

Mouadh Boutaghane

Product management with NetBeans

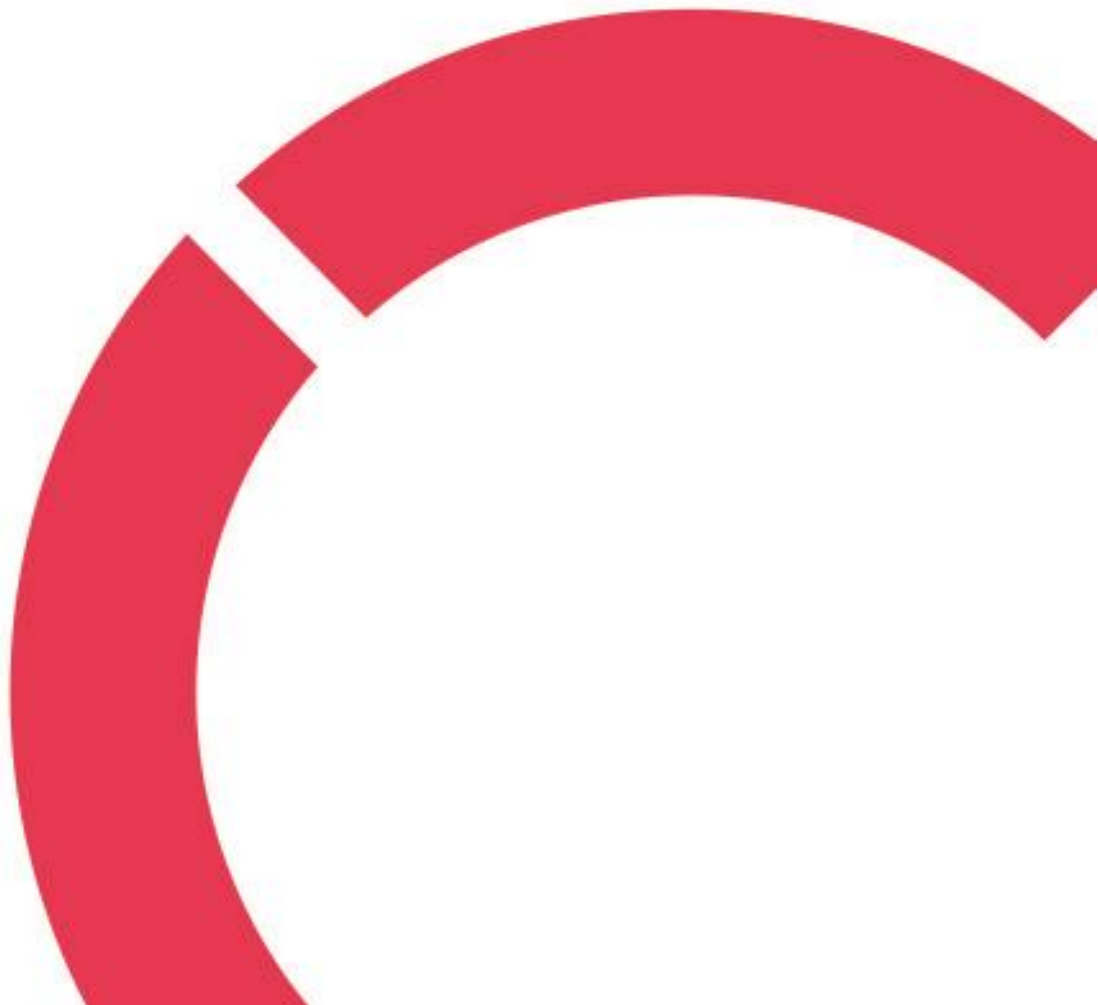
NetBeans

Thesis

CENTRIA UNIVERSITY OF APPLIED SCIENCES

Information Technology

April 2025



ABSTRACT

Centria University of Applied Sciences	Date April 2025	Author Mouadh Boutaghane
Degree programme Information Technology		
Name of thesis Product management with NetBeans.		
Centria supervisor Henry Paananen		Pages 7 + 2
<p>This research analyses the creation of an inventory management desktop system which delivers enhanced tracking capabilities for sellers regarding products and customers with suppliers and optimal stock control. This research explores planning followed by theoretical foundations and incorporates development stages which lead to final product evaluation and improvement suggestions.</p> <p>In this application seller can access the list of (products and product families, customers and suppliers) and add, delete, modify the list, then he can do the calculations, finally he develops plans to look at the stock products, and profit...etc., the main purpose for the thesis is to solve the problem of organizing inventories for any product owner and make their job easier with one application .</p>		
Key words Java,NetBeans,SQL,Desktop Application ,UML,UI,		

CONCEPT DEFINITIONS

SQL

Structured Query Language. Relational databases, data stored in tables

UI

User Interface.

UML

Unified Modelling Language.

JDBS

Java Database Connectivity.

JDK

Software Development Kit

IDE

integrated development environment

JDK

Java Development Kit

GUI

graphical user interfaces

ABSTRACT
CONCEPT DEFINITIONS
CONTENTS

1 INTRODUCTION.....	1
2 OVERVIEW OF PRODUCT.....	2
2.1 Desktop application development.....	2
3 THEORETICAL FRAMEWORK.....	3
3.1 NetBeans.....	3
3.2 SQLite.....	3
3.3 Java Language.....	4
4 PLANNING PROCESS.....	6
4.1 Use Case Diagram.....	7
4.2 Class Diagram.....	9
4.3 CODE STRUCTURE.....	11
4.3 Interacting with the database.....	12
5 DEVELOPMENT PHASE.....	14
5.1 Frontend.....	14
6 EVALUATION AND DISCUSSION.....	16
6.1 Evaluation.....	16
6.2 Discussion.....	17
7 CONCLUSION.....	18

1 INTRODUCTION

The discussion on inventory management includes both finished products and raw materials and semi-finished products. Raw materials together with unfinished products form part of inventory. Inventory management involves developing and carrying out systems to optimize profitability. Optimal inventory management requires owning correct amounts during appropriate period. The production quality suffers when insufficient inventory exists which causes work break. The business costs have two parts from owning more stock than necessary which includes high costs along with dangerous inventory value reduction. Good inventory management requires finding a balanced system which extracts maximum profit while keeping costs at a minimum level. Inventory management benefits from effective modeling and planning tools that let sellers handle their inventory needs through this application. The tool enables sellers to access product and supplier lists before making updates to the data while performing calculations and creating plans that analyze inventory products and profits.

Application development for the desktop will be presented through each stage from Understanding until project completion. The thesis will introduce a discussion about the project content during its initial section. The necessary theory for application development will be explained next before moving onto desktop applications which make up a big portion of programming approaches. The wide scale of the project required specific limitations to proceed. Previously the application follows a phase of planning after which development of the application occurs. The thesis completes with an examination of the product's last stage and proposed modifications.

2 OVERVIEW OF PRODUCT

The selection of thesis topic led to plans for developing an application which presented both construction difficulties and satisfaction in its development. During the development phase the desktop application emerged to assist product owners manage their inventory organization. The following section introduces the project details to readers.

2.1 Desktop application development

A desktop application enables users to perform tasks and access information by working directly on personal computers and laptops through software installations. Users can run desktop applications independently of web browsers since these applications operate outside browser constraints and independently from internet connectivity. Computers use word processors graphics design tools and video editing programs as examples of desktop applications. (Dhruv 2024.)

The advantage of desktop applications is their local data storage yet users must independently arrange backups for data protection. Desktop applications serve as essential tools for efficient offline data management and control which make them important for personal and professional needs.(Pietrelli 2017.)

3 THEORETICAL FRAMEWORK

During the thesis work a Desktop application was built .To accomplish it ,only one software was needed in the begging and its called NetBeans, NetBeans has a built in Database called JDBC which stands for (Java Database Connectivity),the issue with it is that if we try to open the application in a different PC or a different environment we would face many problems such as misconfiguration so another approach was used for the Database which is SQLite, it was selected because its lighter.

3.1 NetBeans

Free and open source NetBeans IDE functions as a platform which supports development of desktop mobile and web applications. The IDE enables developers to build applications in Java among a set of supported languages which also include HTML5 and PHP with C++. The IDE integrates point-to-point development support which assists from the initial project setup through the entire debugging phase and application profiling until deployment. NetBeans IDE functions on Windows, Linux, Mac OS X and several UNIX-based systems running between them. The IDE supports complete JDK 7 technology coverage along with recent Java development features. The IDE represents the initial development environment to offer full support for Java Development Kit version 7 and Java Enterprise Edition Version 7. The IDE delivers complete Java EE support enabled by the latest specifications for Java along with XML Web services and SQL whereas it specifically supports the GlassFish Server as the reference Java EE implementation. (Allen G 2010.)

3.2 SQLite

The SQL database engine SQLite operates as a space-efficient program that functions autonomously without servers and remains self-contained in a single package. The application embeds SQLite to deliver database engine functionality while eliminating server processes and cuts down configuration requirements.(Hipp R 2023.)

SQLite is a database engine. The application provides users with a way to access relational databases via software interaction. A database in SQLite operates using a single file format which differentiates it from other database engines. Database accessibility reaches high levels because users need only duplicate the data storage file to get a database copy and attach the database to share it by email.(Co-decademy Team 2018.)

3.3 Java Language

Java operates as a class-based object-oriented programming language that works for minimal implementation dependencies. WORA serves as the fundamental goal of Java to enable program developers to create applications which operate across every Java-supporting platform without repeating code compilation steps. James Gosling at Sun Microsystems Inc. launched Java in May 1995 before Oracle Corporation took ownership of the technology for building mobile applications and desktop software as well as web-based programs. Java stands out due to its straightforward nature along with fortified security capabilities which drive businesses toward using it for corporate applications. The compilation of Java applications generates byte code which can run through any Virtual Machine built for Java. Java technical syntax matches C/C++ language syntax. With Java users experience convenient development of programming code along with automated code compilation and debugging features. Java provides developers with the tools to generate code libraries along with modular program sections.(Gosling 2025).

3.4 UI&UX

Designing User Interfaces concentrates on the visual components together with interactive elements of products. User component design as well as screen implementation falls under the authority of UI design. A digital product comes to its completion by directly achieving both functional perfection and visual appeal. UX design serves as the user experience design discipline which studies website and application interfaces to determine how they can maintain usability standards. Research explores user requirements to create the logical framework and various workflow structures which they discover through their investigations.(Emil Lamprecht 2023)

3.5 StarUML

StarUML functions as an architectural tool which enables developers to develop UML diagrams for software design purposes. StarUML serves as an essential tool that lets developers and designers create

documents which display system structure alongside behaviour visualization. StarUML allows users to create different UML diagrams including class diagrams as well as sequence diagrams and use case diagrams and others to model distinct elements in software development projects. StarUML serves as a common modelling tool in software development to assist project planning and system architecture design and system architecture communication about the product.(StarUML Team 2025.)

4 PLANNING PROCESS

The planning process began with organizing the functionalities needed for this application. The first step was to use Unified Modelling Language, UML, to show the functionalities of the application with the help of a diagram. (UML-diagrams.) using StarUML software. The Use Case diagram was the first diagram that was done and it can be seen down below in the FIGURE 1.1, the diagram is about the seller who is the main character and the user of the application, he has five attributes that he can manage with the usage of the application (sells, clients, supplier, products and purchases) each one of them can be added, edited, deleted by the user based on the needs.

The application has five main pages that can be navigated through and an example is listed down in the FIGURE 1. In this figure it can be seen that each item can be added, deleted or edited based on the user, and the list of purchases includes the purchase number, the item supplier, the product itself, the quantity and the total amount listed before and after the purchase.

Purchases management

ID

Supplier **ADD**

Product **DELETE**

Quantity **EDIT**

List of Purchases

Purchase Num	Supplier	Product	Quantity	Total

FIGURE 1-Purchase management

4.1 Use Case Diagram

The diagram of use cases serves in UML (Unified Modelling Language) to depict system functional requirements visually. The diagram illustrates how users who we call actors interact with the system to carry out required functionalities which we term use cases. The system functionality takes precedence over internal implementation details in use case diagrams. A user-friendly method for system requirement collection exists through diagrams which show how users interact with the system.(IBM 2023).

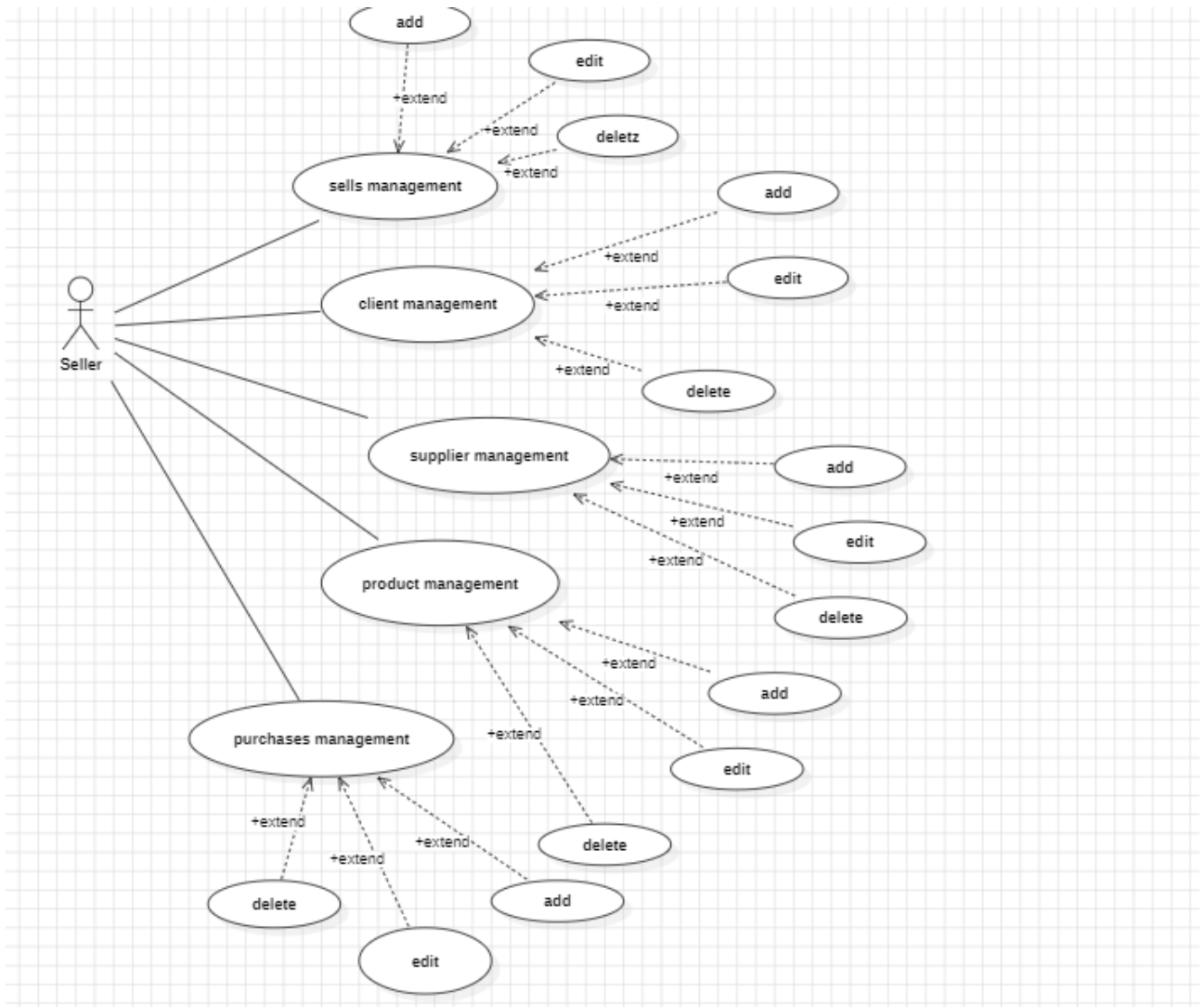


FIGURE 1.1 – Use Case Diagram

The software helps sellers manage their products and clients and suppliers and transactions with high efficiency. The primary framework of the application emerges from six core classes and their relationship model which contains Supplier, Product, Clients, Sells, Purchases along with their connecting data flow. The business operations split between Supplier and Clients for managing contact data and Product for inventory control as Sells and Purchases handle product movement tracking permanently. The system enables its users to carry out fundamental tasks for records such as creating, modifying and removing them. The use case diagram functions to illustrate how users predominantly sellers or admins will interact with the system. The primary business operations covered by the system involve creating new products, altering inventory numbers and maintaining client as well as supplier profiles and tracking sales and purchases while accessing historical transactions. All important business operations

should be accessible through a user-friendly interface through these use cases which assists daily operational practices. The class and use case diagrams present in a structured manner how users experience system features together with the system architectural overview.

4.2 Class Diagram

Software engineering employs this diagram to present system classes and interfaces together with their relationships. The diagram belongs to UML's static component as it does not handle temporal or dynamic elements. A class describes the responsibilities, behavior, and type of a set of objects. Every member of this set represents actual instances of the class. The semantic field links functions and data elements to form sets which contain both components and are named classes. Within object-oriented programming developers make use of classes. Modelling programs with classes helps split complex operations into multiple basic operational units. Through inheritance mechanisms classes demonstrate their parent and child association. The class diagram indicates different class relationships through specialized arcs between specific classes. (Lucid chart. 2023).

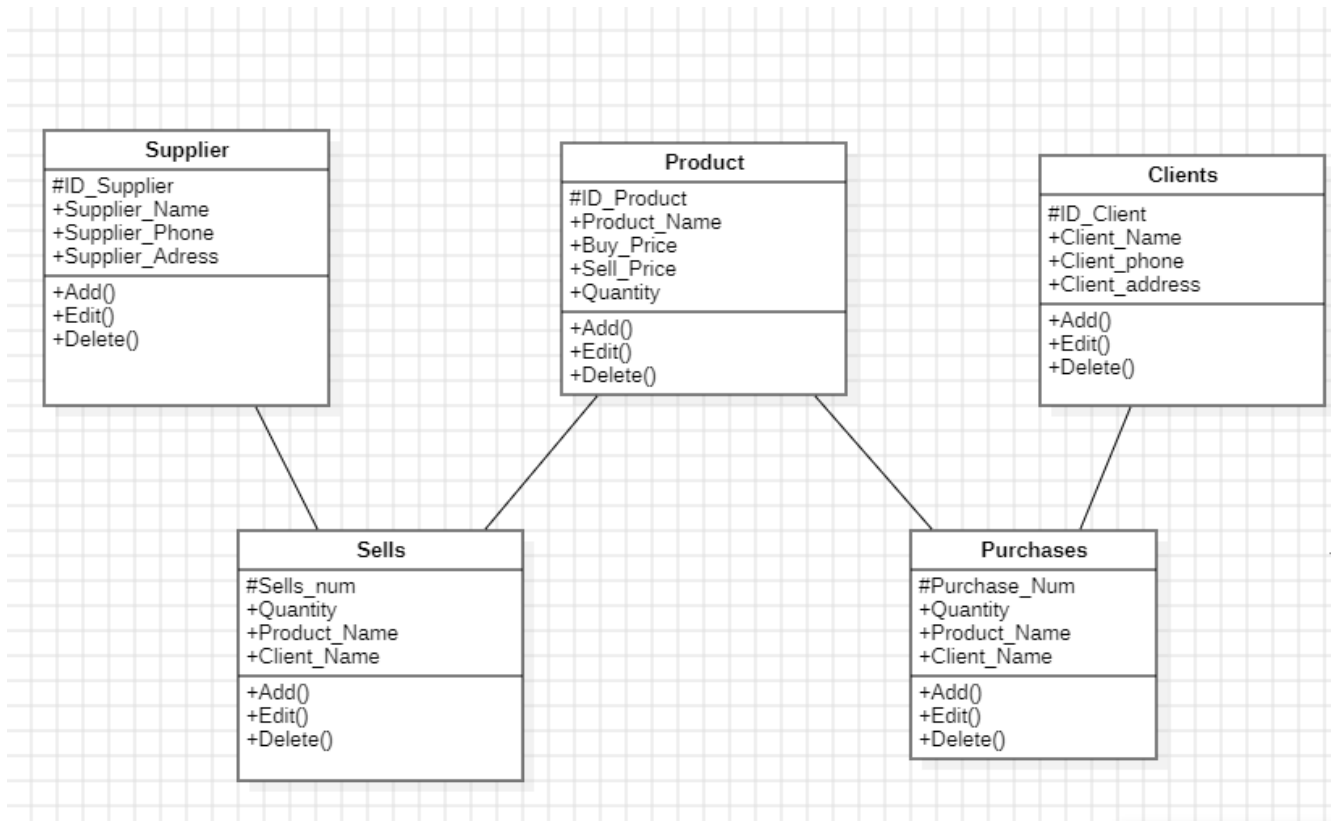


FIGURE 1.2 – Class Diagram

The defined class diagram demonstrates how a software application supports sellers during product organization and management tasks. The system includes six main classes named Supplier and Product together with Clients and Sells and finally Purchases in addition to their relationships. The Supplier class stores provider details that include their contact information and name together with their working address. The Product class controls both inventory description data along with buying prices and selling prices and stock levels. The Clients class stores customer information for transaction tracking. The application uses Sells and Purchases classes to record product transactions where Sells tracks client sales and Purchases handles restocking or procurement activities. This interconnected design allows users to follow all transactions from products to suppliers and finally to their relevant clients. This system design provides streamlined inventory management alongside improved client and supplier record keeping alongside better sales and purchase control.

4.3 CODE STRUCTURE

The code structure is seen below in FIGURE 2. This figure is about the introduction interface and the code is about the loading that is shown in FIGURE 2.1, This Java program creates a splash screen (a temporary loading window) before launching the main application. It visually represents the loading process using a progress bar and dynamic status messages. This is an example to show the code .

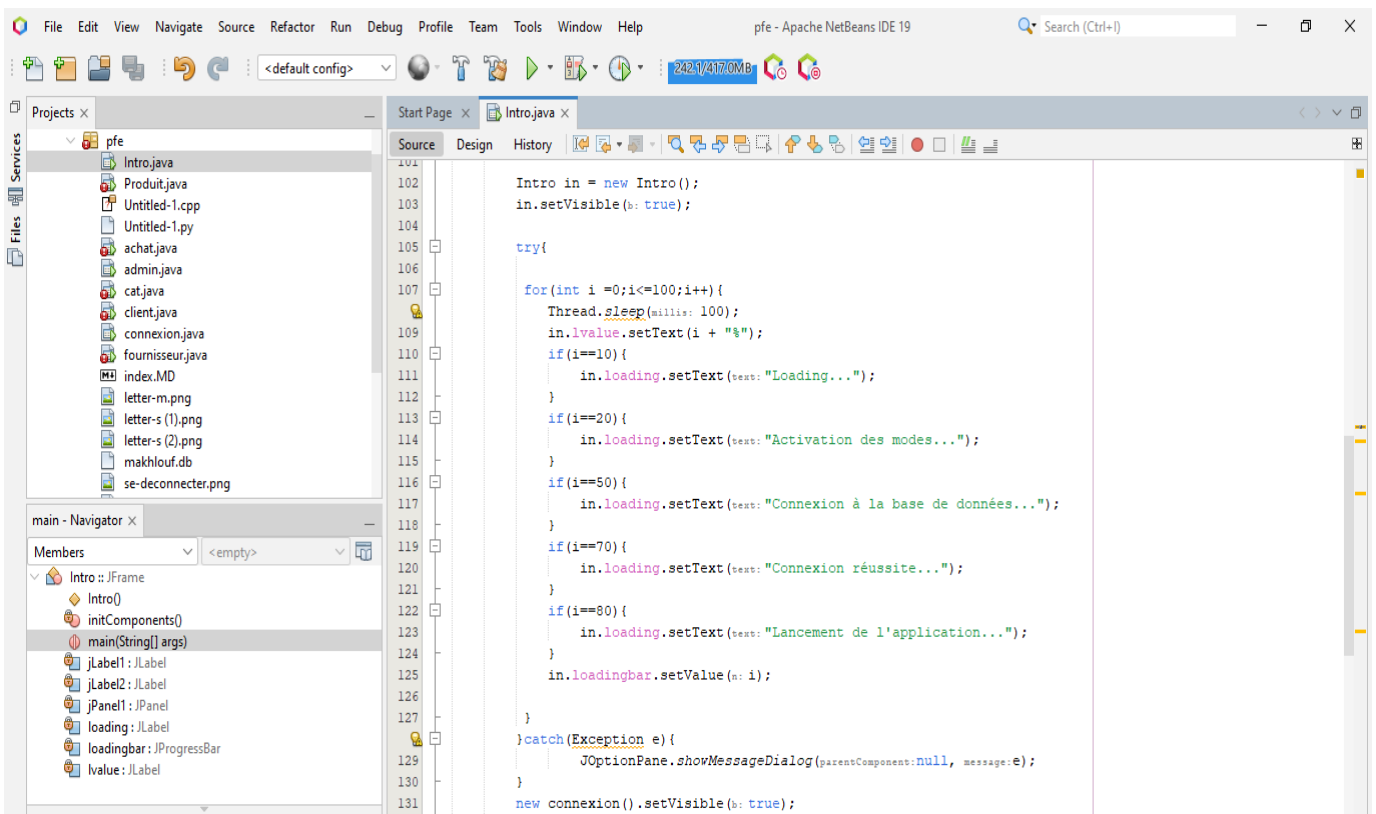


FIGURE 2 – Intro

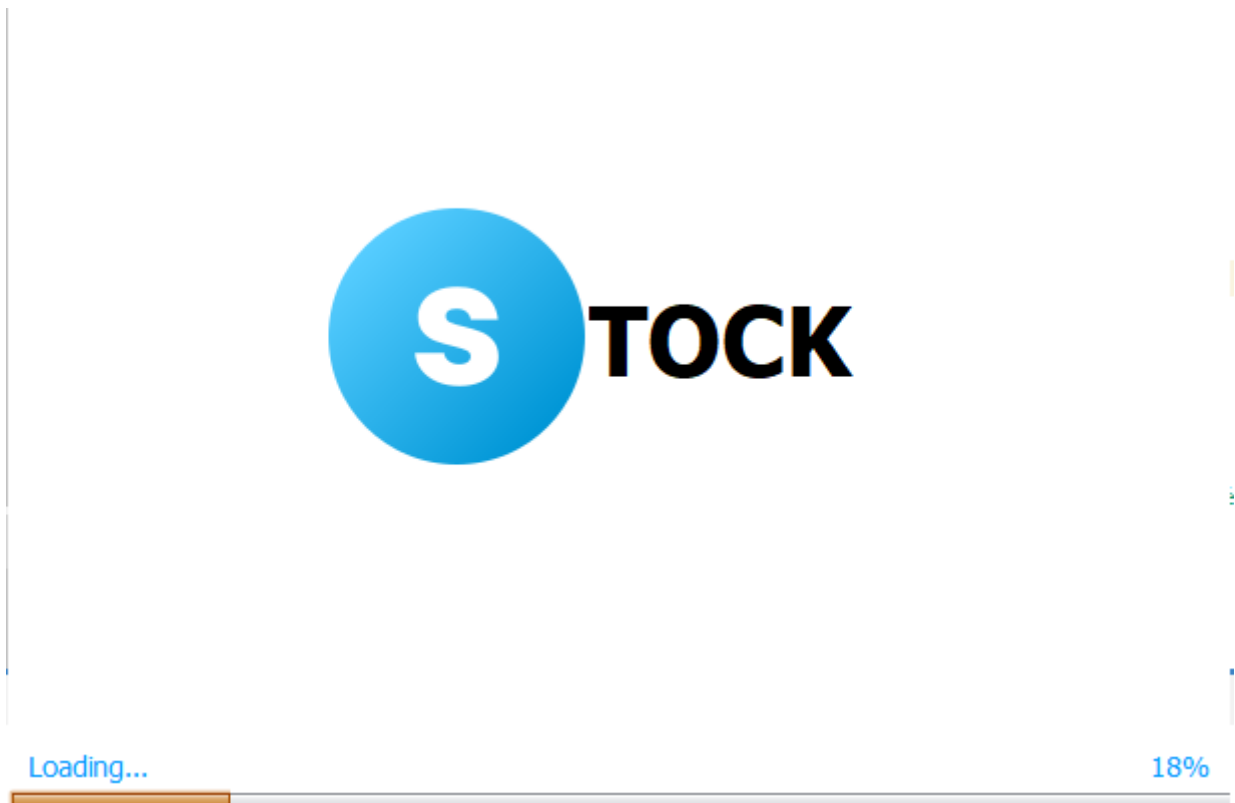


FIGURE 2.1 – Loading

4.3 Interacting with the database

```
9tableseller[CLEAR]CREATE TABLE "seller" (  
  "id" INTEGER PRIMARY KEY AUTOINCREMENT,  
  "nom" VARCHAR(20) NULL ,  
  "phone" VARCHAR(10) NULL ,  
  "address" VARCHAR(50) NULL  
etableproduct[CLEAR]CREATE TABLE "product" (  
  "id" INTEGER PRIMARY KEY AUTOINCREMENT,  
  "nom" VARCHAR(20) NULL ,  
  "price" INTEGER NULL ,  
  "qte" INTEGER NULL ,  
  "categorie" VARCHAR(20) NULL  
9tableclient[CLEAR]CREATE TABLE "client" (  
  "id" INTEGER PRIMARY KEY AUTOINCREMENT,  
  "nom" VARCHAR(20) NULL ,  
  "phone" VARCHAR(10) NULL ,  
  "address" VARCHAR(50) NULL
```

FIGURE 3– A look on the DataBase

This database structure contains three central entities these include sellers and products along with clients. A sellertable contains selling information for both individuals and entities that comprises names and phone numbers along with addresses. The producttable contains details of all offered products with their names and prices and quantities and categorization. Finally, the clienttable stores information about customers, such as their name, phone number, and address.

5 DEVELOPMENT PHASE

The application utilized Java as its main programming language to access capabilities of this platform-agnostic language while running its critical functions. An application data storage system based on SQLite together with Java functions as its programming framework. SQLite offers the main benefit of file-based storage along with lightweight operation for database management which means applications no longer require dedicated database servers. This feature gives the program broad usability since it works across all devices that contain the NetBeans integrated development environment thus enabling streamlined Deployment as well as simplified user experience. Java handled the application Logic and SQLite took responsibility for data management to create the basis of this project development.

5.1 Frontend

The interaction sequence for the application starts with a frontend UI which serves as the main tool for user communication. After users provide their authentication credentials that include individual usernames and passwords they can obtain entry to the system. The application displays an interface containing different attributes along with functionalities after users successfully complete their login. This interface architecture provides simple navigation control which enables users to examine available choices and execute their required tasks for the application framework. A well-designed frontend user interface enables smooth interactions with Java backend and SQLite database functions and accessible features.

Welcome

Connect to continue

User Name:

Password:

Connect **Leave**

FIGURE 4- A login interface

The application provides an uncomplicated yet friendly authentication system which enables users to enter the application. Users can access the login through two buttons and three input fields labeled username and password located within an interface that includes "Connect" and "Leave" buttons. Users of every knowledge level can easily navigate through the system because of the blue background and basic design layout.

6 EVALUATION AND DISCUSSION

This thesis aimed to develop a convenient useful desktop application which manages store products and inventory effectively. The main objective focused on building a tool that streamlines usual complicated work procedures which appear during inventory tracking and organization in a retail situation. The application will undergo extensive testing after development completion to evaluate its functionality as well as its usability features. The critical assessment phase will begin after which a discussion on necessary improvements and enhancements for future development will follow to optimize application functionality according to user needs.

6.1 Evaluation

The project execution met all main objectives because the system maintained operational stability during testing and development phases. The product owners received indispensable inventory management capabilities through the delivered application core features. The system design incorporated user-friendly navigation and users could perform important functions such as tracking products and entering data and tracking inventory without difficulty.

The base functionality of the system operates as expected but the present application version requires additional development work. User interface (UI) requires improvement to create better usability and attractive design for users. Modern users find the UI workable but it needs to update its visual attractiveness along with touch features which modern interfaces typically deliver.

Program development occurred through Java programming language within NetBeans Integrated Development Environment (IDE). The programming team chose Java since it provides dependable software features combined with industry-level programming capabilities. The process of mastering Java was difficult yet it enabled development of an application which remains dependable and expands in capacity. Development with NetBeans proved outstanding as an optimal platform especially for developers who are new to programming. Its accessible interface together with integrated components made the development process much simpler while providing users an easy experience. Developers obtain tools for GUI design together with graphical interface components like buttons and progress bars

through the IDE system. The streamlined development process made coding work more effective and time-efficient resulting in project success.

The present system successfully meets its core functions and delivers operational excellence but multiple future improvements will enhance its potential. A user-friendly interface together with added functionality will make the system competitive for broader adoption which will guarantee its long-term success.

6.2 Discussion

Future improvements remain available even after considering the previous information. The application presents room for improvement mainly through the development of its UI section. The existing UI maintains functional reliability yet it does not achieve complete usability standards that accommodate all users. The user experience would increase with an advanced and customizable program design. Future developments of the application need to build a suitable interface with visually pleasing design elements which meet accessibility needs of various users.

7 CONCLUSION

An application creation was the primary focus of this thesis work to help product owners monitor inventory and boost business operational performance. The selection of NetBeans as development environment happened because the interface provided userfriendly access to design tools through integrated features. The coding phase became a major obstacle despite the successful progress in the design phase. Competent mastery of Java programming language which serves as application implementation tool takes a substantial period to learn. Java knowledge was required as an essential component for reaching the stated objectives in this research work. The project managed to create an operational inventory management system despite the development hurdles that were encountered. The system improves fundamental business operations through inventory level tracking and product data entry and report generation which enhances appropriate decisions and organizational performance. The study solved realworld user needs by designing software that supported small to medium enterprises in resolving their practical challenges thereby proving software development success requires user need alignment. Developing software demands strict planning together with steady testing and sequential development which leads to superior software solutions according to this project's finish. The essential experience demonstrates that superior programming skills with added adaptation capabilities help companies solve unplanned problems. The software application demonstrates that technology based tools hold immense capabilities for business operation efficiency improvements.

REFERENCES

- Allen, G. 2010. NetBeans IDE Overview. Available at: <https://netbeans.apache.org/tutorial/main/kb/docs/ide/overview-screencast-smaller/>. (Accessed: 20.3.2025).
- CareerFoundry. (2023). The Difference Between UX and UI Design: A Layman's Guide. Retrieved from <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/> (Accessed: 17 February 2023).
- Codecademy Team. n.d. What is SQLite? Available at: <https://www.codecademy.com/article/what-is-sqlite>. (Accessed: 10.4.2025).
- Dhruv, S. 2024. How to Build a Desktop Application. Available at: <https://www.aalpha.net/blog/how-to-build-a-desktop-application/>. (Accessed: 20.3.2025).
- Gosling, J. 1996. The Java Programming Language. Available at: <https://www.amazon.com/Java-Programming-Language/dp/0201634554>.(Accessed: 20.3.2025).
- Hipp, D. R. 2023. SQLite: A Self-Contained, Serverless, Zero-Configuration, Transactional SQL Database Engine. Available at: <https://www.sqlite.org/about.html>. (Accessed: 20.3.2025).

IBM. 2023. Use-case diagrams in UML modeling. Available at: <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>. (Accessed: 20.3.2025).

Lucidchart. 2023. UML Class Diagram Tutorial. Available at: <https://www.lucidchart.com/pages/uml-class-diagram>. (Accessed: 20.3.2025).

The myVCF tool enables users to handle high-throughput mutation data through its desktop platform according to Pietrelli and Valenti (2017). *Bioinformatics* (Oxford, England), 33(22), 3676-3678. <https://doi.org/10.1093/bioinformatics/btx475>. (Accessed: 10.4.2025).

StarUML Team. (n.d.). StarUML – A Modern Software Modeling Tool. Retrieved from <https://staruml.io> (Accessed: March 27, 2025).