

Final Year Project
On
Inventory Management System

Submitted By
Tawfiqur Rahman (Roll No:)

In partial fulfillment of the requirement for the degree of
Masters of Computer Science and Engineering (M.Sc.)

October, 2021

JAHANGIRNAGAR UNIVERSITY

**Final Year Project
On
Inventory Management System**

Submitted To

**Submitted By
Tawfiqur Rahman (Roll No:)**

**Under the supervision of
Mr. Dipesh Shrestha**

**In partial fulfillment for the award of the Degree of
Masters of Computer Science and Engineering (M.Sc.)**

October, 2021

Faculty of Management

JAHANGIRNAGAR UNIVERSITY

RECOMMENDATION

This is to certify that the Final Year Project on
Inventory Management System

Submitted By

Anish Singh Maharjan (Roll No: 12080115)

Entitled

“INVENTORY MANAGEMENT SYSTEM”

Has been prepared and approved by this college.

This report is forwarded for examination to Pokhara University.

Mr. Kamal Pd. Regmi
Principal

Faculty of Management
JAHANGIRNAGAR UNIVERSITY

DECLARATION

I hereby declare that the work reported in this project work on “**Inventory Management System**” submitted to Nobel College, Pokhara University is our original work done in the form of partial requirement for the degree of Bachelor’s in Computer Information System (BCIS) under the supervision of Mr. Dipesh Shrestha, Head of department (Management Faculty), Nobel College and supervisor for the final year project Mr. Utsav Neupane. The material contained in the report has not been submitted to any University or Institution for the award of any degree.

Anish Singh Maharjan (Roll No: 12080115)

JAHANGIRNAGAR UNIVERSITY

Department Of Management

CERTIFICATE OF APPROVAL

I have the pleasure in forwarding the project of Mr. Anish Singh Maharjan (Roll No: 12080115) and Mr. Mandip Humagain (Roll No: 12080122) entitled “Inventory Management System” for the award of the degree of Bachelor of Computer Information System of this institute. Mr. Anish Singh Maharjan (Roll No: 12080115) and Mr. Mandip Humagain (Roll No: 12080122) have completed the project work for the full prescribed period under Pokhara University curriculum and the project embodied the result of his investigations conducted during the period they worked as a full time student of this department.

The Final year project has been approved by the following panel of examiners:

S.N	Name	Designation	Signature	Date
1	Mr. Dipesh Shrestha	Head of Department (Management Faculty)		

ACKNOWLEDGEMENT

This project is prepared in the partial fulfillment of the requirement for the degree of Bachelor in Computer Information System (BCIS). The satisfaction and success of completion of this task would be incomplete without heartfelt thanks to people whose constant guidance, support and encouragement made this work successful. On doing this undergraduate project I have been fortunate to have help, support and encouragement from many people I would like to acknowledge them for their cooperation.

Our first thanks goes to Pokhara University for designing such a worthy syllabus and making us do this project. Our next batch of thanks goes to the faculty of Management of Nobel College without whose help our project would have been impossible. This list includes Principal of Nobel College, Mr. Kamal Pd. Regmi, Head of Department (Management Faculty), Mr. Dipesh Shrestha. Our very sincere and heartfelt thanks go to Mr. Utsav Neupane our project supervisors who constantly guided us through the project time period. Without his guidance, our project would have been impossible. Last but not the least I want to thank every direct and indirect hands that are involved in completion of this project.

Finally, our heartfelt thanks goes to Mr. Krian Kurmar Regmi, Training Manager of Leaffrog technology Nepal as an external examiner for our final year project and thanks for sharing his valuable time with us to evaluate our work and provide professional suggestion with us.

This project has been a wonderful experience where I have learnt and experienced many beneficial things.

With Regards

Anish Singh Maharjan (Roll No: 12080115)

Registration No: 2011-2-08-0031

Mandip Humagain (Roll No: 12080122)

Registration No: 2011-2-08-0038

ABSTRACT

This project is aimed at developing a desktop-based application named Inventory Management System for managing the inventory system of any organization. The Inventory Management System (IMS) refers to the system and processes to manage the stock of organization with the involvement of Technology system. This system can be used to store the details of the inventory, stock maintenance, update the inventory based on the sales details, generate sales and inventory report daily or weekly based. This project is categorized by individual aspects for the sales and inventory management system. In this system I are solving different problem affecting to direct sales management and purchase management. Inventory Management System is important to ensure quality control in businesses that handle transactions resolving around consumer goods. Without proper inventory control, a large retail store may run out of stock on an important item. A good inventory management system will alert the wholesaler when it is time to record. Inventory Management System is also on important means of automatically tracking large shipment. An automated Inventory Management System helps to minimize the errors while recording the stock.

LIST OF FIGURES

[illegible]

LIST OF ABBREVIATIONS

IMS

Inventory Management System

DBMS

Database Management System

DFD

Data Flow Diagram

JIT

Just-In-Time

DML

Data Manipulation Language

DDL

Data Definition Language

DCL

Data Control Language

CLR

Common Language Runtime

SQL

Structured Query Language

TABLE OF CONTENTS

RECOMMENDATION	ii
DECLARATION	iii
CERTIFICATE OF APPROVAL	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS	viii
CHAPTER-1: INTRODUCTION.....	1
1.1 Introduction to Inventory Management System	1
1.2 Literature Review.....	1
1.3 Problem Statement	2
1.4 Objective of the Project	2
1.4.1 Primary objective	2
1.4.2 Secondary objective	2
1.6 Features of Project	2
1.7 Scope of the Application	3
CHAPTER-2 BACKGROUND KNOWLEDGE	4
2.1 Architectural Review.....	4
2.1.1 Client tier	4
2.1.2 Business tier	4
2.1.3 Data tier	4
2.2 Database Theory	5
2.2.1 Relational Database	5
2.2.2 Structured Query Language (SQL)	6
2.3 ACID Property	6
CHAPTER-3: ANALYSIS AND DESIGN	8

3.1	Background Research	8
3.2	Requirement Analysis	8
3.3	IMS Requirement	8
3.4	Users Requirement	9
3.4.1	Admin	9
3.4.2	Inventory management.....	9
3.5	Feasibility Analysis	9
3.5.1	Economic Feasibility	9
3.5.2	Technical Feasibility	9
3.5.3	Operational Feasibility	10
3.5.4	Schedule Feasibility	10
3.6	Gantt chart	10
CHAPTER – 4: SYSTEM DESIGN		11
4.1	Process Flow Diagram	11
4.2	Use Case Diagram.....	11
4.2.1	Diagram Building Block	12
CHAPTER – 5: TOOLS AND TECHNOLOGY USED		14
5.1	Development Tools	14
5.1.1	Microsoft Visual Studio	14
5.1.2	MySQL Database Connector	15
5.1.3	.Net Core 3.1	15
5.2	Technology Used	15
5.2.1	.Net Core Structure	15
5.2.2	Compilation to Manage Code	16
5.2.3	JIT compilation	17
5.2.4	The .Net Core Language	17
5.2.5	Data Provider	18

5.2.6 The Connection object	18
5.2.7 The command Object.....	18
5.2.8 React	19
5.3 MySQL	19
CHAPTER- 6: CODING IMPLEMENTATION.....	21
6.1 Application Code Structure.....	21
6.2 Logic	21
6.3 Project Screenshot.....	21
CHAPTER – 7: DEBUGGING AND TESTING	31
7.1 Purpose of Testing	31
7.2 Type of Testing	31
7.2.1 Units Test	31
CHAPTER – 8: CONCLUSION AND LESSON LEARNT	32
8.1 Project Limitation	32
8.2 Conclusion	32
8.3 Lesson Learnt.....	32
8.4 Future Enhancements.....	32

CHAPTER-1: INTRODUCTION

1.1 Introduction to Inventory Management System

The project Inventory Management System is a complete desktop-based application designed on .Net Core technology using Visual Studio Software and React for frontend. 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 web application which has admin component to manage the inventory and maintenance of the inventory system.

This web 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 updated 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 Literature 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, I came up with 2 Inventory Management System existing in the market, which helps to give the knowledge for the development of our project. This application software is only used by the large organization so I 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, I decided to include some of common and key features that should be included in every inventory management system. So, I decided to include those things that help the small organization in a way or other.

1.3 Problem Statement

After analyzing many existing IMS I have now the obvious vision of the project to be developed. Before I started to build the application team had many challenges. I 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 stock management.

1.4 Objective of the Project

1.4.1 Primary objective

The primary objectives of the project are mentioned below:

- To know the fundamentals of the React and Visual Studio with .Net Core 3.1 Technology

1.4.2 Secondary objective

The secondary objectives of this project are mentioned below:

- To develop an application that deals with the day-to-day requirement 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.

1.6 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 login is 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.

Sales details: It show the details about the sales and the remaining stock of sales. It also shows 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.

1.7 Scope of the Application

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

Some of the scopes are:

- Multiple people are responsible in assigning the details or records
- It is security driven.

CHAPTER-2 BACKGROUND KNOWLEDGE

2.1 Architectural Review

This desktop-based application is based on 3-tier architecture of .Net Core. The 3-tier includes the three hierarchy of the flow of 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:

2.1.1 Client tier

The visual part is implemented using all kinds of React 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.

2.1.2 Business 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 by the client's tier.

2.1.3 Data tier

Data layer is also the class which gets the data from the business tier 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 database Connectivity. MySQL database connectivity is used to manage the communication between the middle tier and the backend database by issuing complex database queries.

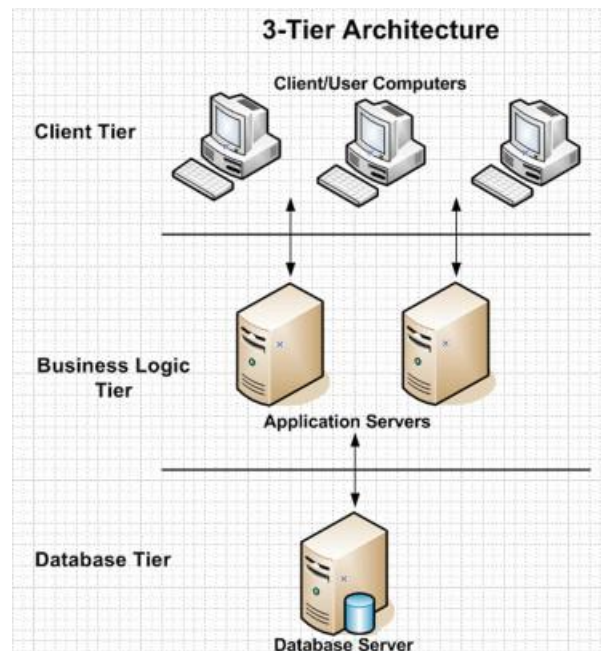


Figure 2.1: Tier Architecture

Source: <https://www.sciencedirect.com/topics/computer-science/presentation-logic>

2.2 Database Theory

A database is a collection of information that is organized so that it can easily be accessed, managed and updated. In one view, database can be classified according to types of content: bibliography, full-text, numeric, and image. In computing, database is sometime classified according to their organizational approach. A distributed database is one that can be dispersed or replicated among different points in a network.

2.2.1 Relational Database

IMS has the relational database model. A relational database is a digital database whose organization is based on the relational model of data. This model organizes data into one or more tables of rows and columns. These tables here have the relation. The relation is maintained by the unique key defined in each row. The key can be primary and foreign depending on their nature of connection. The standard user and application program interface to a relational database is the structured query language (SQL). SQL statement are used both for interactive queries for information from relational database and for gathering data for reports.

Primary Key

The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique or it can be generated by the DBMS. A primary key's main features are:

- It must contain a unique value for each row of data.
- It cannot contain null value.

Foreign Key

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. In foreign key reference, a link is created between two tables when the column or columns that hold the primary key value for one table are referenced by the column or column in another table thereby establishing a link between them. Creating a foreign key manually includes the following advantages:

- Changes to primary key constraints are checked with foreign key constraints in relation table.
- An index enables the Database Engine to quickly find related data in the foreign key tables.

2.2.2 Structured Query Language (SQL)

The structured Query language (SQL) is the set of instructions used to interact with a relational database. In fact, SQL is the only language the most database actually understand. Whenever you interact with such a database, the software translates your commands into SQL statement that the database knows how to interpret. SQL has three major Components:

- Data Manipulation Language (DML)
- Data Definition Language (DDL)
- Data Control Language (DCL)

2.3 ACID Property

Every database transaction obeys the following rules:

Atomicity – Either the effects of all or none of its operation remain (“all or nothing” semantics) when a transaction is completed (committed or aborted respectively). In other words, to the outside world a committed transaction appears (by its effects on

the database) to be indivisible, atomic, and an aborted transaction does not leave effects on the database at all, as if never existed.

Consistency – every transaction must leave the database in a consistent (correct) state, i.e., maintain the predetermined integrity rules of the database (constraints upon and among the database's objects). A transaction must transform a database from one consistent state to another consistent state (however, it is the responsibility of the transaction's programmer to make sure that the transaction itself is correct, i.e., performs correctly what it intends to perform (from the application's point of view) while the predefined integrity rules are enforced by the DBMS). Thus, since a database can be normally changed only by transactions, all the database's states are consistent. An aborted transaction does not change the database state it has started from, as if it never existed (atomicity above).

Isolation – Transactions cannot interfere with each other (as an end result of their executions). Moreover, usually (depending on concurrency control method) the effects of an incomplete transaction are not even visible to another transaction. Providing isolation is the main goal of concurrency control.

Durability – Effects of successful (committed) transactions must persist through crashes (typically) by recording the transaction's effects and its commit event in a non-volatile memory.

CHAPTER-3: ANALYSIS AND DESIGN

3.1 Background Research

I started research by identifying the need of IMS in the organization. Initially I bounded our research to find the general reasons that emerged the needs of Inventory Management System. I used different techniques to collect the data that can clearly give us the overall image of the application. The techniques I used are interview with the developers, visiting online websites that are presented as the templates and visiting some organization to see their IMS application. Basically, the following factors forced us to develop IMS application:

3.2 Requirement Analysis

I collected a number of requirements for project from our primitive research, website visits, and interview to the concerned personnel and their experiences regarding the concepts of its development. I have even visited some organization in Kathmandu valley and analyze its importance and try to develop the project by fulfilling all the weakness that I found in the application. I then decided to build same type of application with different logic flow and new language which will be suitable for the small organization.

3.3 IMS Requirement

The goal for the application is to manage the inventory management function of the organization. Once it is automated all the functions can be effectively managed and the organization can achieve the competitive advantage. Business requirement are discussed in the Scope section, with the following additional details:

- Helps to search the specific product and remaining stock.
- Details information about the product sales and purchase.
- Brief Information of the organization today's status in terms of news, number of present inventories as per the date entered. It helps to identify the total presented inventory in the company

- To know the balance and details of sales distributed in specific date.
- There is proper transaction management of inventory.
- All transaction has specific entry date along with quantity and rate.
- Only admin can login in the page.

3.4 Users Requirement

User requirement are categorized by the user type

3.4.1 Admin

- Able to create new product/manufacturer/supplier/customer/user along with date.
- Able to edit the entry as per entry.
- Able to add, modify and delete the stock entry.

3.4.2 Inventory management

- Able to check the stock available.
- Able to check the balance payment.
- Able to view the remaining sales stock.

3.5 Feasibility Analysis

This software has been tested for various feasibility criterions from various point of views.

3.5.1 Economic Feasibility

The system is estimated to be economically affordable. The system is medium scale desktop application and has affordable price. The benefits include increased efficiency, effectiveness, and the better performance. Comparing the cost and benefits the system is found to be economically feasible.

3.5.2 Technical Feasibility

Development of the system requires tools like:

- Visual Studio 2019
- .Net Core 3.1
- MYSQL 8.0

3.5.3 Operational Feasibility

The system provides better solution to the libraries by adding the typical requirement and necessities. The solution provided by this system will be acceptable to ultimate solution for the stock management.

3.5.4 Schedule Feasibility

The organized schedule for the development of the system is presented in the schedule subsection. The reasonable timeline reveals that the system development can be finished on desired time framework.

3.6 Gantt chart

It is one of the popular ways to illustrate project schedule. A Gantt chart is a graphical representation of a project that shows each activity task as a horizontal bar whose length is proportional to its time for completion. A Gantt chart for the project deliverables within time frame. This project Gantt chart is shown below:

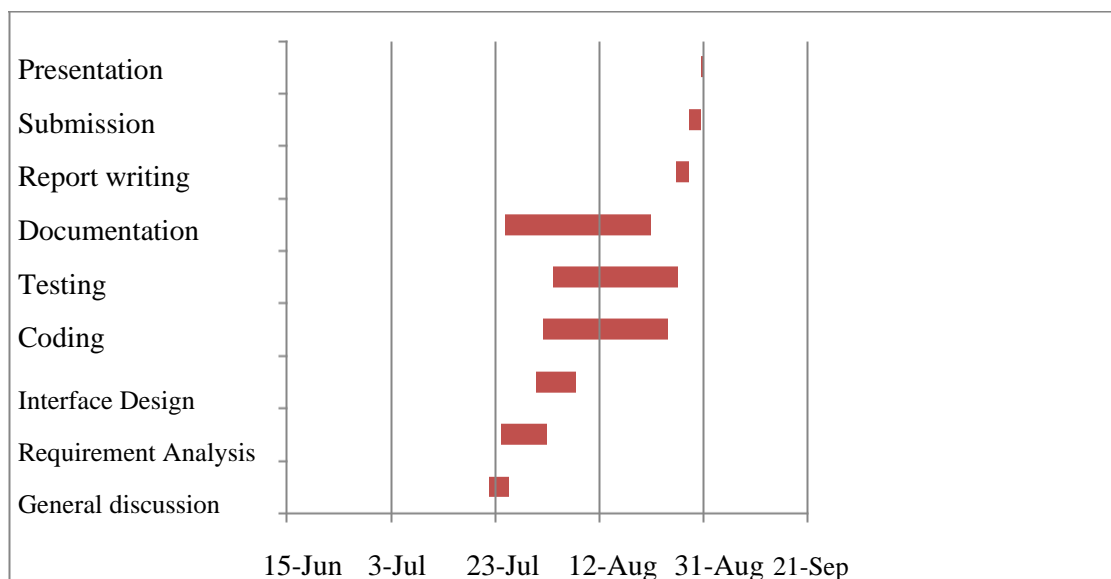


Figure 3.6: Gantt chart

CHAPTER – 4: SYSTEM DESIGN

4.1 Process Flow Diagram

Process Flow Diagram or Flowchart is a diagram which uses geometric symbols and arrows to define the relationships. It is a diagrammatic representation of the algorithm. The Process flow Diagram of our application is shown below:

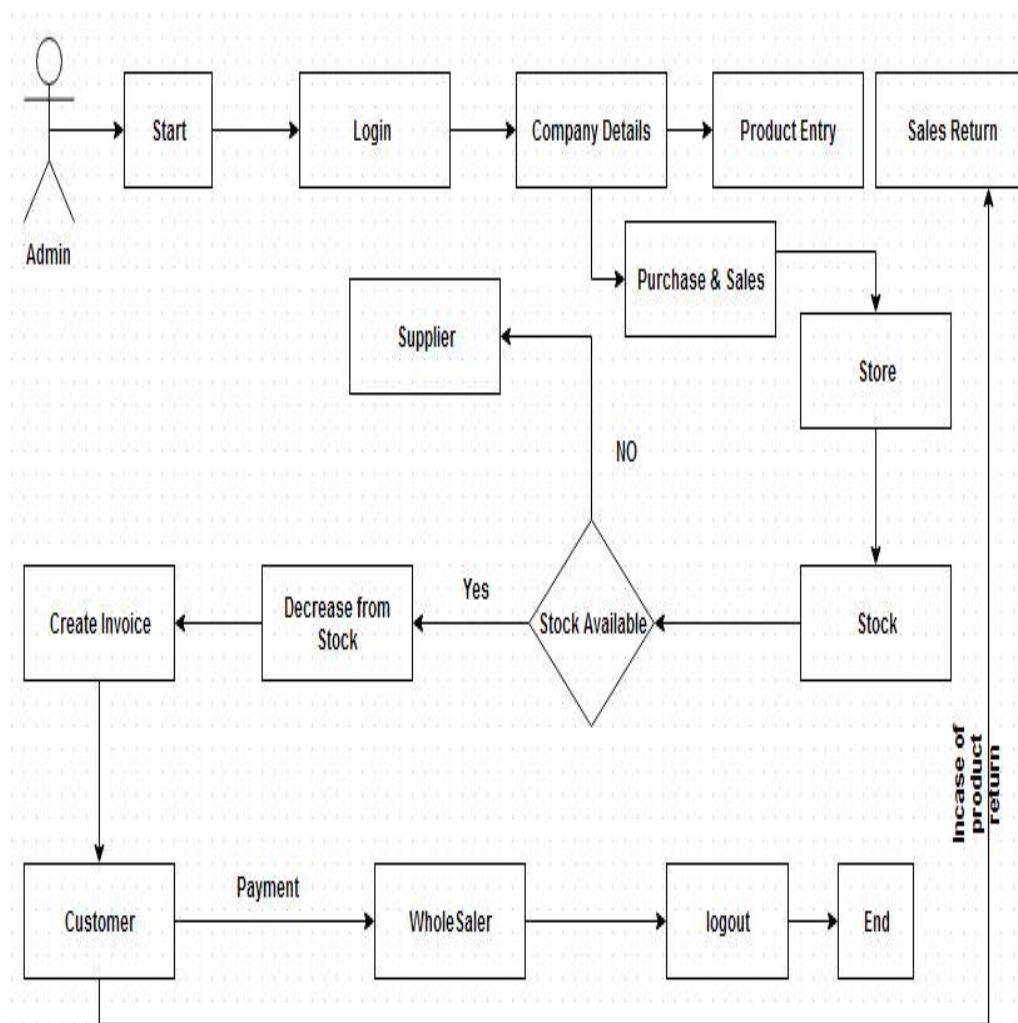


Figure 4.1: IMS Process flow diagram

4.2 Use Case Diagram

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors and their goals. The main purpose of a use case diagram is to show what system functions are performed for which actors.

4.2.1 Diagram Building Block

Use cases

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

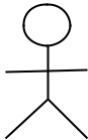
Actor

An actor is a person, organization or external system that plays a role in one or more interactions with the system

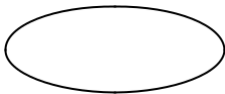
System boundary boxes (optional)

A rectangle is drawn around the use case called the system boundary box to indicate scope of the system.

Actor



Use case



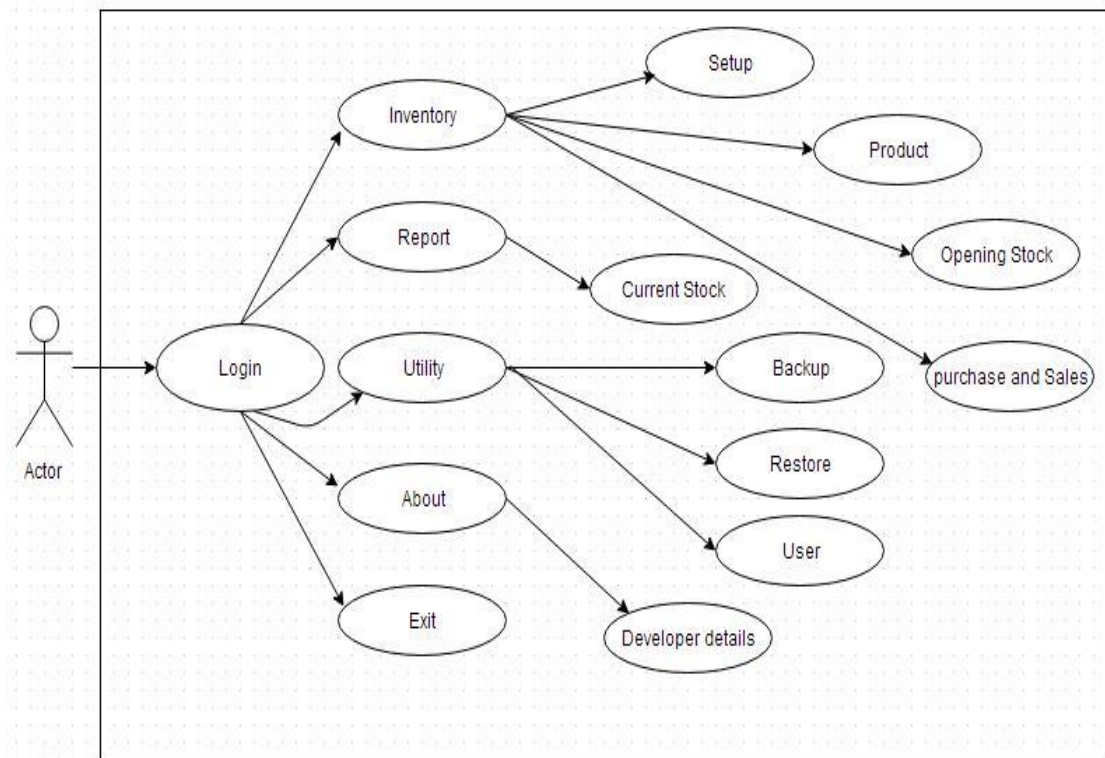


Figure 4.2.1: IMS Use Case Diagram

CHAPTER – 5: TOOLS AND TECHNOLOGY USED

5.1 Development Tools

5.1.1 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 Core Framework, .Net Core Compact Framework and Microsoft Silverlight. Microsoft Visual Studio simplifies the basic tasks of creating, debugging and deploying applications.

Microsoft Visual Studio comes with .Net Core 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 interactive designer. Microsoft Visual Studio offers several tools to make parallel programming simpler: in addition to the Parallel Extensions for the .Net Core Framework and the Parallel Patterns Library for native code, Visual Studio includes tools for debugging parallel applications.

The Visual Studio code editor now highlights references; whenever a symbol is selected; all other usages of the symbol are highlighted. It also offers a Quick Search feature to incrementally search across all symbols in C++, C# and VB.Net Core projects. Quick Search supports substring matches and camel Case searches. The Call Hierarchy feature allows the developers to see all the methods that are called from a current method as well as the methods that call the current one. IntelliSense in Visual Studio supports a consume-first mode which developers can opt into. In this mode, IntelliSense will not auto-complete identifiers; this allows the developer to use undefined identifiers (like variable or method names) and define those later. Visual Studio can also help in this by automatically defining them, if it can infer their types from usage.

I have used Visual Studio Community 2019, v 14.0.23107.10 for developing the Inventory Management System Application.

5.1.2 MySQL Database Connector

MySQL Database Connector provides a graphical management tool for MySQL database. MySQL Database Connector user interface is a subset of MySQL that is available with other editions of SQL Server. MySQL Database Connector call also manage instance of the SQL Database Engine created by any edition of MySQL. Inventory Management System is developed using MySQL 8.0.23.

5.1.3 .Net Core 3.1

The .Net Core 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 Core class library, which includes classes, interfaces, and value types that support an extensive range of technologies. The .Net Core provides a managed execution environment, simplified development and deployment, and integration with a variety of programming languages, including F# and C#.

5.2 Technology Used

5.2.1 .Net Core Structure

The .Net Core 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 Core 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:

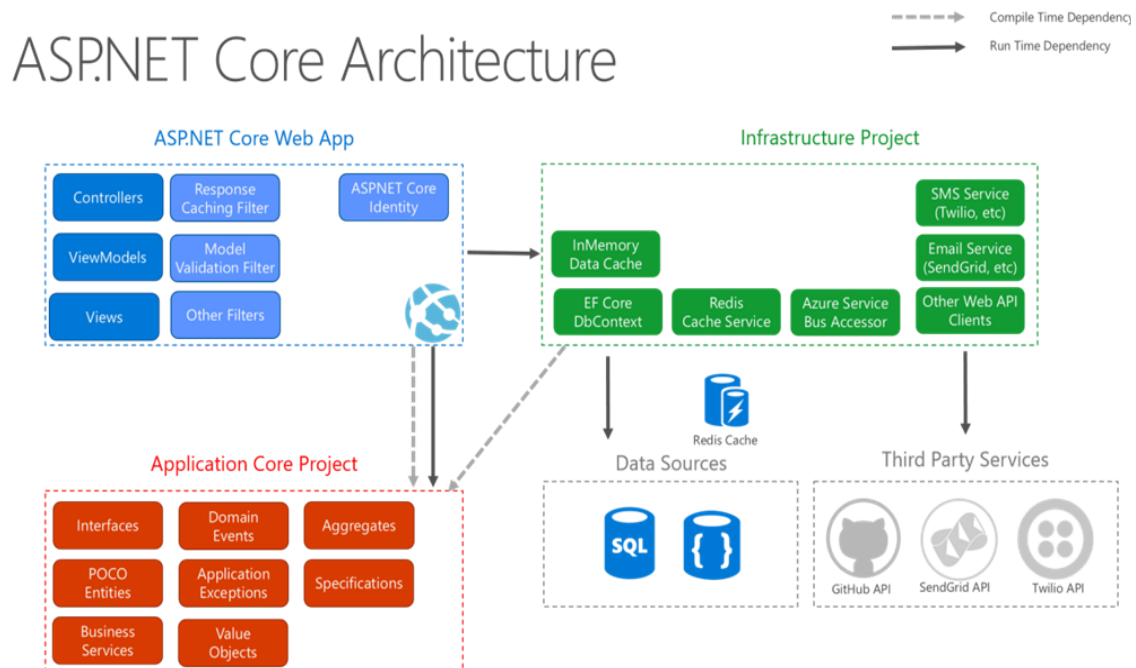


Figure 5.2.1: .Net Core Architecture

Source: <https://www.c-sharpcorner.com/article/asp-net-core-2-architecture-design-pattern-ideology/>

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

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

At the base of the .Net Core 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 Core 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 Core Framework Base Classes is ADO.Net Core, which provides access to and management of data.

The third layer of the framework is ASP.Net Core and Windows Forms. ASP.Net Core 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. Using ASP.Net Core, it's now possible to build robust web applications that are even more functional than Win32 applications of the past.

5.2.2 Compilation to Manage Code

Code that is compiled and targeted to the CLR is known as managed code. Managed code provides metadata that is needed for the CLR to provide the services of multi-

language support, code security, object lifetime management, and memory management. The .Net Core Framework requires that you use a language compiler that is targeted at the CLR, such as the Visual Basic .Net Core, C#, C++ .Net Core, or Jscript .Net Core compilers provided by Microsoft. So how does the code that you typed into Visual Studio .Net Core become the code that the user receives when he is using your application? It is fairly simple and straightforward. Figure below shows a diagram of the compilation process.

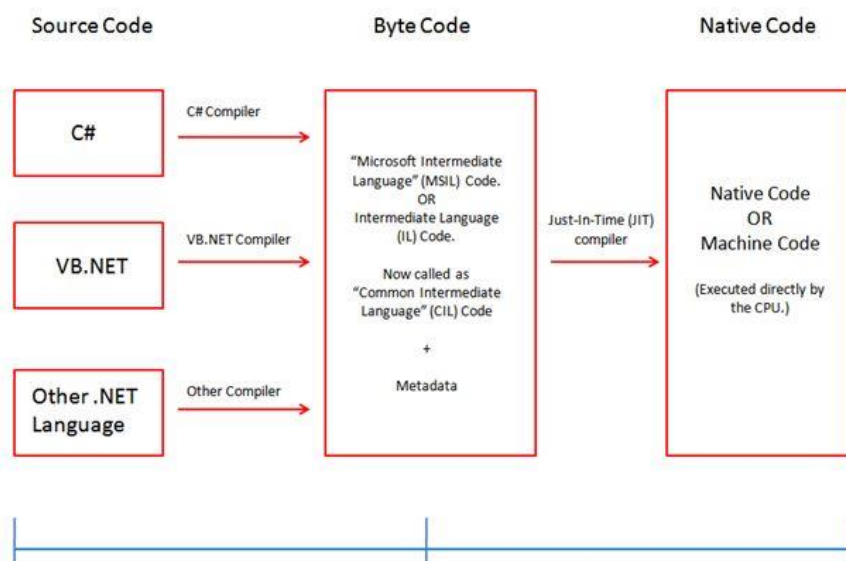


Figure 5.2.2: Compilation to managed code

Source: Teacher's handout

The IL is CPU-independent. This means that IL code is not reliant on the specific computer that generated it. In other words, it can be moved from one computer to another (as long as the computer supports the .Net Core Framework) without any complications. This is what makes XCopy, or just copying over the application, possible.

5.2.3 JIT compilation

The .Net Core Framework contains one or more JIT compilers that compile your IL code down to machine code. This is done when the application is executed for the first time.

5.2.4 The .Net Core Language

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

Now the .Net Core provides you with a language-independent programming platform. You do not have to decide which language would provide a better solution. All languages are now on a level playing field. In .Net Core, no one language is superior to any of the other languages. They all have equal access to everything that .Net Core offers. To be part of the .Net Core 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 provides four languages with the .Net Core Framework:

- Visual Basic .Net Core
- C#
- C++.Net Core and
- Jscript .Net Core.

Microsoft also provides J# (pronounced J-sharp), but in order to use this new language that is basically Java for .Net Core, you need to download the language to install it on your server.

5.2.5 Data Provider

The data provider is responsible for providing and maintaining the connection to the database. A database provider is a set of related components that works together to provide in an efficient and performance driven manner. Each Data provider consists of the following components classes:

- The command object which is used to execute a command.
- The Connection object which provides a connection to the database.
- The Data Reader object which provides a ready only, connects record set.

5.2.6 The Connection object

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

5.2.7 The Command Object

The command object is represented by corresponding classes: SQL Command. Command object are used to execute commands to a database across a data connection. The command objects provide three methods that are used to execute commands on the database.

- ExecuteNonQuery: Executes commands that have no return values such as
- INSERT, UPDATE AND DELETE
- ExecuteScalar: Returns a single value from a database query
- ExecuteReader: Returns a result set by way of a DataReader Objects.

5.2.8 React

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality. Another notable feature is the use of a virtual Document Object Model, or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This process is called reconciliation. This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change. This selective rendering provides a major performance boost. It saves the effort of recalculating the CSS style, layout for the page and rendering for the entire page.

5.3 MySQL

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL website (<http://www.mysql.com/>) provides the latest information about MySQL software. MySQL is a database management system. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications. MySQL databases are relational. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables.

The Main Features of MySQL

This section describes some of the important characteristics of the MySQL Database Software. In most respects, the roadmap applies to all versions of MySQL.

- Uses multi-layered server design with independent modules.
- Designed to be fully multithreaded using kernel threads, to easily use multiple CPUs if they are available.
- Provides transactional and non-transactional storage engines.
- Uses very fast B-tree disk tables (MyISAM) with index compression.
- Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
- Uses a very fast thread-based memory allocation system.
- Executes very fast joins using an optimized nested-loop join.
- Implements in-memory hash tables, which are used as temporary tables.
- Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.
- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.

CHAPTER- 6: CODING IMPLEMENTATION

6.1 Application Code Structure

Inventory Management System was designed using Visual Studio as mentioned earlier following the three-tier application architecture. It provided us with the code editor as a white blank space and the solution explorer where every code file was kept. Code Editor is where the logical I developed into code and kept safe in the solution explorer. In solution explorer I kept every code file by creating the folder and adding those files in a folder that are similar in nature. The main folder was the Inventory Management System. Following are the list of the folders, sub-folders and their corresponding files:

6.2 Logic

Logic is the main component of any application portrayed through the code. Every module in the application includes logic. Most of the logic are common and understandable as I call 3-tier architecture-based system.

6.3 Project Screenshot

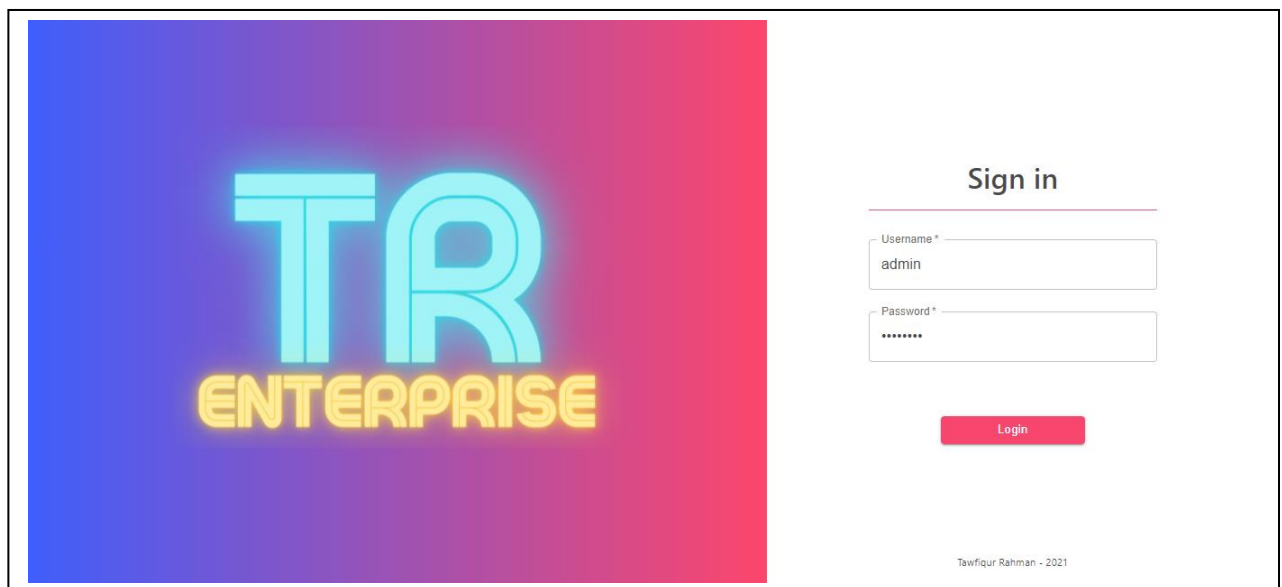


Figure 6.3.1: Admin or login page

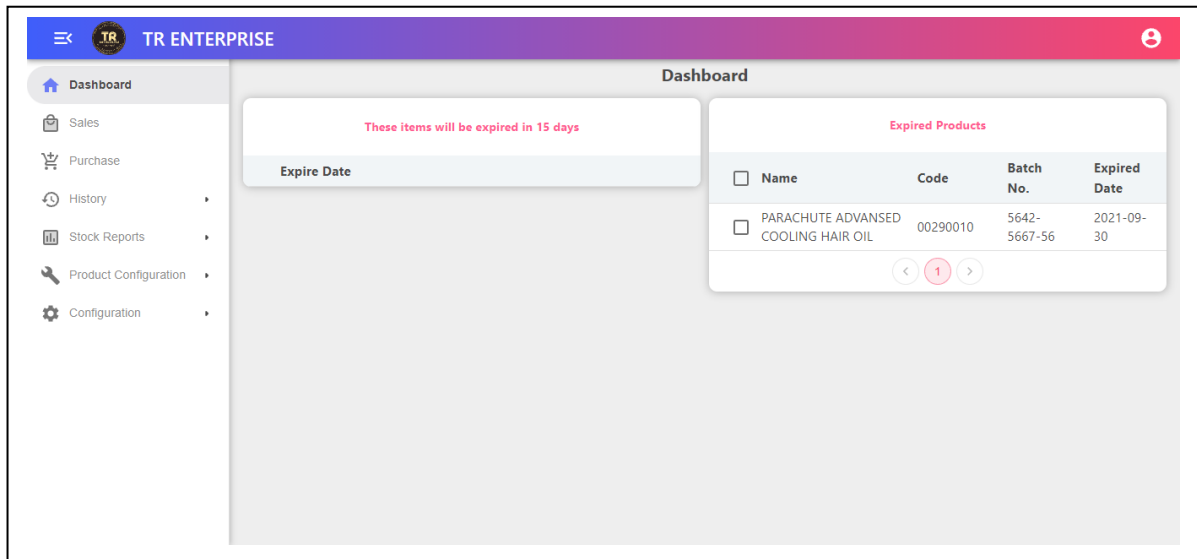


Figure 6.3.2: Dashboard

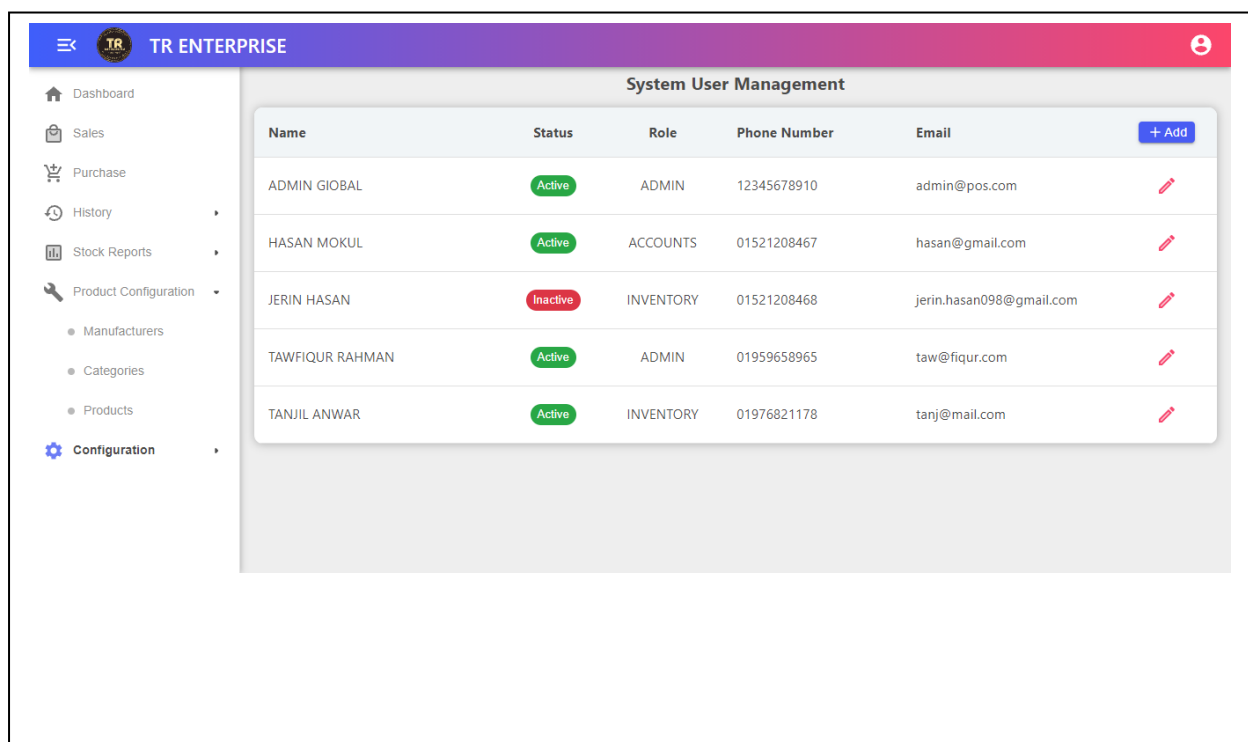


Figure 6.3.3: User Management

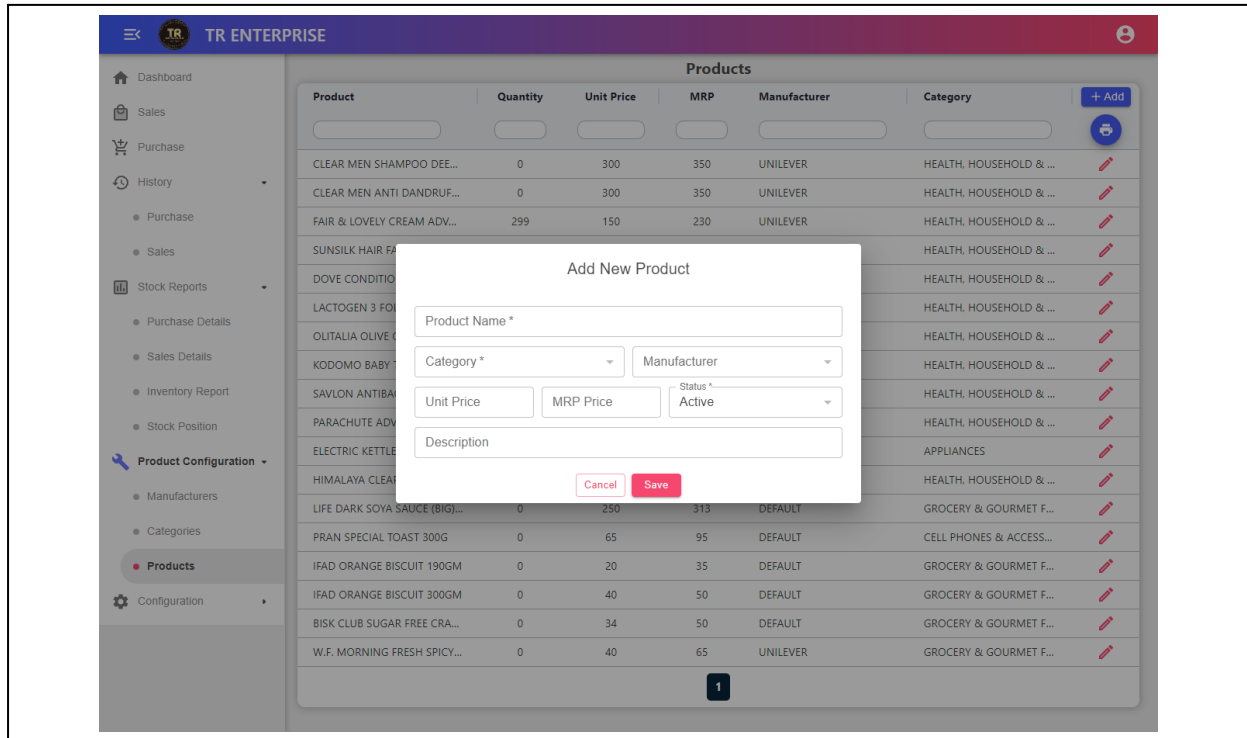


Figure 6.3.4: Product Entry System

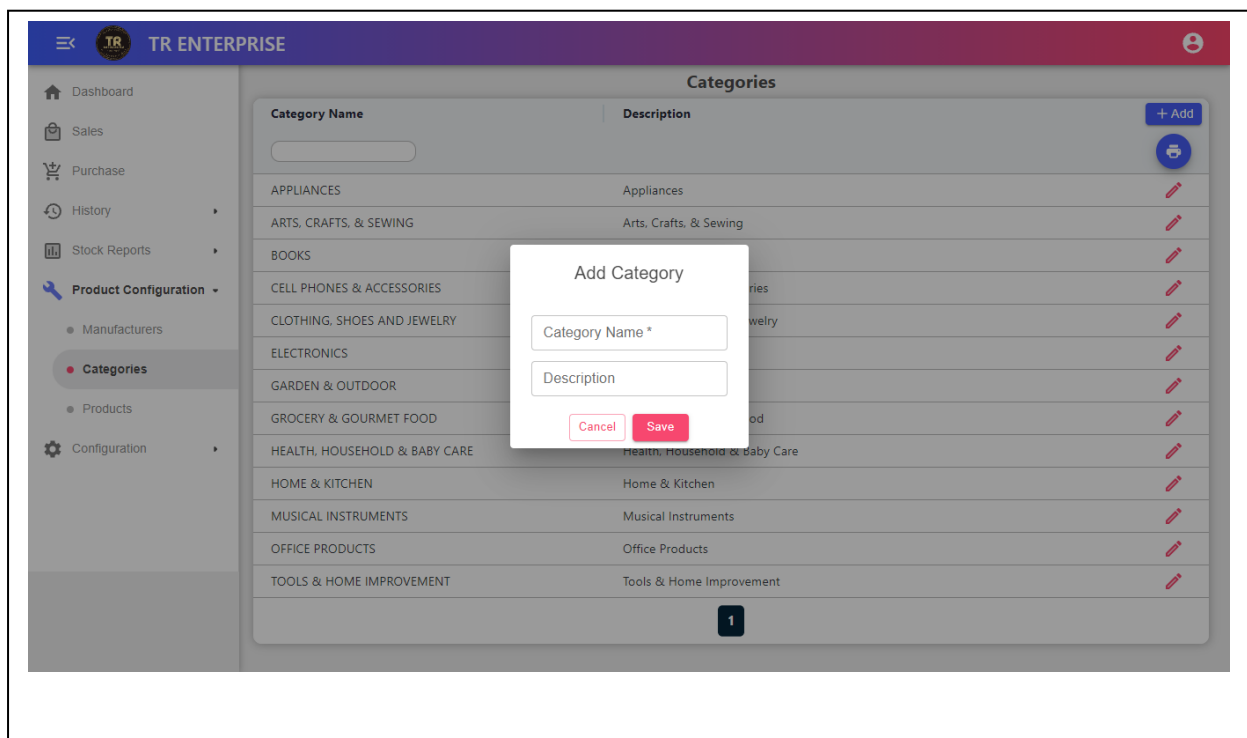


Figure 6.3.5: Category Entry System

TR ENTERPRISE

Suppliers

Dashboard, Sales, Purchase, History, Stock Reports, Product Configuration, Configuration

Supplier Name *, Company name *, Division *, District *, Thana *, Address *, Mobile *, Email, Supplier Type *, Remarks

Cancel, Save

Figure 6.3.6: Supplier Entry System

TR ENTERPRISE

Customers

Dashboard, Sales, Purchase, History, Stock Reports, Product Configuration, Configuration

Customer Name *, Mobile *, Division, District, Thana, Address *, Customer Status *, Remarks

Cancel, Save

Figure 6.3.7: Customer Entry System

Figure 6.3.8: Purchase Module

TR

TR ENTERPRISE

Dashboard

Sales

Purchase

History

Stock Reports

Product Configuration

Configuration

General Customer

Search Customer
MD SHAHIN

Customer Name : MD SHAHIN
Mobile : 01236658987

Add Customer

Search product by Name/Code

Quantity *
Quantity

Discount
Discount

Add

Product Name	Category	MRP	Quantity	Discount (%)	Total	Action
FAIR & LOVELY CREAM	HEALTH, HOUSEHOLD & BABY CARE	230	40	0	9200	
ADVANCED MULTI VITAMIN 100GM	HEALTH, HOUSEHOLD & BABY CARE	270	30	5	(₹100) 7695	

Total Price
16,895

Discount
5 %

Payment
16050.25

Receive By *
Cash & Cash ...

Grand Total: **16,050.25** Due: 0

Submit

Figure 6.3.9: Sale Module

TR ENTERPRISE							
Purchase History							
Search by Invoice # Invoice		Supplier SAGAR TRADERS	From 02/05/2021	To 10/05/2021			
Date	Supplier	Invoice	Total	Discount	Grand Total	Paid	Due
Sep 20, 2021	SAGAR TRADERS	0101000000035	297,000	44,550 (15%)	252,450	252,450	0
Purchase Details							
Product Name	Unit price	Quantity	Discount(%)	Total			
FAIR & LOVELY CREAM ADVANCED MULTI VITAMIN 100GM	150	500	0	75,000			
SAVLON ANTIBACTERIAL WET WIPES 20PCS	60	1000	0	60,000			
PARACHUTE ADVANSED COOLING HAIR OIL	120	100	0	12,000			
DOVE CONDITIONER OXYGEN MOISTURE 180ML	150	1000	0	150,000			
			Net Total	297,000			
			Discount(15%)	44,550			
			Grand Total	252,450			
			Paid	252,450			
			Due	0			
			Paid By	United Commercial Bank Ltd			

Figure 6.3.10: Purchase History

TR ENTERPRISE							
Sales History							
Search by Invoice # Invoice		Customer MD KHALEQ	From 02/05/2021	To 10/05/2021			
Date	Customer	Invoice	Total	Discount	Grand Total	Paid	Due
Sep 20, 2021	MD KHALEQ	0101000000037	96,730	9,673 (10%)	87,057	50,000	37,057
Sales Details							
Product Name	Unit price	Quantity	Discount(%)	Total			
PARACHUTE ADVANSED COOLING HAIR OIL	160	50	0	8,000			
SAVLON ANTIBACTERIAL WET WIPES 20PCS	85	500	0	42,500			
FAIR & LOVELY CREAM ADVANCED MULTI VITAMIN 100GM	230	201	0	46,230			
			Net Total	96,730			
			Discount(10%)	9,673			
			Grand Total	87,057			
			Paid	50,000			
			Due	37,057			
			Paid By	Cash & Cash Equivalents			

Figure 6.3.11: Sales History

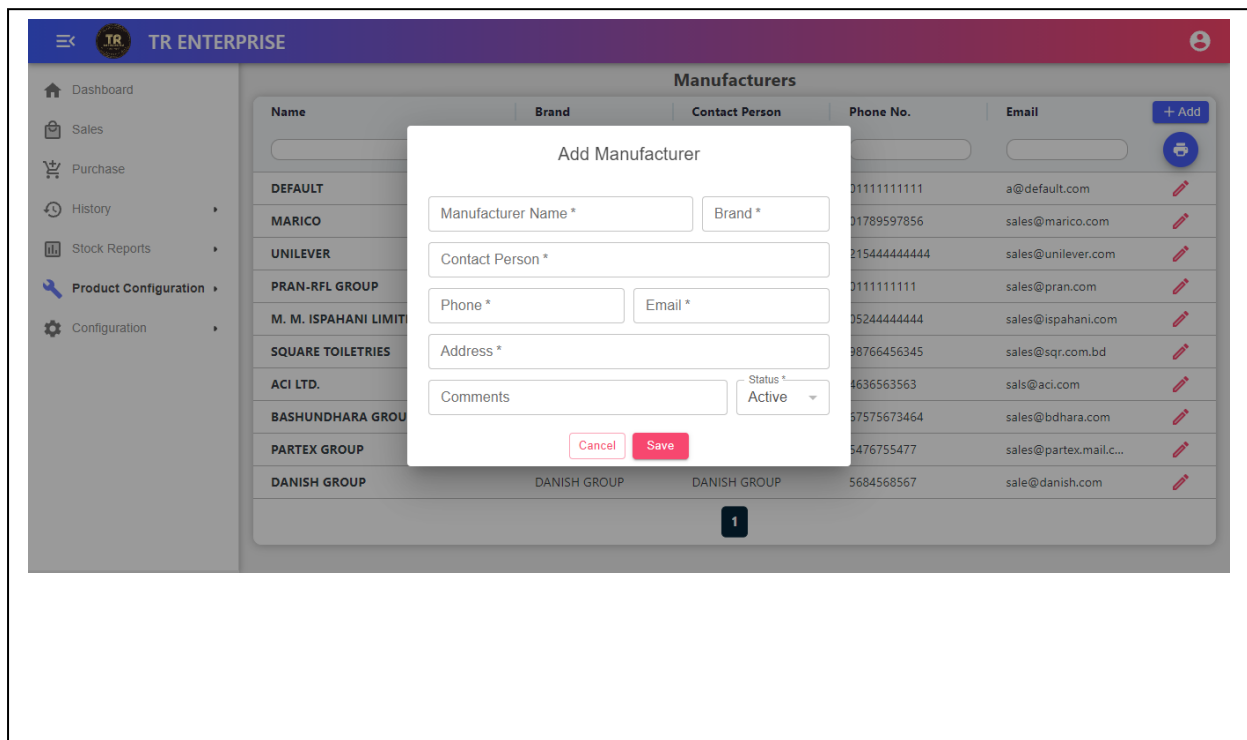


Figure 6.3.12: Manufacturer Entry

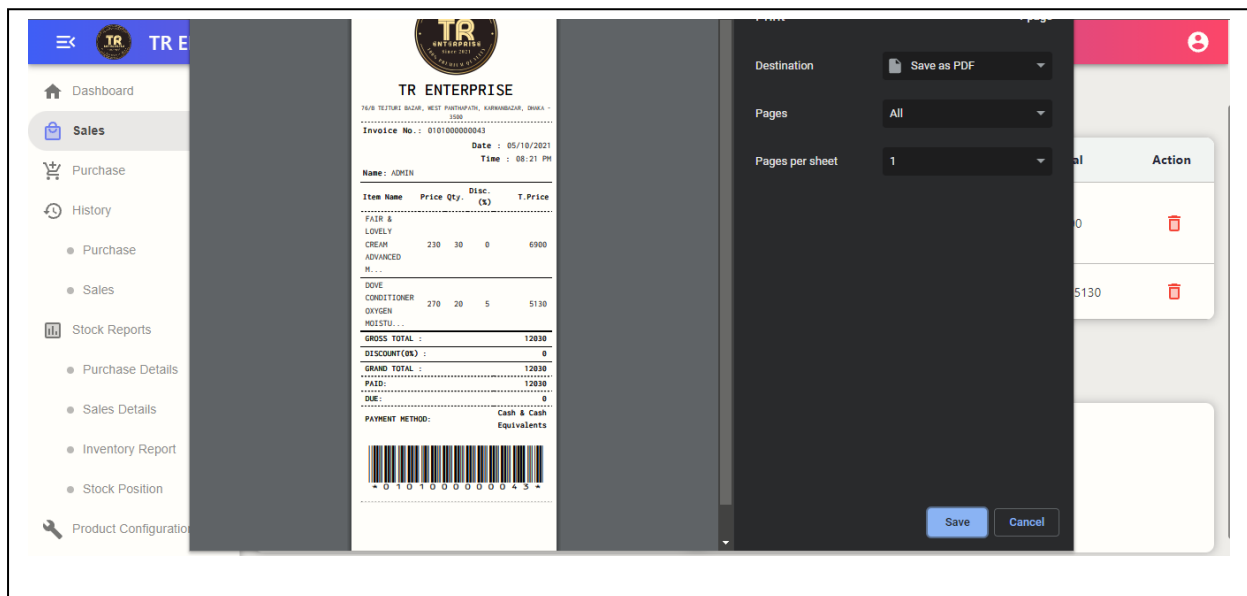


Figure 6.3.13: POS Receipt Generation

TR ENTERPRISE Date : 30/09/21
NEW TECTURE BICAL, WEST PARTHAKRITI, KARNATAKA, INDIA - 576001

Invoice No. : 0010000000035
Supplier Invoice No. : INV-4954

Payment By : ADMN GSDML
Payment To : SAGAR TRADERS
ADMN 12345678910 34, AMBARKHAN, AHMERIGANJ, HARIANGANJ
Payment Method : United Commercial Bank Ltd 5675674543

Sl.	Item	Unit Price	Quantity	Total Price
1	FAIR & LOVELY CREAM ADVANCED MULTI VITAMIN 300GM	250	300	75000
2	SANLON ANTIBACTERIAL WET WIPES 20PCS	40	3000	60000
3	PARACHUTE ADVANCED COOLING HAIR OIL	120	100	12000
4	SOVE CONDITIONER OXYGEN MOISTURE 180ML	150	1000	150000
Net Total				297000
Discount(3.5%)				44550
Discounted Total				252450
Paid				252450
Due				0

Destination: Save as PDF
Pages: All
Pages per sheet: 1

Paid: 252,450 Due: 0

Total: 75,000, 60,000, 12,000, 150,000, 297,000, 44,550, 252,450, 252,450

Save Cancel

Figure 6.3.14: A4 Payment Voucher Generation

TR ENTERPRISE

Purchase Details

Supplier*: SAGAR TRADERS From*: 08/05/2021 To: 10/05/2021

Print

Figure 6.3.15: Purchase Details Module

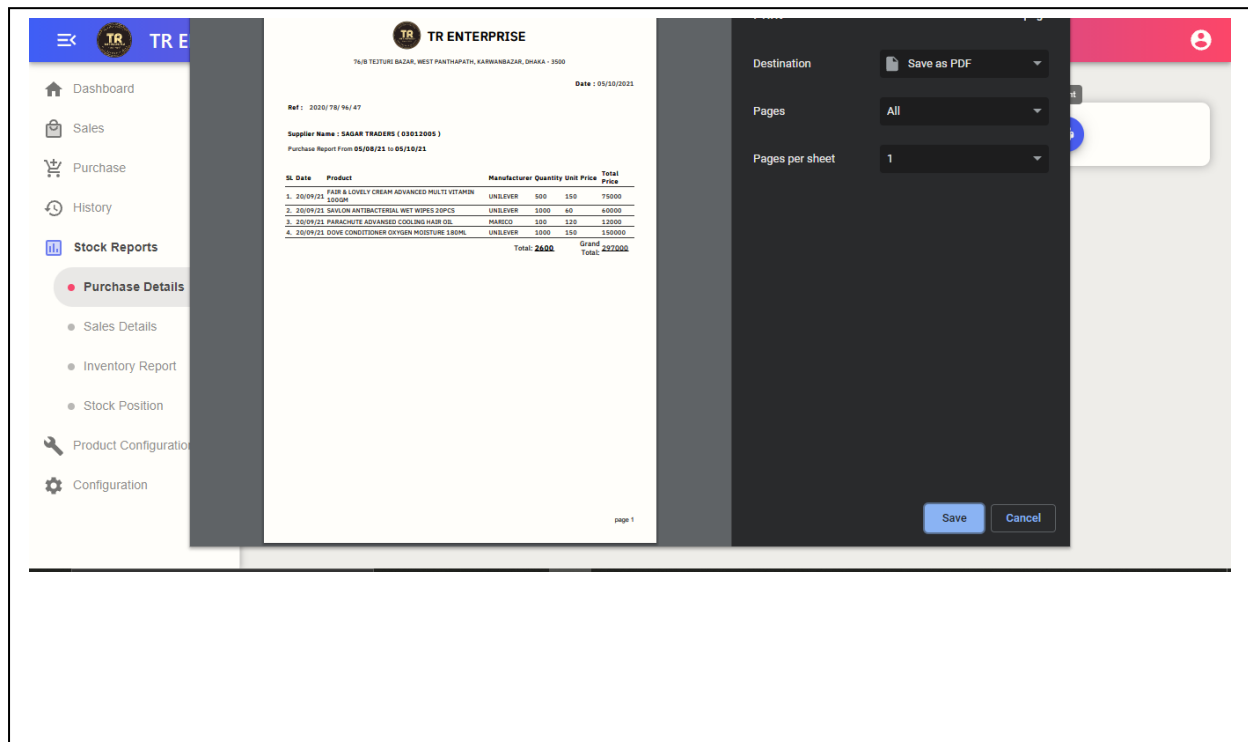


Figure 6.3.16: Purchase Details Module Report

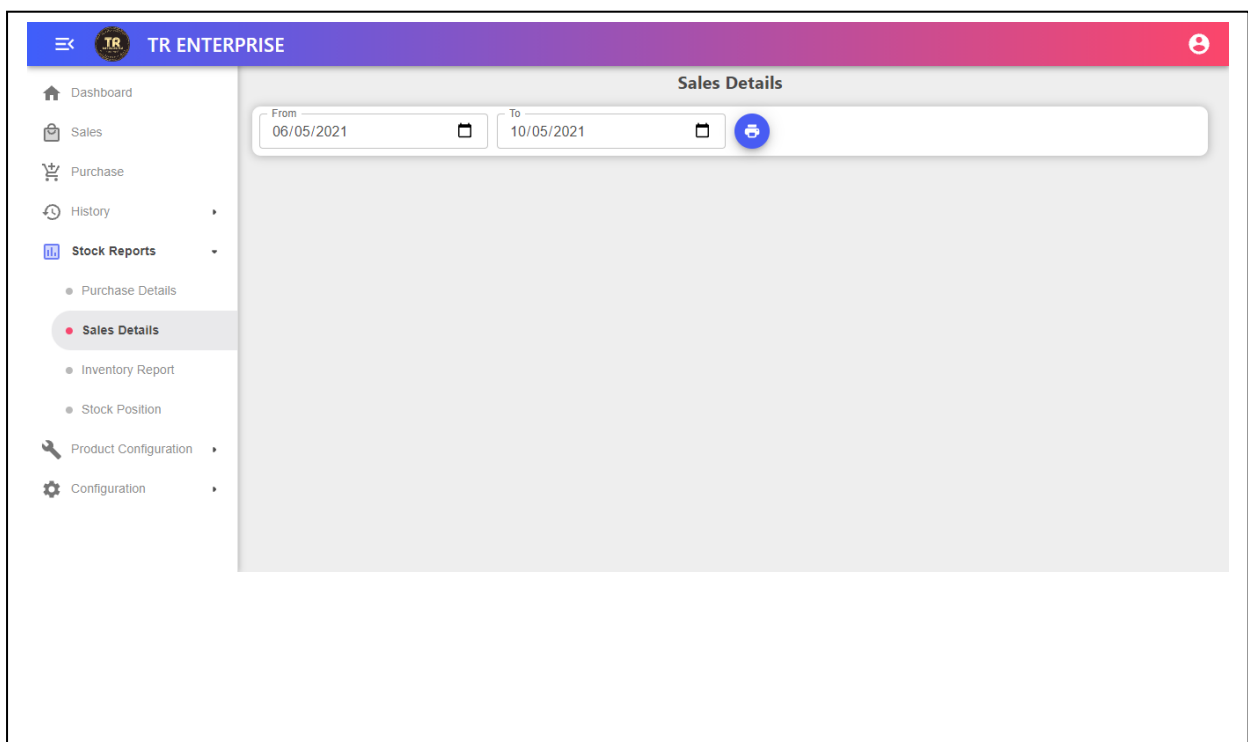


Figure 6.3.17: Sales Details Module

TR ENTERPRISE
T/18 TETURU BAZAR, WEST PANTHAPATH, KARWANBAZAR, DHAKA - 3500

Date : 05/10/2021

Ref : 2020/78/96/47

Sales Report From 05/04/21 to 05/10/21

Sl. No	Date	Product	Customer	Manufacturer	Quantity	MRP	Discount	Total
1.	20/09/21	DOVE CONDITIONER OXYGEN MOISTURE 350ML	UNILEVER	500	270	0	135000	
2.	20/09/21	PARACHUTE ADVANCED COOLING HAIR OIL	MARCO	50	160	0	8000	
3.	20/09/21	SAXON ANTISEPTICAL WET WIPES 20PCS	UNILEVER	500	85	0	42500	
4.	20/09/21	FAIR & LOVELY CREAM ADVANCED MULTI VITAMIN 100GM	UNILEVER	201	230	0	46230	
5.	24/09/21	DOVE CONDITIONER OXYGEN MOISTURE 350ML	UNILEVER	100	270	0	27000	
6.	05/10/21	FAIR & LOVELY CREAM ADVANCED MULTI VITAMIN 100GM	UNILEVER	30	230	0	6900	
7.	05/10/21	DOVE CONDITIONER OXYGEN MOISTURE 350ML	UNILEVER	20	270	0	5400	
					Total: 1461		Grand Total: 221030	

page 1

Destination: Save as PDF
Pages: All
Pages per sheet: 1

Save Cancel

Figure 6.3.18: Sales Details Module Report

TR ENTERPRISE
T/18 TETURU BAZAR, WEST PANTHAPATH, KARWANBAZAR, DHAKA - 3500

Date : 05/10/2021

Ref : 2020/78/96/47

Inventory Report From 05/08/21 to 05/10/21

Sl. No	Date	Product	Manufacturer	Opening stock	In	Out	Closing stock	Unit Price	MRP
1.	20/09/21	PARACHUTE ADVANCED COOLING HAIR OIL	MARCO	0	100	50	50	160	160
2.	20/09/21	FAIR & LOVELY CREAM ADVANCED MULTI VITAMIN 100GM	UNILEVER	0	500	201	299	150	230
3.	20/09/21	SAXON ANTISEPTICAL WET WIPES 20PCS	UNILEVER	0	1000	500	500	85	85
4.	20/09/21	DOVE CONDITIONER OXYGEN MOISTURE 350ML	UNILEVER	0	1000	500	500	150	270
5.	24/09/21	DOVE CONDITIONER OXYGEN MOISTURE 350ML	UNILEVER	500	0	100	400	150	270
6.	05/10/21	FAIR & LOVELY CREAM ADVANCED MULTI VITAMIN 100GM	UNILEVER	299	0	30	269	150	230
7.	05/10/21	DOVE CONDITIONER OXYGEN MOISTURE 350ML	UNILEVER	500	0	20	480	150	270

page 1

Destination: Save as PDF
Pages: All
Pages per sheet: 1

Save Cancel

Figure 6.3.19: Inventory Stock Report

CHAPTER – 7: DEBUGGING AND TESTING

7.1 Purpose of Testing

The purpose of software testing is to access or evaluate the capabilities or attributes of a software program's ability to adequately meet the applicable standards and application need. Testing does not ensure quality and the purpose of testing is not to find bugs. Testing can be verification and validation or reliability estimation. The primary objective if testing includes:

- To identifying defects in the application.
- The most important role of testing is simply to provide information.
- to check the proper working of the application while inserting updating and deleting the entry of the products.

7.2 Type of Testing

I have used one type of testing to ensure the error free features of our software application:

7.2.1 Units Test

This type of testing is the testing of individual software components. It is typically done by the programmer and not by the testers. It requires details information and knowledge about the internal program design and code to perform this.

During unit testing, I carried out various testing task such as the reflection of the unit data on database and its interface. Various types of bugs associated with the component Ire identified and fixed. I use various functional keys to test our software.

In our software unit testing is concerned with the stock units, opening stock units and product units' validation as well as the validation of product units.

CHAPTER – 8: CONCLUSION AND LESSON LEARNT

8.1 Project Limitation

Since this is our first project it has some limitation. Due to less knowledge in particular fields and limited time I am not able to fulfill all our expectations that I expected I could do while the project got started. I hope these limitations are considerable. Some of the project limitations 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 web-based application basically suitable for small organization. It has every basic item which are used for the small organization. I was successful in making the application where I 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.

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 our team learnt are listed as below:

- Learnt about the IMS process.
- Learnt about .Net Core technology, its components and ways to implement them
- Learnt to work in pressure and to be patient.
- Learnt to manage the database under MySQL.

8.4 Future Enhancements

Since this project was started with very little knowledge about the Inventory Management System, I came to know about the enhancement capability during the

process of building it. Some of the scope I can increase for the betterment and effectiveness
oar listed below:

- Interactive user interface design.
- Use of Oracle as its database.
- Online payment system can be added.
- Making the system flexible in any type.
- Sales and purchase return system will be added in order to make return of products.

Lost and breakage