DECLARATION

I MELI ANABELLE GRACE hereby declare that, this work entitle "**DESIGN AND IMPLIMENTATION OF AN INVENTRORY MANAGEMENT SYSTEM (IMS)**" was originally carried out by me. It is an original piece of work, which has not been defended in any HND, HPD, Degree award anywhere, and all borrowed ideas have been acknowledge by means of reference.

NAME: MELI ANABELLE GRACE

Signature________________________________

Date: __________________________________

CERTIFICATION

This is to certify that the project titled “**DESIGN AND IMPLIMENTATION AN INVENTORY MANAGEMENT SYSTEM**” with case of PROMACAM is a record of independent research work done by MELI ANABELLE GRACE, under the supervision of MR. FONILECK KINGSLEY CHE and submitted to CITEC-HITM Yaoundé, for the award of an HND in Computer Software Engineering.

SUPERVISOR: MR. FONILECK KINGSLEY CHE

SIGNATURE OF THE SUPEREVISOR: ___________________________________

Date: ________________________________

DEDICATION

I dedicate this report to my family, the Meli's

ACKNOWLEGEMENT

Special thanks to my supervisor of MR. FONILECK KINGSLEY CHE and the entire staff of CITEC HITM for this laudable initiative whose encouragement and relentless assistance, supervision and guidance were immeasurable in making this project a success. Big thanks goes to my field supervisor and the entire staff of PROMACAM for entrusting me in his enterprise to carry out this projects work.

My sincere gratitude is due to my dear Parents, MR and MRS MELI who helped me succeed in this course by providing both material and moral support. I value my brother's cooperation, who morally supported me in order to ensure that I achieved my goal.

Finally, I would want to convey my sincere thankfulness to God for enabling me to complete this piece of work by His hand of grace.

ABSTRACT

This study presents the design and implementation of an inventory management system using VB.net for PROMACAM, a company that sells laboratory equipment and reagents. The system is developed to help the company automate their inventory management processes, track inventory levels, and streamline their operations. The system uses a database to store product information and transaction data, and a graphical user interface (GUI) to allow users to interact with the system. The inventory management system is designed to support various functions such as adding new products, updating inventory levels, generating reports, and setting reorder points. The system also includes security features to ensure that only authorized personnel can access the system. The implementation of the system has been tested, and it has been found to be effective in improving the inventory management processes of PROMACAM. The system has the potential to reduce costs, improve efficiency, and increase productivity for the company. It equally presents a literature review on the topic of inventory management, and provides theorical, conceptual and empirical review of the study. We continue with the methodology used in the study, including the method used in designing the inventory management system and the systems requirements and design specifications. Furthermore, we shall have the results of the study. That is, a presentation of the designed inventory management system. This includes the features and the functions. We end with the summary and findings of the study. The strength and weaknesses of the inventory management system designed and the recommendations for future improvements.

Key words:
Inventory system
Design of an inventory system
Implementation of an inventory system.

RESUME

Cette étude présente la conception et la mise en œuvre d'un système de gestion des stocks utilisant VB.net pour PROMACAM, une société qui vend du matériel et des réactifs de laboratoire. Le système est développé pour aider l'entreprise à automatiser ses processus de gestion des stocks, à suivre les niveaux de stocks et à rationaliser ses opérations. Le système utilise une base de données pour stocker les informations sur les produits et les données de transaction, et une interface utilisateur graphique (GUI) pour permettre aux utilisateurs d'interagir avec le système. Le système de gestion des stocks est conçu pour prendre en charge diverses fonctions telles que l'ajout de nouveaux produits, la mise à jour des niveaux de stocks, la génération de rapports et la définition de points de commande. Le système comprend également des fonctions de sécurité qui garantissent que seul le personnel autorisé peut accéder au système. La mise en œuvre du système a été testée et s'est avérée efficace pour améliorer les processus de gestion des stocks de PROMACAM. Le système a le potentiel de réduire les coûts, d'améliorer l'efficacité et d'augmenter la productivité de l'entreprise. Il présente également une analyse documentaire sur le thème de la gestion des stocks et fournit une revue théorique, conceptuel et empirique de l'étude. Nous poursuivons la méthodologie utilisée dans l'étude, y compris la méthode utilisée pour concevoir le système de gestion des stocks, les exigences des systèmes et les spécifications de conception. En outre, nous présenterons les résultats de l'étude. Il s'agit d'une présentation du système de gestion des stocks conçu. Cela comprend les caractéristiques et les fonctions. Nous terminerons par le résumé et les conclusions de l'étude. Les forces et les faiblesses du système de gestion des stocks conçu les recommandations pour les améliorations futures.

Mots clés :
Système d'inventaire
Conception d'un système d'inventaire
Mise en œuvre d'un système d'inventaire.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

CDM:	Conceptual Data Model
DBMS:	Database Management System
ICT:	Information and Communication Technology
IS:	Information System
LDM:	Logical Data Model
OMT:	Object Modeling Technique
PDM:	Physical Data Model
SADT:	Structured Analysis and Design Technique
UML:	Unified Modeling Language
UP:	Unified Process
IMS :	Inventory Management System
FR :	Functional Requirement
VB :	Visual Basic
HDD :	Hard Disk Drive
RAM :	Random Access Memory
GHz :	GigaHertz
SQL :	Structured Query Language
IT :	Information Technology
PHP :	Hypertext Preprocessor
OS :	Operating System
SDLC :	Software Development Life Cycle
OTM :	Operation and Technology Management
CPU :	Central Processing Unit
OOM:	Object Oriented Modeling

CHAPTER ONE

INTRODUCTION AND PRESENTATION OF ENTREPRISE

INTRODUCTION

In many firms, the most difficult and complex tasks fall to the IT staff. Some of these positions and duties are crucial to the business's continued existence, profitability, and possibly even its long-term survival. Because we are in a digital world and use the internet for a variety of activities. My project aims at managing the products, sales of the products and the inventory for a company who deals with the selling of laboratory equipment and reactants.

This inventory management system helps companies identify which and how much stock to order at what time. It equally gives the company a complete picture of the inventory and lets you use real-time data to follow the whereabouts and movement of any item. It can considerably eliminate mistakes and inconsistencies in the company's records and orders. Manual spreadsheet is an inventory tracking approach that is difficult to maintain and subject to human mistake.

An inventory management system is a software application that helps businesses track and manage their inventory. Here are some advantages of using an inventory management system:

Increased efficiency: An inventory management system can help businesses increase their efficiency by automating tasks such as inventory tracking, order fulfillment, and reordering.

Improved accuracy: By automating inventory tracking, businesses can reduce the likelihood of human error and improve the accuracy of their inventory records.

Better inventory control: An inventory management system provides businesses with realtime information about their inventory levels, allowing them to make informed decisions about when to order more products and how much to order.

Cost savings: By optimizing inventory levels, businesses can avoid overstocking and understocking, which can result in excess inventory costs or lost sales.

Improved customer satisfaction: With an inventory management system, businesses can ensure that they have the products their customers want in stock, leading to improved customer satisfaction.

Enhanced data analysis: An inventory management system can provide businesses with detailed data about their inventory, such as sales trends, inventory turnover, and product profitability, which can help them, make informed business decisions.

Overall, an inventory management system can help businesses operate more efficiently, reduce costs, and improve customer satisfaction.

Inventory is always dynamic. Inventory management requires constant and careful evaluation of external and internal factors and control through planning and review. Most of the organizations have a separate department or job function called inventory planners who continuously monitor, control and review inventory.

BACKGROUND OF THE STUDY

The historical background, theoretical background, conceptual background, and contextual background of the research are all included in the background.

1.1.1 HISTORICAL BACKGROUND

The historical background covers the background of the study.

1.1.1.1 HISTORICAL BACKGROUND OF THE STUDY.

The concept of inventory management dates back thousands of years to the time when humans first started trading goods. In ancient times, traders used various methods to keep track of their inventory, such as tallying goods on a piece of paper, using clay tablets, or carving marks on wood.

The Industrial Revolution of the 18th and 19th centuries brought about significant changes in inventory management practices. The development of assembly line production methods, mass production, and the emergence of modern manufacturing industries led to the need for more advanced inventory management systems.

In the early 20th century, American engineer Harrington Emerson developed the concept of scientific management, which focused on the efficient use of resources, including inventory. This led to the development of new inventory management techniques, such as the Economic Order Quantity (EOQ) model, which calculates the optimal order quantity of inventory.

In the mid-20th century, the development of computers and information technology led to the emergence of modern inventory management systems. In the 1960s and 1970s, businesses started using mainframe computers to manage their inventory, which provided real-time inventory tracking and control.

In the 1980s and 1990s, inventory management systems became more sophisticated, with the development of barcode technology, which allowed businesses to track inventory more accurately and efficiently. The widespread use of personal computers and the internet in the 1990s and 2000s led to the development of cloud-based inventory management systems, which allow businesses to manage their inventory from anywhere in the world.

Today, inventory management systems are an essential tool for businesses of all sizes, helping them optimize inventory levels, reduce costs, and improve customer satisfaction.

CONCEPTUAL BACKGROUND

The concept of inventory management is closely related to the concept of supply chain management, which is the process of managing the flow of goods and services from suppliers to customers. An inventory management system is an important component of a supply chain management system, as it helps businesses manage their inventory levels and ensure that they have the products their customers need in stock.

An inventory management system also provides businesses with real-time information about their inventory levels, allowing them to make informed decisions about when to order more products and how much to order. This information can help businesses avoid stock outs, reduce inventory-carrying costs, and improve customer satisfaction.

Overall, the conceptual background of an inventory management system is based on the idea that managing inventory levels is an important part of managing a business, and that using technology to track and manage inventory can help businesses operate more efficiently, reduce costs, and improve customer satisfaction.

Here are some terms related to inventory and their explanations:

Stock: Stock refers to shares of ownership in a company that are publicly traded on a stock exchange. When a company goes public, it issues shares of stock to investors in exchange for capital that can be used to fund operations and growth. The value of a company's stock is determined by supply and demand in the market, and can fluctuate based on a range of factors, including company performance, market conditions, and investor sentiment.

Sales: Sales refer to the exchange of goods or services for money or other consideration, such as credit or barter. In a business context, sales typically involve the exchange of products or services between a seller and a buyer. The process of sales typically involves a series of steps, including prospecting, lead generation, qualifying potential customers, making sales presentations, negotiating terms, and closing deals.

THEORITICAL BACKGROUND

Inventory management is the process of overseeing and controlling the flow of goods and materials into and out of a business. An inventory management system is a software-based tool that helps businesses track and manage their inventory more effectively. The theoretical background of inventory management is based on several key concepts and principles.

The cost of carrying inventory: One of the main concepts in inventory management is the cost of carrying inventory. This includes the cost of storing and maintaining inventory, as well as the opportunity cost of tying up capital in inventory that could be used for other purposes. The goal of inventory management is to balance the cost of carrying inventory with the cost of stock outs, or the cost of not having enough inventory to meet customer demand.

Demand forecasting: Another key concept in inventory management is demand forecasting. Businesses use historical data and other factors such as seasonality, trends, and market conditions to forecast demand for their products. This information is used to determine how much inventory to order and when to order it.

CONTEXTUAL BACKGROUND

An inventory management system is based on the idea of tracking and controlling the flow of goods and materials in a business. The system is designed to provide real-time visibility into inventory levels, location, and movement. It is based on a centralized database that stores information about inventory items, including their description, quantity, location, cost, and other relevant details.

The system is typically based on a software application that automates the inventory management process, including tasks such as ordering, receiving, and stocking inventory. It can also help businesses track inventory movement and monitor inventory levels to avoid stock outs or overstocking.

1.2 PROBLEM STATEMENT.

During my internship period at PROMACAM, I realized the employees retained the inventory data in the logbook and it was not properly organized; the general director found it challenging to record the data promptly and safely. Because of their chaotic structure, the corporation had trouble estimating its profits, which is a concern. I decided to design an inventory management system that will allow the business to easily, swiftly, and more securely handle its inventory data.

1.3 OBJECTIVES OF THE STUDY

The objective of this study is divided into two types, the main and the specific objectives.

1.3.1 MAIN OBJECTIVES

The goal of this project is to design an inventory management system to manage the products of PROMACAM.

1.3.2 SPECIFIC OBJECTIVES

The specific objectives of this internship study are:

- To study the various requirements for an inventory management system.
- Classify objects based on their attributes and actions □ Using UML to model the system.
- Using MySQL to develop the database.
- Design the interface using Vb.net.
- Connect the database to the interface

RESEARCH QUESTIONS

The research question is divided into two, the general research question and the specific research question.

1.4.1 GENERAL RESEARCH QUESTION

Will the implementation of an inventory management system have a favorable effect on the sales of the company's products?

1.4.2 SPECIFIC RESEARCH QUESTIONS

Companies that deal with sales have a lot of problem when it comes to handling their inventory. Like regulating what goods arrive, when they arrive, and from which invoice or lot number.

- Will an inventory management system help the company identify which and how much stock to order at what time?
- Can the system help the company track incoming supplies, produced items, damaged goods, sold products and shipped items?

HYPOTHESIS OF THE STUDY

The hypothesis is divided into general and specific.

GENERAL HYPOTHESIS

Putting in place an inventory management system will go a long way to help PROMACAM in manage its inventory.

SPECIFIC HYPOTHESIS

- Users will log into the system to prevent overstock or excess inventory while ensuring they have sufficient supplies of commodities or materials to meet demand
- Users will log into system to easily know the various customers commands.

SIGNIFICANCE OF THE STUDY

This system will help the company in the different aspects:

Improved Efficiency: An inventory management system can help businesses operate more efficiently by automating time-consuming manual tasks, such as counting inventory and tracking items. This can free up time for employees to focus on other areas of the business, such as sales and customer service.

Better Customer Service: An inventory management system can help businesses ensure that they have the right products in stock to meet customer demand. This can help improve customer satisfaction and loyalty, which can lead to increased sales and revenue.

Improved Decision Making: An inventory management system can provide real-time visibility into inventory levels, sales trends, and other important data. This can help businesses make more informed decisions about ordering, pricing, and promotions.

Overall, an inventory management system will help the businesses operate more efficiently, reduce costs, improve customer service, make better decisions, and scale effectively.

JUSTIFICATION OF THE STUDY

This study aims to investigate the impact of implementing an inventory management system on the efficiency and profitability of a small retail business. The study will collect data on inventory levels, order processing times, customer satisfaction, and profitability before and after implementing the system.

DELIMITATION OF THE STUDY

The scope of a study provides a clear and concise overview of the research project and helps to ensure that the study is focused and relevant. It also helps to guide the research process and ensure that the research objectives are met.

THERMATIC SCOPE

The goal of this project, DESIGN AND IMPLEMENTATION OF AN INVENTORY MANAGEMENT SYSTEM, case study PROMACAM, is to assist the company in managing their sales and products.

GEOGRAPHICAL SCOPE

The internship place is located at Total melen, Yaounde Center region.

TIME SCOPE

The internship period took place within two months from June 10, 2022, until August 13, 2022. The idea of this project came to me when a worker complained of being tired of recording the stock manually on a book.

ORGANIZATION OF THE STUDY

This study is organized in five chapters;

- Chapter one provides and introduction to the study and discusses the scope of the study.
- Chapter two that presents the literature of the study.
- Chapter three, we have details of the methodology used in the study.
- Chapter four shows the result of the study, we see the different interfaces of the designed system.
- Chapter five, we have a summary, findings and recommendations on our study.

CHAPTER TWO

LITERATURE REVIEW

INTRODUCTION

A literature review is a critical analysis of existing literature on a particular topic or research question. This chapter provides presentations of the internship activities as well as examining the review by theories, reviews by concepts and review by objective.

2.1 THEORICAL REVIEW

The theories that were before mentioned in chapter one are reviewed in this section of the study. This study's theories include the following:

2.1.1 THE THEORY OF CONSTRAINTS (TOC)

The Theory of Constraints (TOC) is a management philosophy developed by Eliyahu Goldratt that aims to improve organizational performance by identifying and resolving bottlenecks, or constraints, in the production process. In the context of inventory management, the TOC suggests that organizations should focus on identifying and addressing the key constraints that are preventing them from achieving their inventory-related goals.

According to the TOC, the goal of inventory management is to balance inventory levels with customer demand, while minimizing costs and maximizing throughput. The theory suggests that there are three types of inventory that organizations need to manage:

Raw materials: Inventory that is used to produce finished goods.

Work in progress (WIP): Inventory that is being processed or worked on.

Finished goods: Inventory that is ready for sale or distribution.

The TOC suggests that organizations should focus on identifying and resolving constraints in the production process, rather than trying to optimize inventory levels. This involves using a five-step process:

Identify the constraint: The first step is to identify the bottleneck or constraint that is limiting throughput.

Exploit the constraint: The next step is to optimize the use of the constraint, by ensuring that it is working at full capacity.

Subordinate non-constraints: The third step is to ensure that non-constraints are working in a way that supports the constraint.

Elevate the constraint: If the constraint cannot be fully exploited, the next step is to invest in additional resources or equipment to increase capacity.

Repeat the process: Once the constraint has been addressed, the process should be repeated to identify and resolve any new constraints that arise.

Overall, the TOC suggests that organizations should focus on optimizing their production process by identifying and addressing constraints, rather than trying to optimize inventory levels. By doing so, they can improve throughput, reduce costs, and improve overall organizational performance.

2.2 CONCEPTUAL REVIEW

Overall, this conceptual review will provide a comprehensive overview of the key concepts and theories related to inventory management systems, and will highlight the importance of effective inventory management in achieving organizational goals. We shall define various component of the study:

Inventory: Inventory refers to the stock of goods, raw materials, or finished products that a business holds at a particular point in time. It includes all the items that are required for the production of goods or services, as well as the finished products that are ready for sale or distribution.

Stock: Stock refers to shares of ownership in a company that are publicly traded on a stock exchange. When a company goes public, it issues shares of stock to investors in exchange for capital that can be used to fund operations and growth. The value of a company's stock is determined by supply and demand in the market, and can fluctuate based on a range of factors, including company performance, market conditions, and investor sentiment.

Sales: Sales refer to the exchange of goods or services for money or other consideration, such as credit or barter. In a business context, sales typically involve the exchange of products or services between a seller and a buyer. The process of sales typically involves a series of steps, including prospecting, lead generation, qualifying potential customers, making sales presentations, negotiating terms, and closing deals.

Operating system: An operating system (OS) is a software that manages computer hardware and software resources and provides common services for computer programs. It acts as an interface between computer applications and the hardware components of a computer system. The operating system performs a range of functions, including managing memory, running applications, providing input/output (I/O) services, managing network connections, and controlling peripheral devices such as printers and scanners.

Information system: An information system (IS) refers to a combination of hardware, software, data, people, and processes that are used to collect, store, process, and distribute information within an organization. Information systems can be used to support a wide range of business functions, including operations, management, and decision-making.

Database: A database is an organized collection of data that is stored electronically on a computer system. It is designed to facilitate the storage, retrieval, and management of large volumes of information in a structured and efficient manner. Databases can store a wide range of information, including customer records, financial transactions, product inventories, employee data, and more.

2.3 EMPIRICAL REVIEW

This empirical review aims to analyze and evaluate existing research studies related to inventory management systems. Specifically, this review will examine the design and implementation of inventory management systems, their impact on business performance, and the challenges and opportunities associated with these systems. By synthesizing existing research, this review will provide insights into best practices for inventory management system design and implementation, as well as identify areas for future research and development. Below are some reviews on inventory management:

"Impact of Inventory Management Practices on Financial Performance of Sugar Manufacturing Firms in Kenya" by John N. Kihoro and Andrew K. Langat (2017). This study analyzed the relationship between inventory management practices and financial performance in the sugar manufacturing industry in Kenya.

"Assessment of Inventory Management Practice: A Case Study of Jimma University Main Campus Store" by Mengistu Ferede (2019). This study evaluated the inventory management practices of a university campus store in Ethiopia and identified areas for improvement.

"The Impact of Inventory Management Practices on Financial Performance in Food and Beverage Industry in Malaysia" by Nur Azlina Mohd Azmi and Norazah Mohd Suki (2017). This study analyzed the impact of inventory management practices on the financial performance of food and beverage companies in Malaysia.

"Design and Implementation of a Web-Based Inventory Management System for a Retail Business" by Sarita N. Wadhwani and Lijuan Wu (2018). This study described the design and implementation of a web-based inventory management system for a retail business in the United States.

"The Impact of Inventory Management on Supply Chain Performance" by Yasser M. AlSafadi, Rizwan Ullah, and Abdullah Al-Mudimigh (2018). This study analyzed the impact of inventory management on supply chain performance in the manufacturing industry in Saudi Arabia.

These studies demonstrate the diverse range of research that has been conducted on inventory management systems, including their impact on financial performance, supply chain performance, and specific industries or businesses.

2.4 PRESENTATION OF THE ENTERPRISE (INTERSHIP)

The description of the internship site and the internship activities are included in this section of the study.

2.4.1 PRESENTATION OF THE INTERSHIP

PROMACAM is located at Total melen opposite the petrol station Total.

2.4.1.1 HISTORICAL BACKGROUND OF THE INTERNSHIP PLACE.

The successful history of the CEO of PROMACAM really inspired me and it tells me we can start from the bottom and with time reach at the top. Proclaimed a company under Cameroonian law, which sells reagents, equipment, and consumables for medical analysis in laboratories created in September 1992. Is in 1992 that the adventure of the young CEO begins. The CEO has always be caring for the wellbeing of people that is why he decided to engage in the selling of laboratory equipment for good treatment of patients. This adventure does not begin directly with a big company. Like Jeff Bezos, he began in little activities and the combination of all these activities leaded to the birth of PROMACAM. According to the CEO PROMACAM like all beginning companies was difficult to sustain at the beginning because the business environment is not always favorable (corruption, administrative slowness…) but the CEO did not get discourage. According to him, we always have to struggle and reach our objectives.

PROMACAM first years started when the CEO had his bachelor degree. At the end of the university with the Bachelor degree in Economics and business management, he received a sum of 150,000 FRS and he decide to go to Baffousam to buy some capsule for upcoming sales in Yaoundé. After realizing that those selling in Baffousam were getting bigger because they were going to Nigeria to buy their merchandise as wholesalers. He decide to take his capital

150,000frs and the profit he had made so far which gave about 450,000frs and went to Nigeria to buy his own merchandise as a wholesaler to.

After which he came back to Yaoundé and started selling his goods at prices wholesalers of Baffousam would sell theirs, which was now beneficial to him and those buying from him in Yaounde because they were gaining in cost for housing and transport to Baffousam and he was selling at the price wholesalers in Baffousam were selling. After some time he bought a container that he placed in front of the CHU hospital at Total Melen and as the CHU did not function, very well the nurses would come to buy some merchandise from him (gloves, test tubes, syringes...). Then he discovered they were some products the nurses needed but he never had then. Therefore, he made a list of those products and went to Nigeria to buy them.

From there he had knowledge of infusion set and other laboratories equipment's. After buying all these new products, he started associating with other hospitals in Cameroon where he did great sales. Sometime after in 2002, he moved from his container to a more comfortable office where from there he employed some workers because the work some becoming hard for one man. Later the he entered in contact with the German company HUMAN this company is one specialized in the production, constructing, and distribution of laboratory equipment worldwide. Being a serious worker HUMAN made PROMACAM the official distributor of the HUMAN products in Cameroon.

After this, he finally moved to a bigger office, employed more workers, and eventually associated with more hospitals and medical analysis laboratories. His contract with HUMAN made him to be known by other big company, which led to his contract with the French company MEDIFF. From here PROMACAM, aims at producing his own equipment, which would be, used in hospital all this for the better and proper care of patient at hospitals.

2.4.2 ACTIVITIES CARRIED OUT DURING THE INTERNSHIP

WEEK	ACTIVITIES
Week 1 and 2	• We had a complete visit of the company • We were presented to the various department in the company • We were equally presented to the various heads of the different departments

Week 3 and 4	• We attended several meetings both general or departmental meetings • We noted the minutes of the different meetings • We had a briefing on the company (its activities, its functions etc.)
Week 4 and 5	• We were separated to the various departments where we were to work • We were presented to our head of department • We were given the different functions of our department in the company
Week 6,7 and 8	• We were given our different charges and what we were to do what explained in detail. • We started working

2.4.3 THE INTERNSHIP EXPERIENCE

Even though, PROMACAM is not an IT based company. I learned to develop a lot skills both like wise to my specialty and equally like wise to other specialties.

- I learned more about the visual basic programming language through researches.
- Through the historical background of the company and learned how someone can begin with a startup and end up being a great company.
- I equally learned the use of databases and it would be very important to application.
- I learned how to use the database administration tool called PHPMYADMIN.

2.4.4 STRENGTH AND WEAKNESS

This part of the work contains the weakness of the internship place.

2.4.4.1 THE STRENGTH

Working at PROMACAM was a very good experience for me. Since the company deals in the selling of laboratory equipment and reagents, I was very fascinated by the world of medicine. The company delivers equipment to several hospitals in the country and is very successful in its domain.

2.4.4.2 THE WEAKNESSES

The company has very good worker in the various departments but they lack an IT department. Because during my internship I notice that there was no one, there IT inclined. Whenever they had a problem that concerned IT, they will always have to call an expert who will take several time before coming and that is very inconvenient to them.

2.4.5 PROBLEMS ENCOUNTERED

Is true I learned several things for this internship. However, they were also some hard times I had to face.

- The stock room was very and extremely cold which caused me to catch a cold.
- Sometimes departments will delay in sending some documents we demanded which eventually delayed our work.

2.4.6 RECOMMENDATIONS

PROMACAM must decide who will be responsible for checking the stock levels in the working area. To make managing the inventory before entering the application easier, one person will be in charge of overseeing the bringing in and taking out of each item individually.

In addition, PROMACAM needs to establish a software development department to aid in solving IT issues that may be resolved through software for the community. I believe this would be advantageous for the company and would broaden their market reach.

CHAPTER THREE

METHODOLOGY AND MATERIAL USED

3.1 INTRODUCTION

The word "methodology" refers to a process for determining the best solutions to a problem. We use this process to accomplish our primary or targeted goals.

This chapter of our work discusses the research methods and materials to be employed in order to realize our application, which has been deemed the fundamental component of our work. This chapter focuses on the approach and materials we will use to create this application. As we use these resources effectively to create a user-friendly application, we will be led by user needs and observation.

3.2 DESCRIPTION OF THE ARCHITECTURE OF THE SYSTEM OR APPLICATION

The Visual Basic Application model from Visual Basic offers a clear mechanism for managing Windows Forms applications' behavior. This model consists of events that handle the startup and shutdown of the application as well as events that catch unhandled exceptions. Also, it offers assistance with the creation of single-instance apps. Because the application model can be extended, programmers that require additional control can alter it's over ridable methods.

When an application starts up and closes down, it must complete certain tasks. The application may, for instance, show a splash screen, connect to a database, load a previously saved state, and so forth when it first launches. Close database connections, save the current state, and other options are available when the application terminates. Moreover, the program can run particular code in the event of an unexpected shutdown, such as one brought on by an unhandled exception.

It is simple to develop a single-instance application using the Visual Basic Application paradigm. Only one instance of a program can be operating at once, which sets single-instance applications apart from standard applications. Whenever a single-instance program tries to launch a new instance, the first instance is alerted via the StartupNextInstance event that a new launch attempt was made. The command-line arguments for the succeeding instance are included in the notification. The next application instance is thus terminated before any initialization can take place.

This diagram shows how a subsequent instance signals the first instance:

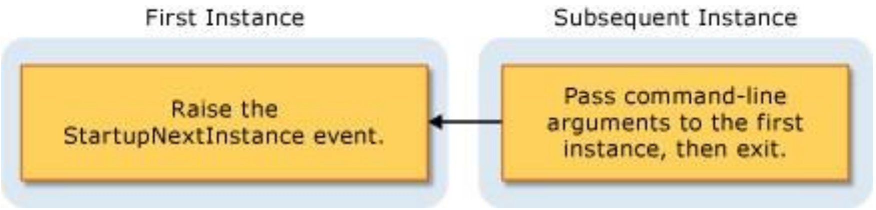

First Instance
Subsequent Instance
Raise the StartupNextInstance event.
Pass command-line arguments to the first instance, then exit.

Figure 1: subsequent instance signals the first instance

3.3 DATA COLLECTION METHOD AND USER’S NEED

A decent .NET application should consider the users' needs as well as the data collection process. Without researching the information system, a .NET application cannot be realized. Therefore, a good .NET application should address users' needs with the aid of the numerous data collection techniques implemented.

3.3.1 OBSERVATION

When working supplies and sales merchandise are delivered to PROMACAM, a specific person is in responsibility of verifying the inventory items that are currently in stock. Even routinely counting each material by hand leads to errors. It takes time since the business owner must review each inventory item every day before placing a new order for the inventory in order to keep track of the inventory. In addition, there is a lack of communication because staff members fail to acknowledge or record when an item has been taken out of the inventory.

The environment is unfriendly, the control mechanism takes a lot of time, is less precise and efficient. Finding specific information requires a lot of time and is a highly monotonous task. It is highly likely for competitors or dishonest people to access these files by employing a close associate of the Manager or through some other ways because the records and information about the customers, merchandise, personnel, and repairs controlled by the Manager are generally held in physical materials. Access to the records cannot be stopped once the files or documents containing the information are located. With the implementation of the proposed method, the tediousness of the manual system, which has been shown to be extremely onerous—can be decreased or completely removed.

3.3.2 USERS' NEEDS

Because a professional developer or programmer cannot create a web application without taking into account the needs of the consumers, a relevant .NET application should be built in accordance with those needs.

Below are the various user's needs I noticed and which he equally complained about;

- Manual recording of client information and product information is done in an unattractive book. As a result, the books are vulnerable to physical harm, information may be lost, and dust particles might gather.
- The manager finds it difficult to manually update the records in the documents as the number of products to be serviced increases.
- The atmosphere is unfriendly and the control system takes a long time. It is also less precise and efficient. Finding specific information requires time and is a very monotonous task.
- The actions of the manager could be severely hampered if information and records kept in files and documents were physically destroyed or stolen.

3.4 FUNCTIONAL REQUIREMENTS

Here, we will take a closer look at some of the fundamental computer parts that are required for our application to run smoothly and effectively for the target user base.

3.4.1 HARDWARE RESOURCES

For the implementation of our system, we used a computer with the following characteristics:

- **Operating System**: Windows 10 professional.
- **HDD**: 120 GB.
- **RAM**: 4GB.
- **Processor**: Intel Celeron 2.20 GHz.

3.4.2 SOFTWARE RESOURCES

A) .NET FRAMEWORK

It is a software development framework developed by Microsoft that runs primarily on Microsoft Windows. It is used to build apps for Linux, macOS, Windows, IOS, Android and more.

D) EDRAWMAX

It is a software engineering tool for system modelling using the Unified Modelling Language, as well as Systems Modelling Language, and classical modelling notations. It is published by MKLabs and is available on Windows, Linux and MacOS.

3.4.2.1 PROGRAMMING LANGUAGES USED AND IDE / CODE EDITOR

1) VISUAL STUDIO:

It is a multi-paradigm, object-oriented programming language, implemented on the .NET framework. Used to create console apps, web and desktop applications for the windows operating system.

2) SQL:

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

3.5 FUNCTIONAL SPECIFICATION

3.5.1 ROLE PLAYED BY EACH ACTOR IN THE SYSTEM

➢ MANAGER:

- **The manager performs staff administration**
  - He keeps track of each new employee's details.
  - He has the ability to edit or remove personnel information. **The manager performs customer administration**
  - Adding the information for each new client - Updating and erasing customer information - Querying any customer.

- **The manager performs inventory management -** Adding additional goods to the stock (list of items).
  - An item's inventory record can be updated or deleted.
- **The manager performs order management**
  - Adding an order to the order list.
  - Creating bills
  - Printing bills

➢ **STAFF:**

- **A staff performs customer administration like:**
  - Adding the information for each new client
  - Updating and erasing customer information
  - Querying any customer
- **A staff performs inventory management**
  - Adding additional goods to the stock (list of items).
  - An item's inventory record can be updated or deleted.

3.5.2 FUNCTIONALITIES OF THE SYSTEM

The criteria for the IMS are listed in this section. In order to accommodate the several user interactions that the system will have, the functional requirements that were gathered from the users have been categorized.

- **Login to the system**: Based on the user's login information (username and password) in the login section, the system will identify the user.
  - **FR01**: Only the designated users with a working login and password are permitted access to the IMS.
- **Product Management**: The user can view their allocated page after logging in and view, add, delete, and update information about objects after seeing it on the system.
  - **FR02**: The user shall be able to view items.
  - **FR03**: The user shall be able to delete items.
  - **FR04**: The user shall be able to update an item’s information. - **FR05**: The user shall be able to print inventory information.
- **Customer Management**: The user will be able to view customers, add customers, delete customers, and update a customer's details in the customer information section after logging into the system.

- **FR06**: The user shall be able to view customers.
- **FR07**: The user shall be able to add customers.
- **FR08**: The user shall be able to delete customers.
- **FR09**: The user shall be able to update a customer's information.

- **Category Management:** The user will be able to view categories, add categories, delete categories and update a category's details in the category information section after logging into the system.
  - **FR10**: The user shall be able to view categories.
  - **FR11**: The user shall be able to add categories.
  - **FR12**: The user shall be able to delete categories.
  - **FR13**: The user shall be able to update a category's information.
- **User Management**: The user will be able to view users, add users, delete users and update a user's details in the user information section after logging into the system. - **FR14**: The user shall be able to view users.
  - **FR15**: The user shall be able to add users.
  - **FR16**: The user shall be able to delete users.
  - **FR17**: The user shall be able to update a user's information.
- **Order Management:** The user will be able to view orders, add orders, delete orders, and update an order's details in the order information section and print orders after logging into the system.
  - **FR18**: The user shall be able to view orders.
  - **FR19**: The user shall be able to add orders.
  - **FR20**: The user shall be able to delete orders.
  - **FR22**: The user shall be able to update order's details.
  - **FR23**: The user shall be able to print an order.

3.6 TECHNICAL SPECIFICATIONS

- The user in the system should have a login page visible when the application is started.
- The user is led to a Home page with the identical selections after logging in.
- The user can go to the product page and add new products, delete products or update information on a product.

- The user can go to the customer page and add new customers, delete customers or update information on a customer.
- The user can go to the category page and add new categories, delete categories or update information on a category.
- The user can go to the user page and add new users, delete users or update information on a user.
- The user can go to the order page and add new orders, delete orders, update information on an order or print an order.

3.7 NON-TECHNICAL SPECIFICATIONS

From the time a customer arrives at PROMACAM to buy products until a manager or staff member login into the online application, everything must be managed perfectly (IMS). It is critical to understand the following:

- Security: the system has to be secured and should only be accessed by authorized people.
- User-friendly interface: the interface of the system should be user friendly.

3.8 RESEARCH DESIGN

Even though industry professionals run many laboratory equipment sellers, when they receive orders, they will require effective management strategies in order to satisfy all of their clients. Nonetheless, these companies are typically run manually. Most of the time, they overlook necessary orders. They consequently tend to decrease production, which leaves clients unsatisfied. Because management is done by hand. Hence, one notices:

- Difficulty managing or monitoring orders when there are many of them.
- A consistent physical insecurity of the data (taking the risk of an accident into account).
- Everyone has access to his or her information.

Thus, happily, the Research Design for this .NET application is coming to address such issues caused by the manual management of this company.

3.9 ANALYSIS METHODS

The scientific approach method is how the researcher is led to create a suitable .NET application, and it involves an analysis that results in the creation of a .NET application like

IMS. In order to create software that closely matches the observations gathered, this first includes gathering knowledge about the field through observation before moving on to the requirement for and information flow within a certain information system.

The object-oriented methods and the functional methods are the two different types of methods that we will systematic examine or present. By providing the justifications for our choice, we will specify the best way appropriate for the realization of our .NET application in the conclusion.

3.10 OBJECT ORIENTED METHODS

The static organization of the objects, their classes, and their relationships are described by object-oriented methods (OOMs). The OMT technique, UML method, and UP are some OOMs that can be mentioned here.

3.10.1 OMT METHOD

The object-modelling technique (OMT) is an approach to object modeling for creating and modeling software. Rumbaugh created it in the early 1990s as a technique for creating object-oriented systems and supporting object-oriented programming. OMT describes the system's object model or static structure. OMT was created as a method for creating software.
According to Rumbaugh, modeling serves the following purposes:

- Testing physical entities before building them (simulation).
- Interaction with consumers
- Visualization (alternative presentation of information).
- Simplificating complexity

3.10.2 UML METHOD

Unified Modeling Language (UML): A common modeling technique used to create all the different software system components from a variety of viewpoints. During the requirements and design phases of the software development life cycle, UML facilitates the identification of software use cases and the creation of models of software architectures (SDLC).

UML has a number of benefits, including:

- Standardized, formal language that promotes clarity and assures stability. This promotes the utilization of the instruments;

- Implementation of the entire object approach's richness;
- Description of all the models, from analysis to software manifestation; Standardization of the object ideas.

Some limits of UML are:

- The semantics of UML is not formalized. Natural language is used to specify it
- Challenging optimization of class selection;
- Several groups of diagrams lack formalization;

Additionally, it offers methods for creating software deployment diagrams that serve as guidance during the distribution stage. The representation of software components is based on diagrams. A picture is worth a thousand words, as the expression goes. We are able to comprehend potential defects or problems in software or business processes better by using visual representations. An organizational structure for UML diagrams is shown below.

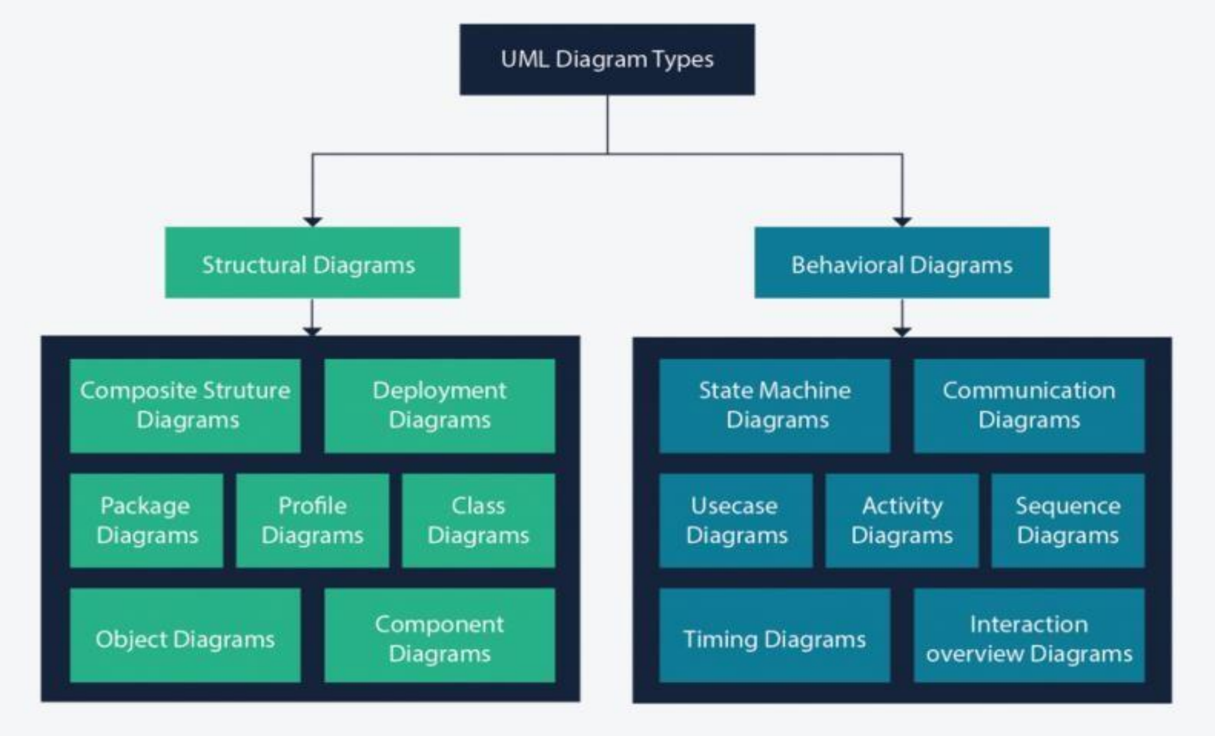


Figure 2:Types of UML diagrams

3.10.3 UP METHOD

Unified Process (UP) is a system process engineering metamodel-compliant, architecture-centric, use-case-driven, iterative, and incremental development process. It makes use of the unified modelling language. Different software systems with varying degrees of administrative and technical complexity can be used with unified process across a range of organizational cultures and domains.

3.11 FUNCTIONAL METHODS

The evolution of procedural languages is where the functional approaches first appeared. They highlight the functions that must be ensured and suggest a hierarchical, downward, and modular strategy by outlining the connections between the various modules, with a focus more on the managements than the data. These techniques incorporated the modelling of the data and the issues developing in real time as systems and programming languages evolved.

3.11.1 SADT METHOD

The Structured Analysis and Design Technique (SADT) consists of a technique for creating these descriptions as well as a graphical language for describing systems. A group of diagrams that explain a system from a specific point of view and with a specific objective make up a SADT model. The analysis and design of systems are guided and disciplined by SADT as a language and approach.

Some advantages of SADT Method are:

- How well it captures the demands of the user.
- Its ability to generate solutions at various levels of abstraction.

Some limits of SADT Method are:

- DE compositional rules are implicit. Analysts claim that the breakdown varies.
- The challenges it faces when attempting to account for non-hierarchical interactions in complex systems.
- Its analysis focuses primarily on the functions, ignoring the coherence of the data.

3.11.2 MERISE METHOD

The information system design and development approach known as MERISE (Méthode d'Etude et de Réalisation Informatique des Systèmes d'Entreprise) is widely utilized in France.

In MERISE, data and processes are treated separately, and the data-oriented view is modelled in three stages, from conceptual to logical to physical. Similar to this, the processoriented perspective goes through the conceptual, organizational, and operational stages. The stages of the life cycle—strategic planning, preliminary research, comprehensive research, development, implementation, and maintenance—are analogous to these steps in the modelling process. It is an analytical technique built on the entity-relationship model. A relational database can be created by utilizing MERISE to create tables with relationships.

The conceptual level, the organizational level, and the physical level are the three stages of abstraction that the Merise method suggests.

The conceptual level: The company's goals are defined at the conceptual level. This level is where the company's goals and burdensome restrictions are stated. It typically represents the beginning of development and the level that is the steadiest. The Conceptual Data Model (CDM) and the Conceptual Treatments Model can be distinguished conceptually (CTM).

The organizational level: The organizational level outlines the structure that should be established in the business to meet the established goals. Its objective is to offer a diagrammatic picture of the company's organizational structure. The Organizational Treatments Model (OTM) and the Logical Data Model (LDM) are both well known (OTM). The second degree of invariance is seen at the organizational level, which is less stable.

The physical level: The mechanisms that will be used to manage the data and activate the therapies are described at the physical level. It is structured around the Operational Treatments Model (OTM) and the Physical Data Model (PDM).

Some advantages of MERISE Method are:

- Merise enables the formalization and understanding of the trade's requirements.
- Merise encourages communication between creator and owner, focusing especially on integrated system development for management.
- Prior to beginning the design process, Merise makes sure that the user's needs are formally outlined within the parameters of a schedule of conditions.

Some disadvantages of Merise Method are:

- Difficulty in maintaining the system.
- Merise is more interested in generic design engineering than in software genius.

3.12 CHOICE OF METHOD

OMT, UML, UP, SADT, and MERISE are some of the key models that can be employed in application design, according to research on this topic. OMT has been selected as the methodology to be used in this effort to design our application and UML will be used as the language. Since UP makes use of UML notations, UML will automatically employ the UP technique. UML was chosen because it fuses the dynamic (behavioural) and static (structural) aspects of the system into one entity to produce positive and desirable outcomes. The static and dynamic entities become interdependent as a result. UML notations are the most important elements in modelling. Efficient and appropriate use of notations is very important for making a complete and meaningful model.

Moreover, it enhances the system's precision and stability. Therefore, employing the UML to MERISE approach to construct our application is quicker. On the other hand, the MERISE method distinguishes between the static and dynamic approaches. It uses data models to represent static systems and treatment models to represent dynamic systems. MERISE is a general approach rather than a method designed specifically for software development like UML, which makes the development of applications slower and more expensive because more materials are needed to achieve the same but less reliable result in quality and quantity.

3.13 APPLICATION OF METHOD

The software system artifacts can be specified, visualized, built, and documented using the pictorial standard and modeling technique known as UML. Therefore, without a shadow of a doubt, UML will aid in our better application realization and functional understanding.

3.13.1 ACTORS

An actor is an outside person or software agent who is close to the system and interacts with it by acting out a role. We have the following actors in our situation:

- ➢ **Main Staff (MANAGER):** He is any administrator of the repair shop who is in charge of the overall management of the day-to-day operations of the shop. He sets up every system to function well and manages access for the other actors (Staff) in CRSMS. He is capable of carrying out the following tasks:

 The manager performs customer administration like:

- Adding details of all new customers.
- Updating and deleting information of customers

 The manager performs product management

- Add new items into the product list
- Delete or update information about a product in the product list.

 The manager performs users management -

Add new users into the users list.

- Delete or update information about a user in the user list.

 The manager performs category management -

Add new categories in the category list.

- Delete or update information about a category in the category list.

 The manager performs order management -

Add new orders in the order list.

- Delete or update information about an order in the order

list. - Print orders

3.13.2 DIAGRAMS

3.13.2.1 SYSTEM LEVEL USE CASE DIAGRAM

Use case diagram is a dynamic or behavior diagram in UML. Use case diagrams illustrate the functionality of a system using actors and use cases. Use cases are a collection of tasks, offerings, and operations the system must handle. The use case of is shown in figure 2 below.

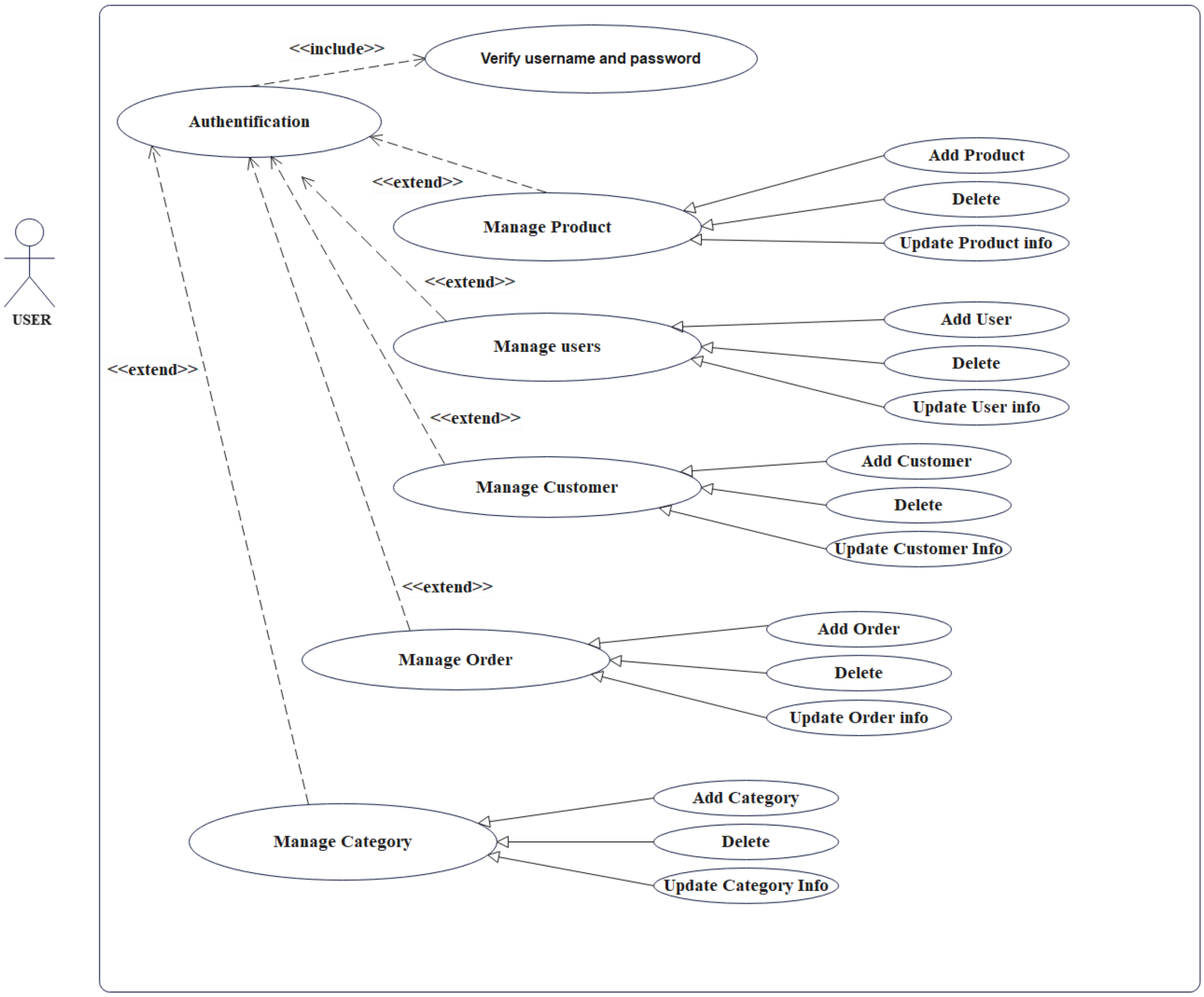


Figure 3:UML System Level Use Case Diagram

3.12.2.2 UML CLASS DIAGRAM

In systems engineering, a use case is a list of steps, typically defining interactions between a role (known in UML as an "actor") and a system, to achieve a goal. The actor can be a human or an external system. The use case diagram of the IMS is shown in figure 3 below.

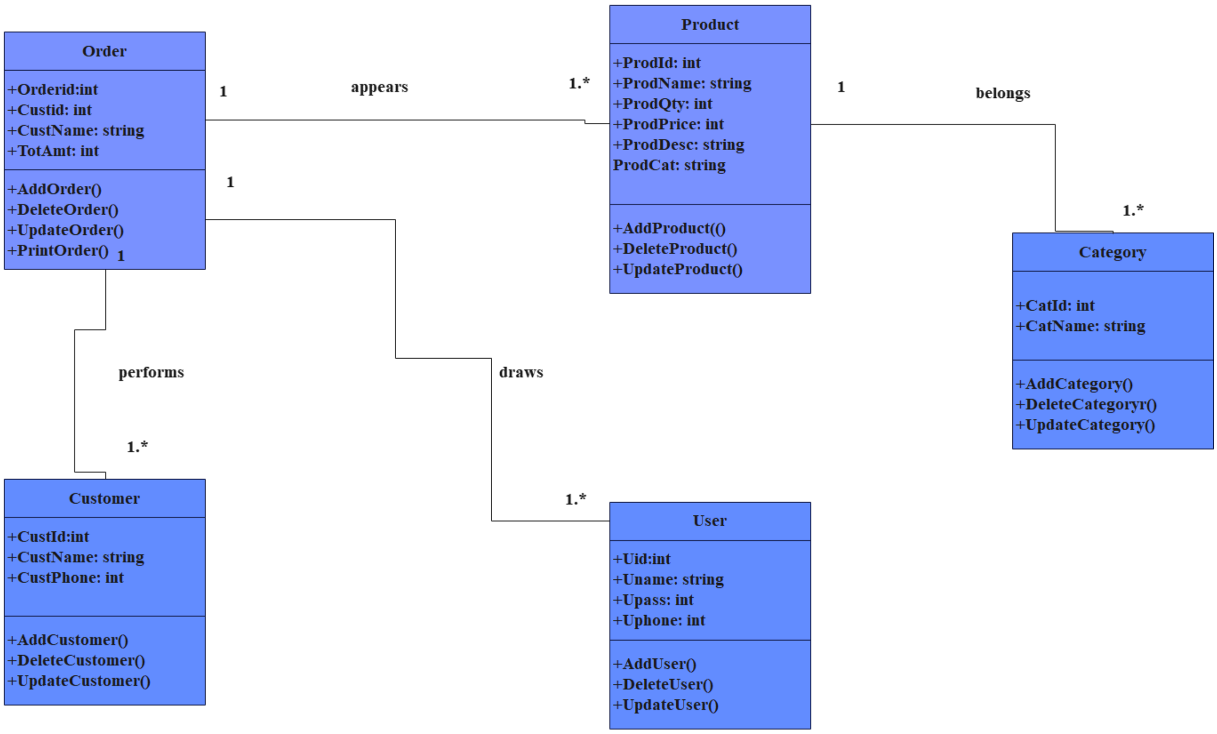


Figure 4:UML Class Diagram of IMS

3.12.2.3 UML ACTIVITY DIAGRAM

In software engineering, UML activity diagrams are a type of visual modeling language used to represent the flow of actions and activities within a system or process. They are often used in software development to illustrate the steps involved in completing a particular task or process. The activity diagram of the IMS is shown in figure 3 below.

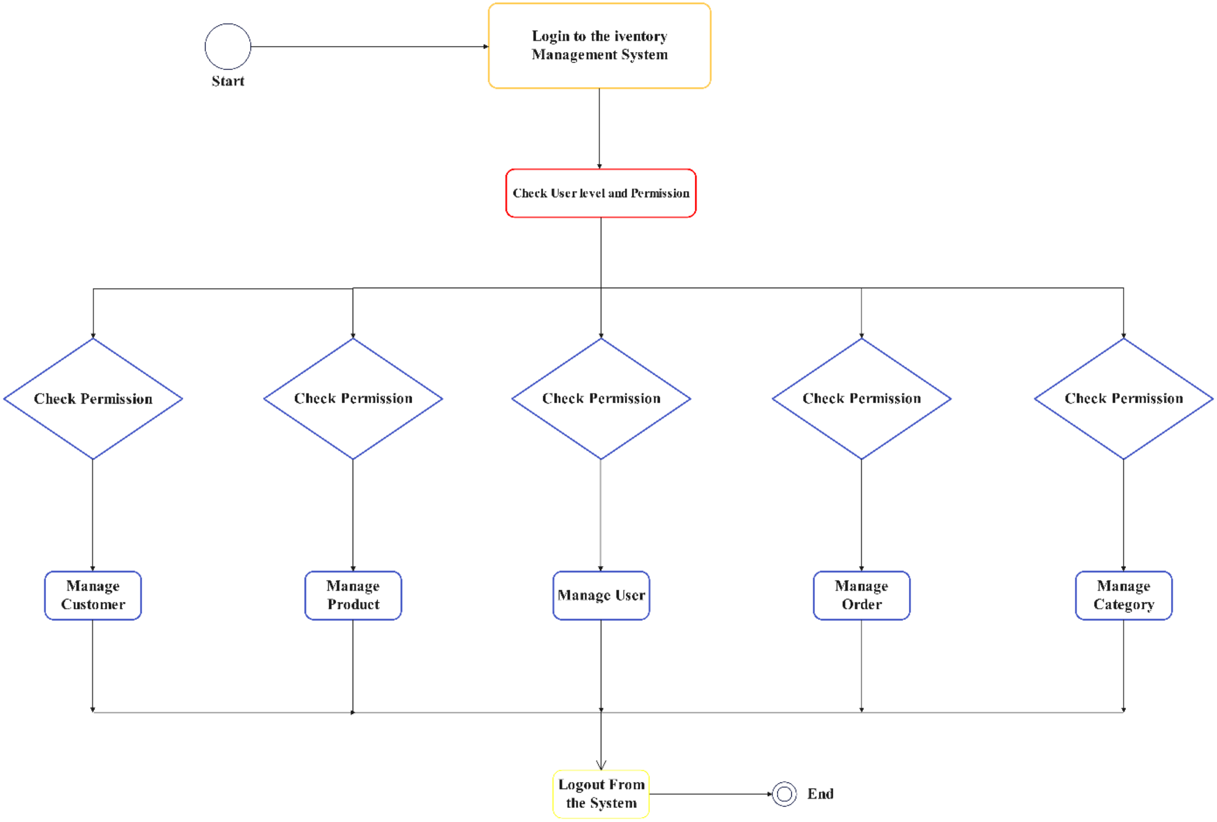


Figure 5:UML Activity Diagram

3.13.3 COMPONENTS OF SYSTEM OR APPLICATION

Our application will be developed around seven (07) components.

1. Login
2. Home
3. Product
4. Customer
5. User
6. Category
7. Order

3.13.3.1 LOGIN

In order to prevent unwanted access to the system, this module is used to log in.

3.13.3.2 HOME

This module serves to access the different forms in the application.

3.13.3.3 PRODUCT

This module list all the products, permits to add a new product, delete a product and update information about a product.

3.13.3.4 CUSTOMER

This module lists every customer and allows editing of a customer's details, adding a new customer, and deleting existing ones.

3.13.3.5 USER

This module lists every user and allows editing of a user's details, adding a new user, and deleting existing ones.

3.13.3.6 CATEGORY

This module lists every category and allows editing of a category's details, add a new category and delete existing ones.

3.13.3.7 ORDER

This module lists every order and allows editing of an order's details, adding a new order, delete existing ones and printing an order.

3.14 VARIOUS MODELS OF THE METHOD USED FOR DESIGN

The process of developing a data model for the information that will be kept in a database is known as data modeling. This data model is a conceptual representation of the relationships between various data objects, the rules, and the data objects themselves. Data modeling facilitates the visual representation of data and ensures that it complies with legal requirements, business regulations, and governmental directives. Data models guarantee data quality while guaranteeing uniformity in naming standards, default values, semantics, and security.

Instead of focusing on the operations that must be carried out on the data, data models place an emphasis on what data is required and how it should be organized. Similar to an architect's blueprint, a data model establishes the relationships between data objects and aids in the construction of a conceptual model.

3.14.1 DATA DICTIONARY

The data dictionary needs to be created after the various entities have been analyzed. It concerns a table in which each piece of information is listed precisely, including its N°, reference, relevance, kind, and size. All of these facts are the result of information that was retrieved and information that was gathered during the interview. These facts are illustrated in the dictionary below.

N0	Reference	Significance	Type	Size
1	ProdId	Product Id	int	3
2	ProdName	Product Name	varchar	30
3	ProdQty	Product Quantity	int	3
4	ProdPrice	Product Price	int	7
5	ProdDesc	Product Description	varchar	30
6	ProdCat	Product Category	varchar	25
7	CustId	Customer Id	int	3
8	CustName	Customer Name	varchar	30
9	CustPhone	Customer Phone	int	9
10	CatId	Category Id	int	3
11	CatName	Category Name	varchar	25
12	UId	User Id	int	3
13	Uname	User Name	varchar	30
14	Upass	User Password	int	10
15	Uphone	User Phone	int	9
16	OrderId	Order Id	int	3
17	TotAmt	Total Amount	int	9

Table2: Data Dictionary

CONCEPTUAL DATA MODEL (CDM)

The system's contents are defined by this data model. Business stakeholders and data architects are usually the ones who develop this model. Organization, definition, and scope setting of business principles are the goals.

Establishing the entities, their qualities, and their relationships is the major goal of this model. There is rarely any information accessible about the actual database structure at this level of data modeling. The three fundamental tenets of data model are:

Entity: A real-world thing.

Attribute Characteristics or properties of an entity.

Relationship: Dependency or association between two entities.

Characteristics of a conceptual data model

- Provides coverage of business principles across the entire organization.
- The conceptual model is created without regard to hardware requirements like location or software requirements like DBMS vendor and technology or hardware specifications like data storage capacity. Data representation with a user's perspective in the "real world" is the main goal.
- These data models are created and constructed with a corporate audience in mind.

Conceptual data models, sometimes referred to as domain models, define fundamental concepts and scope for all stakeholders, resulting in the creation of a shared lexicon.

LOGICAL DATA MODEL (LDM)

Regardless of the DBMS, LDM specifies how the system should be implemented. Business analysts and data architects frequently develop this model. Creating a technical map of the rules and data structures is the goal. Logical data models further enhance the components of the conceptual model. It establishes the links between the data items and specifies the structure of the data elements.

The benefit of the logical data model is that it acts as a foundation for the physical model. The modeling structure, however, is still general. There are no defined primary or secondary keys at this level of data modeling. You must check and modify the connector details that were previously established for relationships at this level of data modeling.

Characteristics of a Logical data model

- Independently created and developed from the DBMS.
- Normalization processes to the model is applied typically till 3NF.
- Specifies the data requirements for a specific project, but depending on its breadth, it might also interface with other logical data models.

3.14.2 RULES TO MOVE FROM CDM TO DATA LDM

There are several processes involved in translating high-level notions into a more intricate and organized representation of data when transitioning from a Conceptual Data Model (CDM) to a Logical Data Model (LDM). Following are some guidelines for switching from a CDM to an LDM:

Rule 1: List the entities that apply to the application or system being represented. Entities are things or ideas with data attached to them.

Rule 2: Determine the connections between things by defining relationships. Relationships explain the connections between the different entities.

Rule 3: Ensure that the entities are normalized so that each has a unique primary key and all attributes are atomic (i.e., indivisible).

Rule 4: To give the model additional structure and detail, provide the entities properties. The traits or characteristics of an entity are described by its attributes.

Rule 5: Specify the characteristics' data types. Data formats and sizes are described by data types.

Rule 6: Provide any restrictions or guidelines that apply to the model's data. Business rules, rules for data validation, and other rules that relate to the data can all be considered constraints.

Rule 7: Verify the model to make sure it appropriately captures the restrictions and requirements of the system being modeled. To make sure the model complies with these specifications, refine it as necessary.

Rule 8: The model's entities, relationships, characteristics, data types, and constraints should all be documented. This documentation will be helpful for reference and upkeep in the future.

3.15 SOFTWARE USED

3.15.1 WINDOWS 10

As a member of the Windows NT family of operating systems, Windows 10 is a line of software developed by Microsoft. It is the replacement for Windows 8.1 and was made generally available for retail sale on July 29, 2015. It was released to manufacture on July 15, 2015.

3.15.2 MySQL SERVER

An open-source relational database management system (RDBMS) called MySQL server is widely used for storing, managing, and organizing structured data. Many users can access and modify data contained in the database thanks to a server-based software program.

The client-server architecture of the MySQL server allows client programs to communicate with the server in order to view and modify database data. With access control measures, the server offers a centralized location for data storage while also guaranteeing data consistency and upholding data security.

3.15.3 VISUAL STUDIO

Microsoft's Visual Studio is an integrated development environment (IDE) for creating software programs for a variety of operating systems and web-based platforms, including Windows, macOS, iOS, and Android. Using a range of programming languages, like as C++, C#, Visual Basic, JavaScript, TypeScript, and others, it offers developers a complete set of tools and functionality for the creation, debugging, and deployment of programs.

To increase coding productivity and accuracy, Visual Studio comes with a code editor that has tools like IntelliSense, code completion, and syntax highlighting. The tools for version control, testing, and debugging are also included, enabling developers to locate faults and address them effectively.

3.15.4 EDRAW MAX

Users can create expert-looking diagrams, charts, and other visual representations of information with the flexible diagramming software Edraw Max. It is made by the Chinese software business EdrawSoft and works with the Windows, Mac, and Linux operating systems.

For several types of diagrams, such as flowcharts, org charts, mind maps, network diagrams, floor plans, and more, Edraw Max offers a large selection of templates and symbols. It is a strong tool for creating intricate and dynamic diagrams because it also comes with advanced capabilities like data linkage, real-time collaboration, and multi-page diagrams.

3.16 PROGRAMMING LANGUAGES

3.16.1 VB.NET

Based on the original Visual Basic language created by Microsoft, VB.NET (Visual Basic.NET) is an object-oriented programming language. It is a component of the.NET framework and is utilized to create desktop, web, game, and other software applications.

A variety of development environments, such as Visual Studio, which offers a complete set of tools and functionality for developing and distributing applications, can be used to create VB.NET applications. Being extremely portable, VB.NET code may be converted into Microsoft Intermediate Language (MSIL), which can then be run on any system that supports the.NET framework.

3.17 HARDWARE USED

We utilized a computer with the following features to develop our system (platform):

- A laptop computer brand HP
- Processor: Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz   2.50 GHz
- RAM: 8Go
- Hard Disk: 465.76Go
- Screen: 13.5 inches
- Operating System: Microsoft Windows 10 Pro

3.18 MODULES OF THE DESIGNED SYSTEM

Our VB.NET application will be developed with the help of the following modules:

- Data base management system placed in a local server (MySQL server)
- An application which will interact with the database server. This application will present the following modules:
    - A login page
    - A Home page
    - A module that displays all the products

- A module to add a new product, delete a product and update a product’s detail
- A module that displays all the customers
- A module to add a new customer, delete a customer and update a customer’s details
- A module that displays all the users
- A module to add a new user, delete a user and update a user’s details
- A module that displays all the categories
- A module to add a new category, delete a category and update a category’s details
- A module that displays all the orders
- A module to add a new order, delete an order, update an order’s details, print, and order.

3.19 PHYSICAL ORGANIZATION (STRUCTURE) OF THE APPLICATION

In this section, we are going to see the global and physical structure of the application in order to see the internal functioning of IMS application as shown in figure 5 below

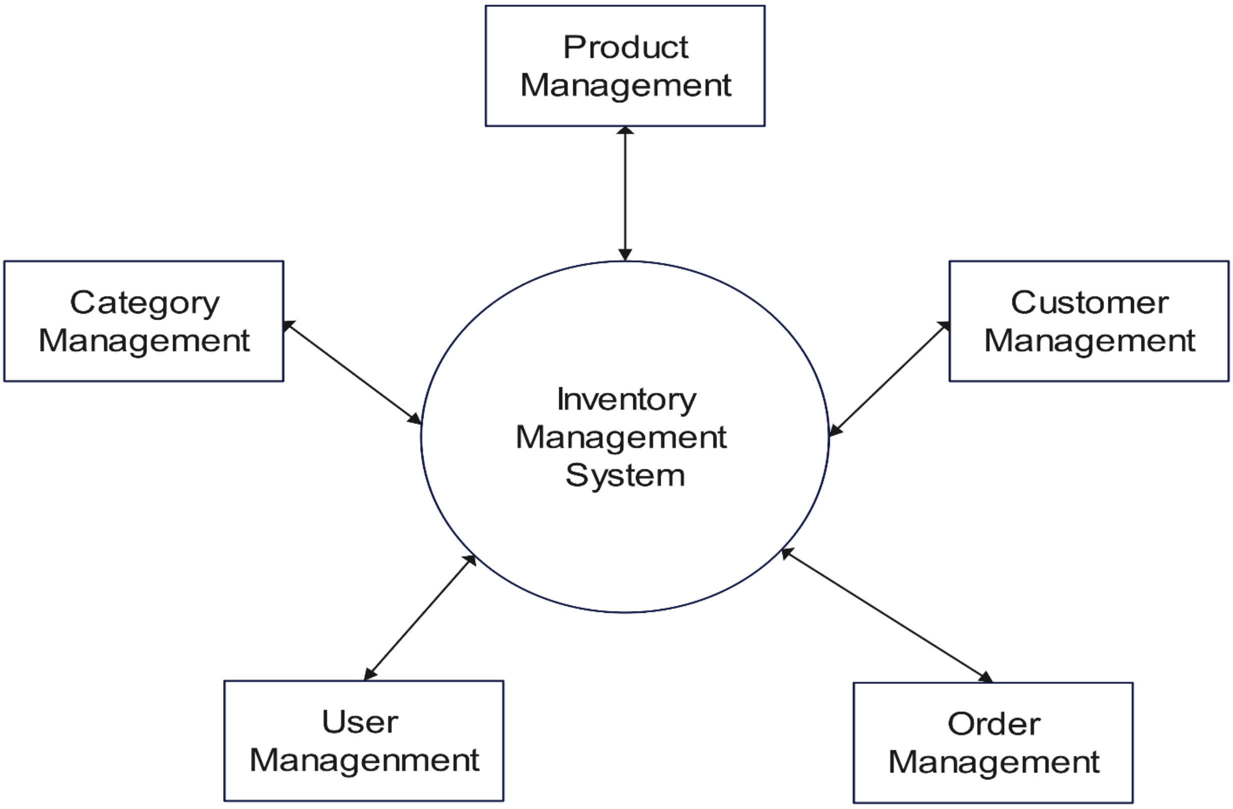


Figure 6:Physical Organization of the application

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 INTRODUCTION

The problem was thoroughly examined at this point in our dissertation, the appropriate technique was employed to identify solutions to the issues the users raised and in this, chapter we will provide the remedies and discuss them. In addition, we will be able to observe the rules that the system's manager has been playing by looking at the outcomes of his actions.

4.2 RESULTS

In terms of outcomes, we will demonstrate the various print screens of our web application in order to determine its functionalities, how it appears in terms of layout, and to gain an understanding of how the system operates internally with the aid of the rules played by the aforementioned actors.

4.3 PRESENTATION OF SCENARIOS AND A BRIEF EXPLANATION OF EACH SCENARIO

4.3.1 THE LOGIN PAGE

The login page of the application is the first screen that the user encounters when accessing the application. It prompts the user to enter their username and password in order to verify their identity and gain access to the application’s features and functionality.

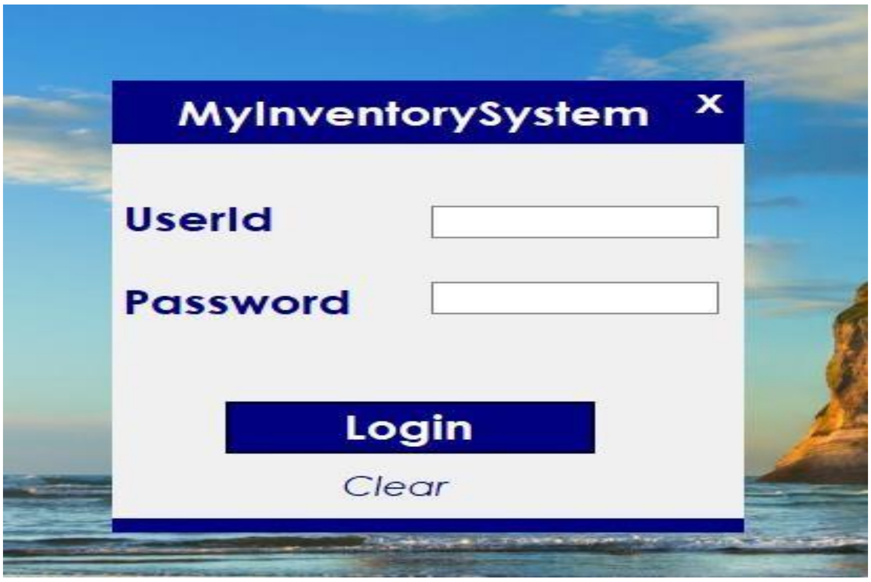


Figure 7: login Page of the application

4.3.2 HOME PAGE

The home page of the system is design to provide users to access the different forms the system and make it easy to manage their products, orders, users, categories, and customer.

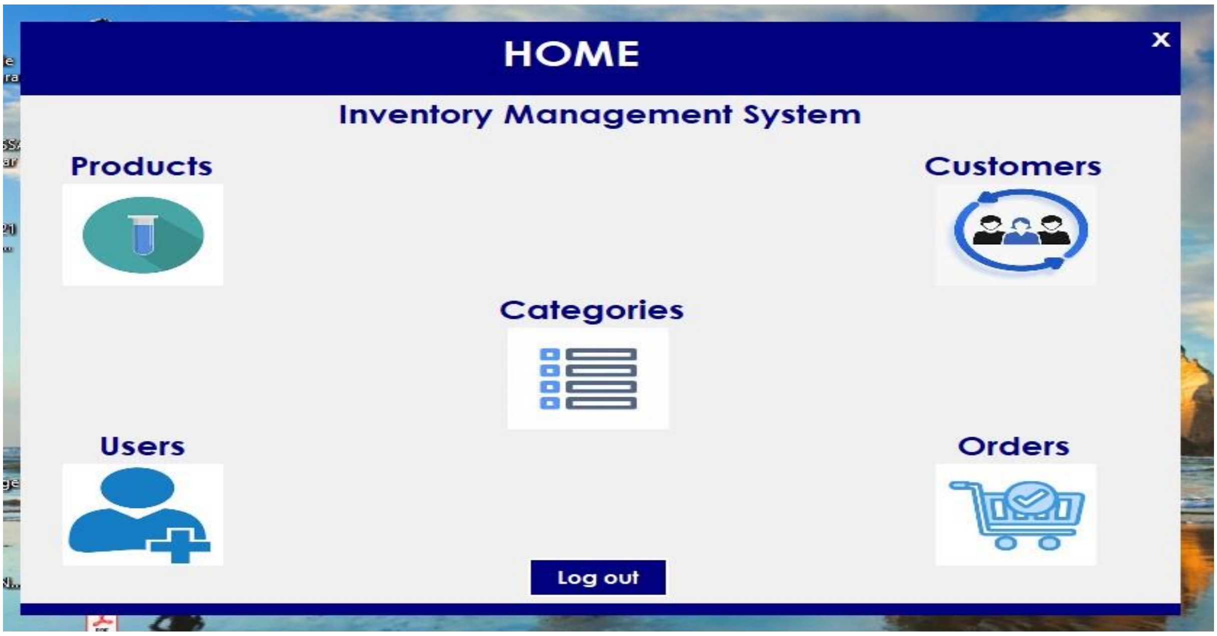
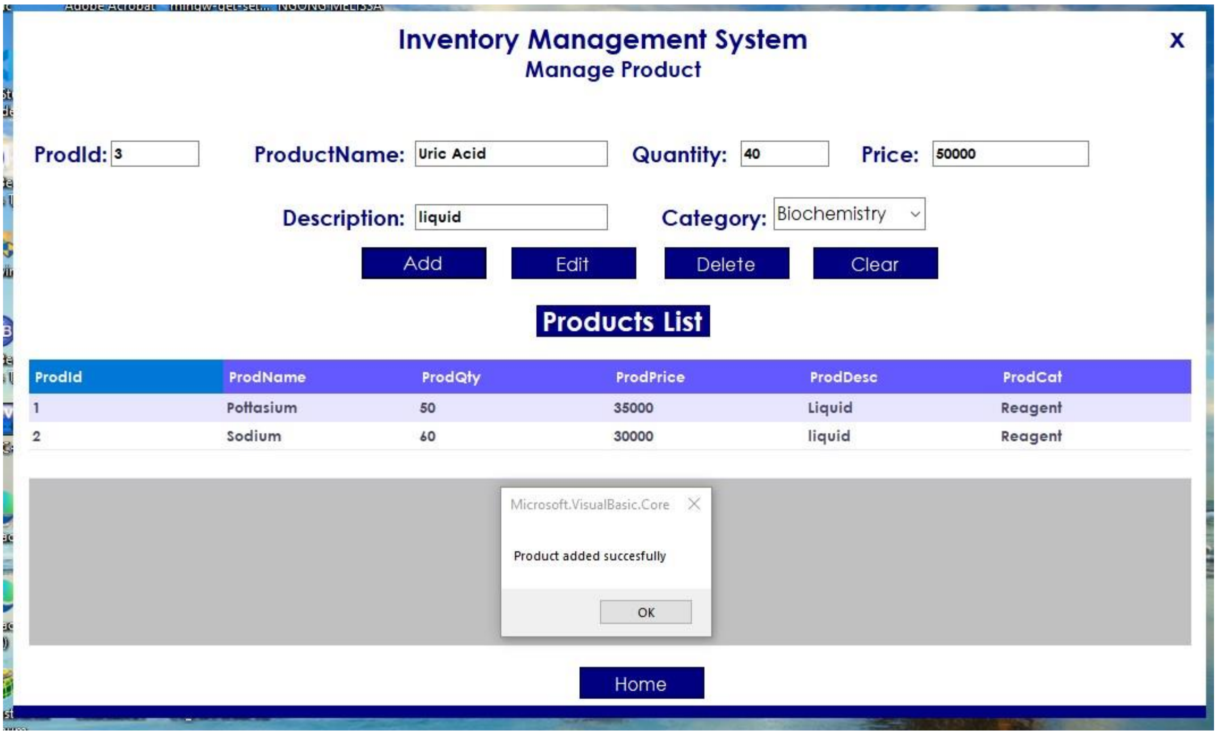


Figure 8: Figure 8: Home Page of the application

4.3.3 PRODUCT MANAGEMENT FORM

This is where a product can be added, deleted and updated. To delete a particular product, the user enters the product Id in the ProdId text box and click on the button delete to update a product information, the user fills in the different text boxes of the product information and the user modifies the information and click on the update button.



Inventory Management System

Manage Product

ProdId: 3 ProductName: Uric Acid Quantity: 40 Price: 50000

Description: liquid Category: Biochemistry

Products List

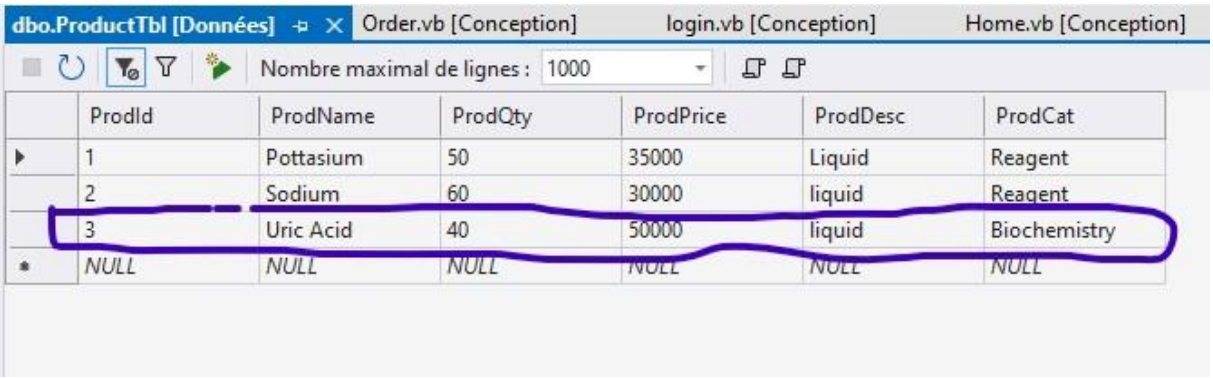
ProdId	ProdName	ProdQty	ProdPrice	ProdDesc	ProdCat
1	Pottasium	50	35000	Liquid	Reagent
2	Sodium	60	30000	liquid	Reagent

Microsoft.VisualBasic.Core

Product added succesfully

Figure 9: Product Form of the application

When the user fills the product information and clicks on add, the information about the product is recorded in the database in the product table as shown in figure 10 below:



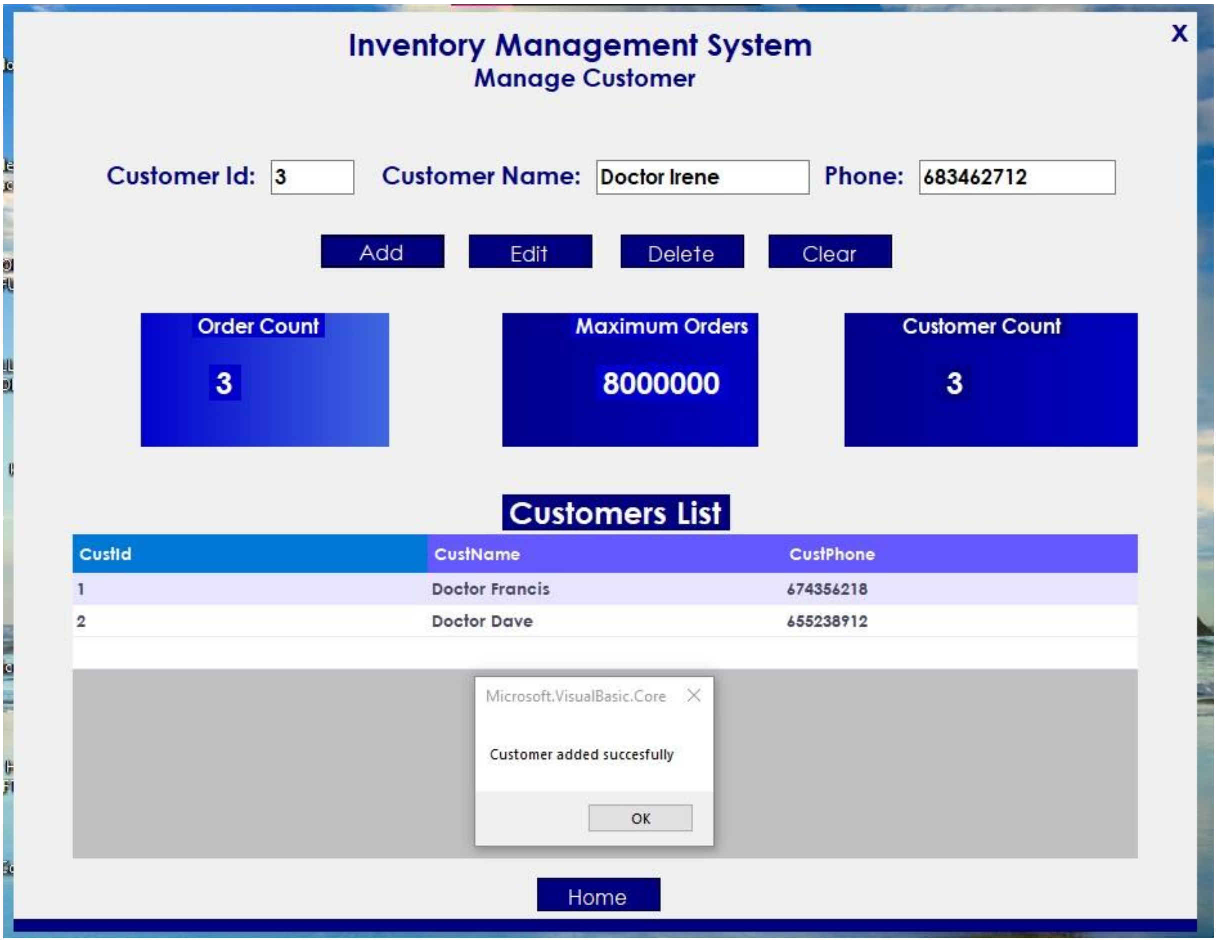
Nombre maximal de lignes : 1000

	ProdId	ProdName	ProdQty	ProdPrice	ProdDesc	ProdCat
▶	1	Pottasium	50	35000	Liquid	Reagent
	2	Sodium	60	30000	liquid	Reagent
	3	Uric Acid	40	50000	liquid	Biochemistry
*	NULL	NULL	NULL	NULL	NULL	NULL

Figure 10: Adding a new product “Uric Acid” in the database.

4.3.4 CUSTOMER MANAGEMENT FORM

Here, a customer can be added, deleted, or updated. While updating a customer's information, the user fills out the appropriate text boxes, makes any necessary changes, and clicks the update button. The user enters the customer Id in the CustomerId text box and clicks the delete button to remove a specific customer.



Inventory Management System

Manage Customer

Customer Id: 3 Customer Name: Doctor Irene Phone: 683462712

Customers List

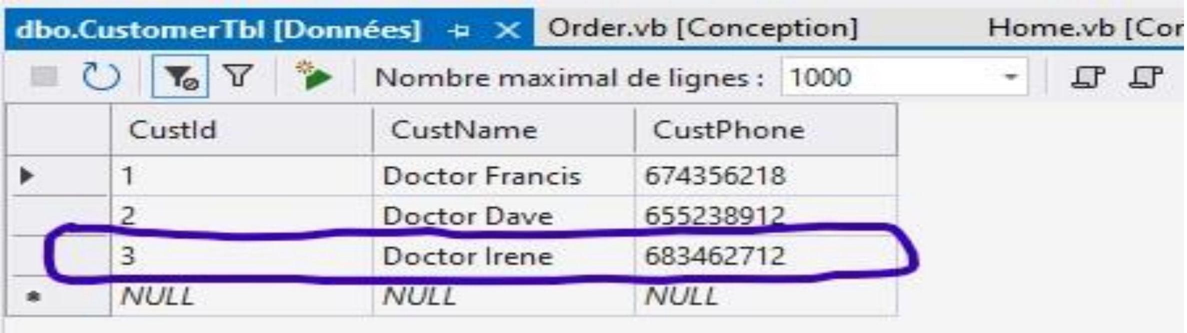
CustId	CustName	CustPhone
1	Doctor Francis	674356218
2	Doctor Dave	655238912

Microsoft.VisualBasic.Core

Customer added succesfully

Figure 11: Customer Form of the application

When the user fills the customer's information and clicks on add, the information about the customer is recorded in the database in the customer table as shown in figure 12 below:



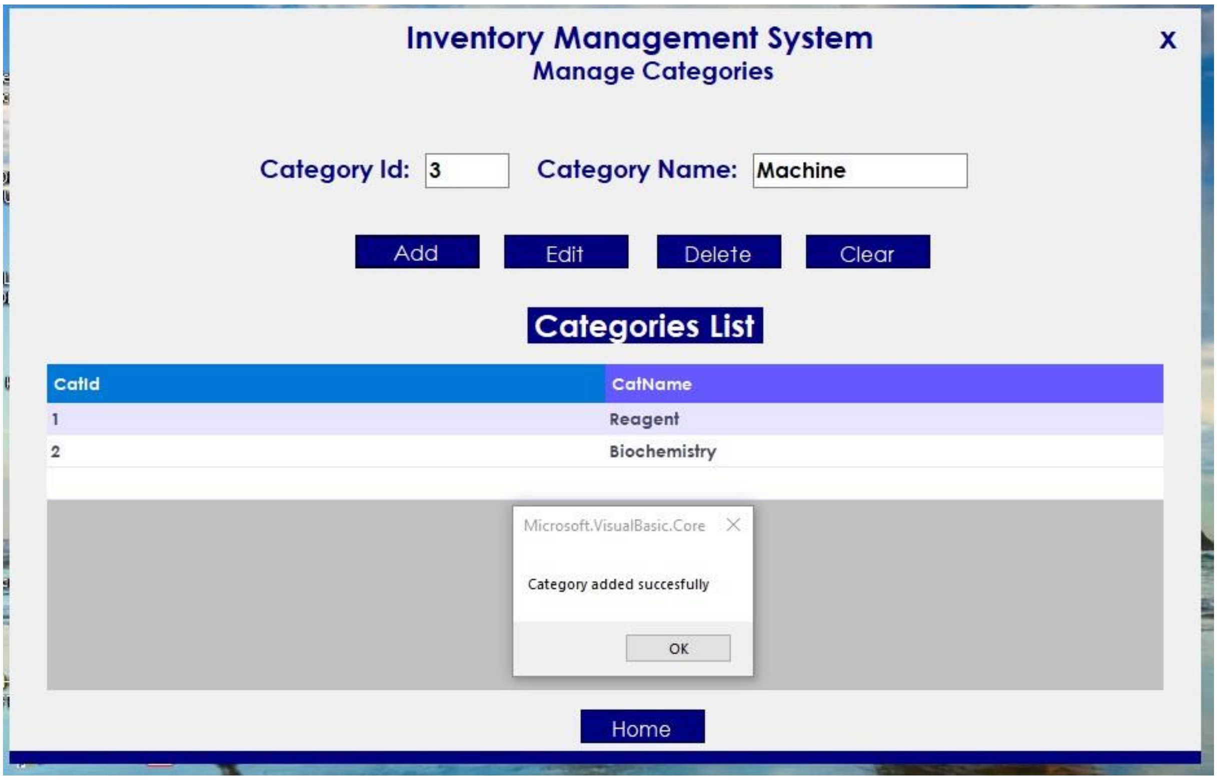
Nombre maximal de lignes : 1000

CustId	CustName	CustPhone
1	Doctor Francis	674356218
2	Doctor Dave	655238912
3	Doctor Irene	683462712
NULL	NULL	NULL

Figure 12: Adding a new customer "Doctor Irene" in the database.

4.3.5 CATEGORY MANGEMENT FORM

A category can be added, deleted, or updated in this form. The user fills up the proper text boxes, makes any required adjustments, and then clicks the update button to update a category's information. To delete a specific category, the user enters the category Id in the categoryId text field and clicks the delete button.



Inventory Management System

Manage Categories

Category Id: 3

Category Name: Machine

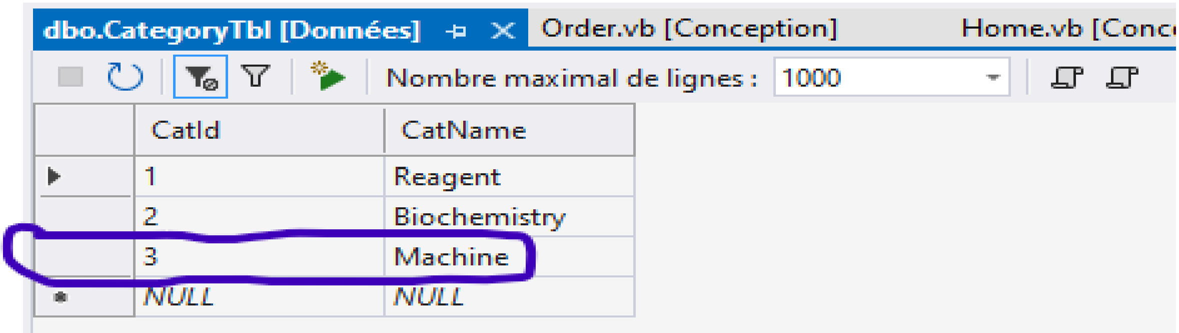
CatId	CatName
1	Reagent
2	Biochemistry

Microsoft.VisualBasic.Core

Category added succesfully

Figure 13: Category Form of the application

When the user fills the category's information and clicks on add, the information about the category is recorded in the database in the category table as shown in figure 14 below:



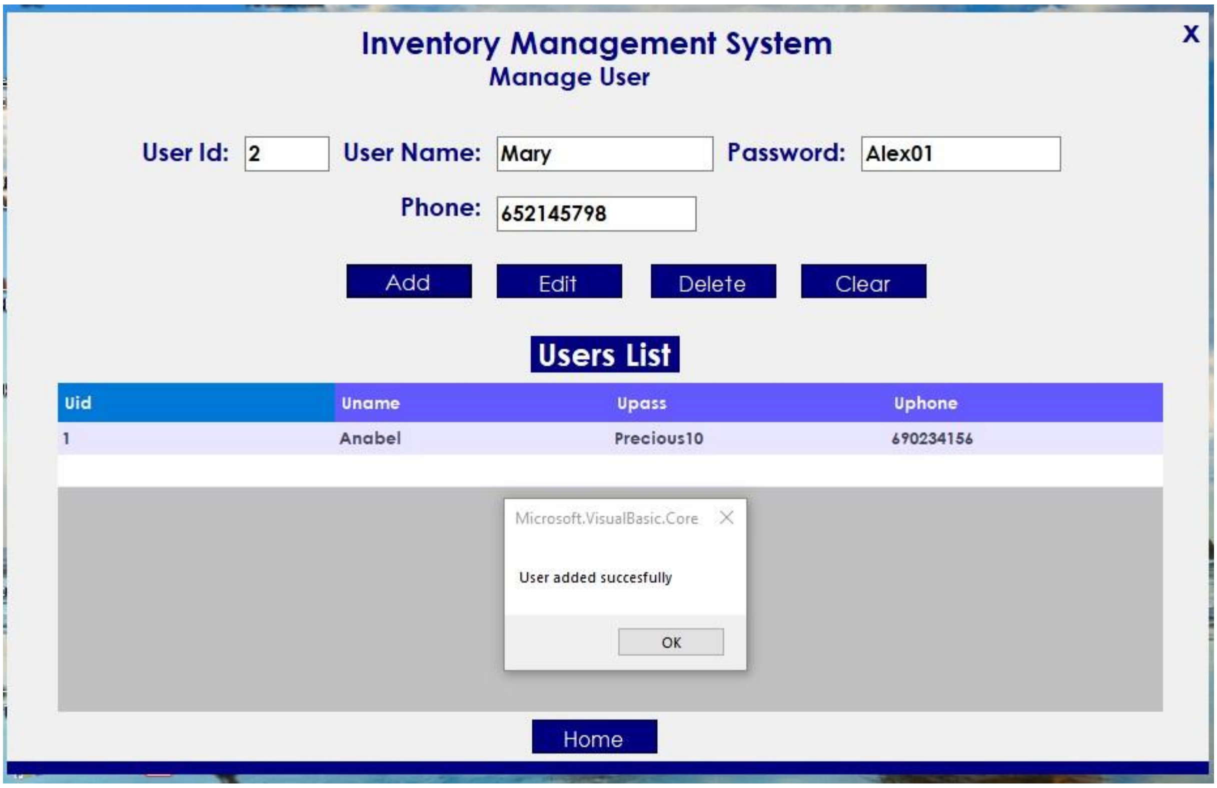
Nombre maximal de lignes : 1000

	CatId	CatName
▶	1	Reagent
	2	Biochemistry
	3	Machine
*	NULL	NULL

Figure 14: Adding a new category "Machine" in the database.

4.3.6 USER MANAGEMENT FORM

A user may be added, deleted, or updated here. To update a user’s information, the user fills out the various text boxes and edits the information before clicking on the update button. To delete a specific user, the user enters the user Id in the user Id text box and clicks on the delete button.



Inventory Management System

Manage User

User Id: 2 User Name: Mary Password: Alex01

Phone: 652145798

Users List

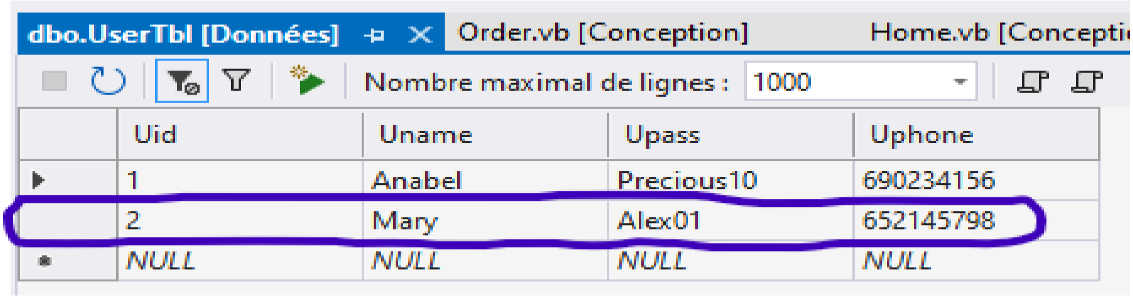
Uid	Uname	Upass	Uphone
1	Anabel	Precious10	690234156

Microsoft.VisualBasic.Core

User added succesfully

Figure 15: User Form of the application

When the user fills the user’s information and clicks on add, the information about the user is recorded in the database in the user table as shown in figure 16 below:



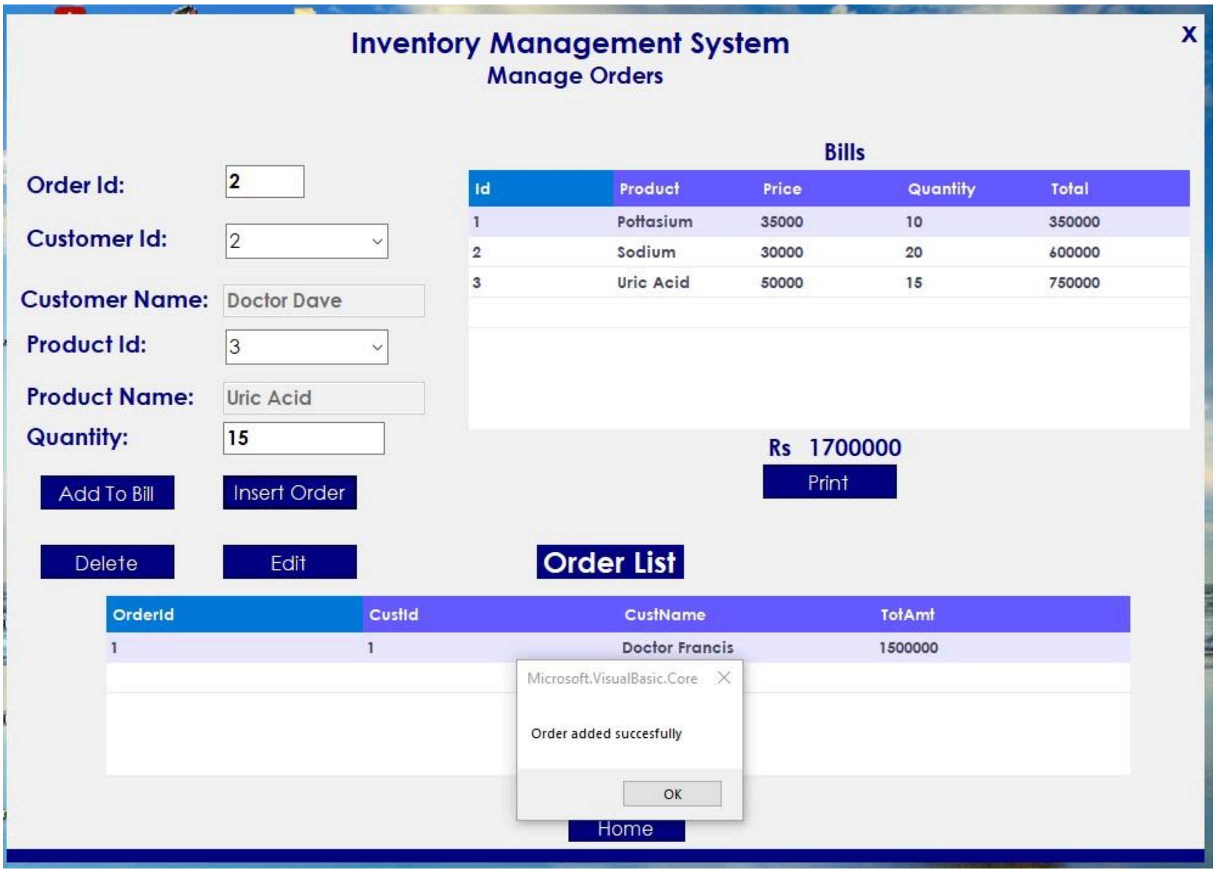
Nombre maximal de lignes : 1000

Uid	Uname	Upass	Uphone
1	Anabel	Precious10	690234156
2	Mary	Alex01	652145798
NULL	NULL	NULL	NULL

Figure 16: Adding a new user “Mary” in the database

4.3.7 ORDER MANAGEMENT FORM

Here, an order can be updated, deleted, added and printed. Before clicking the update button, the user must fill out the various text boxes and make any necessary modifications to their information. The user enters the order Id in the order Id text box and clicks the delete button to remove a specific order. The user can equally print an order by clicking the print button.



Order Id: 2

Customer Id: 2

Customer Name: Doctor Dave

Product Id: 3

Product Name: Uric Acid

Quantity: 15

Bills

Id	Product	Price	Quantity	Total
1	Pottasium	35000	10	350000
2	Sodium	30000	20	600000
3	Uric Acid	50000	15	750000

Rs 1700000

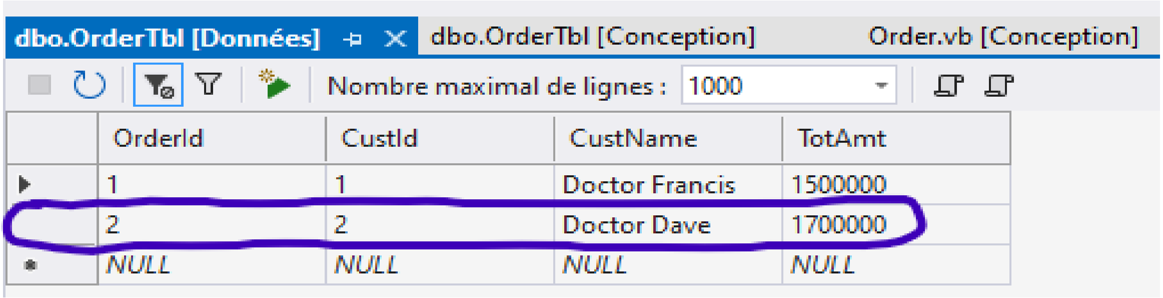
OrderId	CustId	CustName	TotAmt
1	1	Doctor Francis	1500000

Microsoft.VisualBasic.Core

Order added succesfully

Figure 17: Order Form of the application

When the user fills the order's information and clicks on add, the information about the order is recorded in the database in the order table as shown in figure 18 below:

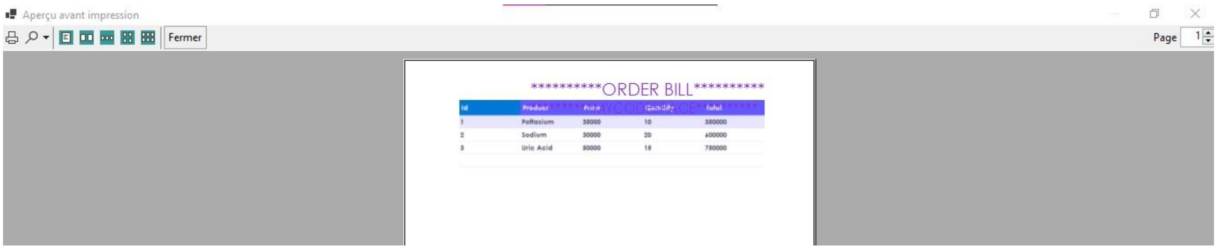


Nombre maximal de lignes : 1000

OrderId	CustId	CustName	TotAmt
1	1	Doctor Francis	1500000
2	2	Doctor Dave	1700000
NULL	NULL	NULL	NULL

Figure 18: Adding a new order in the database.

The user can print the customer's order by clicking on the print button. This is shown on figure 19 below:



**********ORDER BILL**********

Id	Product	Price	Quantity	Total
1	Pottasium	35000	10	350000
2	Sodium	30000	20	600000
3	Uric Acid	50000	15	750000

Figure 19: Printing the customer's order.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATION

SUMARRY AND FINDINGS

This chapter describes the objectives of the system stipulated in earlier chapters, limitations of the system, conclusion and recommendation of the system.

The main goal of this study was to develop a desktop application that could assist PROMACAM in managing the volume of goods coming in and going out, the computation of sales, the management of customers, the management of products, and the management of users. Gathering requirements is the first step in this. To complete this project, we began by outlining the issue, outlining our suggested fix, and laying out our goals. By meeting our objectives, which largely involved creating a desktop application for inventory management at PROMACAM, we have completed the project.

The successful implementation of an inventory management system requires careful planning and consideration of factors such as the type of inventory being managed, the size and complexity of the organization, and the needs and preferences of end-users.

5.2 DIFFICULTIES (LIMITATIONS) ENCOUNTERED

Many obstacles sought to derail this project during the development of this application, but the writer's dedication and perseverance were up to the task.

The following are some difficulties that were faced.

Problems with the Platform's Design

During the design of this application, I encountered some difficulties and some include:

Memory management: VB.net applications require efficient memory management to prevent issues such as memory leaks or crashes.

Debugging and testing: Identifying and fixing bugs or errors in a VB.net application can be time-consuming and challenging, particularly in larger or more complex applications.

5.3 RECOMMENDATION

I will have some recommendations for PROMACAM, which are as follows:

- PROMACAM must have a department, which will be responsible for checking the stock levels in the working area. To use the inventory management system to manage and equally be in charge of overseeing the bringing in and taking out of each item individually.
- PROMACAM needs to establish a software development department to aid in solving IT issues that may be resolved through software for the company. I believe this would be advantageous for the company and would broaden their market reach.
- Information about the organization ought to be computerized as well.

REFERENCES (SYSTEM OF REFERENCING)

J. Glenn Brookshear (1988). Computer science overview PP 493 9TH Edition. Benjamin/Cummings Publishing Company.

Sam Lightstone, Toby Teory, Tom Nadeu, (Fourth Edition, 1999) Database Modeling &Design : Logical Design William H. Mitchell, UML, the Unified Modeling Language.

Pascal, R. (2003). UML par la pratique 2ème Edition Pascal Roques Publisher Groupe EYROLLES.Pp 1-10

Hugh Darwen (March 2010). An Introduction to Relational Database Theory.

(Rumbaugh) Bernard ESPINASSE (1980) Sadt Méthodes fonctionnelles (1980). Présentation générale de la méthode orientée objet : O.M.T.pp 10-30

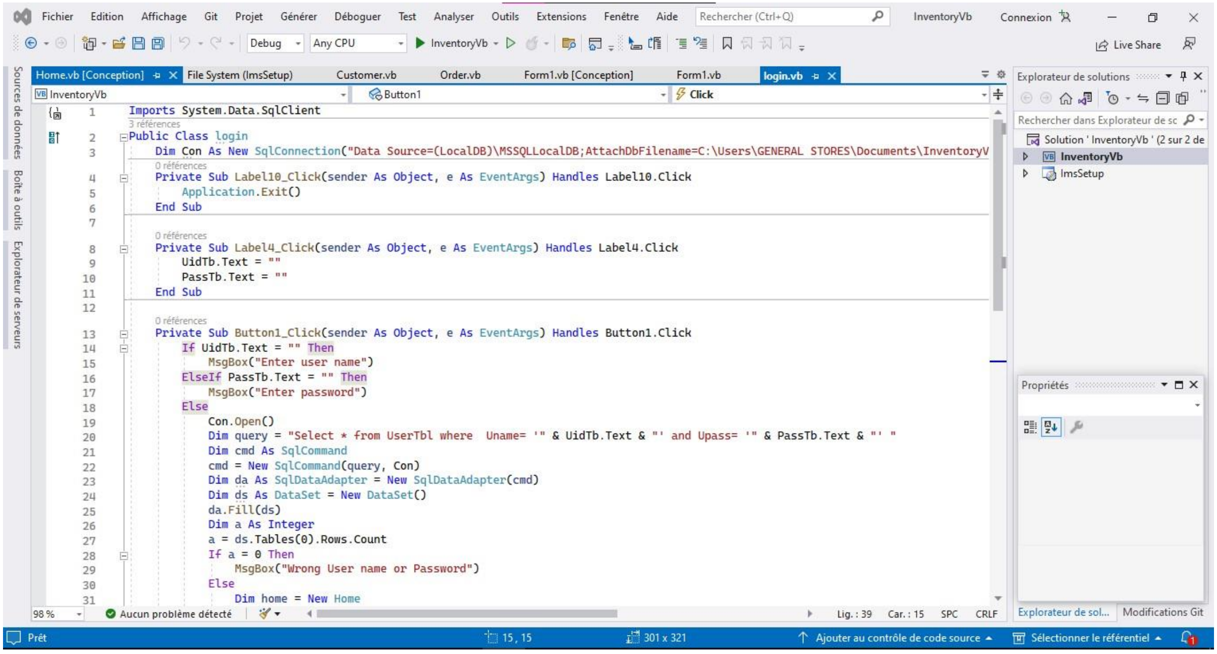
Angappa Gunasekaran (IGI Global, 2007), Modeling and Analysis of Enterprise Information Systems (MERIS)

WEBOGRAPHY

https://www.pearson.com/us/higher-education/product/Brookshear-Computer-Science-AnOverview-10th-Edition/9780321524034.html accessed 3rd July 2017 http://databasemanagement.wikia.com/wiki/DBMS_Functions access 3rd July 2017 https://www.techopedia.com/definition/1234/relational-database-rdb accessed 28th December 2016 https://www.bayt.com/en/specialties/q/47871/advantages-and-disadvantages-of-dbms-overtraditional-file-processing-system/ accessed 28th December 2016 https://www.allacronyms.com/MERISE/M%C3%A9thode_d%27Etude_et_de_R%C3%A9alisation_Informatique_pour_les_Syst%C3%A8mes_d%27Entreprise

APPENDIX

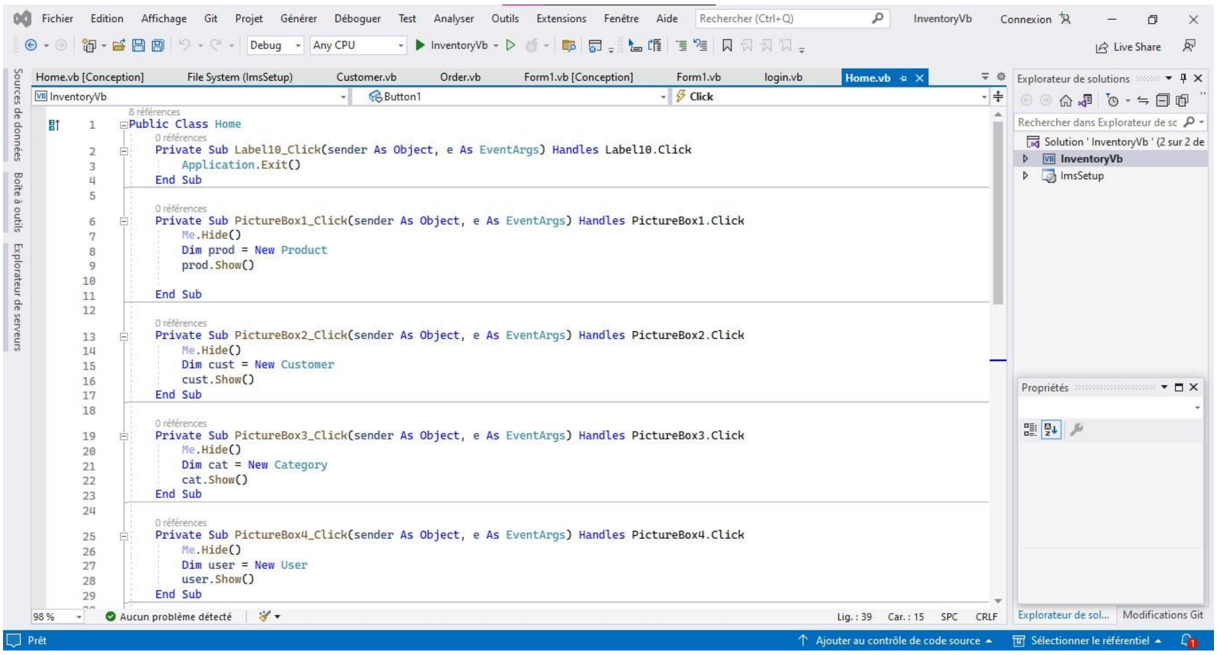
Login page



```
Imports System.Data.SqlClient
3 références
Public Class login
    Dim Con As New SqlConnection("Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\GENERAL STORES\Documents\InventoryV
    0 références
    Private Sub Label10_Click(sender As Object, e As EventArgs) Handles Label10.Click
        Application.Exit()
    End Sub

    0 références
    Private Sub Label4_Click(sender As Object, e As EventArgs) Handles Label4.Click
        UidTb.Text = ""
        PassTb.Text = ""
    End Sub

    0 références
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If UidTb.Text = "" Then
            MsgBox("Enter user name")
        ElseIf PassTb.Text = "" Then
            MsgBox("Enter password")
        Else
            Con.Open()
            Dim query = "Select * from UserTbl where  Uname= '" & UidTb.Text & "' and Upass= '" & PassTb.Text & "' "
            Dim cmd As SqlCommand
            cmd = New SqlCommand(query, Con)
            Dim da As SqlDataAdapter = New SqlDataAdapter(cmd)
            Dim ds As DataSet = New DataSet()
            da.Fill(ds)
            Dim a As Integer
            a = ds.Tables(0).Rows.Count
            If a = 0 Then
                MsgBox("Wrong User name or Password")
            Else
                Dim home = New Home
```

Figure 20: Login page source code.

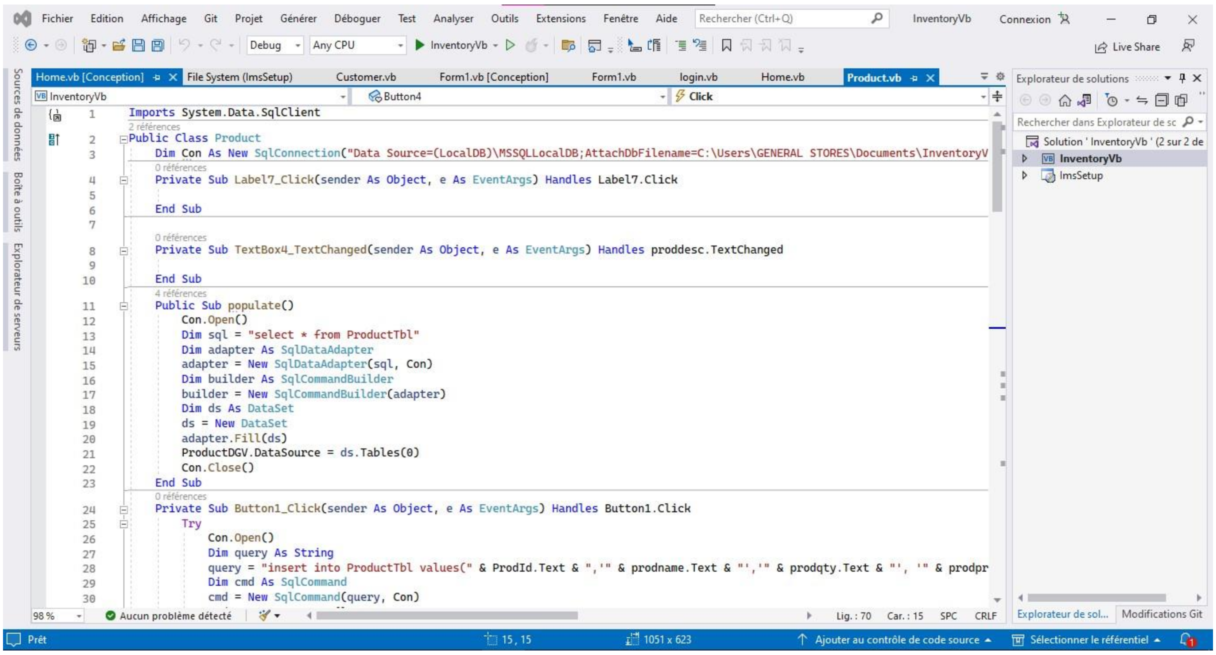
HOME PAGE



```
8 références
Public Class Home
    0 références
    Private Sub Label10_Click(sender As Object, e As EventArgs) Handles Label10.Click
        Application.Exit()
    End Sub

    0 références
    Private Sub PictureBox1_Click(sender As Object, e As EventArgs) Handles PictureBox1.Click
        Me.Hide()
        Dim prod = New Product
        prod.Show()

    End Sub

    0 références
    Private Sub PictureBox2_Click(sender As Object, e As EventArgs) Handles PictureBox2.Click
        Me.Hide()
        Dim cust = New Customer
        cust.Show()
    End Sub

    0 références
    Private Sub PictureBox3_Click(sender As Object, e As EventArgs) Handles PictureBox3.Click
        Me.Hide()
        Dim cat = New Category
        cat.Show()
    End Sub

    0 références
    Private Sub PictureBox4_Click(sender As Object, e As EventArgs) Handles PictureBox4.Click
        Me.Hide()
        Dim user = New User
        user.Show()
    End Sub
```

Figure 21: Home page source code.

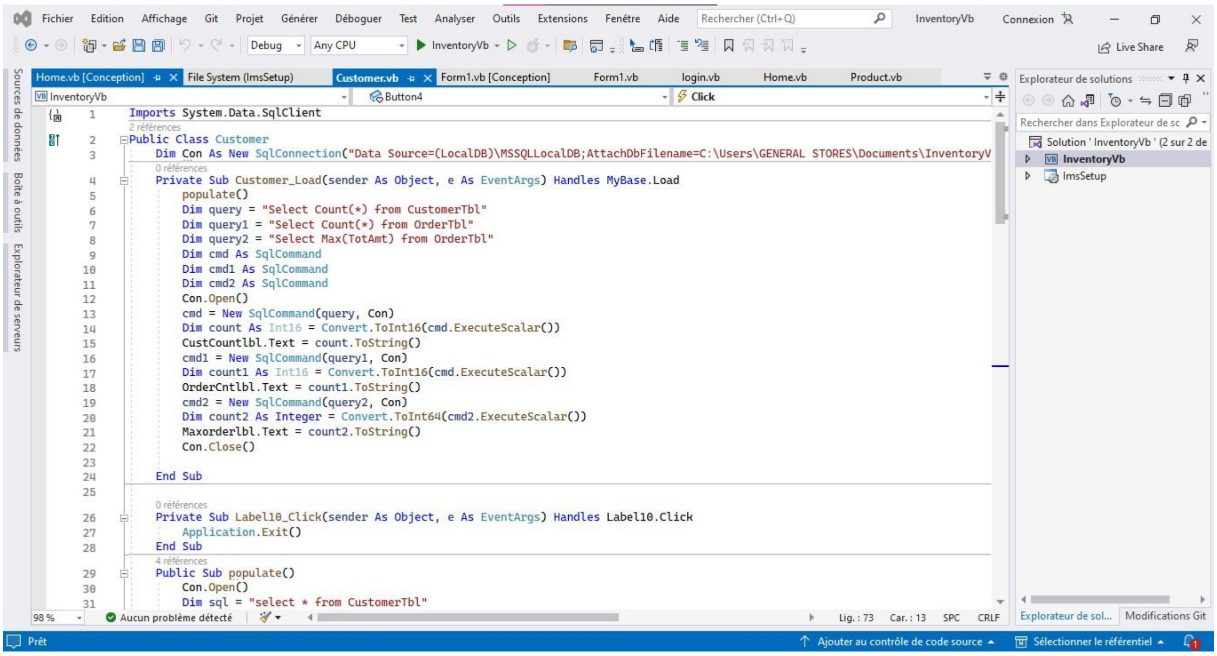
Product form



```vb
Imports System.Data.SqlClient
Public Class Product
    Dim Con As New SqlConnection("Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\GENERAL STORES\Documents\InventoryV
    Private Sub Label7_Click(sender As Object, e As EventArgs) Handles Label7.Click

    End Sub

    Private Sub TextBox4_TextChanged(sender As Object, e As EventArgs) Handles proddesc.TextChanged

    End Sub
    Public Sub populate()
        Con.Open()
        Dim sql = "select * from ProductTbl"
        Dim adapter As SqlDataAdapter
        adapter = New SqlDataAdapter(sql, Con)
        Dim builder As SqlCommandBuilder
        builder = New SqlCommandBuilder(adapter)
        Dim ds As DataSet
        ds = New DataSet
        adapter.Fill(ds)
        ProductDGV.DataSource = ds.Tables(0)
        Con.Close()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Try
            Con.Open()
            Dim query As String
            query = "insert into ProductTbl values(" & ProdId.Text & ",'" & prodname.Text & "','" & prodqty.Text & "', '" & prodpr
            Dim cmd As SqlCommand
            cmd = New SqlCommand(query, Con)
```

Figure 22: Product form source code.

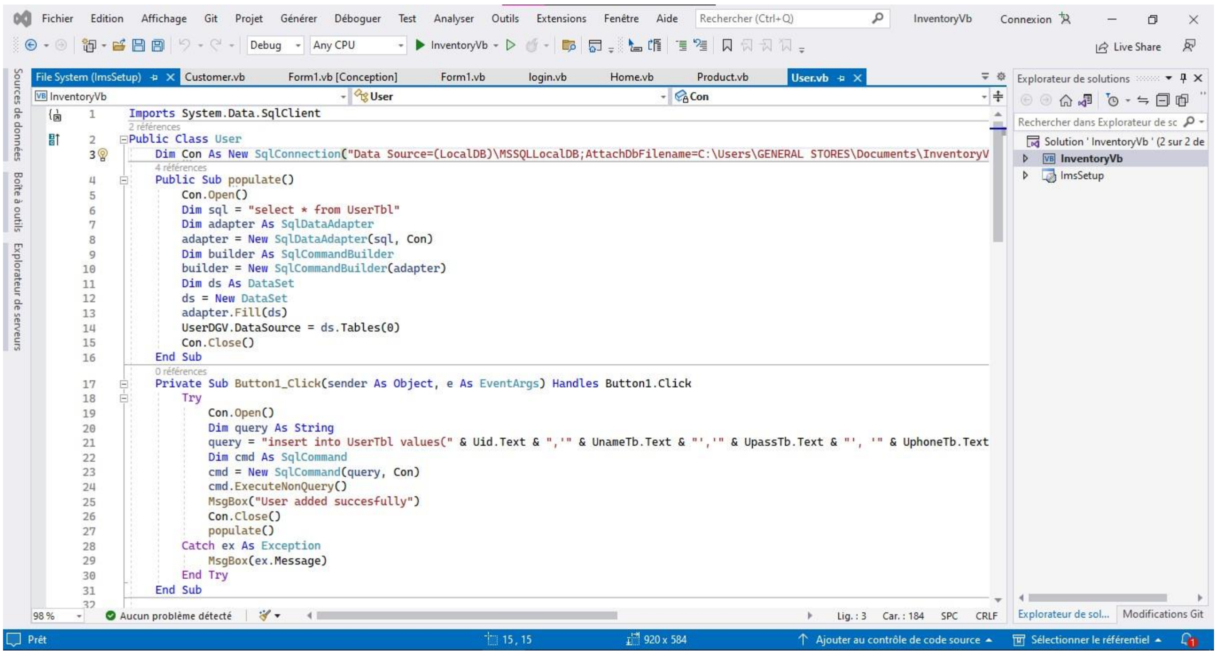
Customer form



```vb
Imports System.Data.SqlClient
Public Class Customer
    Dim Con As New SqlConnection("Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\GENERAL STORES\Documents\InventoryV
    Private Sub Customer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        populate()
        Dim query = "Select Count(*) from CustomerTbl"
        Dim query1 = "Select Count(*) from OrderTbl"
        Dim query2 = "Select Max(TotAmt) from OrderTbl"
        Dim cmd As SqlCommand
        Dim cmd1 As SqlCommand
        Dim cmd2 As SqlCommand
        Con.Open()
        cmd = New SqlCommand(query, Con)
        Dim count As Int16 = Convert.ToInt16(cmd.ExecuteScalar())
        CustCountlbl.Text = count.ToString()
        cmd1 = New SqlCommand(query1, Con)
        Dim count1 As Int16 = Convert.ToInt16(cmd.ExecuteScalar())
        OrderCntlbl.Text = count1.ToString()
        cmd2 = New SqlCommand(query2, Con)
        Dim count2 As Integer = Convert.ToInt64(cmd2.ExecuteScalar())
        Maxorderlbl.Text = count2.ToString()
        Con.Close()

    End Sub

    Private Sub Label10_Click(sender As Object, e As EventArgs) Handles Label10.Click
        Application.Exit()
    End Sub
    Public Sub populate()
        Con.Open()
        Dim sql = "select * from CustomerTbl"
```

Figure 23: Customer form source code.

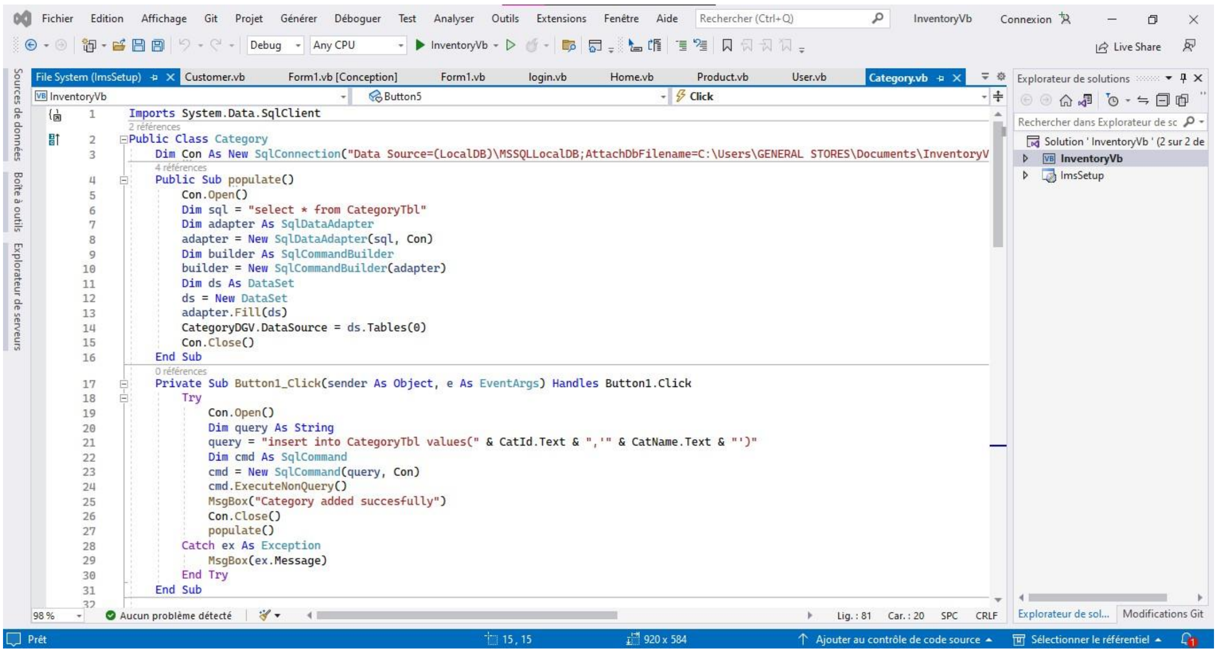
User form



```vb
Imports System.Data.SqlClient
Public Class User
    Dim Con As New SqlConnection("Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\GENERAL STORES\Documents\InventoryV
    Public Sub populate()
        Con.Open()
        Dim sql = "select * from UserTbl"
        Dim adapter As SqlDataAdapter
        adapter = New SqlDataAdapter(sql, Con)
        Dim builder As SqlCommandBuilder
        builder = New SqlCommandBuilder(adapter)
        Dim ds As DataSet
        ds = New DataSet
        adapter.Fill(ds)
        UserDGV.DataSource = ds.Tables(0)
        Con.Close()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Try
            Con.Open()
            Dim query As String
            query = "insert into UserTbl values(" & Uid.Text & ",'" & UnameTb.Text & "','" & UpassTb.Text & "', '" & UphoneTb.Text
            Dim cmd As SqlCommand
            cmd = New SqlCommand(query, Con)
            cmd.ExecuteNonQuery()
            MsgBox("User added succesfully")
            Con.Close()
            populate()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
```

Figure 24: User form source code.

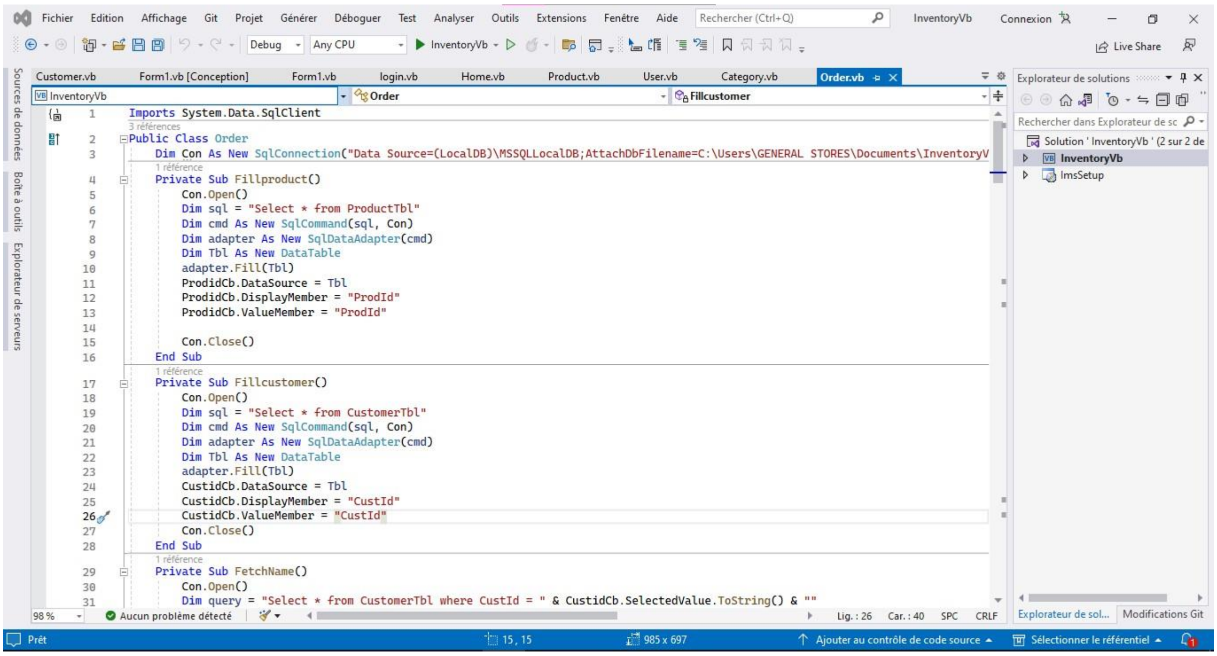
Category form



```vb
Imports System.Data.SqlClient
Public Class Category
    Dim Con As New SqlConnection("Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\GENERAL STORES\Documents\InventoryV
    Public Sub populate()
        Con.Open()
        Dim sql = "select * from CategoryTbl"
        Dim adapter As SqlDataAdapter
        adapter = New SqlDataAdapter(sql, Con)
        Dim builder As SqlCommandBuilder
        builder = New SqlCommandBuilder(adapter)
        Dim ds As DataSet
        ds = New DataSet
        adapter.Fill(ds)
        CategoryDGV.DataSource = ds.Tables(0)
        Con.Close()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Try
            Con.Open()
            Dim query As String
            query = "insert into CategoryTbl values(" & CatId.Text & ",'" & CatName.Text & "')"
            Dim cmd As SqlCommand
            cmd = New SqlCommand(query, Con)
            cmd.ExecuteNonQuery()
            MsgBox("Category added succesfully")
            Con.Close()
            populate()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
```

Figure 25: Category form source code.

Order form



```
Imports System.Data.SqlClient
Public Class Order
    Dim Con As New SqlConnection("Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\GENERAL STORES\Documents\InventoryV
    Private Sub Fillproduct()
        Con.Open()
        Dim sql = "Select * from ProductTbl"
        Dim cmd As New SqlCommand(sql, Con)
        Dim adapter As New SqlDataAdapter(cmd)
        Dim Tbl As New DataTable
        adapter.Fill(Tbl)
        ProdidCb.DataSource = Tbl
        ProdidCb.DisplayMember = "ProdId"
        ProdidCb.ValueMember = "ProdId"

        Con.Close()
    End Sub
    Private Sub Fillcustomer()
        Con.Open()
        Dim sql = "Select * from CustomerTbl"
        Dim cmd As New SqlCommand(sql, Con)
        Dim adapter As New SqlDataAdapter(cmd)
        Dim Tbl As New DataTable
        adapter.Fill(Tbl)
        CustidCb.DataSource = Tbl
        CustidCb.DisplayMember = "CustId"
        CustidCb.ValueMember = "CustId"
        Con.Close()
    End Sub
    Private Sub FetchName()
        Con.Open()
        Dim query = "Select * from CustomerTbl where CustId = " & CustidCb.SelectedValue.ToString() & ""
```

Figure 26: Order form source code.