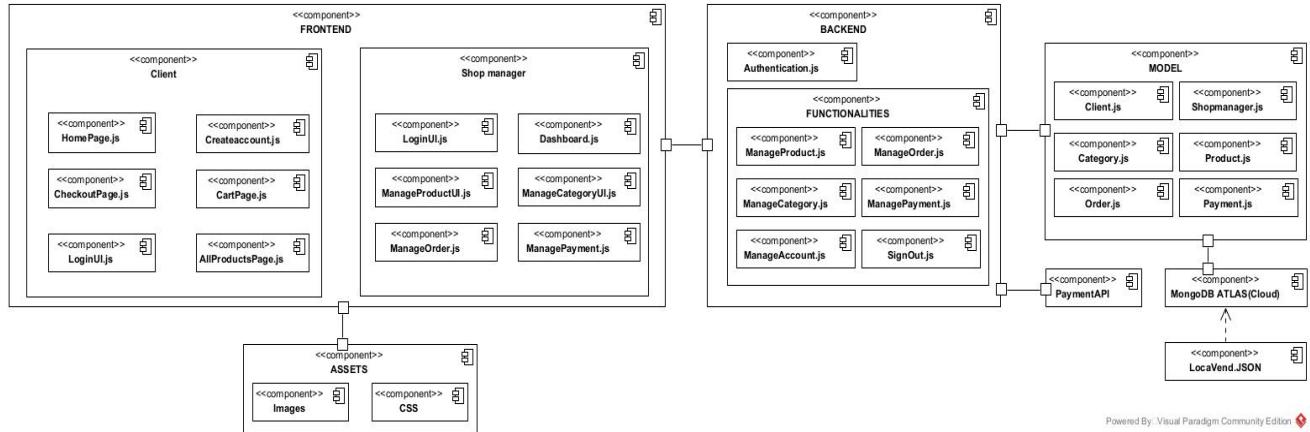
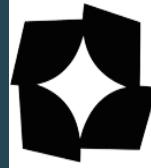


Figure 33: System's Component diagram

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



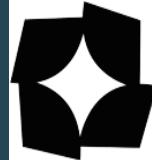
CONCLUSION

In this phase, we focused on both the logical and physical architecture of the application. We began with an overview of the physical architecture, which allowed us to present the system's deployment. Finally, we discussed the logical architecture, describing the various components that make up our system.



CHAPTER 6: FUNCTIONALITY TESTING

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Preamble

In this section, we focus on testing our application to ensure the highest possible quality for users and to prevent potential issues from becoming bottlenecks. The goal is to verify that the software meets specified requirements and performs its expected functions.

Content:

INTRODUCTION

I. APPLICATION FUNCTIONALITIES

II. TESTS SHOWCASES

CONCLUSION

CONCLUSION

RESULTS?

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



INTRODUCTION

Software testing is a comprehensive process that ensures that software applications are reliable, secure, and user-friendly. It encompasses a range of techniques and methodologies, each targeting different aspects of software to provide a quality product.



I. APPLICATION FUNCTIONALITIES

- **Order and Payment**

Clients can browse available products from the shop, place orders, and make payments using Mobile Money (MoMo).

- **Manage Orders**

The shop manager can view, accept, or cancel client orders, depending on product availability.

- **Manage Products**

The shop owner can add, update, or remove products such as phones, laptops, and accessories from the online catalog.

- **Account Management (Admin)**

The admin oversees all system users (clients and delivery staff), ensuring smooth operation and accurate records.

These modules collectively form a comprehensive platform for gas distribution, ensuring that clients can efficiently place orders and suppliers and admins can manage the system smoothly.



II. TESTS SHOWCASES

Software testing can be performed at various stages of development, such as **Unit Testing**, **Integration Testing**, **System Testing**, and **Acceptance Testing**, using different methods like **Black Box** or **White Box** testing.

In this section, we will focus on **Integration Testing** to ensure different components (APIs, databases) communicate and work together, and **System Testing** to verify the entire system, from frontend to backend, functions as a whole.

This approach ensures that both the connections between components and the overall system function as intended.



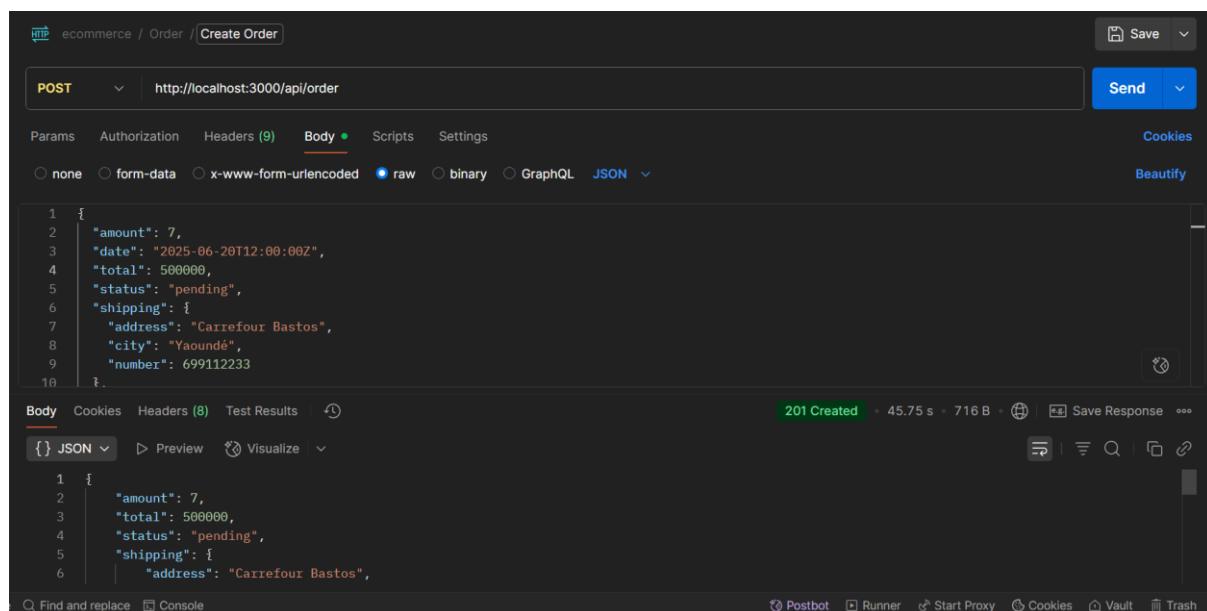
i.) Integration testing

API ↔ Database Interaction

These tests ensure that the API correctly interacts with the database by performing CRUD operations. We use Postman to test the different API end points. Below are the tests carried out on our API's end points:

Order Management tests

- Create Order (**POST /api/order**):
- This endpoint allows the client to place an order. We tested the creation of an order with valid data (Shipping address, products, payment info) and verified that it was correctly saved in the database



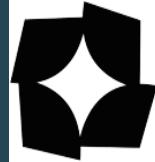
The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** http://localhost:3000/api/order
- Body:** Raw JSON (selected)
- Request Body:**

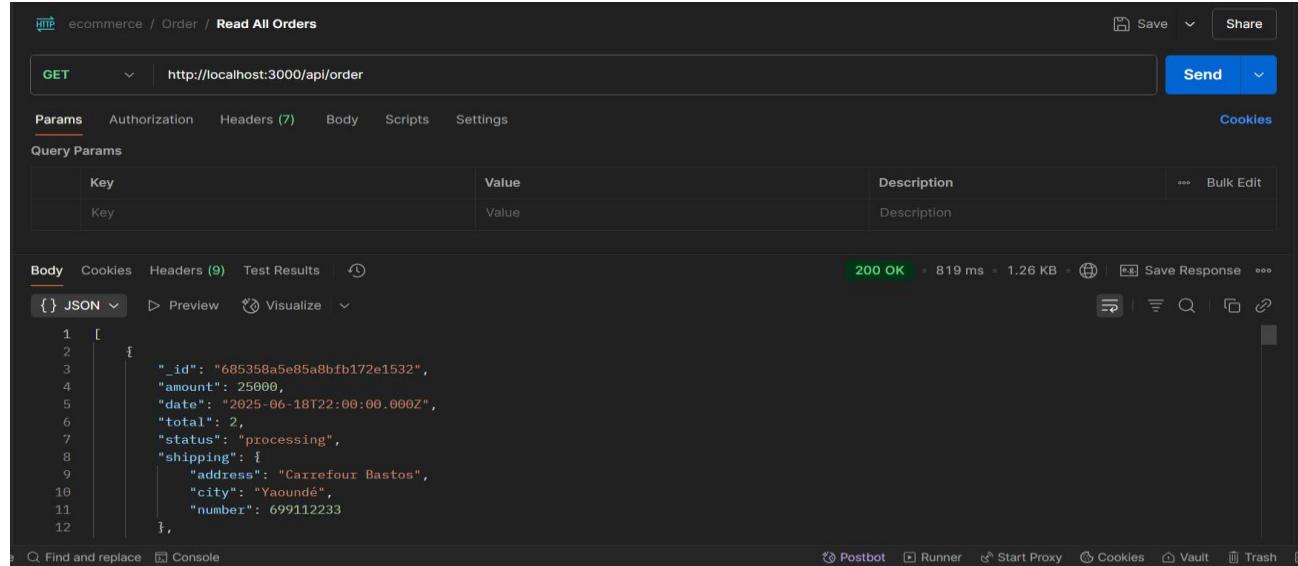
```
1 {
2   "amount": 7,
3   "date": "2025-06-20T12:00:00Z",
4   "total": 500000,
5   "status": "pending",
6   "shipping": {
7     "address": "Carrefour Bastos",
8     "city": "Yaounde",
9     "number": 699112233
10 }
```
- Response:** 201 Created
- Time:** 45.75 s
- Size:** 716 B
- Save Response** button

Figure 34: POST Request to /api/order/ to create an order

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



- Get All Orders (**GET /api/order**): This endpoint enables the shop manager to retrieve all existing orders. We verified that all order records were fetched correctly from the database

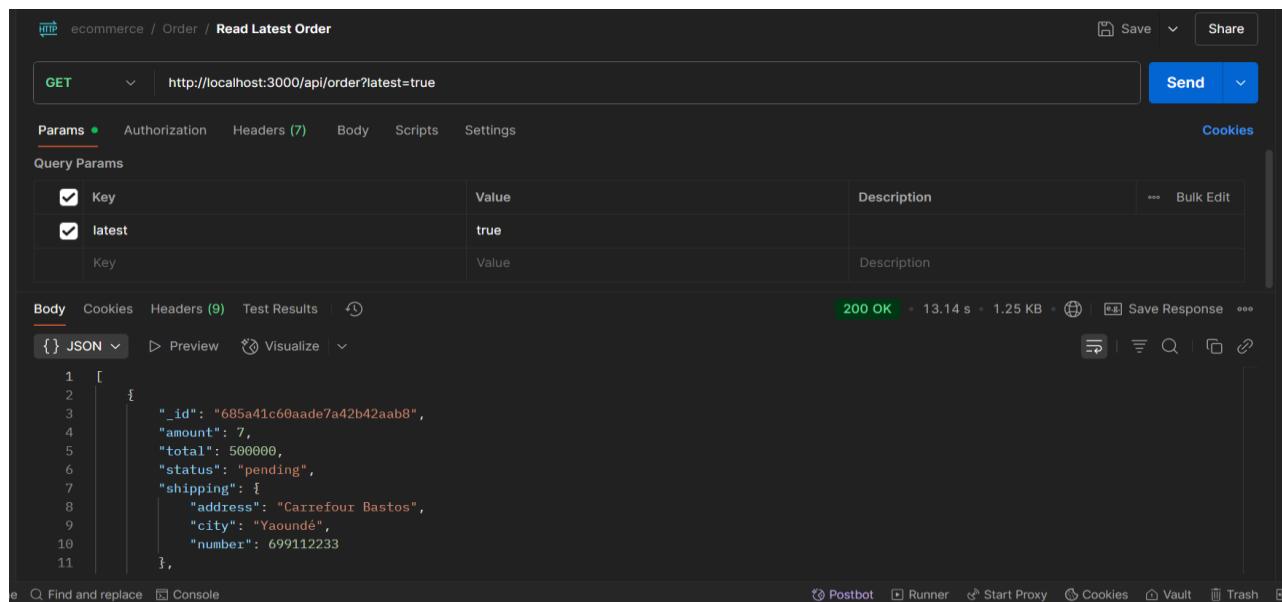


```

1 [
2   {
3     "_id": "685358a5e85a8fb172e1532",
4     "amount": 25000,
5     "date": "2025-06-18T22:00:00.000Z",
6     "total": 2,
7     "status": "processing",
8     "shipping": {
9       "address": "Carrefour Bastos",
10      "city": "Yaoundé",
11      "number": 699112233
12    }
13  ]
  
```

Figure 35: GET Request to `/api/order` to retrieve a list of all orders

- Get Latest Orders (**GET /api/order?latest=true**): This test checks that the API returns the most recent orders made by clients. We ensured the orders were sorted by creation date in descending order.



```

1 [
2   {
3     "_id": "685a41c60aade7a42b42aab8",
4     "amount": 7,
5     "total": 500000,
6     "status": "pending",
7     "shipping": {
8       "address": "Carrefour Bastos",
9       "city": "Yaoundé",
10      "number": 699112233
11    }
12  ]
  
```

Figure 36: Get Request to `s/api/order?latest=true` with parameter (`latest = true`) to get the most recent orders

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Product Retrieval by ID – API Test

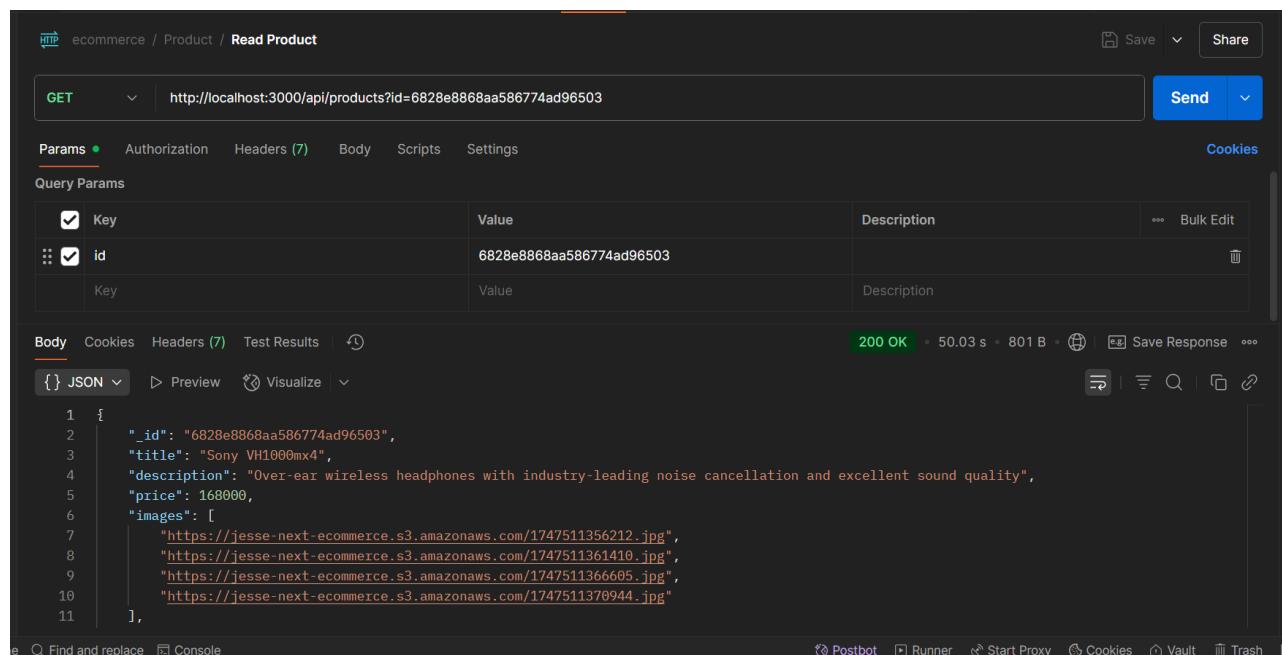
- **Get Products by IDs (POST /api/cart)**

This endpoint allows the frontend to retrieve full product details based on a list of product IDs in the client's cart.

We tested this by sending an array of product IDs to the API and verified that:

- Only products matching the given IDs were returned.
- Each returned product included correct attributes (e.g., title, price, image, stock).
- The response was accurate even when duplicate IDs were included.

This ensures the client's cart displays real-time product information fetched directly from the database.



The screenshot shows a Postman API client interface. The URL in the header is `http://localhost:3000/api/products?id=6828e8868aa586774ad96503`. The 'Params' tab is selected, showing a single query parameter `id` with the value `6828e8868aa586774ad96503`. The 'Body' tab shows the JSON response:

```

1 {
2   "_id": "6828e8868aa586774ad96503",
3   "title": "Sony VH1000mx4",
4   "description": "Over-ear wireless headphones with industry-leading noise cancellation and excellent sound quality",
5   "price": 168000,
6   "images": [
7     "https://jesse-next-commerce.s3.amazonaws.com/1747511356212.jpg",
8     "https://jesse-next-commerce.s3.amazonaws.com/1747511361410.jpg",
9     "https://jesse-next-commerce.s3.amazonaws.com/1747511366605.jpg",
10    "https://jesse-next-commerce.s3.amazonaws.com/1747511370944.jpg"
11  ],

```

Figure 37: API response displaying product details for particular product Id

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



We tested https://sandbox.momodeveloper.mtn.com/collection/v1_0/requesttopay

API endpoint, which request a payment from a client(payer) on behalf of a supplier.

The screenshot shows the Postman interface with the following details:

- HTTP Method:** POST
- URL:** https://sandbox.momodeveloper.mtn.com/collection/v1_0/requesttopay
- Headers (13):**
 - X-Reference-Id: f6dce776-5fb3-4fb7-b2f9-9951cdad14e4
 - X-Target-Environment: sandbox
 - Ocp-Apim-Subscription-Key: 030005d1c7eb4830a30422109751b818
 - Content-Type: application/json
- Body:** Key Value Description
- Response Status:** 202 Accepted
- Response Time:** 268 ms
- Response Size:** 182 B
- Actions:** Save Response, ...

Figure 38:Test to request payment from a client on behalf of a supplier

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



CONCLUSION

Testing is an integral part of the software development life cycle, performed to verify the correctness, reliability, and robustness of the implemented functionality. This phase is critical in software development, as it validates the implemented features and identifies any issues for resolution.



CHAPTER 7: USER GUIDE

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



Preamble

A user guide explains how to use a software application in a language that a nontechnical person can understand. Thus, it enables the user to easily use the application to familiarize with the software and discover its functionalities.

Content:

INTRODUCTION

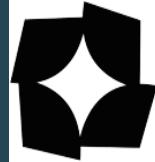
- I. DESCRIPTION OF APPLICATION
- II. INSTALLATION GUIDE
- III. USER MANUAL

CONCLUSION

CONCLUSION

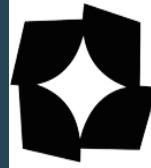
ACKNOWLEDGEMENTS

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



INTRODUCTION

A user guide is a detailed document that offers step-by-step instructions for effectively using a product or service. It provides a wide range of information to help users understand and utilize the product efficiently. This section will provide a brief overview of our web application, followed by step-by-step instructions on how to set it up, and conclude with detailed explanations of its main features and how to use them.



I. DESCRIPTION OF APPLICATION

i.) Purpose

Locavend is a web-based application designed to support the digital management of a local vendor's shop, focusing on the sale and supply of electronic products such as mobile phones, laptops, and accessories. The platform allows clients to place orders online and enables the shop manager to handle product inventory, view client orders, and update order statuses. The goal is to modernize the operations of a traditional shop, reduce manual workload, and provide a more convenient shopping experience for clients.

ii.) Target Audience

LocalVend is designed for the following key actors:

Clients

Individuals looking to purchase electronic products from the shop.
Benefit: They can easily browse items, place orders, and pay using mobile payment options without visiting the physical store.

Shop manager

The local vendor responsible for managing the shop's inventory and processing orders.

Benefit: Simplifies daily operations through digital inventory tracking, real-time order management, and improved customer service

By addressing the operational needs of the shop manager and the purchasing convenience of clients, **Locavend** offers a practical solution for modernizing local retail in Cameroon.



II. INSTALLATION GUIDE

After developing the **Locavend** application, some setup steps are required to deploy and run it. Since Locavend is a web-based application using **MongoDB Atlas** (cloud database) and **Next.js**, we do **not** need to install a local database server like MySQL or use XAMPP. Instead, we only need to set up and run the application server (Next.js) and connect it to MongoDB Atlas through environment variables.

Required Environment

- ✓ A MongoDB Atlas account (cloud database)
- ✓ Node.js and npm installed
- ✓ A modern browser (e.g., Chrome, Firefox)

a.) Database Setup with MongoDB Atlas

1. Create a MongoDB Atlas Account.

Go to <https://www.mongodb.com/cloud/atlas> and create a free account.

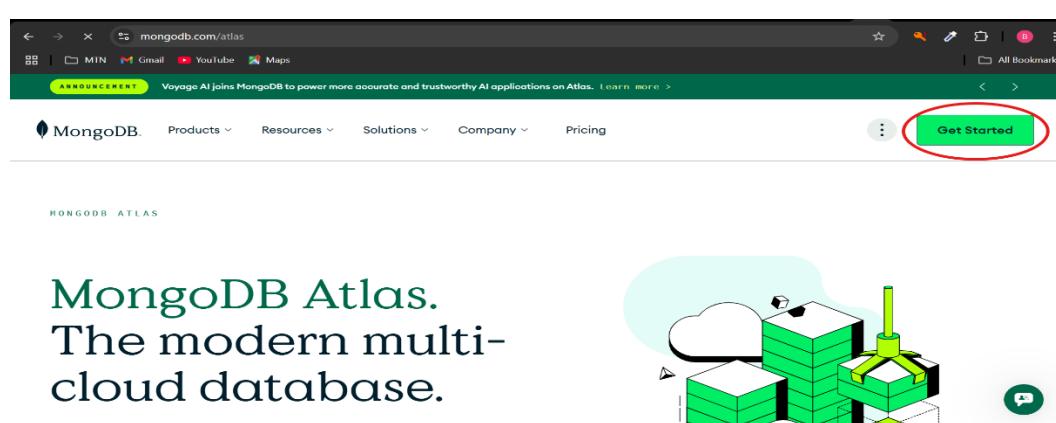


Figure 39: Creating MongoDB Atlas Account

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



2. Create a New Cluster

- a. Click “Build a Database”
- b. Choose **Shared** cluster (free tier)
- c. Select a region close to you and create the cluster.

The screenshot shows the MongoDB Atlas Data Services Overview page. On the left, there's a sidebar with sections for DATABASE (Clusters), SERVICES (Atlas Search, Stream Processing, Triggers, Migration, Data Federation), and SECURITY (Quickstart, Backup). The main area has tabs for next-commerce (selected), Data Services, and Charts. Under the Data Services tab, there's an 'Overview' section with a 'Clusters' card. This card shows 'Cluster0' with options to 'Connect', 'Edit configuration', 'Browse collections', and 'View monitoring'. To the right of the card is a 'Data Size: 297.71 KB' indicator. At the top right of the card is a 'Create cluster' button, which is circled in red. Below the card is a 'Toolbar' with tabs for Resources (5), Tips (4) (which is selected), and Performance. The 'Performance' tab shows links for 'Performance (1)', 'Cost (1)', and 'Resilience (2)'. There's also a green feedback icon at the bottom right of the page.

Figure 40: Creating cluster

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



3. Create a Database and Collection

a. Go to Database > Browse Collections

b. Click Create Database

c. Give it a name like **locavendDB** and a collection name like **orders**.

The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with 'Clusters' selected under 'DATABASE'. A red circle highlights the '+ Create Database' button at the top of this sidebar. Another red circle highlights the '+' icon next to the 'test' database in the list below. The main panel shows the 'test' database details: LOGICAL DATA SIZE: 9.62KB, STORAGE SIZE: 252KB, INDEX SIZE: 288KB, and TOTAL COLLECTIONS: 7. Below this, a table lists collections: accounts, admins, categories, orders, products, sessions, and users. A green 'CREATE' button is visible at the top right of the main panel.

Figure 41: Creating Database (Bigger red circle) and collection (smaller red circle)

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



4. Allow Access from Anywhere

- a. Go to **Network Access**
- b. Click “Add IP Address”
- c. Choose “Allow access from anywhere” (0.0.0.0/0).

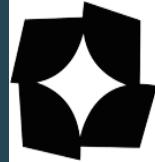
The screenshot shows the MongoDB Atlas interface for managing network access. On the left, there's a sidebar with various options like Stream Processing, Triggers, Migration, Data Federation, SECURITY (Quickstart, Backup), Database Access (Network Access highlighted with a green oval), and New On Atlas (5). The main area is titled 'Network Access' and shows the 'IP Access List' tab. It displays a table of IP addresses with columns for IP Address, Comment, Status, and Actions (Edit, Delete). There are four entries: 129.0.205.157/32, 129.0.226.29/32, 41.205.90.17/32, and 129.0.174.81/32, all marked as Active. Above the table, two yellow info boxes appear: one about the current IP address not being added, and another about connecting from a list of IP addresses. A red circle highlights the 'Add Current IP Address' button at the top right.

IP Address	Comment	Status	Actions
129.0.205.157/32		Active	<button>Edit</button> <button>Delete</button>
129.0.226.29/32		Active	<button>Edit</button> <button>Delete</button>
41.205.90.17/32		Active	<button>Edit</button> <button>Delete</button>
129.0.174.81/32		Active	<button>Edit</button> <button>Delete</button>

Figure 42: Allow access(IP Adress) from anywhere (Application Server)

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



5. Get Your Connection String

- Go to **Database** > Click **Connect**
- Choose “Connect your application”
- Copy the connection URI like this:

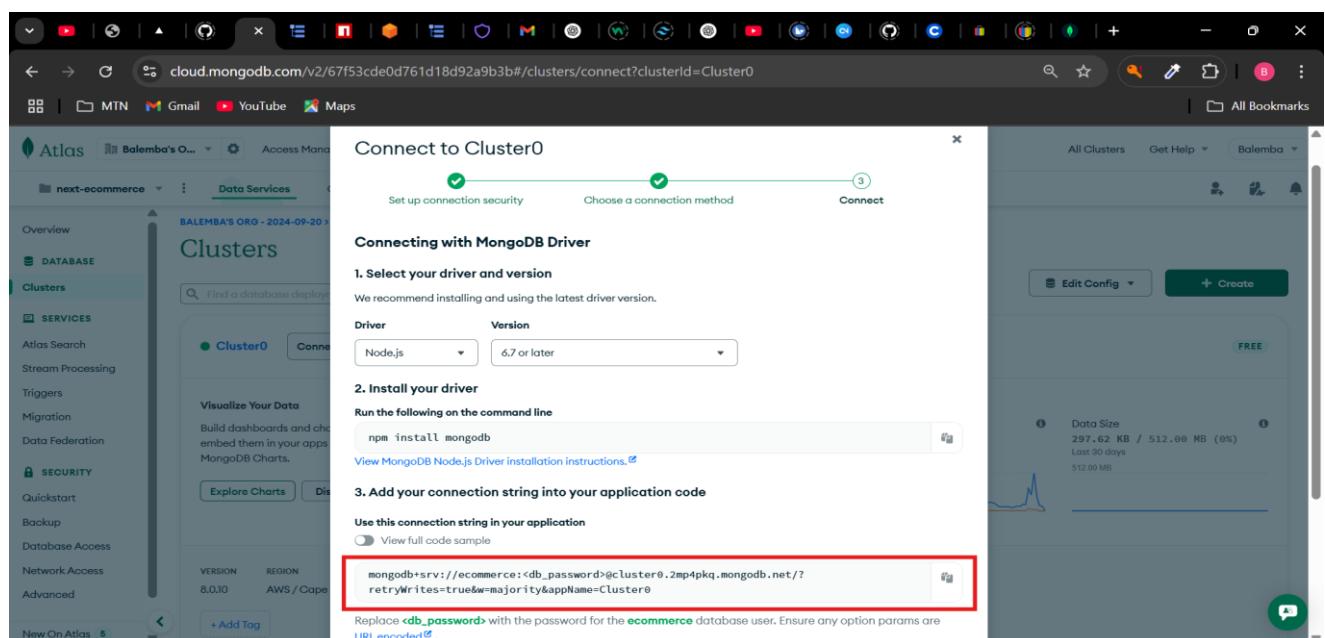


Figure 43: Getting connection string to connect backend to cluster in Mongodb Atlas

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



- Paste this URI into your .env.local file in your Next.js project as:

```
ecommerce-front > env
1 MONGODB_URI="mongodb+srv://ecommerce:ecommerce%23123@cluster0.2mp4pkq.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"
2
```

Figure 44: Inserting Connection URI to env.local file into Next.js project

b.) Application Server (Next.js)

- Install Node.js and npm (Recent compatible versions)

The screenshot shows the official Node.js website at nodejs.org/en. The main heading is "Run JavaScript Everywhere". Below it, a section explains what Node.js is: "Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts." A red box highlights the "Install Node.js" button. To the right, there is a code editor window titled "Create an HTTP Server" containing sample JavaScript code for starting a local server. A "Copy to clipboard" button is visible at the bottom of the code editor.

Figure 45: Nodejs website for download and installation

The screenshot shows the npm website at npmjs.com. The main slogan is "Build amazing things". A search bar labeled "Search packages" is present. At the bottom, a small note reads: "We're GitHub, the company behind the npm Registry and npm CLI. We".

Figure 46: Install npm

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



2. Clone or Download the Project to Target machine

Open your terminal or command prompt and run:

```
C:\Users\acer\Desktop>git clone https://github.com/Balemba-Jeez/PersonalProject.git
```

Figure 47: Downloading project to target machine

3. Install Project Dependencies

Navigate into the project folder and run:

```
C:\Users\acer\OneDrive\Desktop\Defense\Level3\PersonalProject\Code>npm install
```

Figure 48: Installing Project or Application Dependencies

4. Start the Development Server

```
C:\Users\acer\OneDrive\Desktop\Defense\Level3\PersonalProject\Code\ecommerce-admin>npm run dev
> ecommerce-admin@0.1.0 dev
> next dev

  ▲ Next.js 15.2.4
    - Local:   http://localhost:3000
    - Network: http://192.168.8.101:3000
    - Environments: .env

  ✓ Starting...
  ✓ Ready in 3.7s
  ○ Compiling _/error ...
  ✓ Compiled _/error in 3.3s (407 modules)
  ✓ Compiled in 610ms (303 modules)
  GET /_next/static/webpack/b44a5bd25280ad39.webpack.hot-update.json 404 in 4287ms
  ▲ Fast Refresh had to perform a full reload. Read more: https://nextjs.org/docs/messages/fast-refresh-reload
  ○ Compiling /orders ...
  ✓ Compiled /orders in 1487ms (565 modules)
  GET /_next/static/webpack/b44a5bd25280ad39.webpack.hot-update.json 404 in 1449ms
  ▲ Fast Refresh had to perform a full reload. Read more: https://nextjs.org/docs/messages/fast-refresh-reload
  GET /orders 200 in 2027ms
  orders []
  Invalid DOM property 'class'. Did you mean 'className'?
  Invalid DOM property 'fill-rule'. Did you mean 'fillRule'?
  Invalid DOM property 'clip-rule'. Did you mean 'clipRule'?
  Invalid DOM property 'stroke-width'. Did you mean 'strokeWidth'?
  Invalid DOM property 'stroke-linecap'. Did you mean 'strokeLinecap'?
  Invalid DOM property 'stroke-linejoin'. Did you mean 'strokeLinejoin'?
  orders []
  GET /orders 200 in 352ms
```

Figure 49: Starting the development server

Written by: BALEMBABA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



III. USER MANUAL

In this part, the steps of carrying out the operations will be explained in detail. All this will be done through tests and explanations with screenshots in support to allow users not only to know how to actually do manipulations, but also to allow them to have an overview of the results they should expect.

1. Connecting to the platform

Connecting to the platform is not complicated at all. However, it is essential to meet a number of prerequisites, namely:

- ✓ Have a terminal (computer, tablet or smartphone);
- ✓ Have installed a web browser (preferably the most recent version) in your terminal; Enter the URL: <http://localhost:3001>. You will obtain this page

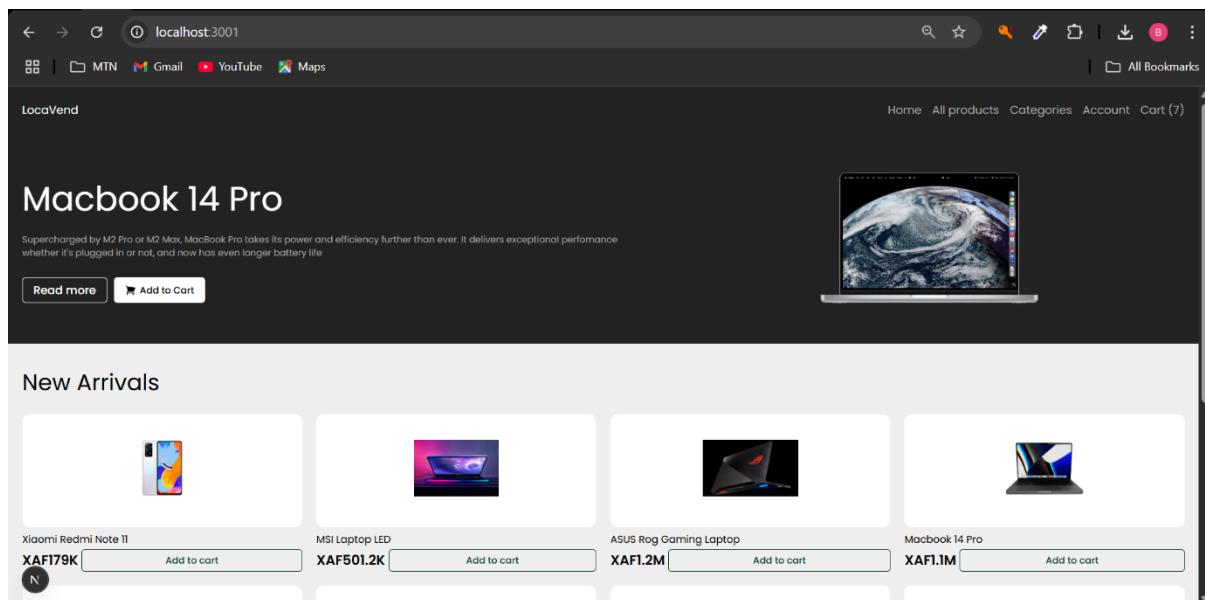
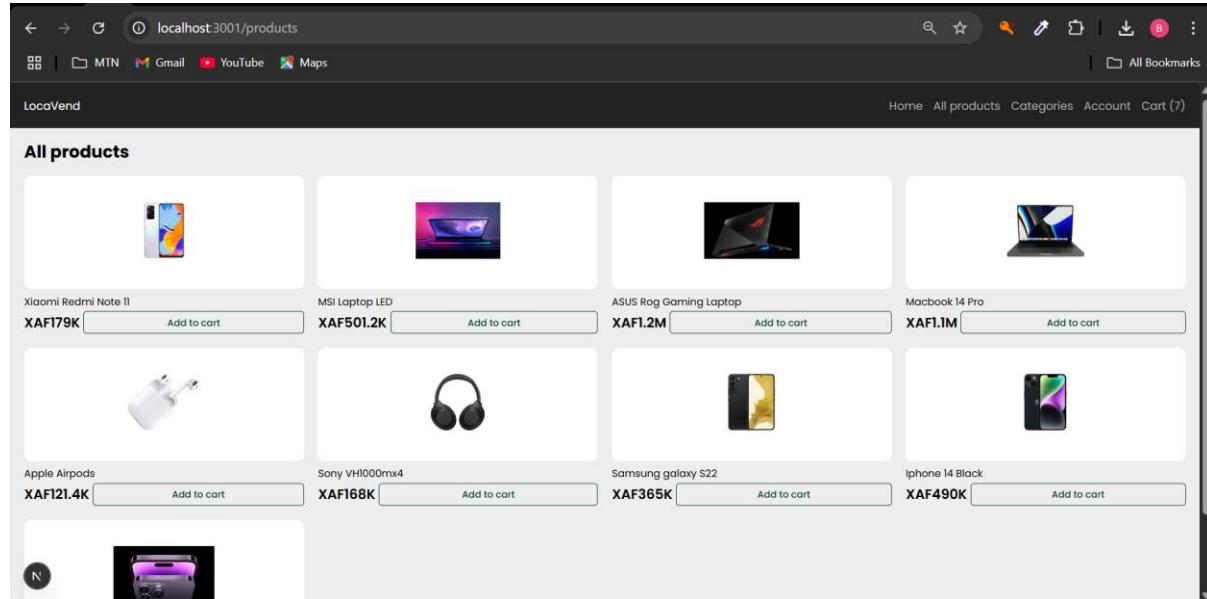


Figure 50: Home page with featured and New arrival products

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



2. Here we have All products page from which the client or guest can add a product to cart or click on product to view product details.

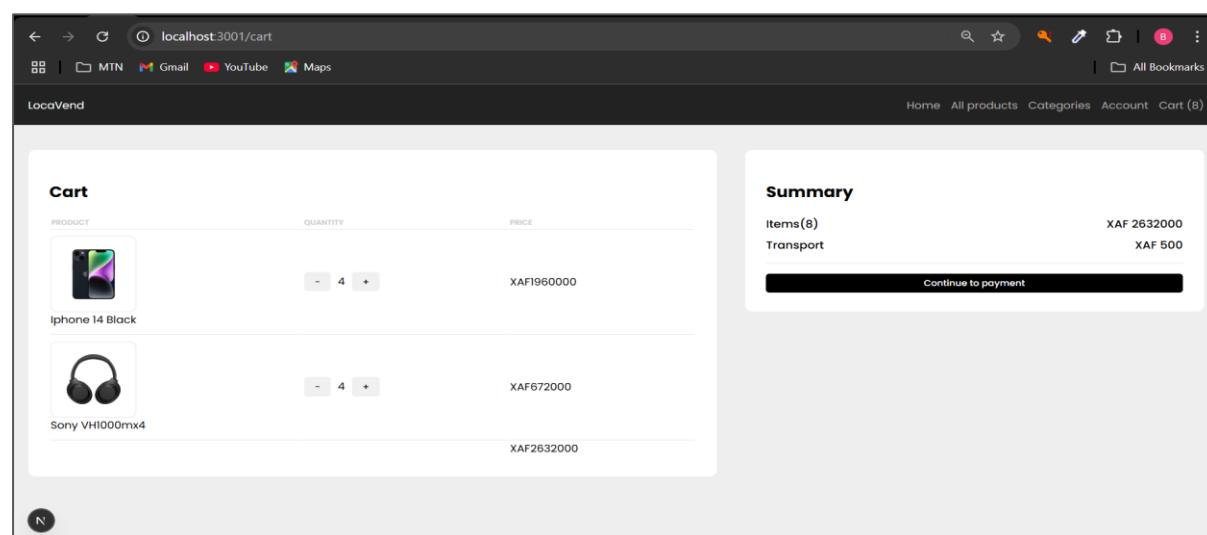


The screenshot shows a web browser displaying the 'All products' page of a website named 'LocaVend'. The page features a grid of eight product cards. Each card includes a small image of the product, its name, a price in XAF, and an 'Add to cart' button. The products listed are:

- Xiaomi Redmi Note 11 | XAF179K | Add to cart
- MSI Laptop LED | XAF501.2K | Add to cart
- ASUS Rog Gaming Laptop | XAF1.2M | Add to cart
- Macbook 14 Pro | XAF1.1M | Add to cart
- Apple Airpods | XAF121.4K | Add to cart
- Sony VH1000mx4 | XAF168K | Add to cart
- Samsung galaxy S22 | XAF365K | Add to cart
- Iphone 14 Black | XAF490K | Add to cart

Figure 51: All product listing page

3. Cart Page



The screenshot shows a web browser displaying the 'Cart' page of the 'LocaVend' website. The page is divided into two main sections: 'Cart' on the left and 'Summary' on the right.

Cart:

PRODUCT	QUANTITY	PRICE
Iphone 14 Black	4	XAF1960000
Sony VH1000mx4	4	XAF672000

Summary:

Items(8)	XAF 2632000
Transport	XAF 500

A 'Continue to payment' button is located at the bottom of the summary section.

Figure 52: Cart page

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



4. Payment/Checkout page

The screenshot shows a web browser window for LocaVend. The URL is localhost:3001/payment. The page is titled "Review order". It displays a table of items with columns for PRODUCT, QUANTITY, and PRICE. Two items are listed: "iPhone 14 Black" (XAF 1960000) and "Sony VHI000mx4" (XAF 672000). Below the table is a "Summary" section showing "Items(8)", "Transport", and "Order total" (XAF 2632000). The "Order total" is broken down into XAF 2632000, XAF 500, and XAF 2632500. A "Confirm and pay" button is at the bottom. On the left, there's a "Ship to" section with input fields for Name, Email, City, Street Address, and CN e.g. CED038798L. At the bottom left is a "Pay with" section with a "Mobile money" option selected.

Figure 53: Payment/Checkout page

5. Shop manager Dashboard

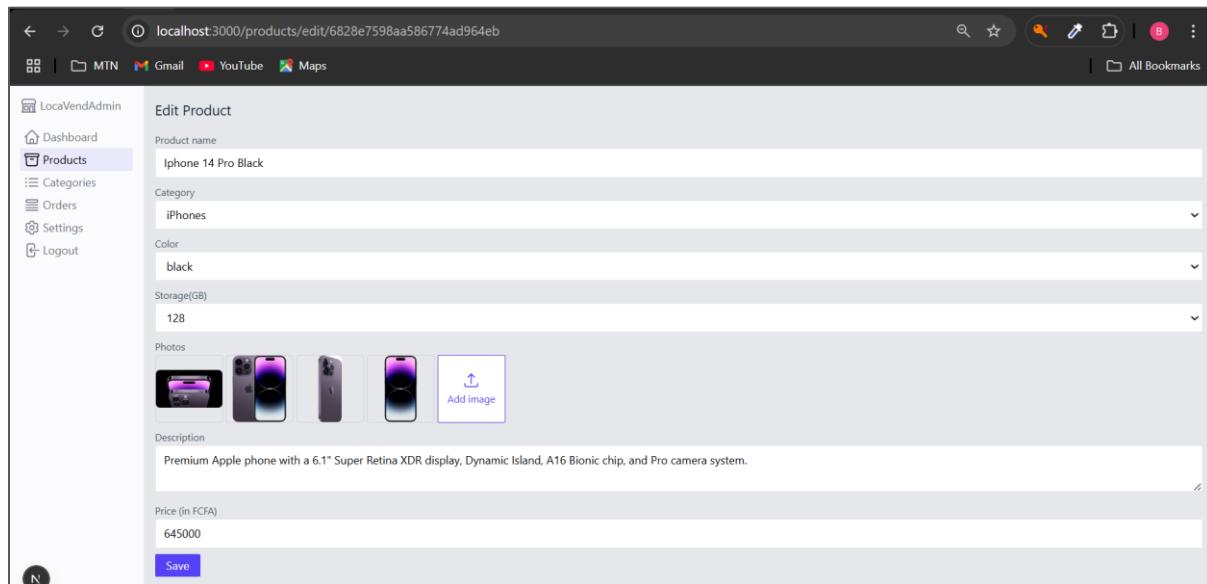
The screenshot shows a web browser window for LocaVendAdmin. The URL is localhost:3000. The page is titled "Hello, balembajessy". The dashboard has a sidebar with links for Dashboard, Products, Categories, Orders, Settings, and Logout. The main area is divided into sections: "Total" (Products: 9, Orders: 3, Revenues: 25.5K), "Orders" (Today: 1, This Week: 1, This Month: 3), "Recent orders" (Order ID: #685a41c6, Products: 1, Time: 4h ago, Status: Pending), and "Recent products" (Sony VHI000mx4, Headphones, XAF 168000, Date Added: 2025-06-19).

Figure 54: Shop manager Dashboard

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



6. Edit Product page



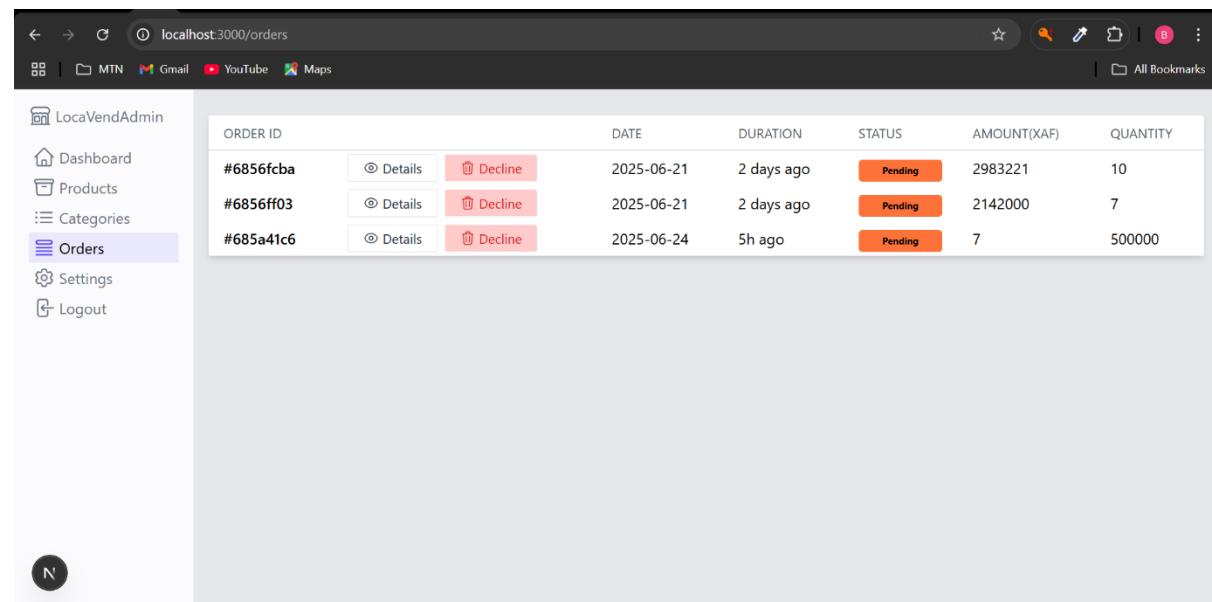
The screenshot shows the 'Edit Product' page of a web-based inventory system. The left sidebar shows navigation links: Dashboard, Products (selected), Categories, Orders, Settings, and Logout. The main form has the following fields:

- Product name: Iphone 14 Pro Black
- Category: iPhones
- Color: black
- Storage(GB): 128
- Photos: Displays five small images of an iPhone 14 Pro and an 'Add image' button.
- Description: Premium Apple phone with a 6.1" Super Retina XDR display, Dynamic Island, A16 Bionic chip, and Pro camera system.
- Price: (in FCFA) 645000

A blue 'Save' button is at the bottom.

Figure 55: Delivery Person Dashboard

7. Order page listing all recent orders.



The screenshot shows the 'Orders' page of the web-based inventory system. The left sidebar shows navigation links: Dashboard, Products, Categories, Orders (selected), Settings, and Logout. The main area displays a table of recent orders:

ORDER ID		DATE	DURATION	STATUS	AMOUNT(XAF)	QUANTITY
#6856fcba	Details Decline	2025-06-21	2 days ago	Pending	2983221	10
#6856ff03	Details Decline	2025-06-21	2 days ago	Pending	2142000	7
#685a41c6	Details Decline	2025-06-24	5h ago	Pending	7	500000

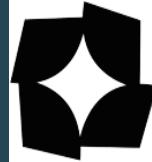
Figure 56: Order page

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



CONCLUSION

As we conclude this final phase of the report, we have outlined the procedure for deploying the application. The main interfaces have also been presented to help users become familiar with the platform's functionalities and appearance, ensuring a smooth onboarding experience.

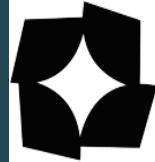


GENERAL CONCLUSION

In conclusion, this project focused on the development of **Locavend**, a web-based application designed to support the purchase and supply management activities of a local vendor in Cameroon. The system was built to address common challenges such as manual tracking of product orders, lack of transparency in sales processes, and poor coordination between clients and shop managers. By streamlining operations such as order creation, product management, and order status tracking, **Locavend** enhances both efficiency and user experience. The application provides a centralized and digital platform for managing customer orders and inventory, contributing to the modernization of local shop management and improving the overall reliability and convenience for both the vendor and clients.



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



ANNEX

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



BIBLIOGRAPHY

- UML 2 For Dummies 1st Edition 2003 by James A. Schardt and Michael Jesse Chonoles.
- GanntProject Introduction for Beginners. 65 pages;
- Javascript_tutorial by tutorialspoint. 50 pages.



WEBOGRAPHY

- <https://www.worldtravelguide.net/guides/africa/cameroon/shopping-nightlife/> | Local shops in Cameroon.
- <https://www.w3schools.com/js/default.asp> | JavaScript Tutorial
- <https://nextjs.org/docs> | Next js basics
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects Standard built-in objects.
- <https://tailwindcss.com/> | Tailwind CSS basics.

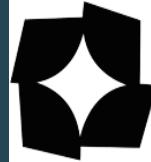


TABLE OF CONTENTS

DEDICATION	i
ACKNOWLEDGEMENTS	ii
SUMMARY	iii
LIST OF FIGURES	iv
LIST OF TABLES	vii
GLOSSARY	vii
ABSTRACT	ix
RÉSUMÉ	x
GENERAL INTRODUCTION	1
TECHNICAL PHASE.....	2
• Introduction	2
• Conclusion	15
CHAPTER 1: EXISTING SYSTEM.....	3
• Preamble	4
• Introduction	5
• Conclusion	29
I. PRESENTATION OF THEME	
• Theme	6
• Definition of Terms	6
II. STUDY OF THE EXISTING SYSTEM.....	7

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



• i. Delimitation of the Field of Study	7
• ii. Description of the Existing System	8
III. CRITICISMS OF THE EXISTING SYSTEM	9
IV. PROBLEMATIC	10
V. LIMITATIONS AND PROPOSED SOLUTIONS	10
• Conclusion	11
CHAPTER 2: SPECIFICATION BOOK.....	12
• Preamble	13
• Introduction	14
• Conclusion	28
• Introduction	29
I. CONTEXT AND JUSTIFICATION.....	15
• i. Context	15
• ii. Justifications	16
II. PROJECT OBJECTIVES.....	17
• i. General Objective	17
• ii. Specific Objectives	17
III. EXPRESSION OF NEEDS.....	19
• i. Functional Needs	19
• ii. Non-Functional Needs	20
IV. PROJECT PLANNING.....	21
• i. Project Plan	21
• ii. Gantt Diagram	22

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



V. ESTIMATED COST OF PROJECT.....	23
• i. Software Resources	23
• ii. Hardware Resources	25
• iii. Human Resources	25
• Total Estimated Cost	26
• iv. Participants	27
VI. PROJECT CONSTRAINTS	28
VII. DELIVERABLES	28
• Conclusion	29
CHAPTER 3: ANALYSIS	30
• Preamble	31
• Introduction	32
I. METHODOLOGY	51
• a. Comparative Study between MERISE and UML	33
• b. Comparative Study of Unified Processes	36
◦ i. Presentation of UP Method	36
◦ ii. Presentation of the 2-TUP Method	39
◦ iii. Comparative Study between 2TUP and UP	41
◦ iv. Justification on the Choice of Analysis Method	41
II. MODELLING	42
• a. Use Case Diagram	42
◦ i. Definition	42
◦ ii. Formalism	42
◦ iii. Components of a Use Case Diagram	43

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

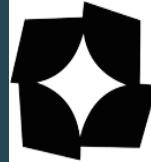
DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



○ iv. General Use Case Diagram	45
○ v. Client Manage Cart Specific Use Case Diagram	46
○ vi. Shop manager Manage Order Specific Use Case Diagram	47
• b. Communication Diagram	53
○ i. Definition	53
○ ii. Formalism	53
○ iii. Components of a Communication Diagram	56
• c. Sequence Diagram	58
○ i. Definition	58
○ ii. Formalism	58
○ iii. Components of a Sequence Diagram	59
• d. Activity Diagram	63
○ i. Definition	63
○ ii. Formalism	63
○ iii. Components of an Activity Diagram	64
CHAPTER 4: CONCEPTION PHASE	71
• Preamble	72
• Introduction	73
A. Capture of Technical Needs	74
• I. Class Diagram	77
○ i. Definition	77
○ ii. Formalism	77
○ iii. Components of a Class Diagram	78
• II. State Machine Diagram	81
○ i. Definition	81
○ ii. Formalism	81

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025

DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



○ iii. Components of a State Machine Diagram	82
● III. Package Diagram	84
○ i. Definition	84
○ iii. Components of a Package Diagram	85
○ iv. System Package Diagram	86
● Conclusion	87
 CHAPTER 5: REALIZATION PHASE	88
● Preamble	89
● Introduction	90
● I. Deployment Diagram	91
○ i. Definition	91
○ ii. Formalism	91
○ iii. Components of a Deployment Diagram	92
● II. Component Diagram	94
○ i. Definition	94
○ ii. Formalism	94
○ iii. Components of a Component Diagram	95
○ iv. System Component Diagram	96
● Conclusion	97
 CHAPTER 6: FUNCTIONALITY TESTING	98
● Preamble	99
● Introduction	100
● I. Application Functionalities	101
● II. Tests Showcases	102
○ i. Integration Testing	103
■ API ↔ Database Interaction	103

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025



DEVELOPMENT OF A WEB BASED INVENTORY AND SALES MANAGEMENT SYSTEM FOR A LOCAL ELECTRONICS VENDOR



• Conclusion	107
CHAPTER 7: USER GUIDE	108
• Preamble	109
• Introduction	110
• I. Description of Application	111
○ i. Purpose	111
○ ii. Target Audience	111
• II. Installation Guide	112
○ a. Database Setup with MongoDB Atlas	142
• III. User Manual	119
• Conclusion	123
GENERAL CONCLUSION	124
ANNEX	XIV
BIBLIOGRAPHY	XV
WEBOGRAPHY	XVI
TABLE OF CONTENTS	XVII

Written by: BALEMBA JESSY NJEA MASSOMA, 3rd year student, Software Engineering at AICS Cameroon Paul Biya Technological centre of Excellence, Academic year 2024-2025