Inventory Management System

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted by

RIYA GOYAL
(2200290140127)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Komal Salgotra

Teaching Assistant

Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, GhaziabadUttar
Pradesh-201206

(MARCH 2024)

CERTIFICATE

Certified that **Riya Goyal (2200290140127)** has/ have carried out the project work having "**Inventory Management System**" (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Riya Goyal (2200290140127)

This is to certify that the above statement made by the candidate is correct to thebest of my knowledge.

Date:

Komal Salgotra   **Dr. Arun Tripathi**
 Assistant Professor   **Head**
Department of Computer Applications   **Department of Computer Applications**
KIET Group of Institutions, Ghaziabad   **KIET Group of Institutions, Ghaziabad**

<<Riya Goyal>>
ABSTRACT

The project aims to create a desktop-based application called Inventory Management System for inventory management of all organizations. Inventory Management System (IMS) refers to the processes and procedures that manage an organization's inventory through the collaboration of technology. The system can be used for storing content information, stock management, updating inventory according to sales content, creating stock and reporting information on a weekly or weekly basis. This project separates many aspects of sales and inventory management. In this system, we solve different problems related to direct sales management and purchasing management. Product management is important to ensure quality control for businesses that impact customers. Without proper inventory management, supermarkets can run out of essential items. Good inventory control will alert sellers when information needs to be collected. Inventory control is also an important part of inventory management. Automated inventory management systems help reduce errors when collecting inventory.

ACKNOWLEDGEMENTS

Table of Contents

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION TO INVENTORY MANAGEMENT SYSTEM

The project Inventory Management System is a complete desktop based application designed on .Net technology using Visual Studio Software. The main aim of the project is to develop Inventory Management System Model software in which all the information regarding the stock of the organization will be presented. It is an intranet based desktop application which has admin component to manage the inventory and maintenance of the inventory system.

This desktop application is based on the management of stock of an organization. The application contains general organization profile, sales details, Purchase details and the remaining stock that are presented in the organization. There is a provision of updating the inventory also. This application also provides the remaining balance of the stock as well as the details of the balance of transaction.

Each new stock is created and entitled with the named and the entry date of that stock and it can also be update any time when required as per the transaction or the sales is returned in case. Here the login page is created in order to protect the management of the stock of organization in order to prevent it from the threads and misuse of the inventory.

1.2 LITRATURE REVIEW

Products are considered as the business resources for the organization. This includes managing the product with appropriate way to review any time as per the requirement. Therefore it is important to have a computer based IMS which has the ability to generate reports, maintain the balance of the stock, details about the purchase and sales in the organization. Before developing this application we came up with 2 Inventory Management System existing in the market, which helps to give the knowledge for the development of our project. These application software are only used by the large organization but so we came up with the application which can be used by the small company for the management of their stock in the production houses. After analyzing the other inventory management system we decided to include some of common and key features that should be included in every inventory management system. So we decided to include those things that help the small organization in way or other.

1.3 PROBLEM STATEMENT

After analysing many existing IMS we have now the obvious vision of the project to be developed. Before we started to build the application team had many challenges. We defined our problem statement as:

To make desktop based application of IMS for small organization. To make the system easily managed and can be secured.

To cover all the areas of IMS like purchase details, sales details and stockmanagement.

1.4 OBJECTIVE OF THE PROJECT

1.4.1 Primary objective

The primary objectives of the project are mentioned below:

- To fulfill the requirement for achieving the Bachelor's degree of Computer Information System.
- To know the fundamentals of the .Net Technology and Visual Studio with the.Net Framework

1.4.2 Secondary objective

The secondary objectives of this project are mentioned below:

- To develop an application that deals with the day to dayrequirement of any production organization
- To develop the easy management of the inventory
- To handle the inventory details like sales details, purchase details and balance stock details.
- To provide competitive advantage to the organization.
- To provide details information about the stock balance.
- To make the stock manageable and simplify the use of inventory in the organization.

1.5 FEATURES OF PROJECT

This application is used to show the stock remaining and details about the sales and purchase. It gives the details about the stock on daily based and weekly based. The details components are described below:

- **Login page:** As application starts the login page appears. Admin loginis determined by the username and password that has all the authority to add, update and delete the stock of the organization as per the requirement.

- **Create Stock:** We can create stock if we need to extend or we havemore than one stock. We can create the stock along with the date.

- **Sales details:** It show the details about the sales and the remainingstock of sales. It also show the details about the sales in return.

- **Purchase details:** It shows the details about the purchase made by the organization along with the price and dates

1.6 SCOPE OF THE APPLICATION

Inventory Management System (IMS) is targeted to the small or mediumorganization which doesn't have many stock or warehouses i.e. only to those organization that has single power of authority.

Some of the scope are:

- Only one person is responsible in assigning the details or records
- It is security driven.
- Stock can be added as per the requirement.

CHAPTER 2
SYSTEM ANALYSIS

2.1 REQUIREMENT ANALYSIS

2.1.1 PROBLEM DEFINITION

Demand is frequently unpredictable in inventory systems, and lead times can often vary. Managers frequently keep a safety supply to minimize shortages. In such cases, it's difficult to say what order amounts and reorder points will result in the lowest total inventory cost. The inventory issue refers to the general issue of deciding how much inventory to keep on hand in expectation of possible demand. Loss occurs when a business is unable to meet demand or when commodities are stocked for which there is no demand.

2.1.2 PERFORMANCE REQUIREMENTS

The following performance characteristics should be taken care of while developing the system:

- **Security**: The system shall use appropriate security measures to protect data from unauthorized access.
- **Performance**: The system shall be able to handle a large number of users and transactions without experiencing significant performance degradation.
- **Scalability**: The system shall be scalable to accommodate future growth in the number of users and transactions.

2.2 SYSTEM REQUIREMENTS

2.2.1 Functional Requirements

- **Product Management**
- Users should be able to add, edit, and delete products.
- Users should be able to view product information, including product name, description, quantity, and price.

- **Inventory Management**
- The system should track inventory levels for all products.
- Users should be able to view inventory levels for individual products or for all products in a specific category.
- The system should generate reports on inventory levels, including low inventory reports and stock-out reports.

- **Purchase Order Management**
- Users should be able to create and manage purchaseorders.
- The system should generate purchase orders in PDFformat.
- The system should track the status of purchase orders and generate reports on purchase order history.

- **Sales Order Management**
- Users should be able to create and manage sales orders.
- The system should generate sales orders in PDF format.
- The system should track the status of sales orders and generate reports on sales order history

- **Customer and Vendor Management**
- Users should be able to add, edit, and delete customers and vendors.
- Users should be able to view customer and vendor information, including contact information and billing and shipping addresses.
- The system should generate reports on customer and vendor activity.

2.3 Hardware Requirements

- **Processor:** Core i3 7Gen
- **RAM:** 8GB
- **Hard Disk:** 128GB

2.4 Software Requirements

- Visual Studio Code

2.5 FEASIBILITY STUDY

2.5.1 ECONOMIC FEASIBILITY

The proposed "INVENTORY MANAGMENT" is economically feasible because

- The system requires very less time factors.
- The system will provide fast and efficient automated environment instead of slow and error prone manual system, thus reducing both time and man power spent in running the system.
- The system will have GUI interface and very less user training is required to learn it.
- The system will provide service to view various information for proper managerial decision making.

2.5.2 BEHAVIORAL FEASIBILITY

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. Therefore it is understandable that the introduction of a candidate system requires special efforts to educate and train the staff. The software that is being developed is user friendly and easy to learn. In this way, the developed software is truly efficient and can work on any circumstances, tradition, locales.

2.5.3 TECHNICAL FEASIBILITY

Technical feasibility centers around the existing computer system (Hardware and Software etc) and to what extend it support the proposed addition. For example, if the current computer is operating at 80 percent capacity - an arbitrary ceiling - then running another application could overload the system or require additional Hardware. This involves financial considerations to accommodate technical enhancements. If the budgets is a serious constraint, then the project is judged not feasible. In this project, all the necessary cautions have been taken care to make it technically feasible. Usinga key the display of text/object is very fast.

2.6 ARCHITECTURAL REVIEW

This desktop based application is based on 3-tier architecture of .Net Framework. The 3-tier includes the three hierarchy of the flowof programming logic from user interface to database and again database to user interface with the desired information requested by the clients. In between there involves the logic layer for effectively and correctly manipulating

the request. The 3-tier includes the following :-

2.6.1 Client tier

The visual part is implemented using all kinds of swing components, which does not make database calls. The main function of this tier is to display information to the user upon user's request generated by user's inputs such as firing button events. For example, inventory list will display when user click "display" button if he or she wants to know the list of stock remaining in the organization.

2.6.2 Business Logic tier

The middle tier, business logic, is called by the client to make database queries. It provides core function of the system as well as connectivity to the data tier, which simplify tasks that were done bythe clients tier.

2.6.3 Database tier

Data layer is also the class which gets the data from the businesstier and sends it to the database or gets the data from the database and sends it to business tier. This is the actual DBMS access layer or object layer also called the business object. The database backend stores information which can be retrieved by using the MySQL Connectivity. MySQL connectivity is used to manage the communication between the middle tier and the backend database by issuing complex database queries.

CHAPTER 3

DESIGN

Designing is the most important phase of software development. It requires a careful planning and thinking on the part of the system designer. Designing software means to plan how the various parts of the software are going to achieve the desired goal. It should be done with utmost care because if the phasecontains any error then that will effect the performance of the system, as a resultit may take more processing time, more response time, extra coding workload etc.

Software design sits at the technical kernel of the software engineering process and is applied regardless of the software process model that is used. After the software requirements have been analyzed and specified, software design is the first of the three technical activities Designing, Coding and Testing that are required to build and verify the software. Each activity transforms informationin such a manner that ultimately results in validated computer software.

3.1 DESIGN GOALS

The following goals were kept in mind while designing the system:

- Make system user-friendly. This was necessary so that system could be used efficiently and system could act as catalyst in achieving objectives.
- Make system compatible i.e. It should fit in the total integrated system. Future maintenance and enhancement must be less.
- Make the system compatible so that it could integrate other modules of system into itself.
- Make the system reliable, understandable and cost-effective.
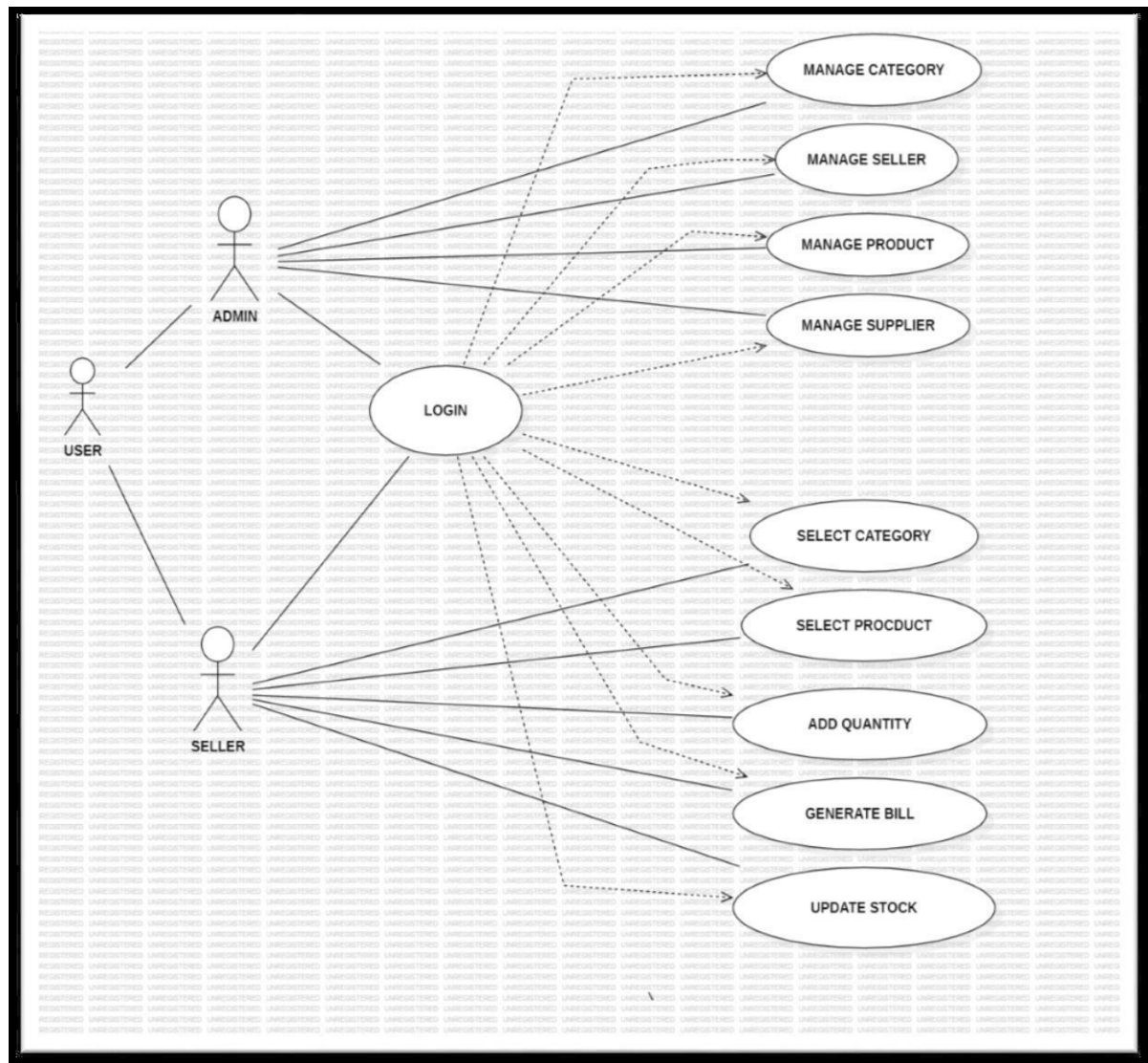
3.2 USE CASE DIAGRAM

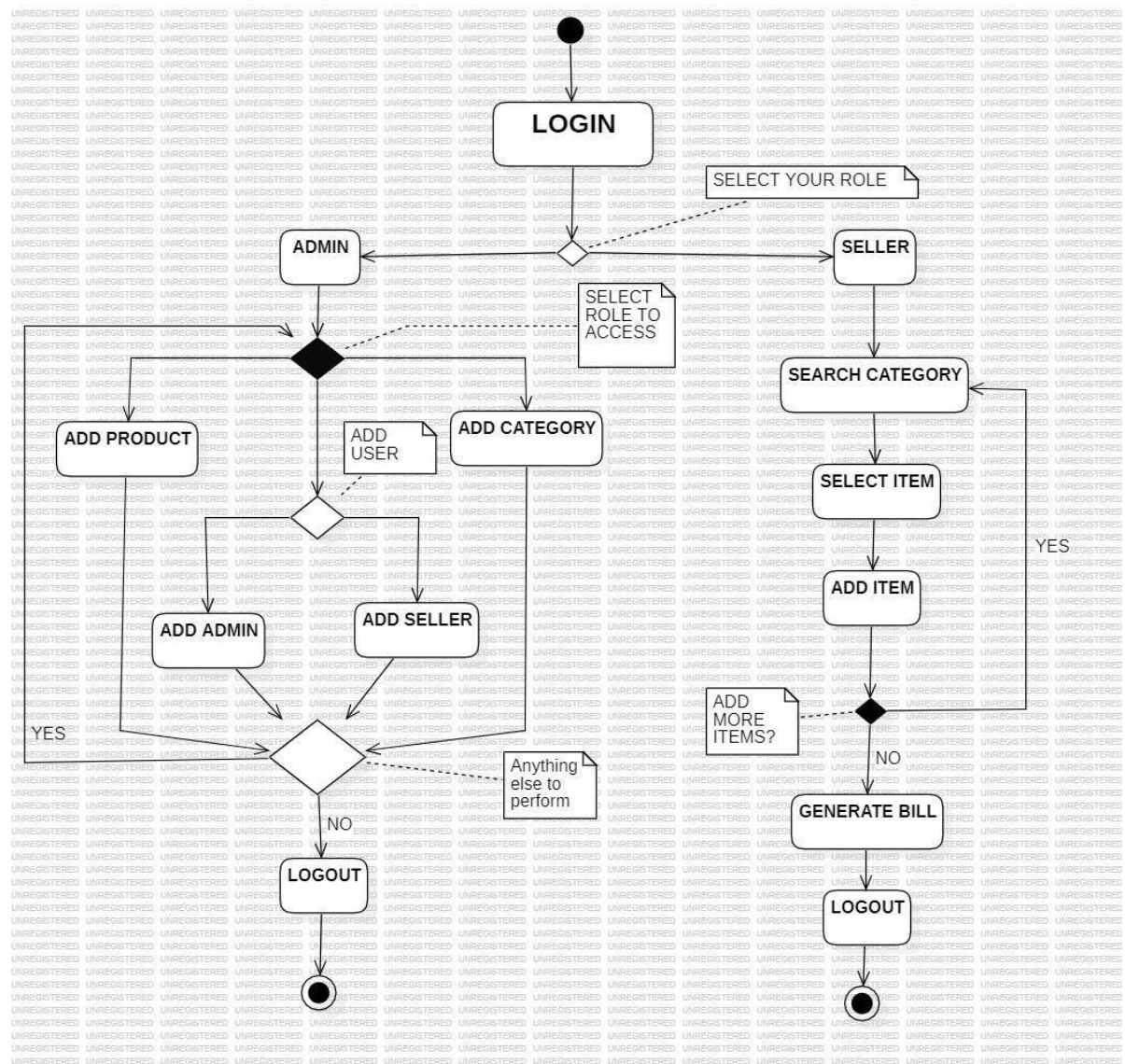**Fig 3.2 Use Case Diagram**

3.3 FUNCTIONAL FLOW OF THE SYSTEM

Fig 3.3 Functional Flow of System

3.4 ENTITY RELATIONSHIP DIAGRAM
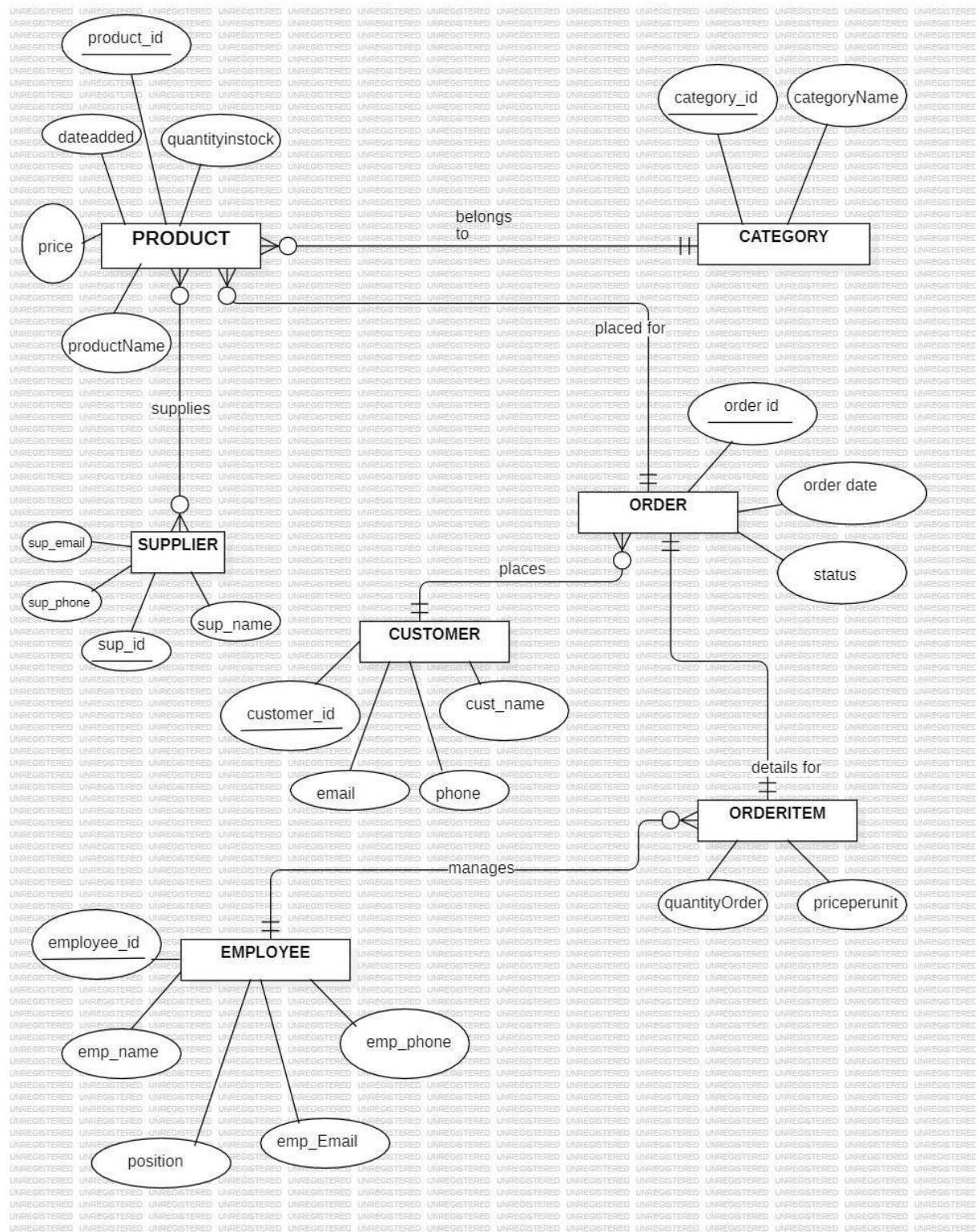

Fig 3.4 Entity Relationship Diagram

3.5 DATA FLOW DIAGRAM

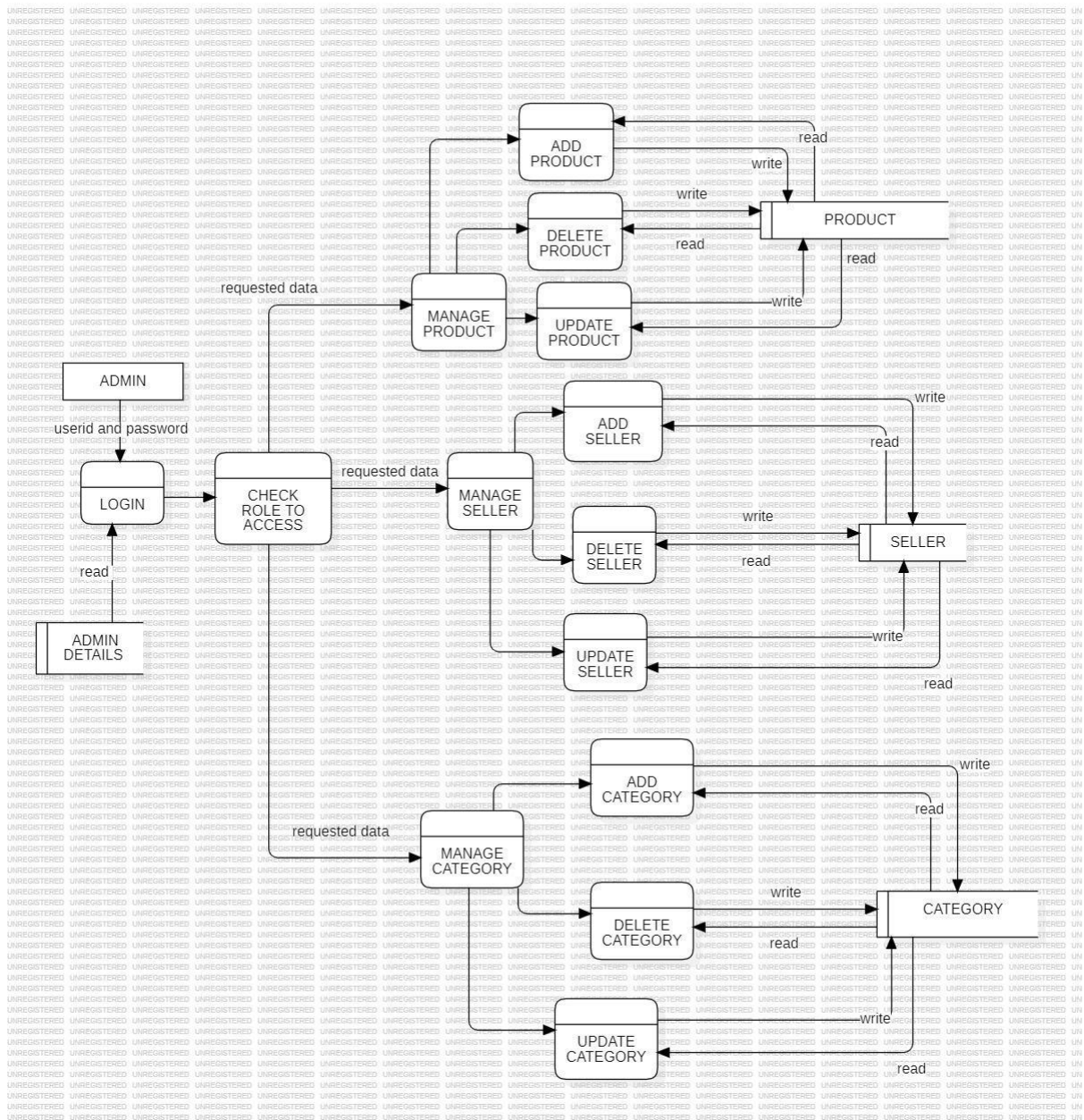Data Flow diagram of individual processes:

1.   Admin Process

Fig 3.5 Data Flow Diagram

CHAPTER 4

TECHNOLOGY USED

4.1 HARWARE REQUIREMENTS

S. N.	Description
1.	PC with 5 GB or more Hard disk.
2.	PC with 2 GB RAM.
3.	PC with core i3 or above processor.

4.1.1PC with 5 GB or more Hard disk:

This specifies the storage requirement for the PC. It should have a hard disk with a capacity of 5 gigabytes (GB) or more. This is where you store your operating system, software applications, and data.

4.1.2 PC with 2 GB RAM:

This sets the minimum random-access memory (RAM) requirement for the PC. It should have at least 2 gigabytes of RAM. RAM is essential for running applications and the operating system efficiently.

4.1.3PC with core i3 or above processor:

This specifies the processor requirement for the PC. It should have an Intel Core i3 processor or a more powerful one. The processor is a crucial component that determines the computer's overall speed and performance.

4.2 SOFTWARE REQUIREMENTS

4.2.1 Microsoft visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Form applications, websites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Window, Windows Mobile,Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. Microsoft Visual Studio simplifies the basic tasks of creating, debugging and deploying applications.

Microsoft Visual Studio comes with .NET Framework and supports applications targeting Windows. It supports IBM DB2 and Oracle databases, in addition to Microsoft SQL Server. It has integrated support for developing Microsoft Silverlight applications, including an interactivedesigner. Microsoft Visual Studio offers several tools to make parallel programming simpler: in addition to the Parallel Extensions for the .NET Framework and the Parallel Patterns Library for native code, Visual Studio includes fools for debugging parallel applications.

4.2.2 Microsoft SQL server Management Studio Express

Microsoft SQL Server Management Studio Express (SSMSE) provides a graphical management tool for SQL Server Express Edition. SSMSE user interface is a subset of SQL Management Studio that is available with other editions of SQL Server. SSMSE call also manage instance of the SQL Server Database Engine created by any edition of SQL Server. Inventory Management System is developed using Microsoft SOL Server 2023.

4.2.3 .NET Framework 4.5

The .NET Framework is a development platform for building apps for Windows, Windows Phone, Windows Server, and Microsoft Azure. It consists of the common language runtime (CLR) and the .NET Framework class library, which includes classes, interfaces, and value types that support an extensive range of technologies. The .NET Framework provides a managed execution environment, simplified development and deployment, and integration with a variety of programming languages, including Visual Basic and Visual C#.

4.3 TECHNOLOGY USED

4.3.1.NET Framework Structure

The .Net architecture is basically segregated in to three layers namely top, middle and bottom layer. The bottom layer is CLR, it is the heart of .NET Framework which provides the runtime environment in which programs are executed. The middle layer comprises the next generation of standard system services are brought under the control of the framework, making them universally available and standardizing their usage across languages. The top layer includes user and program interfaces as figure:
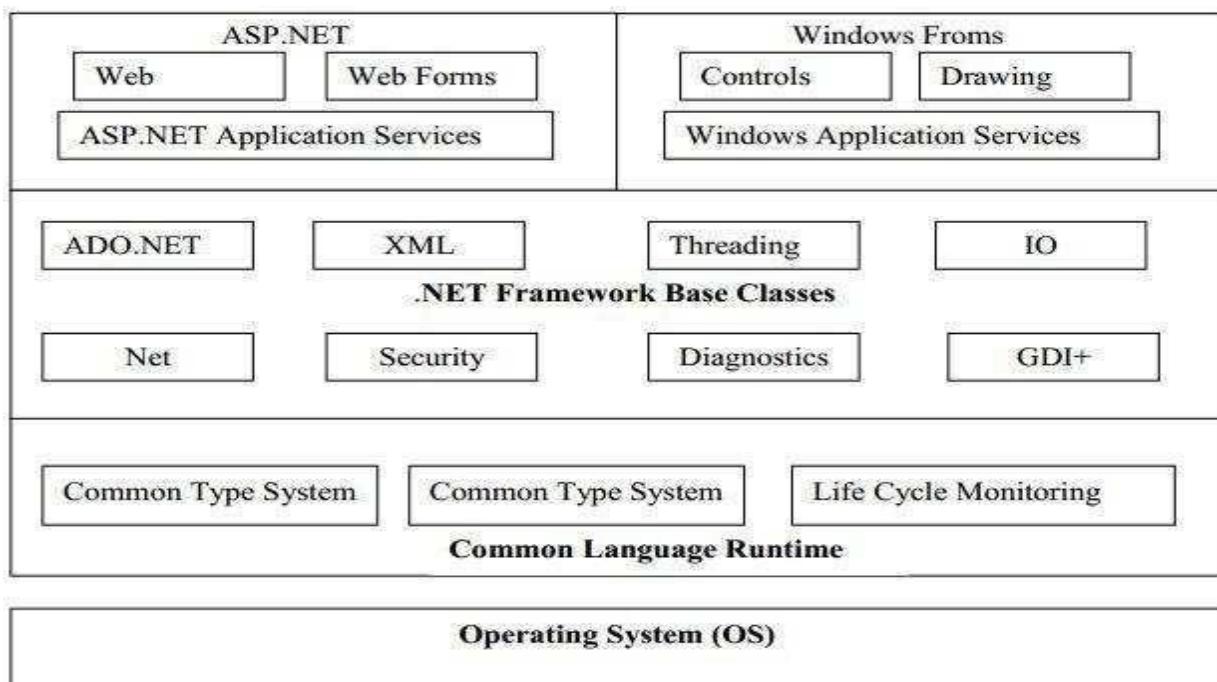
Fig 4.3.1 .Net Framework Architecture

The first thing that you should notice when looking at this diagram is that the .NET Framework sits on top of the operating system. There has also been a lot of talk about

.NET being ported over by some third-party companies so that a majority of the .NET Framework could run on other platforms as well.

At the base of the .NET Framework is the Common Language Runtime (CLR). The CLR is the engine that manages the execution of the code. The next layer up is the

.NET Framework Base Classes. This layer contains classes, value types, and interfaces that you will use often in your development process. Most notably within the .NET Framework Base Classes is ADO.NET, which provides access to and management of data.

The third layer of the framework is ASP.NET and Windows Forms. ASP.NET should not be viewed as the next version of Active Server Pages after ASP 3.0, but as a dramatically new shift in Web application development.

4.3.2 The .NET Language

In the past, you chose the development language for an application based upon the functionality that you were trying to perform. Some languages were more powerful than others, but at the same time they might have required a higher level of understanding and were, in most cases, more difficult to program in.

To be part of the .NET Framework, a language only has to follow certain rules. The biggest and most important rule for inclusion is that the language needs to be an object-oriented language. Microsoft providesfour languages with the .NET

4.3.3 FRAMEWORK:

- Visual Basic
- C#.NET

4.3.4 The Connection Object

The connection object created the connection to the database. Microsoft Visual Studio .NET provides two types of connection classes: the SQL connection object, which is designed specifically to connect to Microsoft SQL Server.

4.4 MICROSOFT SQL SERVER

Microsoft SQL Server is an application used to create computer databases for the Microsoft Windows family of server operating systems. Microsoft SQL Server provides an environment used to generate database that can be accessed from workstations, the Internet, or other media such as a personal digital assistant (PDA). Microsoft SQL Server is used to create desktop, enterprise, and web-based database applications. It is used at different levels and with various goals.

SQL Server makes simpler and easier to deploy, manage, and optimize enterprise data and analytical applications. An enterprise data management platform, it performs a single management console that enables data administrators anywhere in your organization to

monitor, manage, and tune all of the databases and associated services across your enterprise. It provides an extensible management infrastructure that can be easily programmed by using SQL management objects, enabling users to customize and extend their management environment and independent software vendors to build additional tools and functionality to further extend the capabilities that come out of thebox.

CHAPTER 5

TESTING AND DEBUGGING

Software testing is a critical element of the ultimate review of specification design and coding. Testing of software leads to the uncovering of errors in the software functional and performance requirements are met. Testing also provides a good indication of software reliability and software quality as a whole. The result of different phases of testing are evaluated and then compared with the expected results. If the errors are uncovered, they are debugged and corrected. A strategy approach to software testing has the generic characteristics:

- Testing begins at the module level and works "outwards" towards the integration of the entire computer-based system.

- Different testing techniques are appropriate at different points of time.

- Testing and debugging are different activities, but debugging must be accommodated in the testing strategy

5.1 UNIT TESTING

The module interface is tested to ensure that information properly flows into and out of the program unit under test. The unit testing is normally considered as an adjunct step to coding step. Because modules are not a standalone program, drivers and/or stubs software must be developed foreach unit. A driver is nothing more than a "main program" that accepts test cases data and passes it to the module. A stub serves to replace the modules that are subordinate to the modules to be tested. A stub may do minimal data manipulation, prints verification of entry and returns.

Approaches used for Unit Testing were:
- **Functional Test**: Each part of the code was tested individually and the panels were tested individually on all platforms to see if they are workingproperly.

- If they all work individually, they should work when we put them together. The problem of course is "putting them together". This can be done in two ways:

- **Top-down integration:** Modules are integrated by moving downwards through the control hierarchy, beginning with main control module are incorporated into the structure in either a depth first or breadth first manner.

- **Bottom-up integration:** It begins with construction and testing with atomic modules i.e. modules at the lowest level of the program structure. Because modules are integrated from the bottom up, processing required for the modules subordinate to a given level is always available and the need of stubs is eliminated.

5.1.1 Testing includes Verification and Validation

- **Verification :-** is a process of confirming that software meets its specification.
- **Validation :-** is the process of confirming that software meets the customer's requirements.

5.2 SYSTEM TESTING

System testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software.

5.2.1 System Testing Process:

System Testing is performed in the following steps:

- Test Environment Setup: Create testing environment for the better quality testing.
- Create Test Case: Generate test case for the testing process.
- Create Test Data: Generate the data that is to be tested
- Execute Test Case: After the generation of the test case and the test data, test cases are executed.
- Defect Reporting: Defects in the system are detected.
- Regression Testing: It is carried out to test the side effects of the testing process.
- Log Defects: Defects are fixed in this step.
- Retest: If the test is not successful then again test is performed.

5.3 ACCEPTANCE TESTING

It is formal testing according to user needs, requirements, and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers, or other authorized entities to determine whether to accept the system or not.

Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use.

5.4 DEBUGGING

Debugging occurs as a consequence of successful testing i.e. when a test case uncovers an error, debugging is the process that results in identifying the location of error ad the removal of error. The poorly understood mental process that connects a symptom to cause is debugging.

This process will always have one of the two outcomes.

- The cause will be found, corrected and then removed or
- The cause will not be found. In the latter case the person performing debugging may suspect a cause, design a test case to help validate his suspicion, and then work towards the correction of errors in the interactive fashion.

Following three approaches of debugging were used:

- Debugging by Induction
- Debugging by Deduction
- Backtracking

CHAPTER 6

IMPLEMENTATION

Once the system was tested, the implementation phase started. A crucial phase in the system development life cycle is successful implementation of new system design. Implementations simply mean converting new system design into operation. This is the moment of truth the first question that strikes in every one's mind that whether the system will be able to give all the desires results as expected from system. The implementation phase is concerned with user training and file conversion.

The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of computer system Implementation is used here to mean the process of converting a new or revised system design into an operational one. Conversion is one aspect of implementation. The other aspects are the post implementation review and software maintenance.

There are three types of implementations

- Implementation of a computer system to replace a manual system

- Implementation of a new computer system to replace an existing one

- Implementation of a modified application to replace an existing one.

6.1 MODULES

In computer software, a module is an extension to a main program dedicated to a specific function. In programming, a module is a section of code that is added in as a whole or is designed for easy reusability

The proposed system of "Online Inventory Management System" has thefollowing modules

1. Login
2. Admin

6.2 FLOWCHARTS OF MODULES

6.2.1 FLOWCHART OF LOGIN MODULE
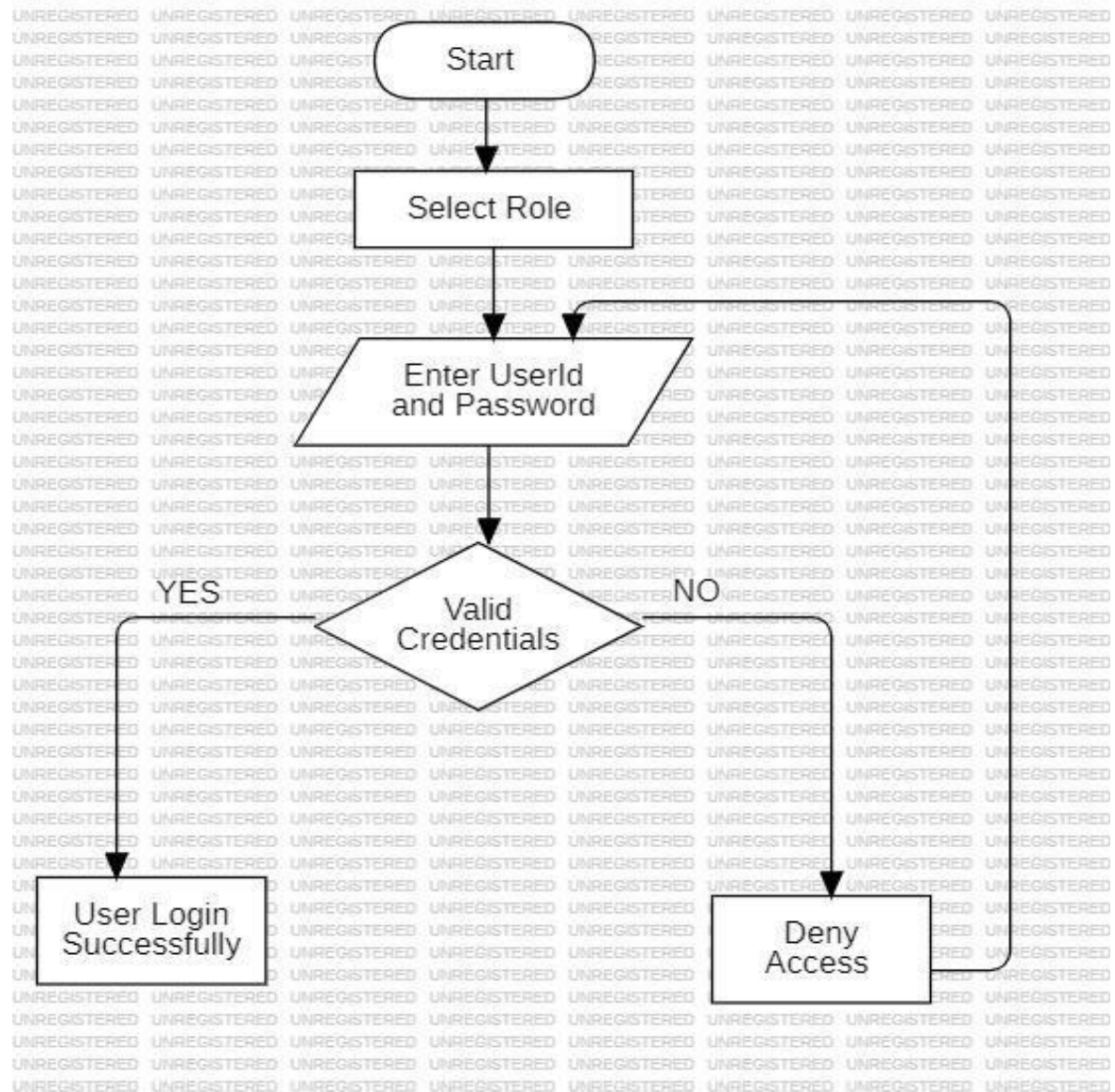
6.2.2 FLOWCHART OF ADMIN MODULE
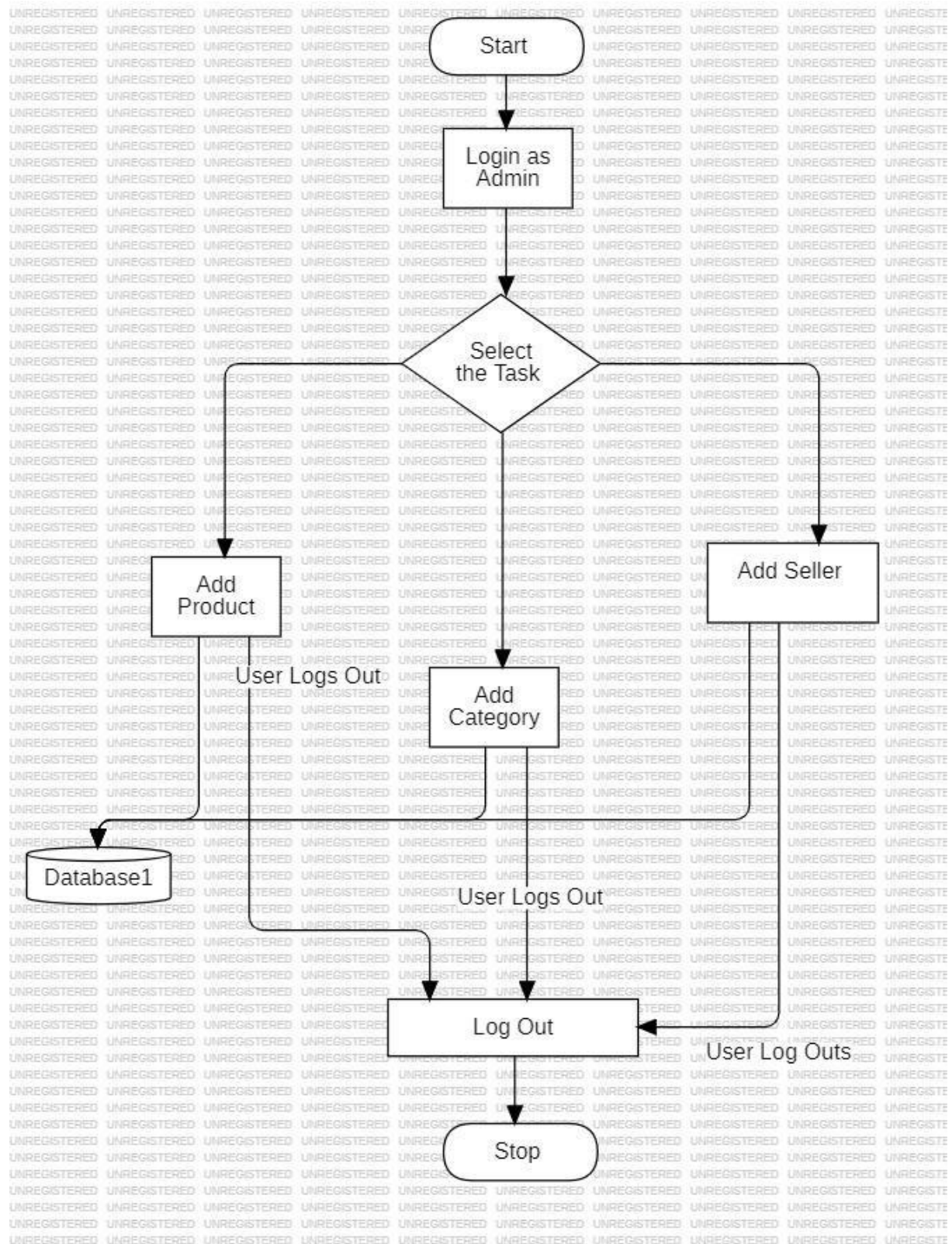
6.2.3 FLOWCHART OF SELLER MODULE
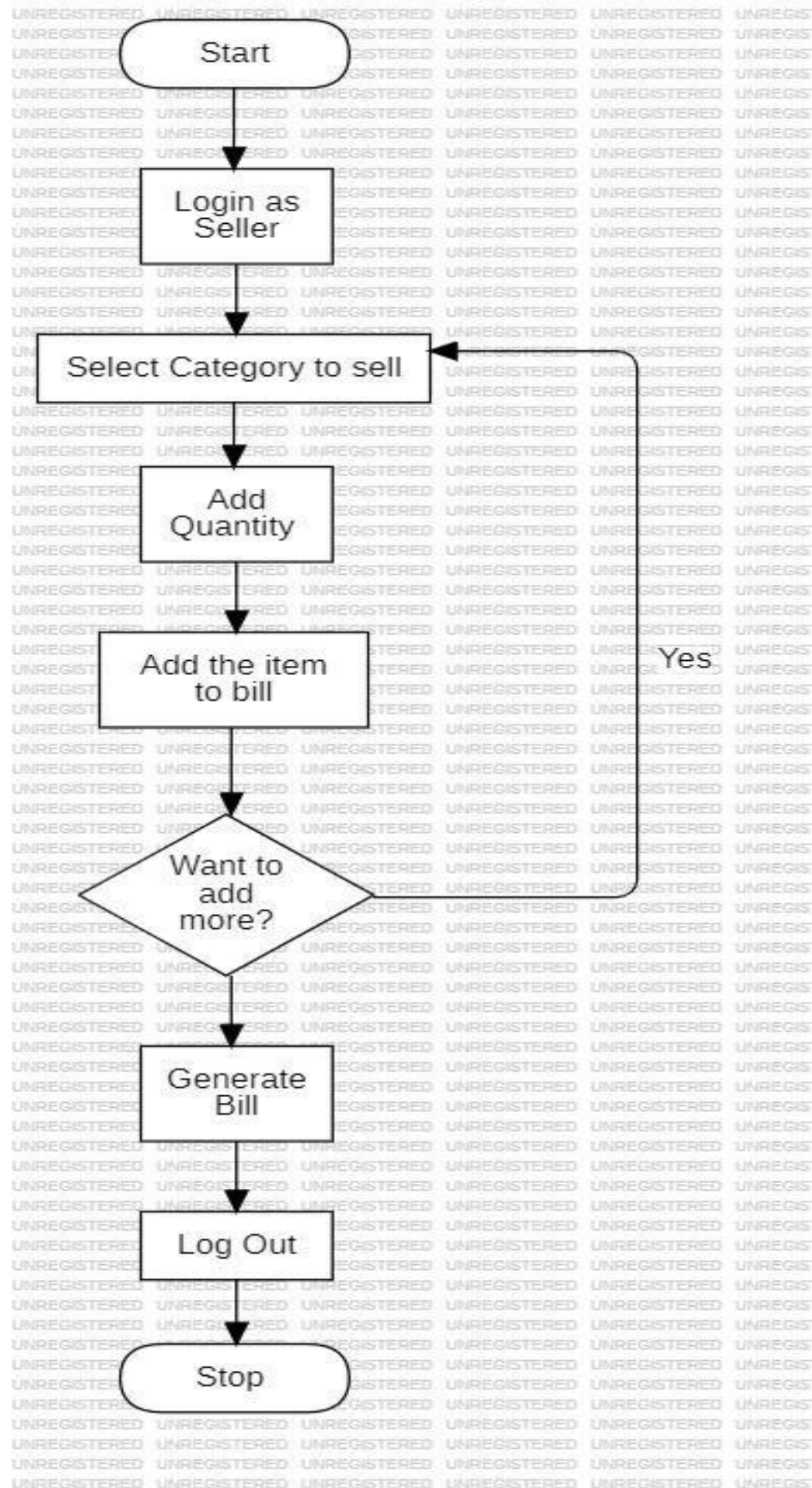
CHAPTER 7
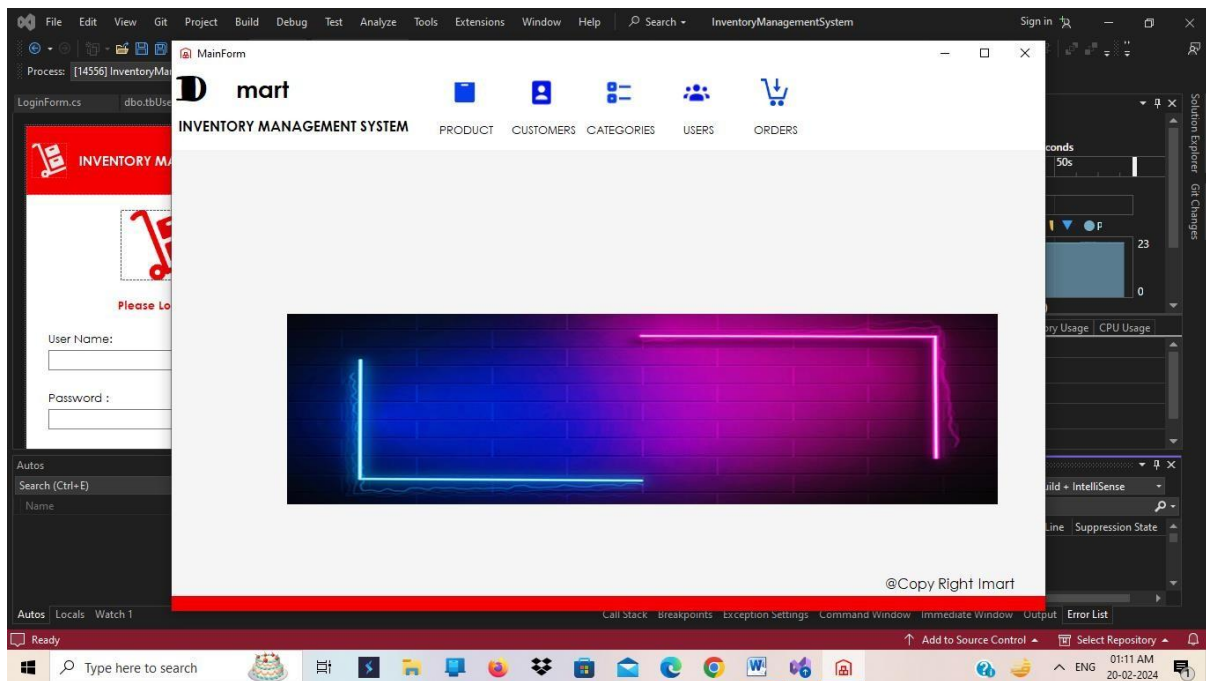
PROJECT SCREENSHOTS

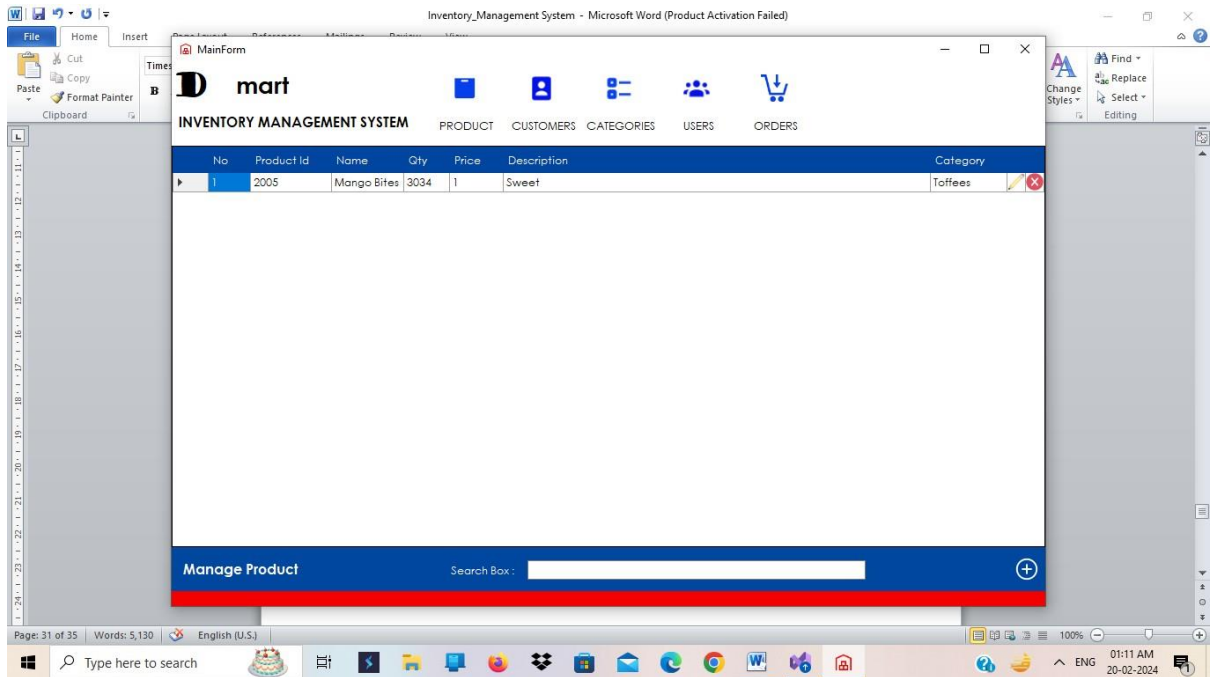## 7.1 LOGIN PAGE:
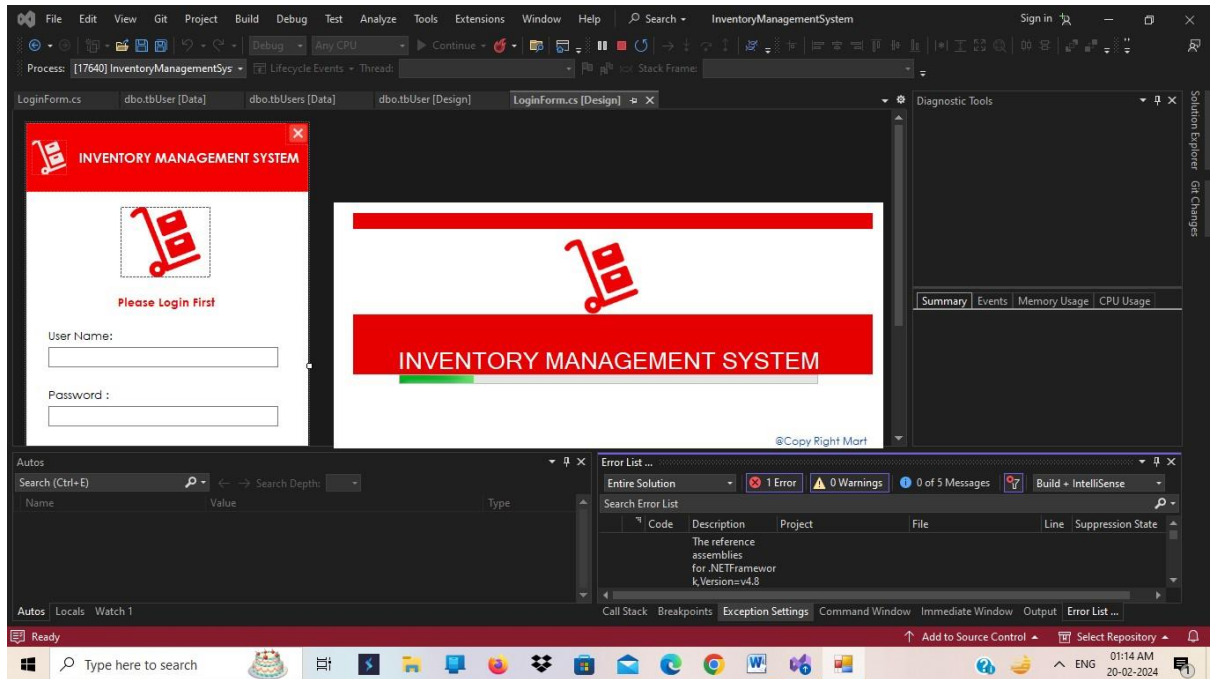
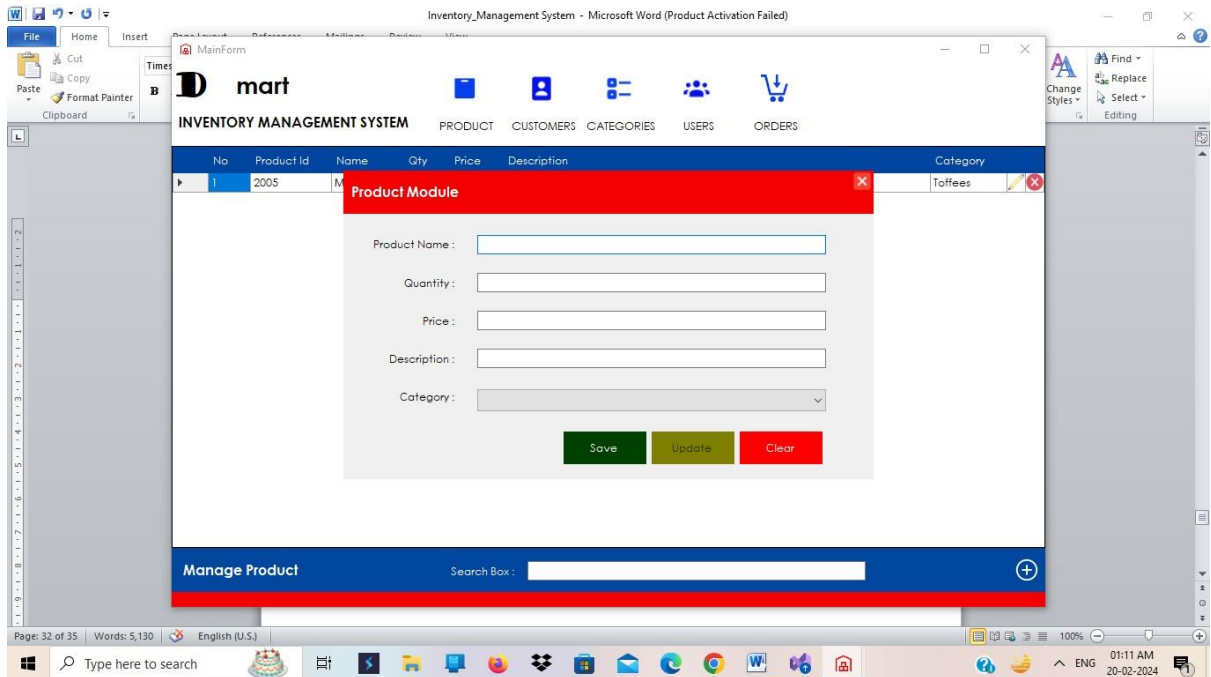## 7.2 LOGIN PAGE ACCESS GRANTED:
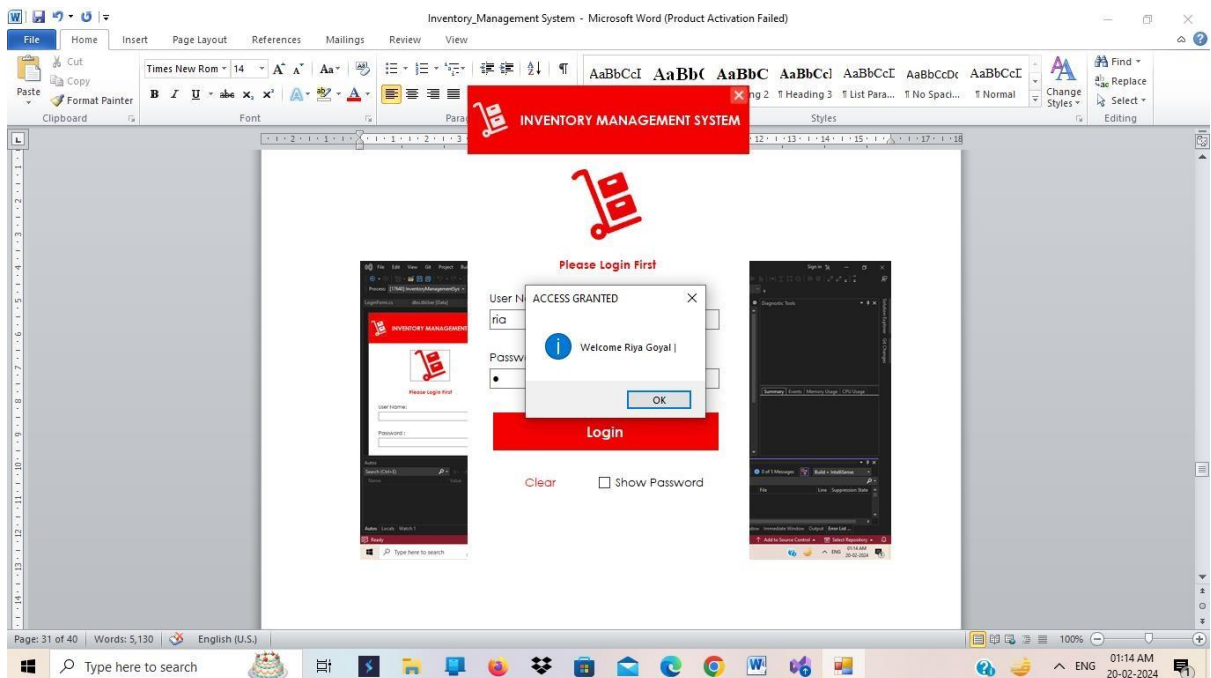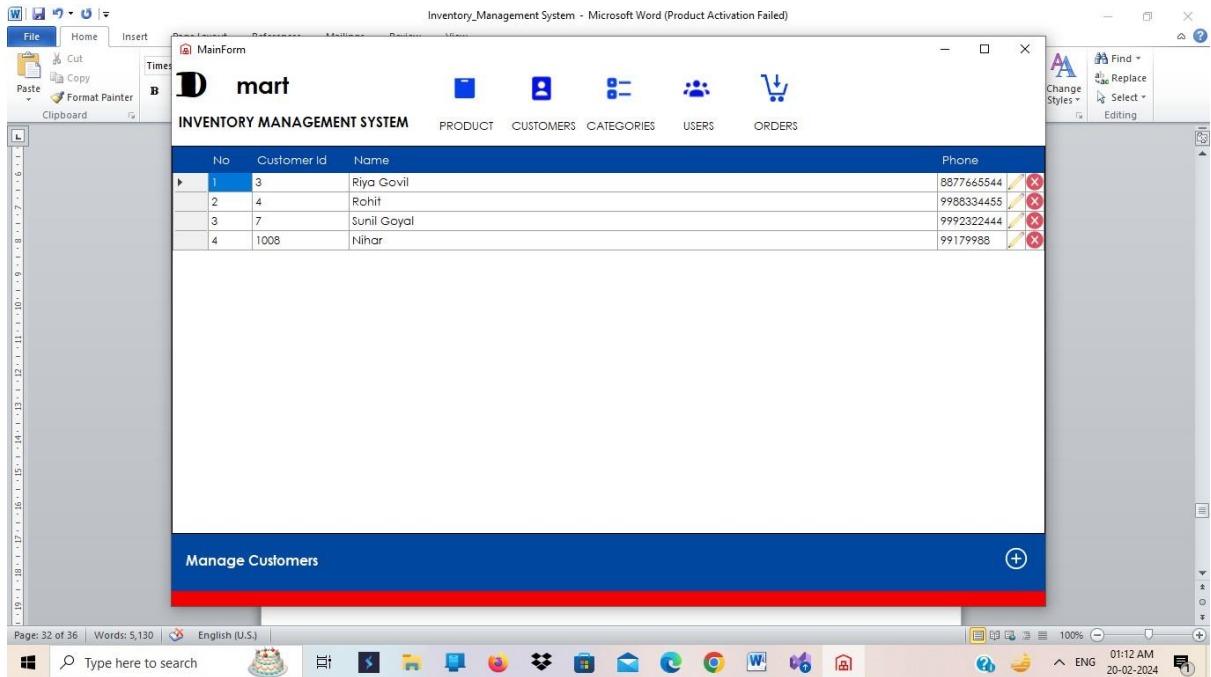
7.3 EXIT FROM LOGIN PAGE:
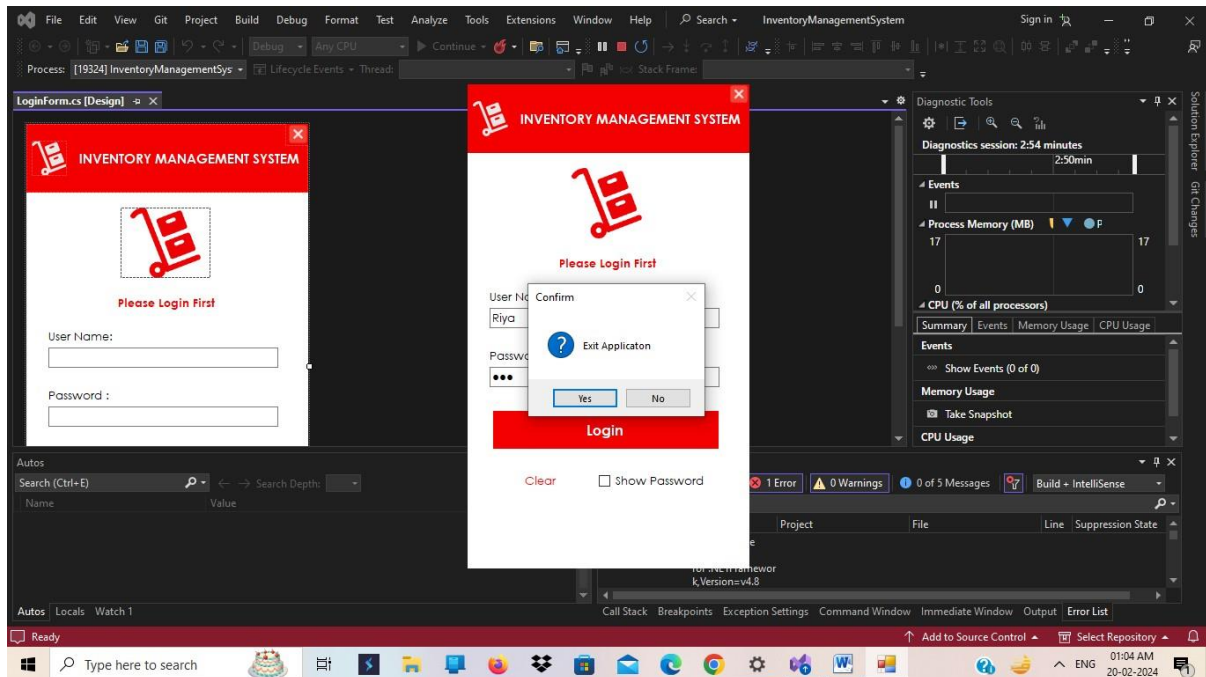
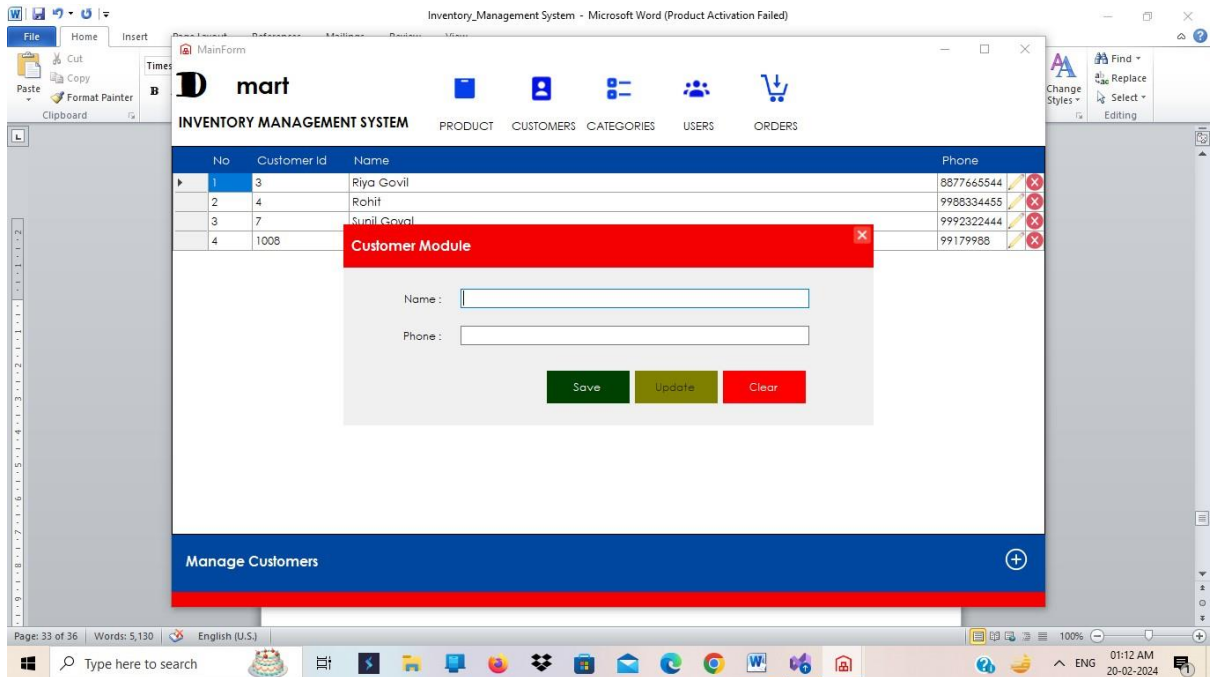## 7.4 INVENTORY:

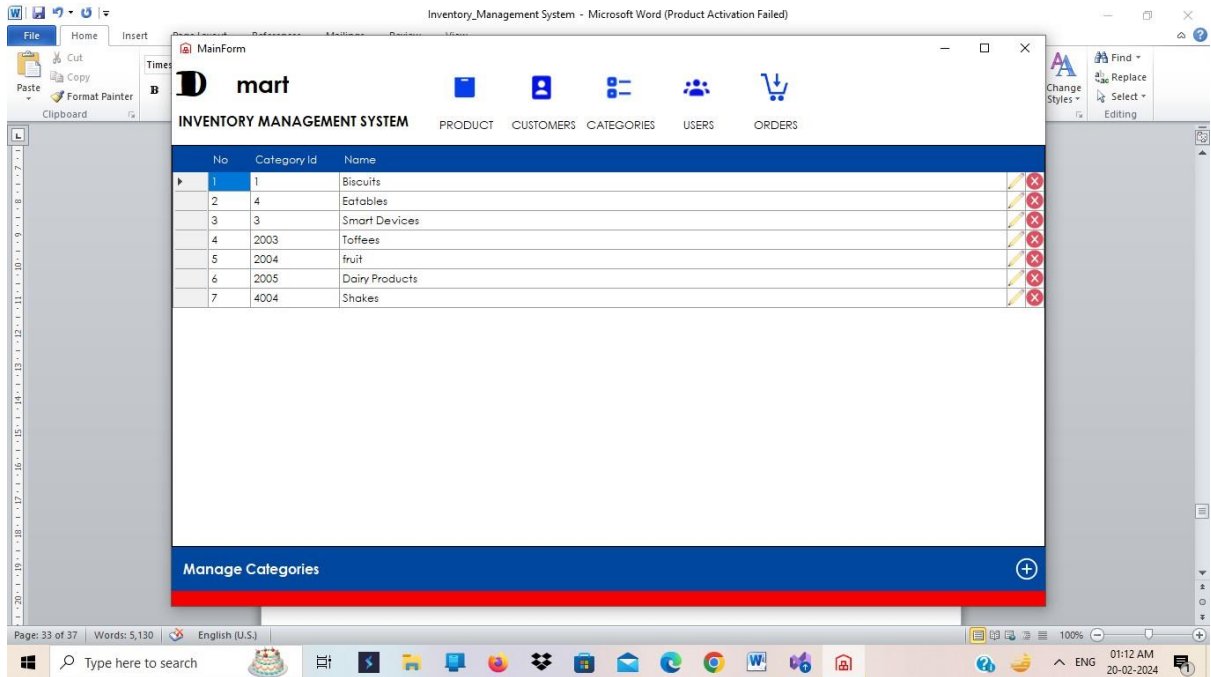## 7.5 PRODUCT IN THE INVENTORY:
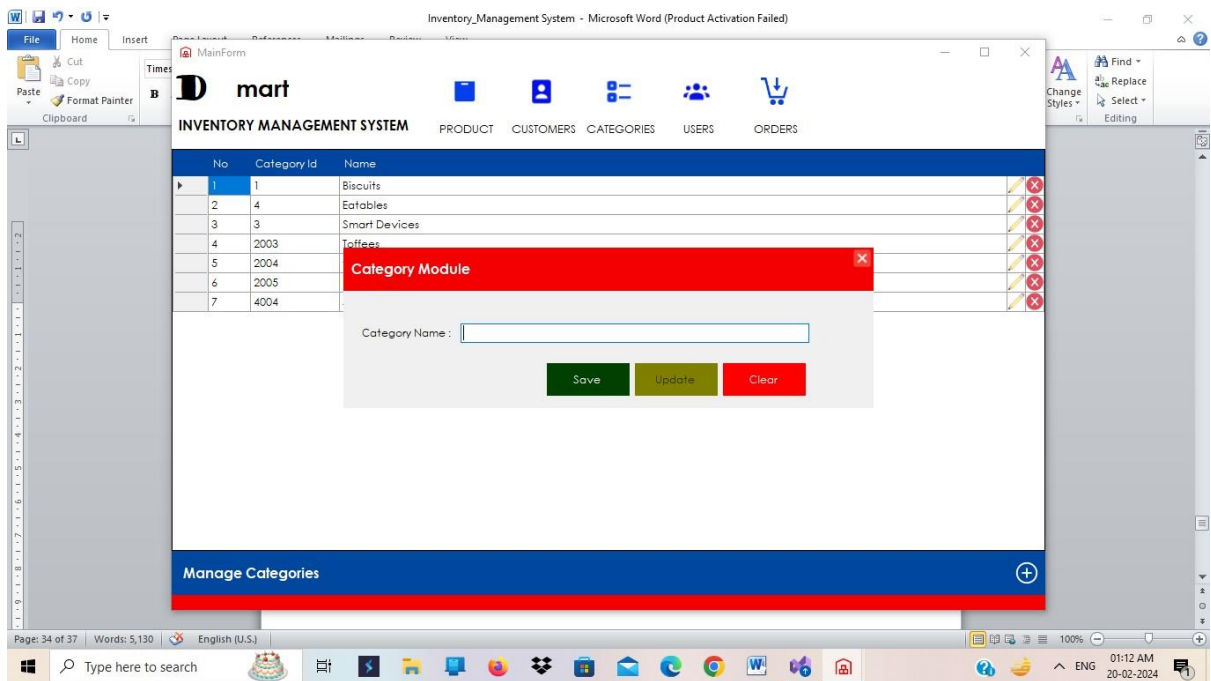

7.6 INSERT NEW PRODUCT:
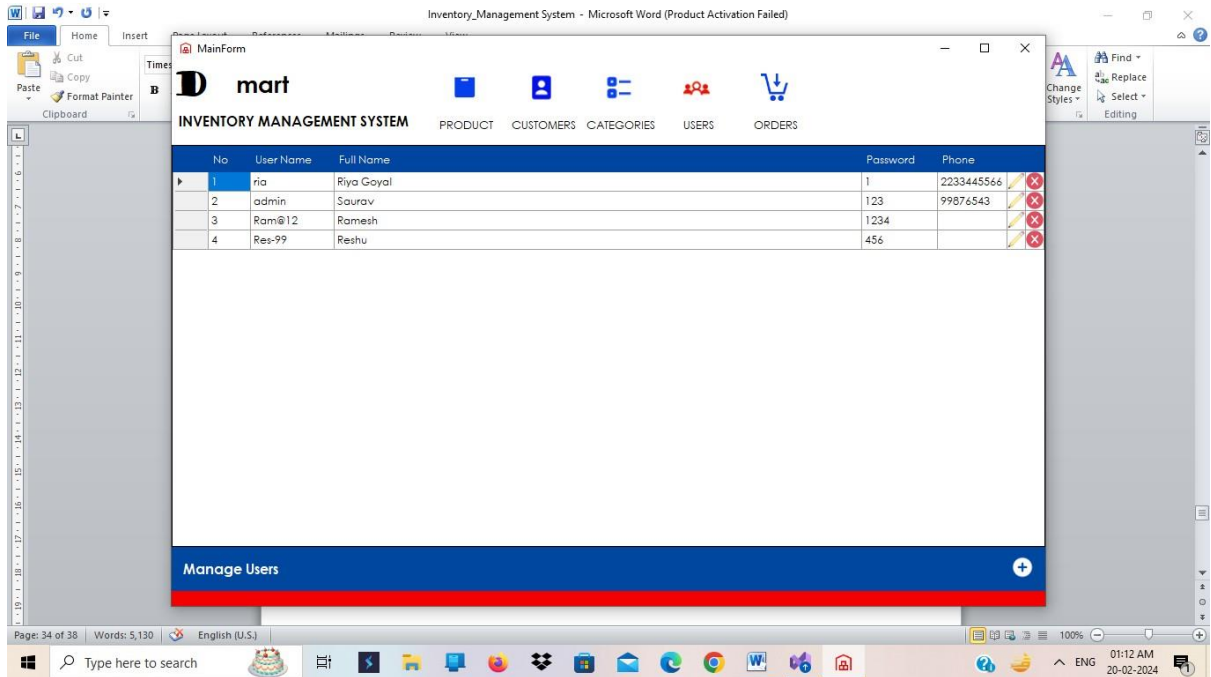
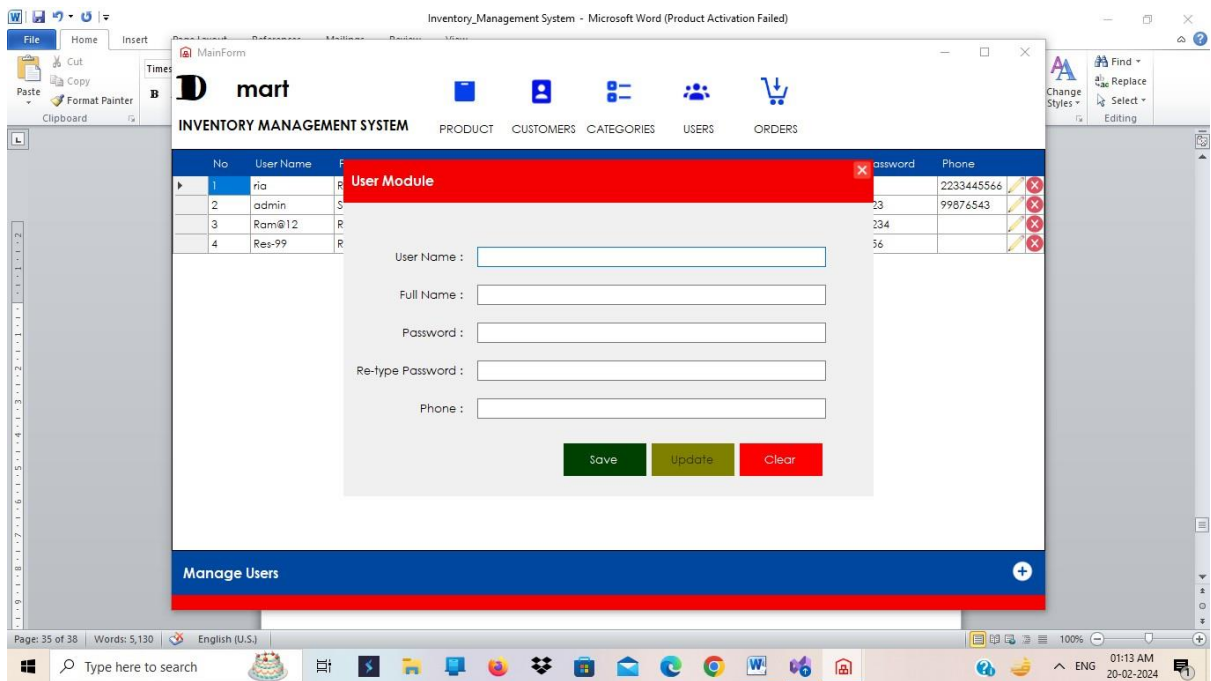## 7.7 SELECT THE CUSTOMER:

7.8 INSERT NEW CUSTOMER:

7.9 CATEGORY:

7.10 ADD CATEGORY:

7.11 USERS:

7.12 ADD USERS:

7.13 ORDERS:

7.14 INSERT ORDER:

CHAPTER 8

CONCLUSION

8.1 LIMITATION OF THE PROJECT

Due to less knowledge in particular fields and limited time we were not able to fulfil all our expectations that we expected we could do while theproject got started.

We hope these limitations are considerable. Some of the projectlimitations are:

- This application is not suitable for those organization where there is large quantity of product and different level of warehouses
- This software application is able to generate only simple reports.
- Single admin panel is only made.
- It is not suitable for large organization

8.2 CONCLUSION

To conclude, Inventory Management System is a simple desktop-based application basically suitable for small organization. It has every basic item which are used for the small organization. Our team is successful in making the application where we can update, insert and delete the item as per the requirement. This application also provides a simple report on daily basis to know the daily sales and purchase details. This application matches for small organization where there small limited if godwits.Through it has some limitations, I strongly believe that the implementation of this system will surely benefit the organization.

8.3 LESSON LEARNT

Doing something for long time periods always gives good lesson.Some of the things that I learnt are listed as below:

- Learnt about the IMS process.
- Learnt about Python technology, its components and ways toimplement them
- Learnt to work in pressure and to be patient.
- Learnt to manage the database under SQLite through Django

8.4 FUTURE ENHANCEMENTS

Since this project was started with very little knowledge about the Inventory Management System, we came to know about the enhancementcapability during the process of building it. Some of the scope we can increase for the betterment and effectivenesslisted below:

- Interactive user interface design.
- Manage Stock Godown wise.
- Online payment system can be added.
- Making the system flexible in any type.
- Sales and purchase return system will be added in order to makereturn of products

BIBLIOGRAPHY

1. Smith, J. (2020). "Effective Inventory Management in the Digital Age." Journal of Supply Chain Management, 25(3), 45-62.

2. Brown, A. (2018). "The Role of Cloud Computing in Modern Inventory Management Systems." International Journal of Logistics Technology, 12(2), 87-104.

3. White, L. K. (2019). "E-commerce and Inventory Control: A Case Study Analysis." Journal of Business Technology, 18(4), 301-318.

4. https://www.tradegecko.com/

5. https://www.zoho.com/inventory/