THE SUPERIOR COLLEGE LAHORE

Faculty of Computer Science & IT
Final Year Project

POS Jewellery Software

Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Tayyaba Khurram	MCSW-S19-006	MCS	03214649065	tayyabakhurrum2@gmail.com
Urwa Shafq	MCSM-S19-009	MCS	0347 4068910	Mcsm-s19-009@superior.edu.pk

Rohail Shahzad
Supervisor

Project Report
POS Jewellery Software

Change Record

Table 1 Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Tayyaba Khurram, Urwa Shafq	1.0		Proposal	
Tayyaba Khurram, Urwa Shafq	2.0		Template 3	
Tayyaba Khurram, Urwa Shafq	3.0		None	
Tayyaba Khurram, Urwa Shafq	4.0		The Project Plan for the desktop application will be used to define the work that will be done for this project. It is also indicative of who will do each project. The Statement of Work, Resource List, Roles, Risks, and Schedule will all be stated in this document.	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

_____

HEAD OF THE DEPARTMENT

Comments: _____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

Date:_____ Signature:_____

Dedication

This work is dedicated to my Parents because my Parents motivate me to do this and give me suggestion and discuss everything about the market, support us to make this project, and help me. We would like to dedicate this dissertation to our parents who

always stand with us in every step of life.

Acknowledgements

We are really thankful to our supervisor who has guided us to make such a wonderful Desktop Application. Who taught us how to manage our work and tell us the pros and cons of our idea. He deals with us politely and gives us suggestions. He has the best communication level and is very supportive, gave me and my group more or more motivation and confidence. He did not refuse our idea but he enhances that idea.

Executive Summary

We are providing the Point-of-Sale(POS) Retail Jewellery software to create Cash Receipts, Invoices, and receive A/R Payments. In addition, you can create returns for Receipts and Invoices. The POS system replaces the cash register in a retail store. The POS system can track sales by various payment methods like cash, cheque and credit cards.

The POS system requires that you have the Accounts Receivable and the Inventory modules. For example, POS will update inventory sold figures in Inventory, or the salesperson's commission in Accounts Receivable. In addition, if you want to create Sales Orders, you will need the Order Entry (O/E) module. Other modules are optional.

The POS system replaces your manual cash drawer with a personal-computer. In addition, you can attach accessories like the POS Cash-drawer and the Column Receipt Printer.

We are providing IT based solution to different Jewellery shop holders so that their manual system can be automated. The system will provide all accessories and expense details. We are targeting a shop as a pilot and the whole market for overall business.

Today people prefer automated system overall manual system. We have also a noticed problem that normally Jewellery shop holders are still using manual systems through which they face lots of problems as data loss, payments issue, etc. There are very few shop holders who are using software-based solution. We are making desktop Application for Jewellery shop holders and its customers. We are also going to add a feature of add all detail of product, purchase detail, sale detail, expense detail, and how much is the monthly profit have we made this month and also the loss detail. The project objective is to convert the manual system into automated system.

Tables Of Contents

List of Figures

Chapter 1
Introduction

Chapter 1: Introduction

POS is known as a point of sale or point of purchase where a transaction is carried out once the customer picks the product for buying. If you are a retailer or restaurateurs, you cannot run a demanding business without having a proper POS setup. The complete setup of POS system includes computer, credit card reader, receipt printer, cash drawer and POS software. It is very useful for keeping the track record of inventory and the number of sales made per day.There are many benefits of using this method including better user experience, retention of customers, price customization of every product, and keeping the track record of the previous purchase history of the customers. This process refrains you from using old traditional methods of carrying out sale process in which cash registers were used.

Today people prefer automated systems overall manual systems. We have also a noticed problem that normally Jewellery shop holders are still using manual systems through which they face lots of problems as data loss, payments issue, etc. There are very few shop holders who are using the software-based solution. We are making desktop Application for Jewellery shop holders. We are also going to add a feature of add all detail of product, purchase detail, sale detail, expense detail, and how much is the monthly profit have we made this month and also the loss detail. And we are targeting desktop application for entire market.

1.1. Background

The existing system was not very effective & was highly time-consuming. The current system works manually. The existing system was somewhat paper-based (paper-work). Which takes time. The rate at which the work is done is very slow due to the non-utilization of a computerized system. It does not provide better data facilities as well as data availability on a fingertip. Inconsistency was the major problem in the existing system as there is no proper facility was provided to update the data. In case the user wants to find out details of particular Jewellery whole database records are displayed. It is difficult to remember all data Consumes a large volume of paperwork Large storage space is required to keep to the files and register in proper coordination.

1.2. Motivations and Challenges

The target clients of the project are the common people of Pakistan or other countries. The friendly interface would be provided for the users that will guide them, how they get benefited from the services that the Desktop Application provides. According to their need, the prominent features of our desktop app motivate the user to use this desktop ap

1.3. Goals and Objectives

Generally, the main goal of this research is to salvage the increasing problem of the said company by developing a computer-based pos and inventory system to replace their old manual and traditional way of doing it.

Specifically, the system aims to;

- Achieve accuracy and reliability of calculation and totaling of products purchased.
- Reduce time of the organization in inventorying their products.
- Easily modify the products details.
- Reduce pressure on the labor and man power from the repetitive and dull job of doing manual process of getting the products.

1.4. Literature Review/Existing Solutions

A point of sale is a point at which a sale is made, the ownership (and usually the possession) is conveyed from the retailer to the buyer, and indirect taxes become payable. A point of sale is, commonly, a retail outlet. A point of sale system is a supply net administration system for customer management that delivers real-time control of merchandise in stock and sale analysis. POS systems deliver valuable, near-real-time information on sales, such a system can be used to update inventory stock status and generate purchase orders as needed.

Products are considered as the business resources for the organization. This includes managing the product inappropriate 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. This application software is 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 the 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 a way or other.

1.5. Gap Analysis

Our product is a new brand in the market we will provide a plate form for shop holders who manages their inventory. We will introduce a unique product which is so much different we will provide the services.

1.6. Proposed Solution

This project is to develop for Sales and Purchase Inventory Management System for a Jewellery Shop. This system can be used to store the details of the inventory, update the inventory based on the sale details, purchase detail, produce receipts for sales, generate sales and inventory reports periodically, etc. and the admin component (used by the administrators for performing admin level functions such as adding new items to the inventory, changing the price of an item, etc.)

- Detail of product,
- Purchase detail,
- Sale detail,
- Expense detail,
- Stoke details.
- Order different Jewellery accessories

1.7. Project Plan

The Project Plan for the desktop application will be used to define the work that will be done for this project. It is also indicative of who will do each project. The Statement of Work, Resource List, Roles, Risks, and Schedule will all be stated in this document.

1.7.1. Work Breakdown Structure

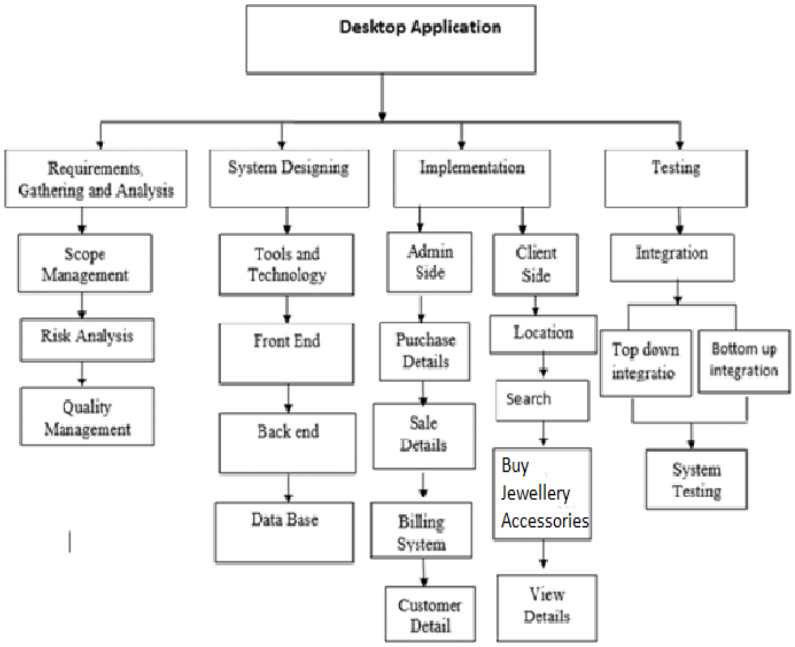


Figure 1 WBS

1.7.2. Roles & Responsibility Matrix

Table 2 Roles & Responsibility

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Planning	Proposal	Proposal Defense	15 Days	Tayyaba Khurram, Urwa Shafq
2	Documentation	Report	Making Report	3 months	Tayyaba Khurram, Urwa Shafq
3	Front End, Back End	Front End: .NET Framework & C# Back End: SQL Server	Interface Back End	5 months	Tayyaba Khurram, Urwa Shafq

1.7.3. Gantt Chart

Figure 2 Gantt chart

1.8. Report Outline

In this chapter, we discuss existing system problems and also describe their solutions in our project also discussed the background of the proposed system. We discuss different kinds of challenges about our project and discuss how to work on our project step by step to meet

The timeline and show how that project is used to help the people around us.

Chapter 2

Software Requirement Specification

Chapter 2: Software Requirement Specifications

2.1. Introduction:

In this chapter, we will discuss the purpose of the project we will discuss the feature of the project and explain the implementation and design intended Audience and reading suggestions product perspective and many more things about the project.

2.1.1. Purpose

We would be targeting Jewellery shop holders for using a manual inventory system. Our product is for Jewellery shop holders. We are going to target lower class shop holders who don't have any idea about the rules and regulations of the business; we have also a noticed problem that normally jewellery shop holders are still using manual systems through which they face lots of problems as data loss, payments issue, etc. There are very few shop holders who are using the software-based solutions and then I have developed a Desktop Application.

2.1.2. Document Conventions

We use the Italic Calibri font to write the Dedication and the Acknowledgment at the start of the document.

We use the Calibri font family with the 12pt font size to write all the details of the document.

We use the Calibri font family with the 14pt font size to and bold to write all the headings in the document.

We use the Calibri font family with the 11pt font size to write all the description of the figures and tables of the document.

2.1.3. Intended Audience and Reading Suggestions

This document is written for developers of the system, project manager, client, and complete staff of a retail store including an IT engineer who will look after the system after it is installed and operational. The document is written properly so the reader should read this document in the defined order.

2.1.4. Product Scope

The scope of this system is to provide user efficient working environment and more output can be generated through this. This system provides a user-friendly interface resulting in knowing each

and every usability feature of the system. This system helps in tracking records so that past records can be verified through them and one can make decisions based on the past records. This system is developed in such a way that even a naive user can also operate the system easily. The calculations are made very quickly and the records are directly saved into databases and the databases can be maintained for a longer period of time. Each record can be retrieved and can be verified for future transactions. Also, this system provides a high level of security for data leaking as only admin people can access the database no changes can be made to it until it verifies the user login id and password. We also have an operator login through which operators can take orders but can't make changes in the database. Limited access is available to the operator.

The following are the few features that we will provide in our system:

•Detail of product,

•Purchase detail,.

•Sale detail,

•Expense detail,

•Stoke details

•Order in Different Jewellery accessories,

2.1.5. References

2.2. Overall Description:

We will define the product perspective, product functions and features of our system user class and characteristics.

2.2.1. Product Perspective

The software described in this SRS is the software for a complete Desktop Application. The system is based on just software there is no hardware include in this product.

Our main perspective to provide a platform to such fame's from where they can get opportunities like Money without any interest and provide a safe and profitable platform for shop holders.

2.2.2. Product Functions

The main function of our product is

- Detail of product
- Purchase detail
- Sale detail
- Expense detail
- Stoke details.
- Order in Different Jewellery accessories

2.2.3. User Classes and Characteristics

The end-users of this project fall into three user class categories,

- unskilled
- partly skilled
- Highly skilled

2.2.3.1. Unskilled user:

Unskilled user is those type of user who hasn't relevant prior skill or education other than basic abilities to operate a system.

2.2.3.2. Partly skilled user:

They should be able to use the system and further be able to train others with minimal training themselves. They must be able to explain all elements of the user interfaces except the server. This class of users would be expected to have a junior-high-school certificate education or equivalent.

2.2.3.3. Highly skilled user:

The software should not be needlessly complex, but it is still expected not to be entirely 'plug and play'. This class of users is expected to have a high-school certificate or equivalent, as well as extensive computer experience.

2.2.4. Operating Environment

- .NET Framework & C# is use for front-end in our project.
- SQL Server is use for back-end in our project.
- Local Host is used for server in our project.

2.2.5. Design and Implementation Constraints

Tools and frameworks for the usage of front end, back end and database

- For front end we are using C# & .NET frameworks for Win-Form Applications
- SQL server for writing back-end
- We are using MSSQL server for database and we will run it locally

2.2.6. User Documentation

The Desktop Application we are making is quite user-friendly there is no such need for the manuals, but for the ease of the people when we will upload different materials such as manual, we will also upload a video that will be telling how to use it. Secondly, we shall describe the whole in the description area. These are enough for the user to understand it.

2.2.7. Assumptions and Dependencies

- Each user must have a valid user id and password
- Users must log in to the system to access any record.
- Only the Administrator can delete records.
- Client/User runs an operating system that supports minimum requirements.
- Information retrieval from the Database is important.

2.3. External Interface Requirements

2.3.1. User Interfaces

Before implementing the actual design of the project, a few user interface designs were constructed to visualize the user interaction with the system. This built-in

interface creation capability has had the future re benefit of the standard on the user interface. User can move from one program to another and see the same basic interface tools with allowing them to concentrate solely on the using capabilities of the Application.

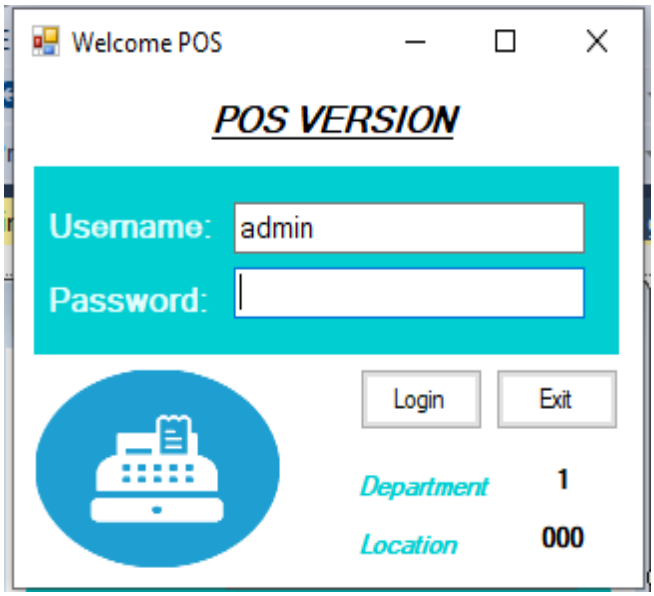


Figure 3 User Interface

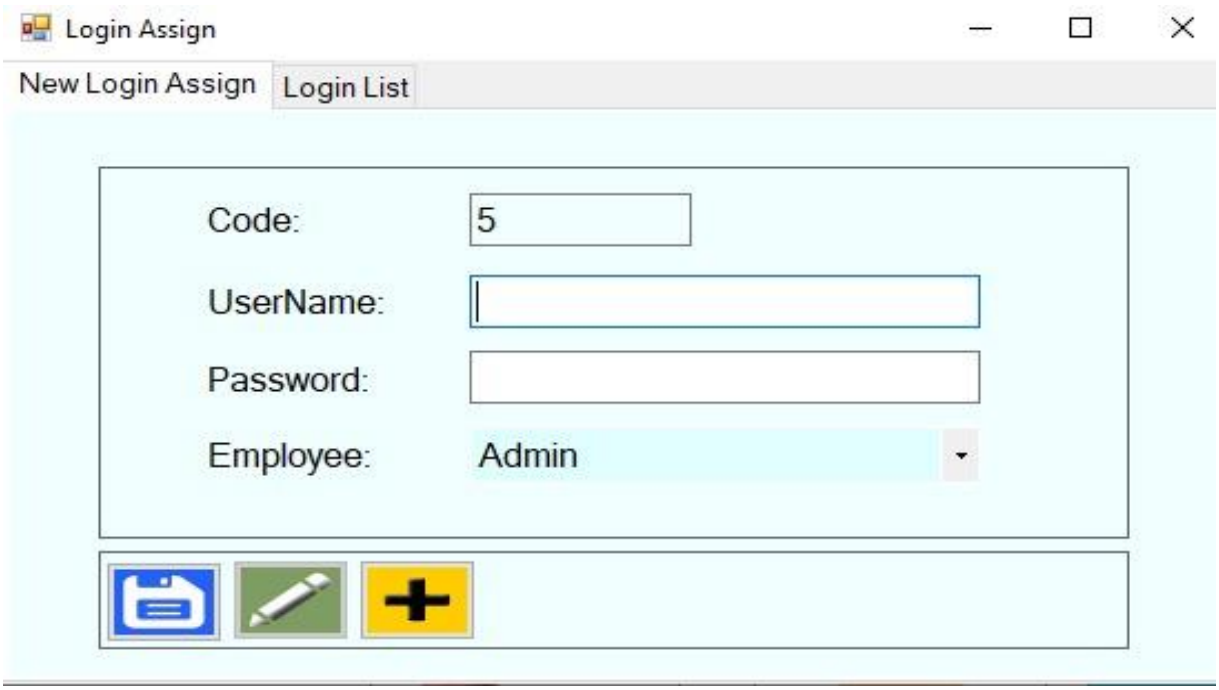


Figure 4 User Interface

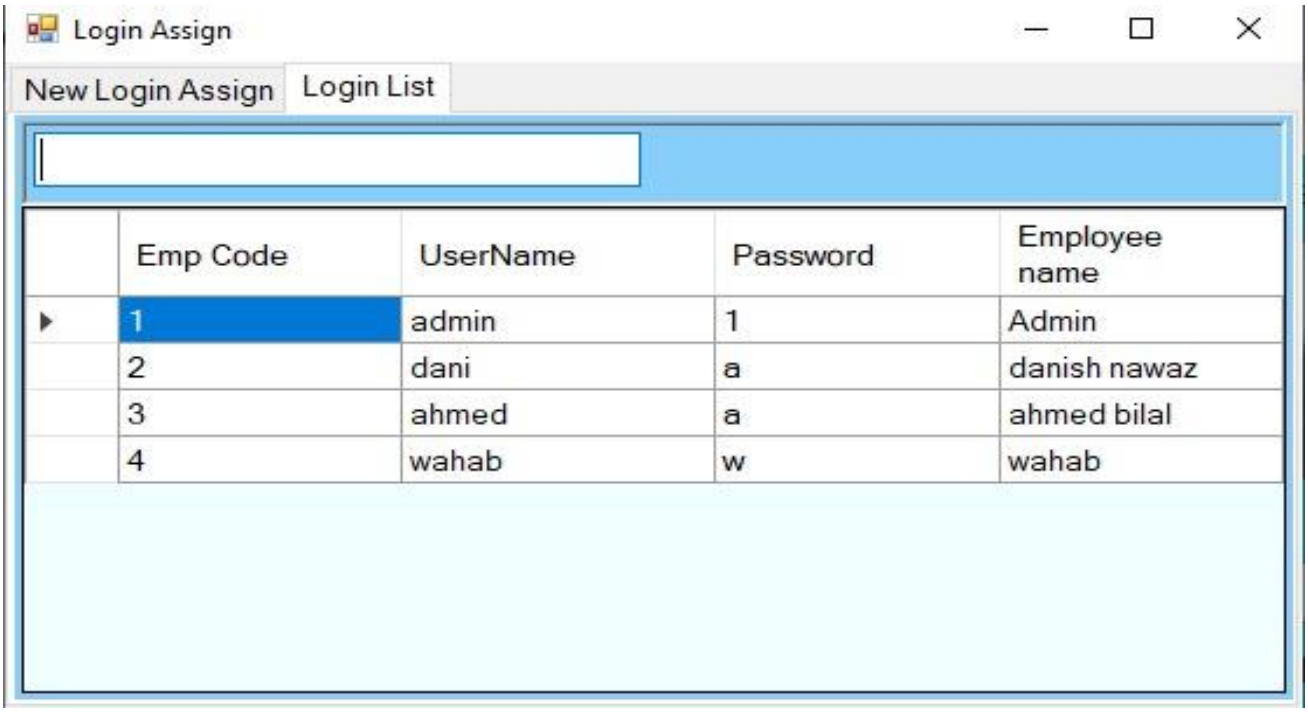


Figure 5 User Interface

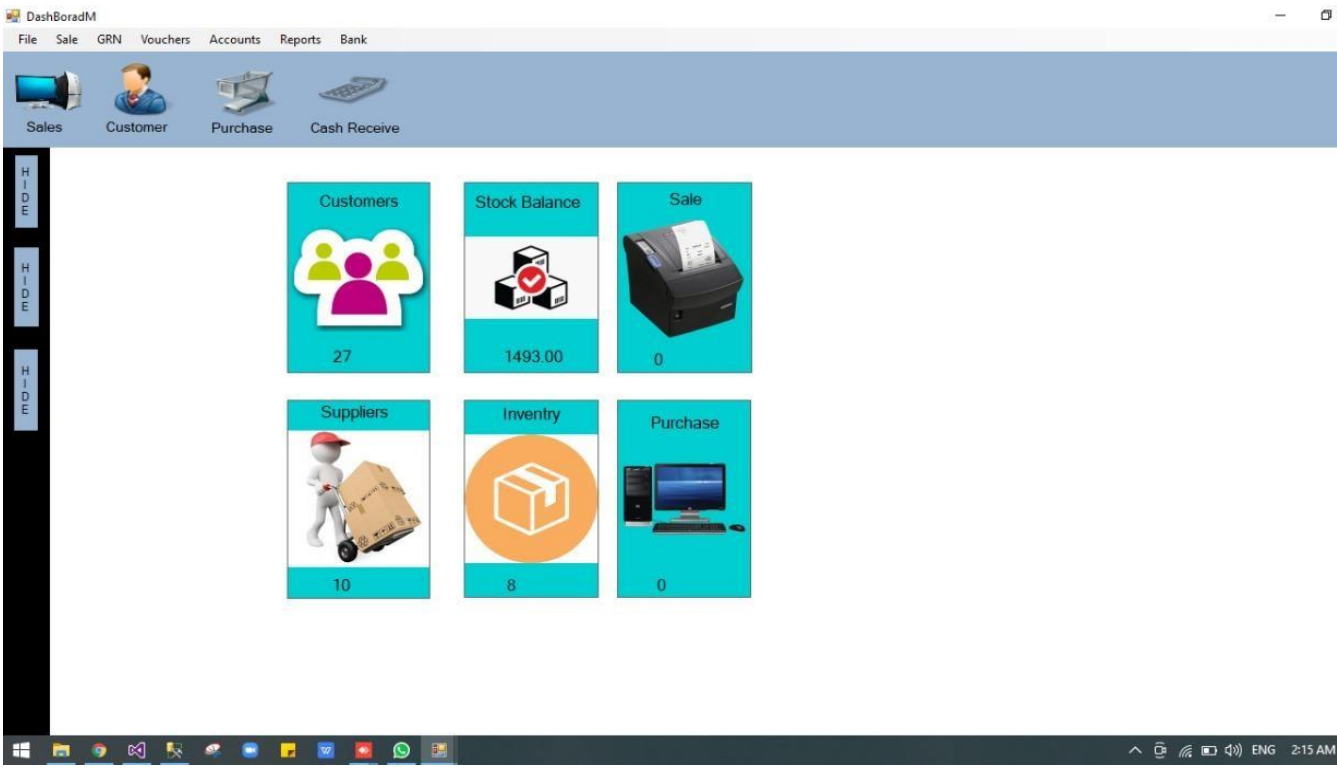


Figure 6 User Interface

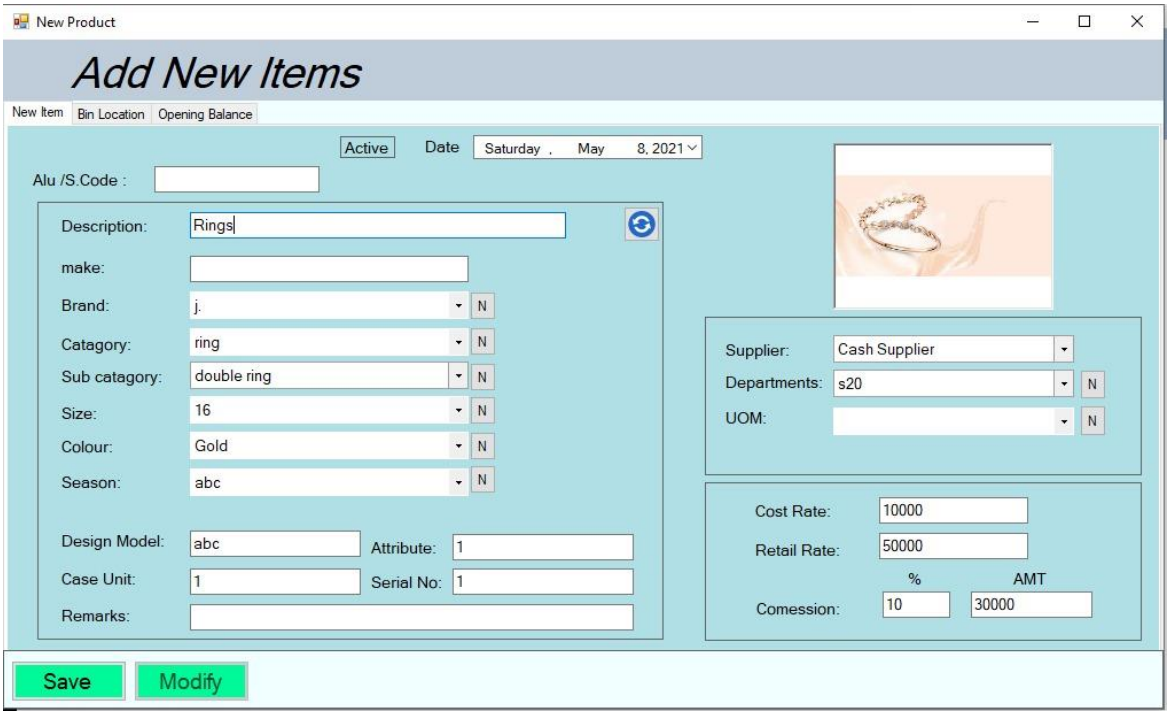


Figure 7 User Interface

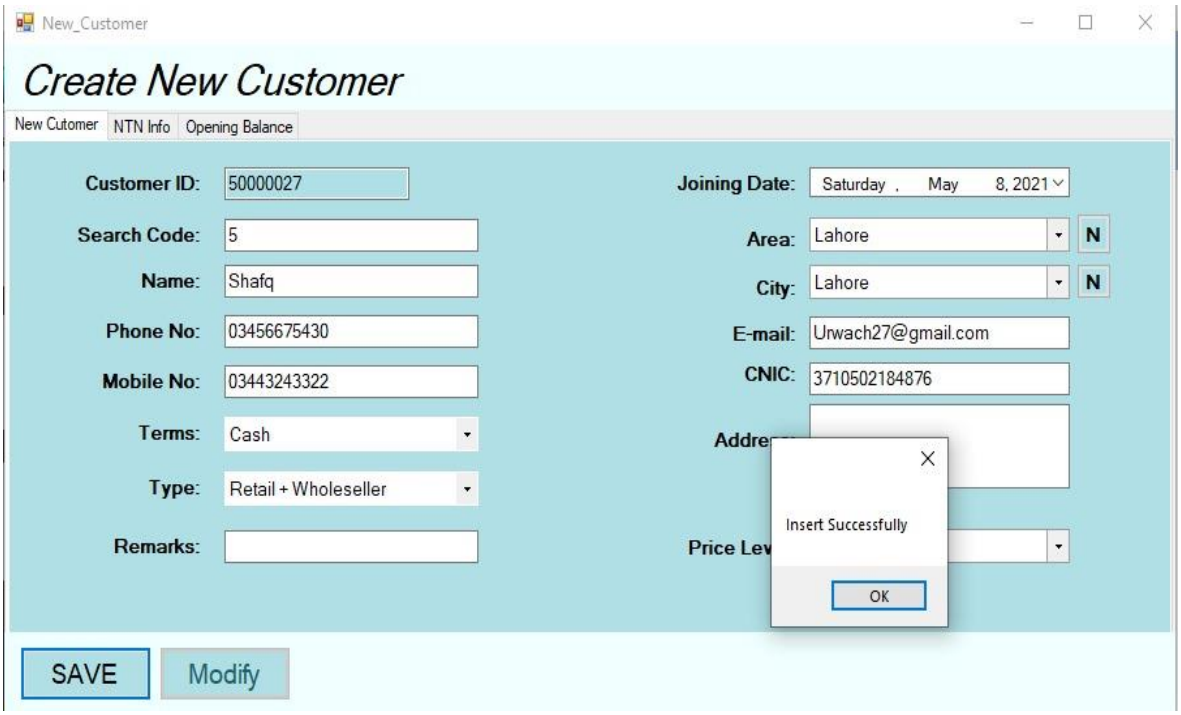


Figure 8 User Interface

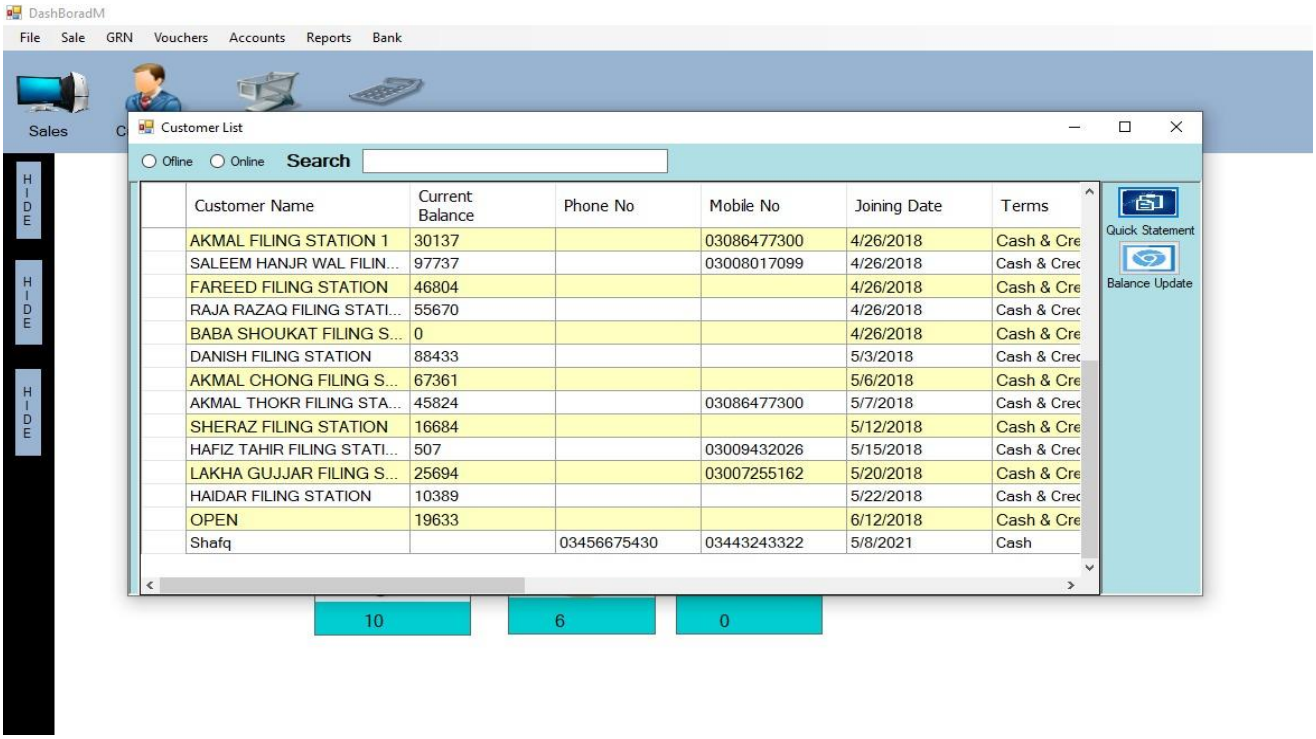


Figure 9 User Interface

Figure 10 User Interface

Figure 11 User Interface

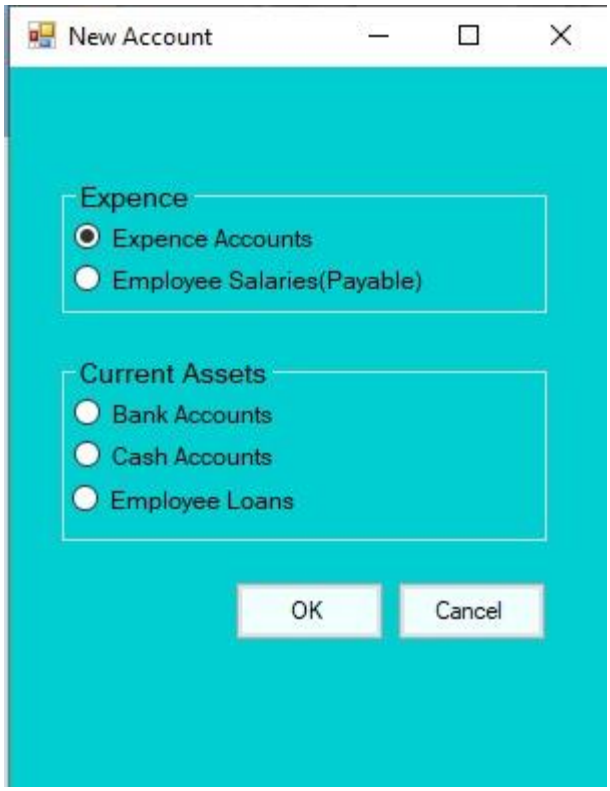


Figure 12 User Interface

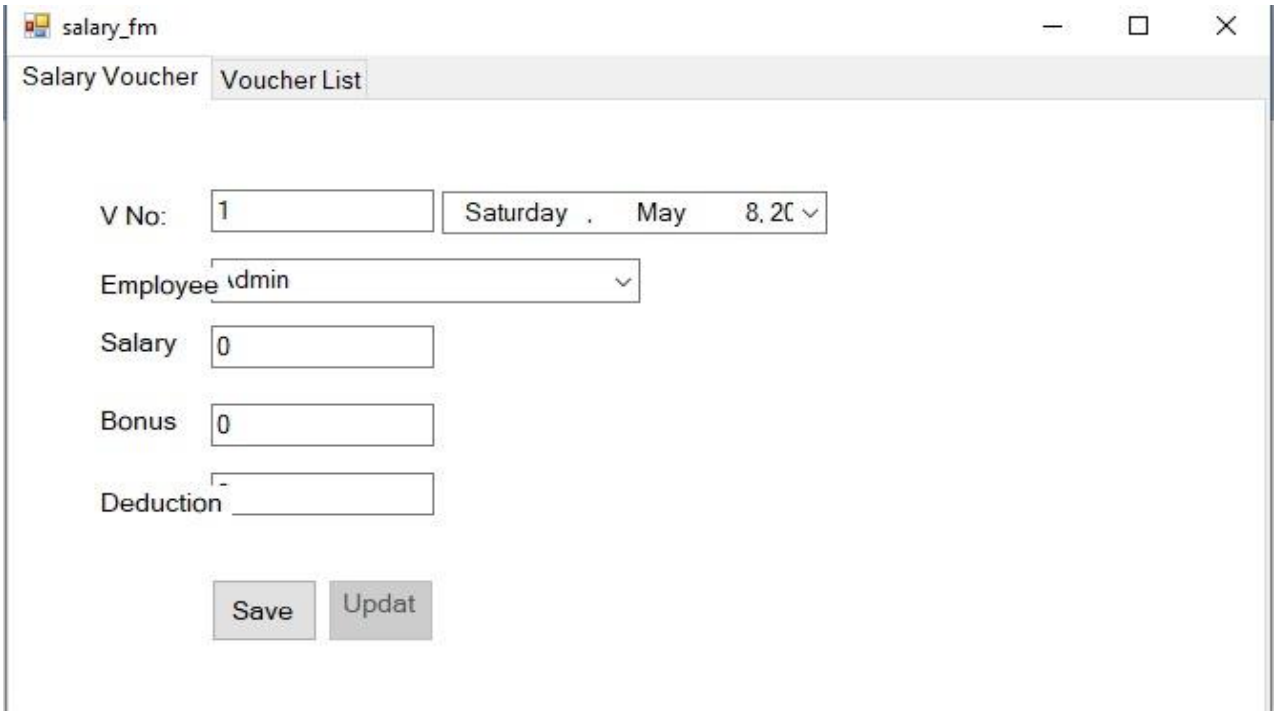


Figure 13 User Interface

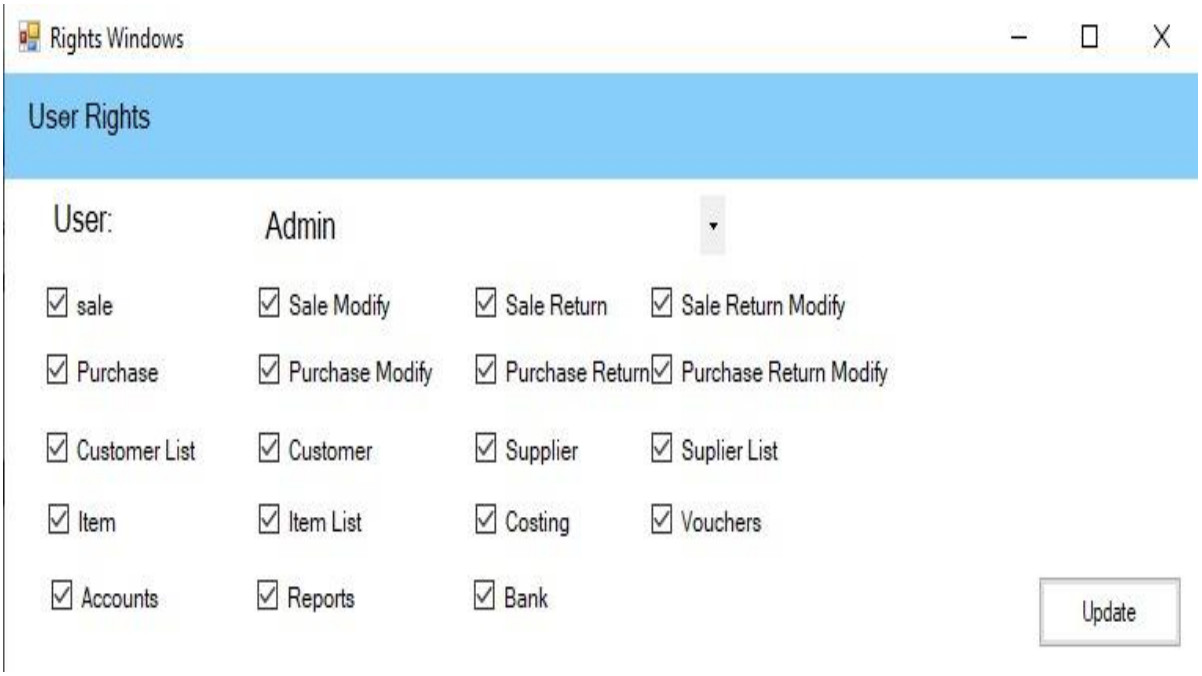


Figure 14 User Interface

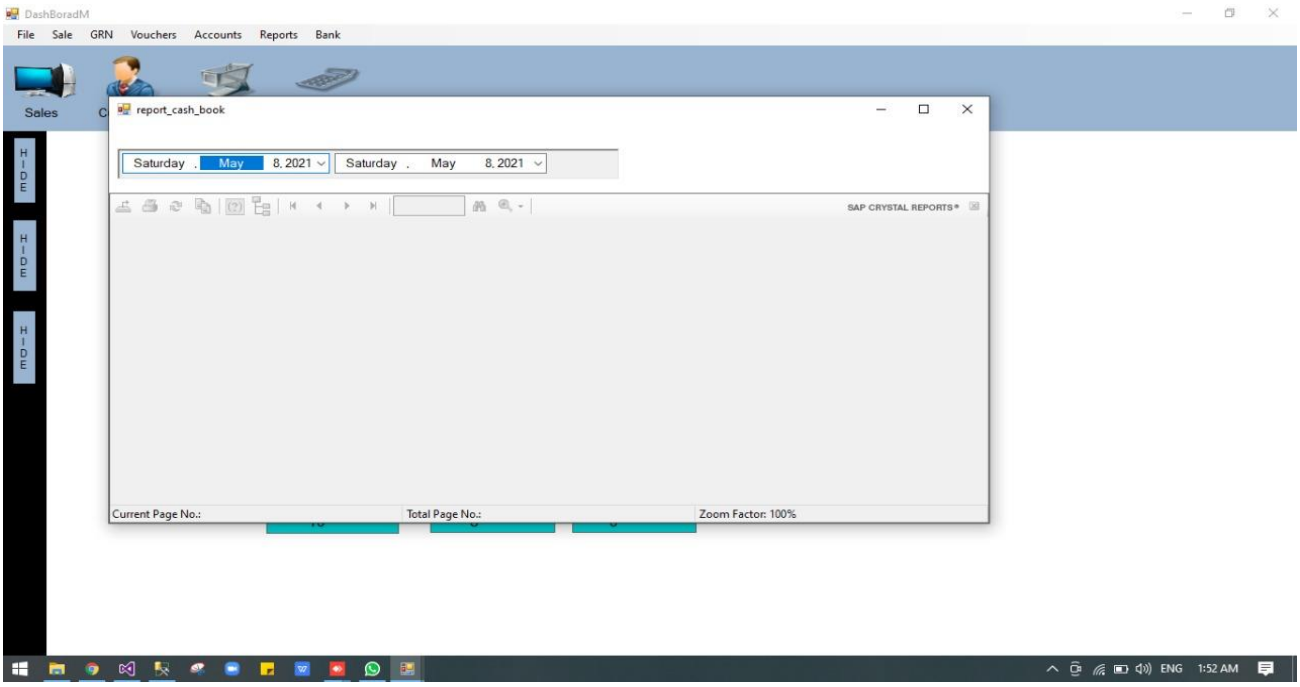


Figure 15 User Interface

2.3.2. Hardware Interfaces

- POS service requires a PC able to run windows 7 or higher.
- POS server side service requires the ability to grow dynamically based upon number of clients and frequency of communications but does not initially require a large amount of space or resources to run.
- A functional mouse and keyboard are required for fully functionality of both the client and server sides of POS.
- The system enables to maintain the computerized record of customers.
- It will also perform quick search for the availability of the stock.
- The system requirement specification (SRS) will also provide a detailed information on the internal and external view of the system as well as interfaces required by the desktop app.
    - Processor : Intel Core Duo 2.0 GHz or higher. RAM : Minimum1 GB or Greater.
    - Hard disk : 20 GB.
    - Basic system which is capable of using internet.

2.3.3. Software Interfaces

Operation System: Windows 7 or higher.

2.3.4. Communications Interfaces

In this project, we aim to help people to manage their records like purchase records, sales records, product details, customer records etc.The Software Requirement Specification states system interfaces, user interfaces, software and hardware interfaces, communication interfaces and functional and nonfunctional requirements in detail. Developer of the project will make use of this document throughout the implementation process.

2.4. System Features

The system must be able to show information to user in real time. The system must be able

- The system shall provide a registration page.
- The system shall provide a login page.
- The system must be able to show product details.
- Database should be working fine
- *The system supports customers purchased receipt.*
- *System can search the product from the stock according to customers demand.*
- *System can add stock.*
- *System can update stock.*
- *System can delete stock.*
- *System can show the stock report.*
- *System can show the sales report.*
- *System can register new staff.*
- *System can add customer service.*
- *System can update customer service.*
- *System can view all the service records according to product specific ID.*
- *System can update password (Admin & Staff)*
- *Simple GUI for the user.*
- *A menu bar for complete menu for available functionalities in system.*
- *Search bar to ease the user to search their desire information.*
- *After adding items into the record there must be a button to place/cancel the items from record.*
- *After this process at the end, there are different suitable payment methods for user to avail like online payment by their bank accounts/credit cards.*
- *At the end their must be a receipt showed up at the screen of user with complete details like order number, delivery date, price etc.*
- *If POS is used by a single user then there is no need for an internet connection but if it is used by multiple users then a LAN network is needed.*

2.4.1. Search:

2.4.1.1. Description and Priority

This feature is used by the user to search for their required information in the database. The system will show all the data related to that searched word. The priority of this feature is very high because the user must have to search in order to maintain the records or for some special purpose.

2.4.1.2. Stimulus/Response Sequences

The user first registers himself then login. After that the user click on the search bar then searches the required data.

2.4.1.3. Functional Requirements

Functional requirements describe a function of software or its components. A function explains the behavior, a set of inputs processing and output.

REQ-SF1-1: user should see the sign of the search.

REQ-SF1-2: the system has to search for the keywords the user entered.

2.5. Other Nonfunctional Requirements:

Functional requirement is referring to the functionalities must be applying to a system. The system must be able to show information product.

2.5.1. Performance Requirements

The Desktop Application must perform best when the user is using it. There should not be any lagging in the Desktop Application. So performance must be ensured. The developer must keep these non-functional requirements in the mind as well.

2.5.2. Safety Requirements

The implementation of Desktop Application based on the purchases and sales of Jewllery and thus provides safety and improves the quality of service provided to the customer. Therefore, we can conclude that POS will be used for saving all the records of the customers.

2.5.3. Security Requirements

In regard to authorization, payment, and bank account information, all sensitive information is sent over encrypted channels and databases that are managed by other established services. POS Jewellery

Desktop Application does not store or directly manipulate any sensitive banking information. Rather, it only asks for and archives usernames, passwords, and information directly related to the POS Jewellery Desktop Application. This information is also encrypted and not available for public access

2.5.4. Software Quality Attributes

Software Quality Attributes are features that facilitate the measurement of performance of a software product by Software Testing professionals, and include attributes such as availability, interoperability, correctness, reliability, learnability, robustness, maintainability, readability, extensibility, test-ability, efficiency, and portability. High scores in **Software Quality Attributes** enable software architects to guarantee that a software application will
perform as the specifications provided by the client.

2.5.4.1. Availability

This attribute is indicative as to whether an application will execute the tasks it is assigned to perform. Availability also includes certain concepts that relate to software security, performance, integrity, reliability, dependability, and confidentiality. In addition, top-notch availability indicates that a software-driven system will repair any operating faults so that service outage periods would not exceed a specific time value.

2.5.4.2. Interoperability

Software-driven systems could be required to communicate and act in tandem to solve certain tasks. Interoperability describes the ability of two systems to engage in the exchange of information via certain interfaces. Therefore, Software Quality Assurance engineers must examine the interoperability attribute in terms
of both syntactic and semantic interoperability.

2.5.4.3. Performance

This attribute pertains to the ability of a software-driven system to conform to timing requirements. From a testing point of view, it implies that Software Testing engineers must check whether the system responds to various events

Within defined time limits. These events may occur in the form of clock events, process interruptions, messages, and requests from different users, and others.

2.5.4.4. Test-ability

Software test-ability indicates how well a software-driven system allows Software Testing professionals to conduct tests in line with predefined criteria. This attribute also assesses the ease with which Software Quality Assurance engineers can develop test criteria for a said system and its various components. Engineers can assess the testability of a system by using various techniques such as encapsulation, interfaces, patterns, low coupling, and more.

2.5.4.5. Security

This attribute measures the ability of a system to arrest and block malicious or unauthorized actions that could potentially destroy the system. The attribute assumes importance because security denotes the ability of the system to protect data and defend information from unauthorized access. Security also includes authorization and authentication techniques, protection against network attacks, data encryption, and such other risks. It is imperative for Software Testing professionals to regularly conduct updated security checks on systems.

2.5.4.6. Usability

Every software-driven system is designed for ease of use to accomplish certain tasks. The attribute of usability denotes the ease with which users are able to execute tasks on the system; it also indicates the kind of user support provided by the system. The most well-known principle for this property is KISS (Keep It Simple Stupid). In addition, Software Quality Assurance engineers must test software to check whether it supports different accessibility types of control for people with disabilities. Usability has a critical and long standing bearing on the commercial fortunes of a software application or package.

2.5.4.7. Functionality

This attribute determines the conformity of a software-driven system with actual requirements and specifications. Most Software Testing professionals view this attribute as crucial and a foremost requirement of a modern application, and would therefore advocate the performance of tests that assess the desired functionality of a system in the initial stages of Software Testing initiatives.

2.5.5. Business Rules

In this Desktop Application, database should be working otherwise this application did not work.

2.6. Other Requirements

2.6.1. Non-Functional Requirement:

- The system can save stock into the database safely.
- The system can support all the PC (Personal Computer).
- The system can create a backup database file after every transaction (sales, stock, service, update of authentication details).
- Stock should be added after end of sales per day.
- For security issues only admin can change the password on behalf of staffs.
- Staffs can only access this system for sales, service and checking reports.

Chapter 3

Use Case Analysis

Chapter 3: System Analysis

In this chapter, we will see the use cases of our project with detail description in the form of fully dressed use cases. Use cases are the most important diagram of the whole system that explains the system very easily.

3.1. Use Case Model:

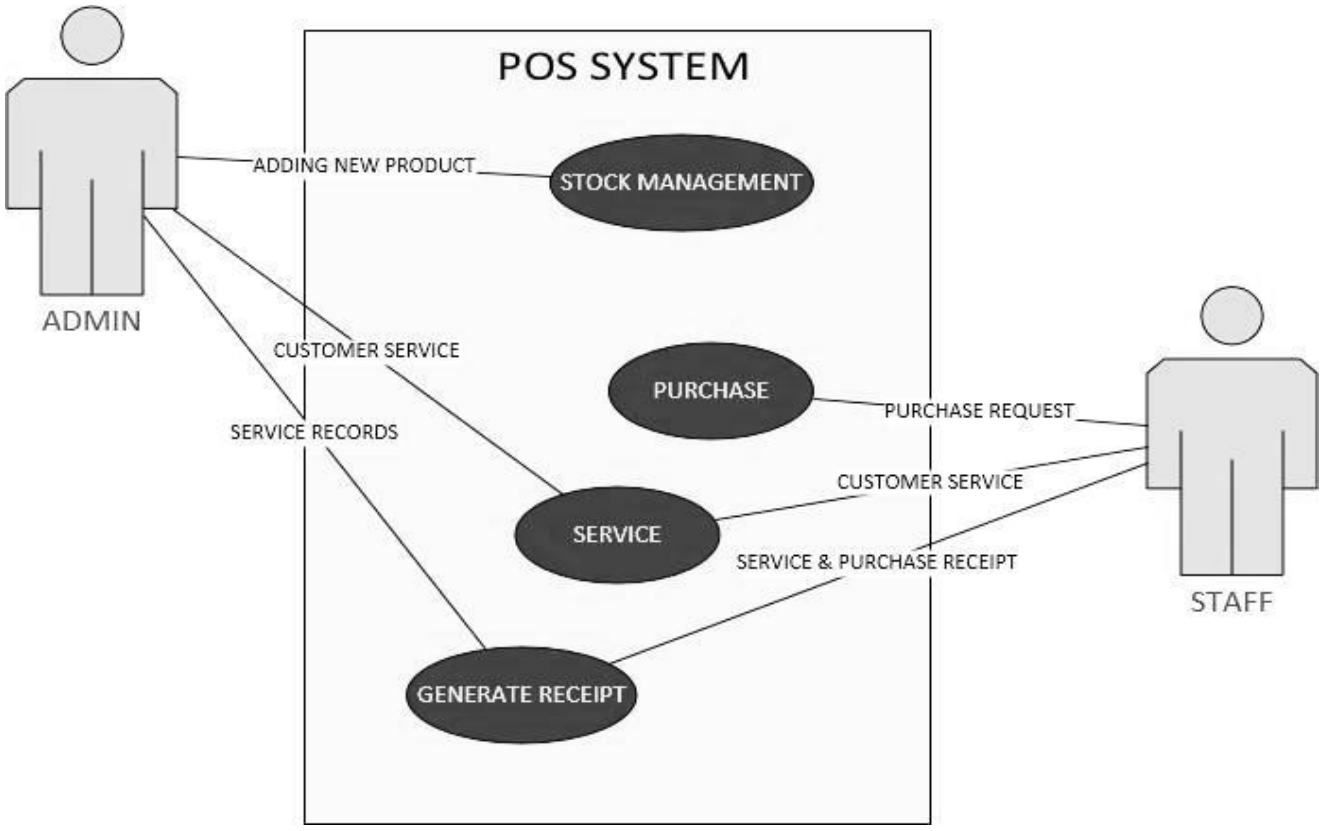
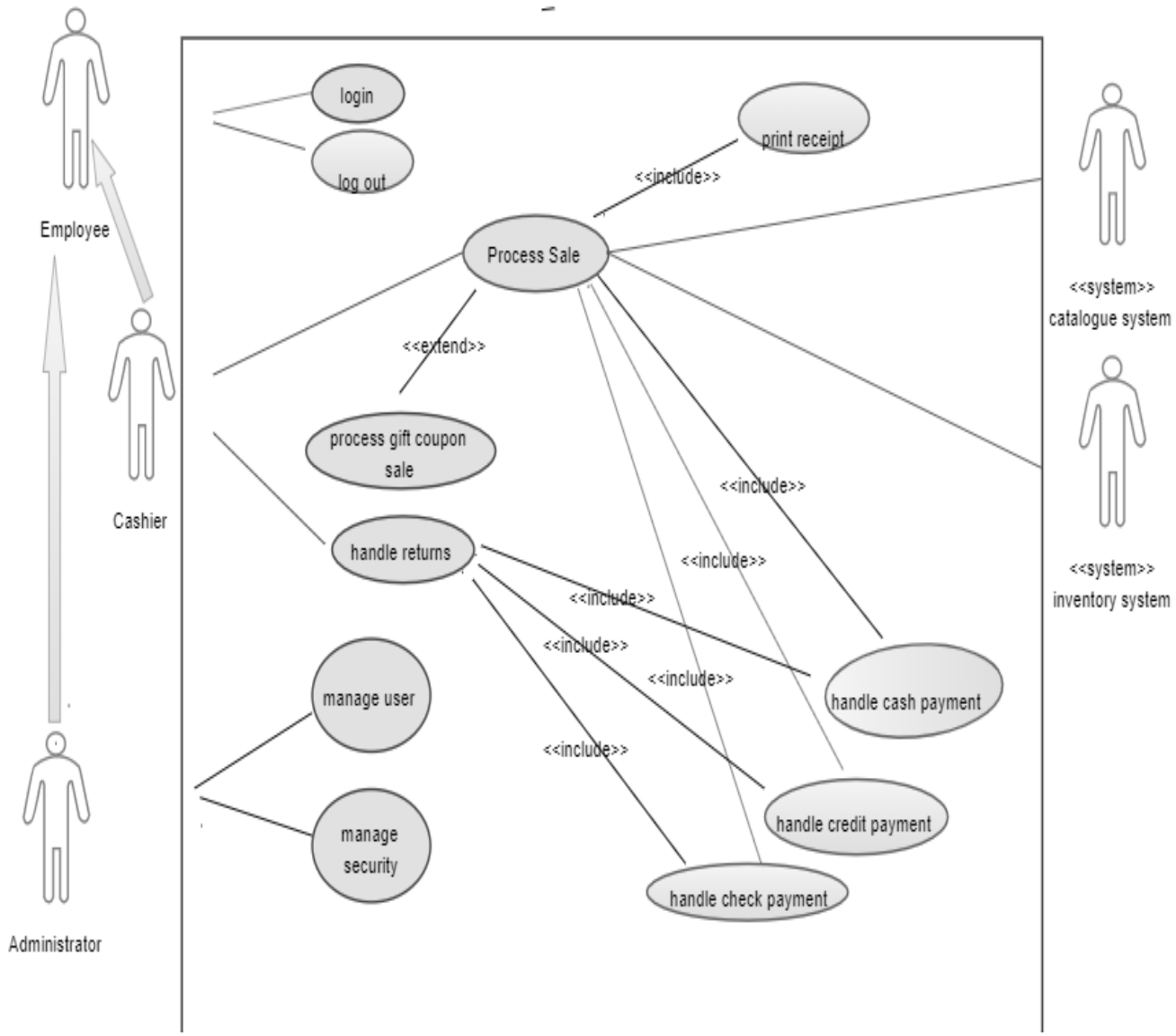
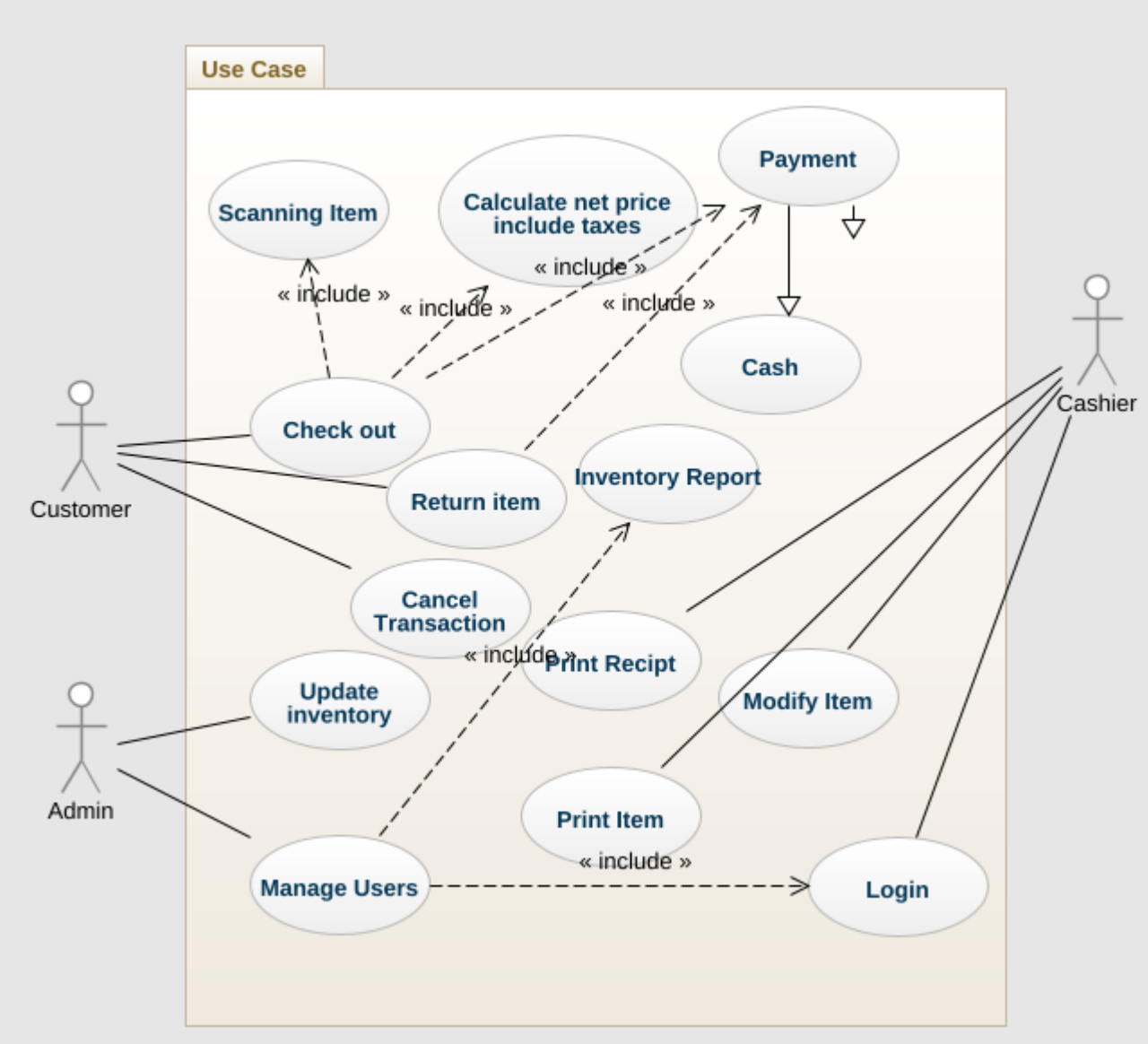


Figure 16 Use Case Model



Use case Diagram


Use Case
Payment
Scanning Item
Calculate net price include taxes
« include »
« include »
« include »
« include »
Cash
Cashier
Check out
Customer
Return item
Inventory Report
Cancel Transaction
« include »
Print Recipt
Update inventory
Modify Item
Admin
Print Item
« include »
Manage Users
Login

3.2. Fully Dressed Use Cases:

Full Dressed Use Case 1:

Table 3 Fully Dress Use Case 1

USE Case	Search
Stake Holder	Admin, user
Priority	1
Precondition	None
Post condition	Can view product
Extension	System down
Non -Functional Requirement	price,
Status	Pass

Full Dressed Use Case 2:

Table 4 Fully Dress Use Case 2

USE Case	Sell
Stake Holder	Staff, admin
Priority	2
Precondition	Add detail of product
Post condition	None
Extension	System failure
Non -Functional Requirement	Price, quality
Status	Pass

Full Dressed Use Case 3:

Table 5 Fully Dress Use Case 3

USE Case	Bill
Stake Holder	Staff, admin
Priority	3
Precondition	Add detail
Post condition	None
Extension	System failure,
Non -Functional Requirement	Price, quality
Status	Pass

Full Dressed Use Case 4:

Table 6 Fully Dress Use Case 4

USE Case	Purhase
Stake Holder	Staff, admin
Priority	3
Precondition	Add detail
Post condition	None
Extension	System failure,
Non -Functional Requirement	Price, quality
Status	Pass

Full Dressed Use Case 5:

Table 7 Fully Dress Use Case 5

USE Case	Invoice
Stake Holder	Staff, admin
Priority	3
Precondition	Add detail
Post condition	None
Extension	System failure,
Non -Functional Requirement	Price, quality
Status	Pass

Chapter 4

System Design

Chapter 4: System Design

This chapter is about the diagrams that are needed to make the software more easily and more efficiently. There will be different diagrams showing different ways the project is completing.

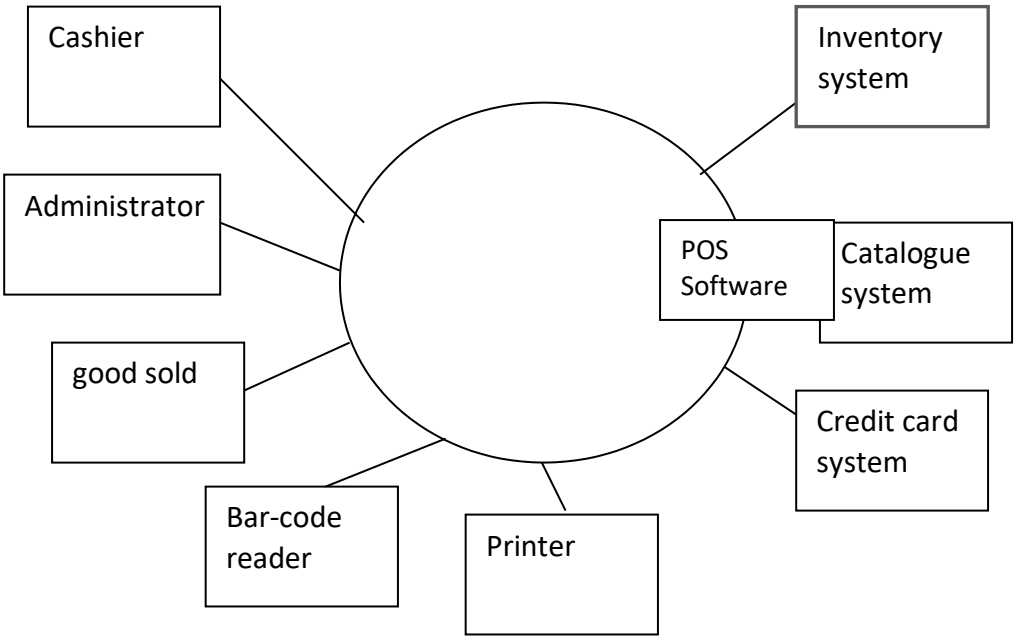


Figure 17 System Diagram

4.1. Architecture Diagram

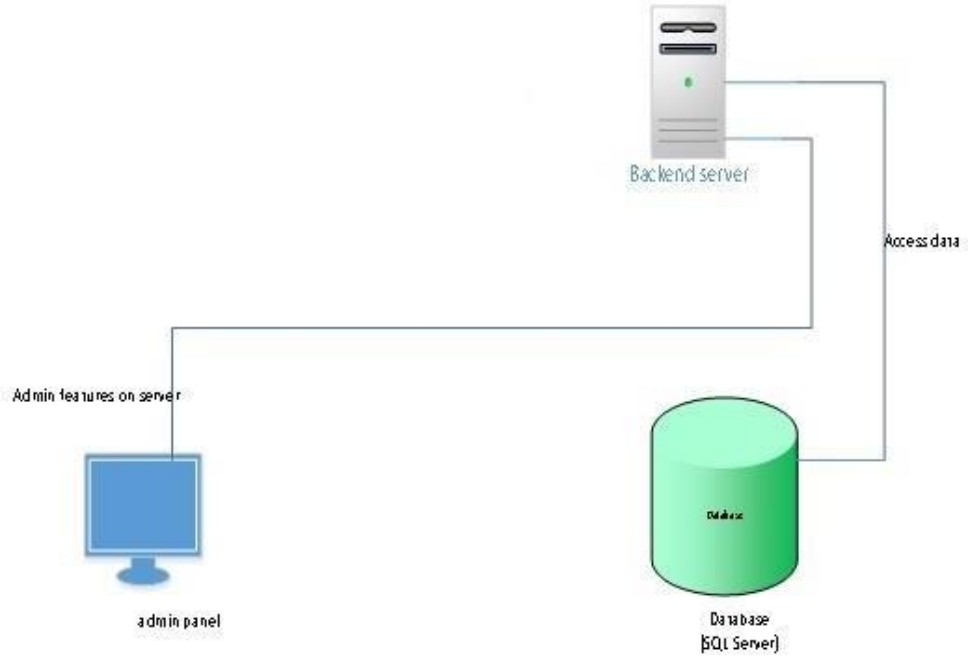


Figure 18 Architecture Diagram

4.2. Domain Model

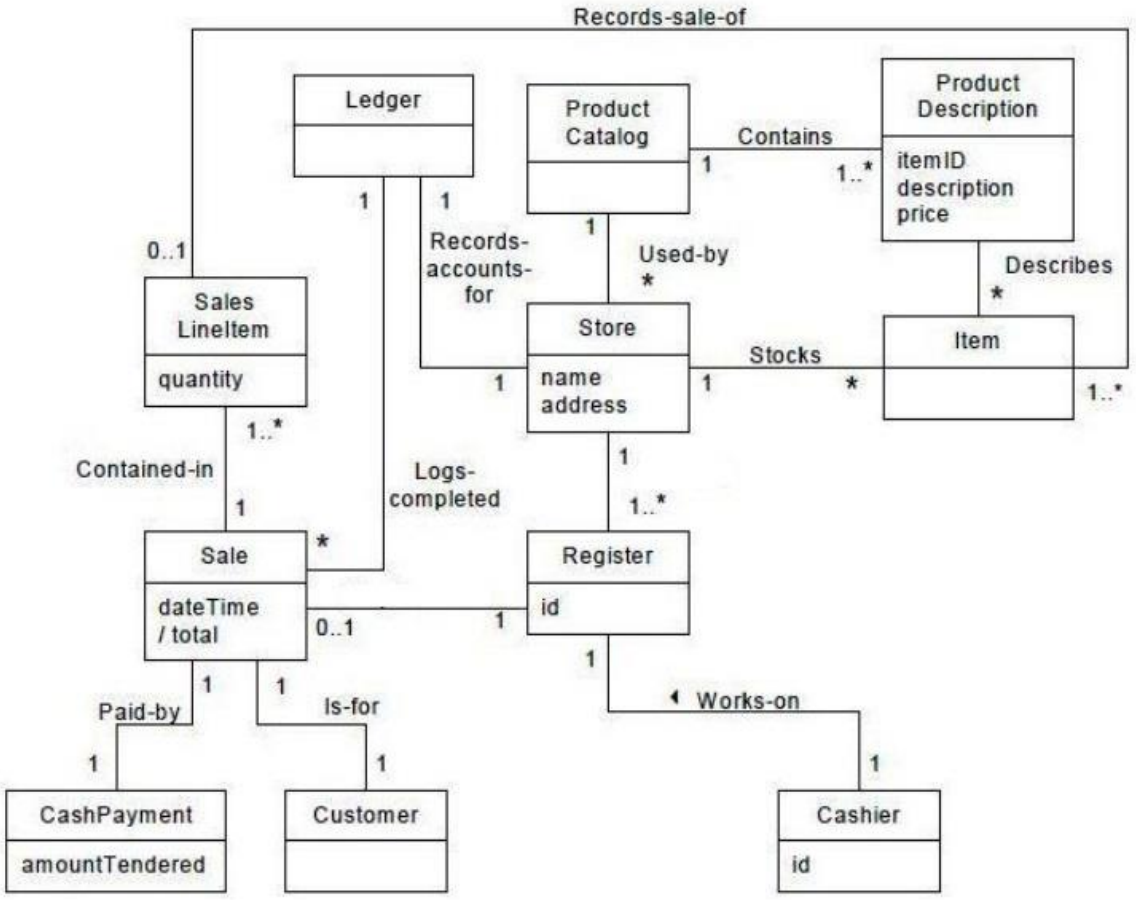


Figure 19 Domain Model

4.3. Entity Relationship Diagram with data dictionary

Table 1 Data Dictionary

Entity	Attributes	Data Type	Primary Key	Foreign Key	Null able
User	Id	Int	yes		
	Name	Varchar			
	Email	Varchar			
	Phone no	Number			
	address	Varchar			
Login	Lid	Int	Yes		
	Name	Varchar			
	password	Varchar			
Stock	SID	Int	yes		
	S type	Varchar			
product	P id	Int	Yes		
	P Name	Varchar			
	P item	Int			
	P cust id	Int			
	P type	Varchar			
Payment	Pay id	Int	yes		
	Pay amt	Float			
	Pay date	Date		Yes	
Sales	Sal id	Int		Yes	
	Sale cust id	Int		yes	
	Sale amt	Float			
	Sale type	Varchar			
Customer	Cust name	Varchar			
	Cust id				
Suplier	Supl name	Varchar			
	Supl id	Int	Yes		

ER Diagram

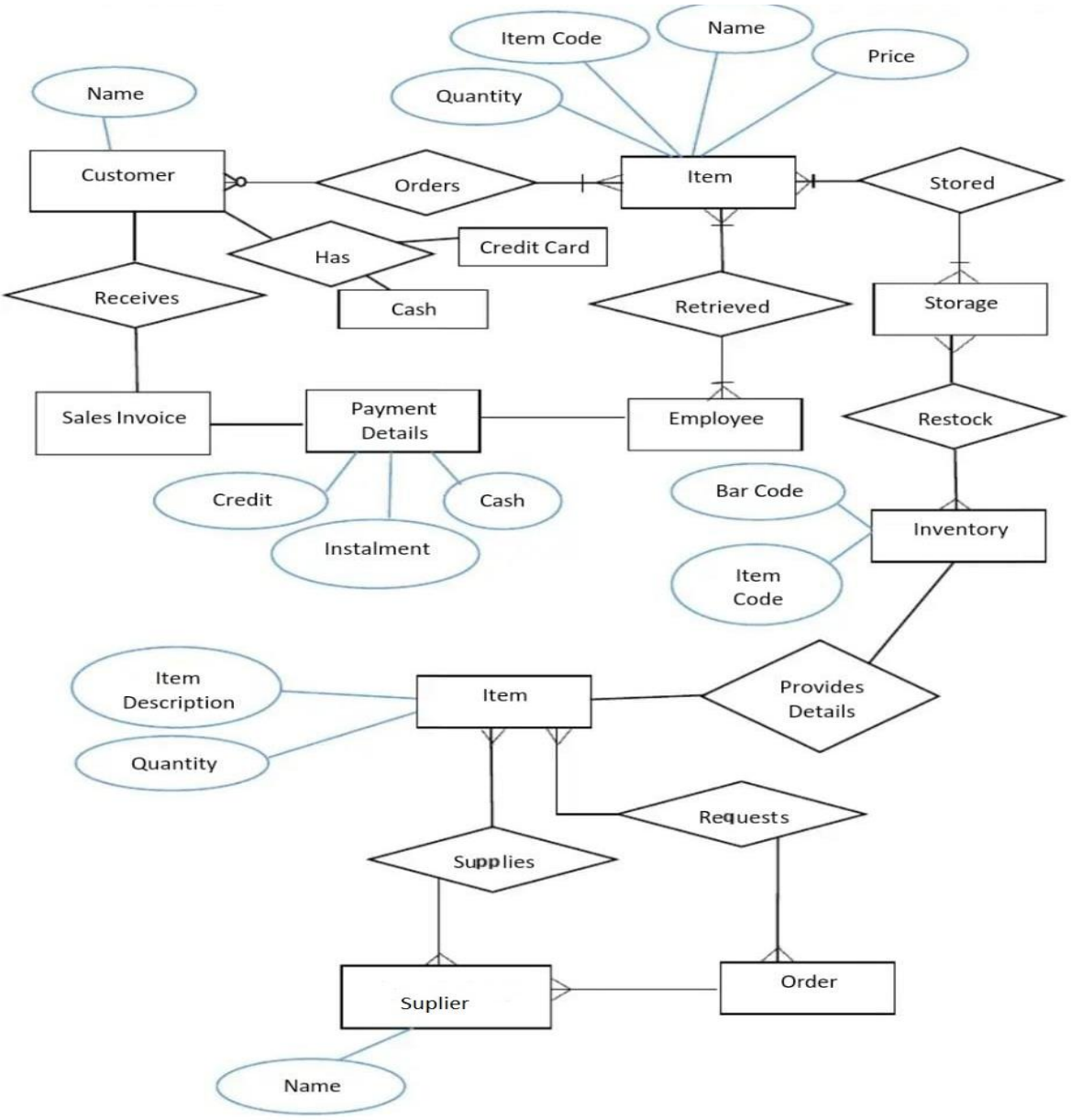
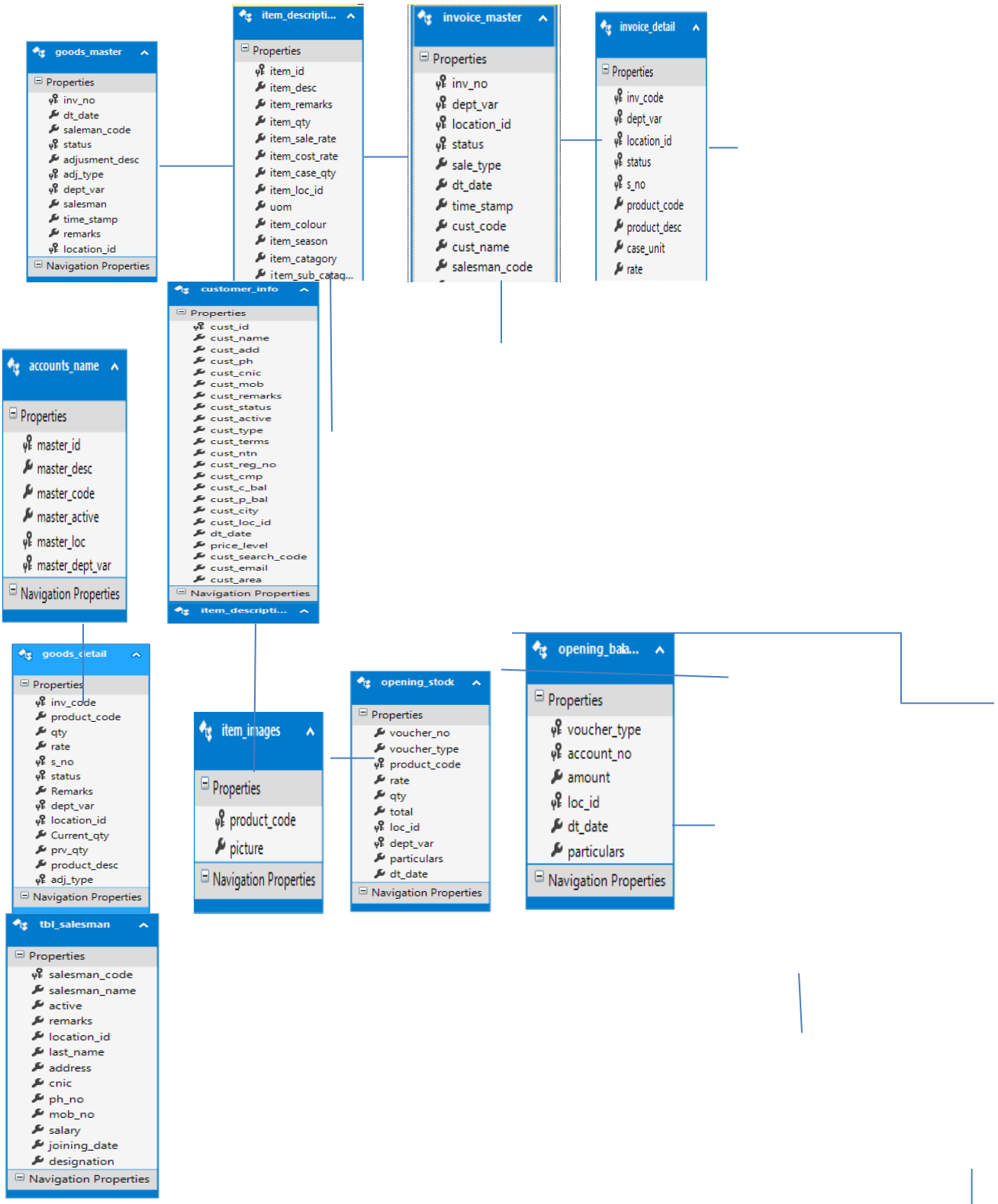


Figure 20 ERD

4.4. Class Diagram



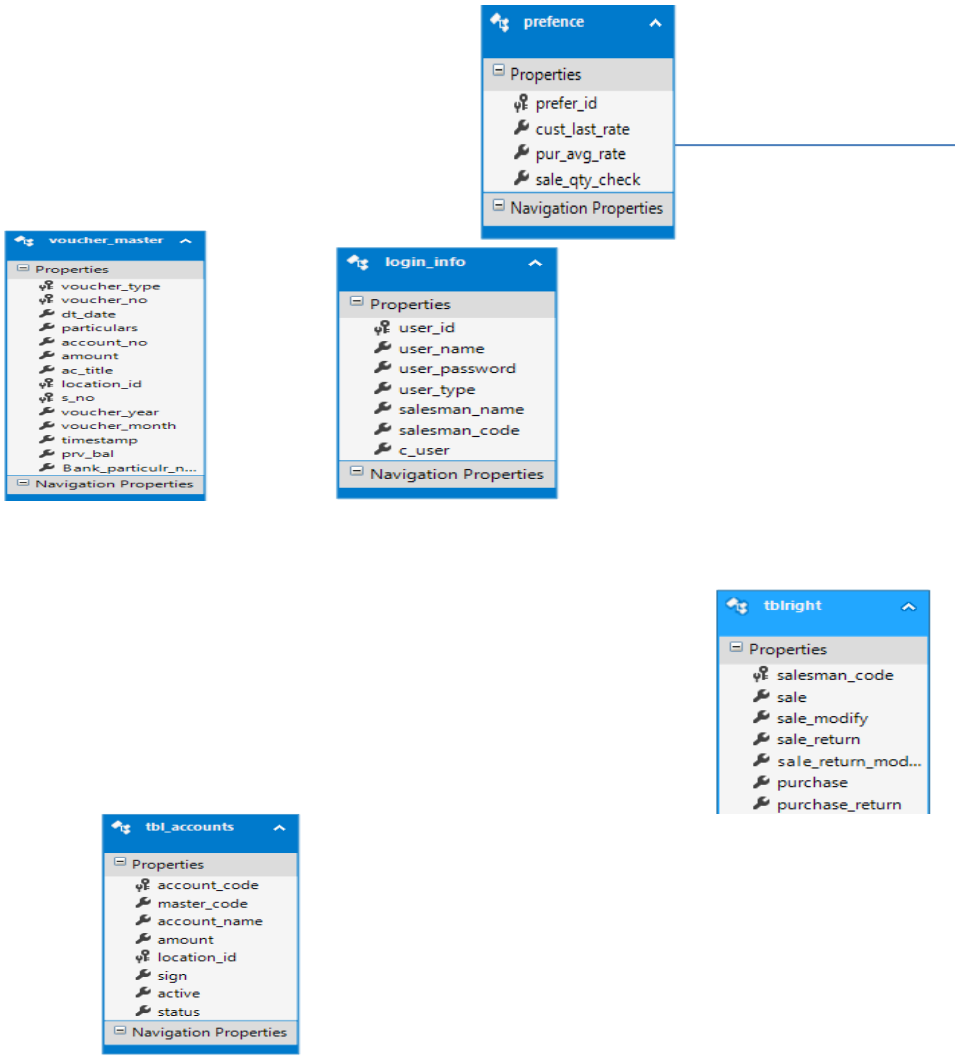


Figure 21 Class Diagram

4.5. Sequence / Collaboration Diagram

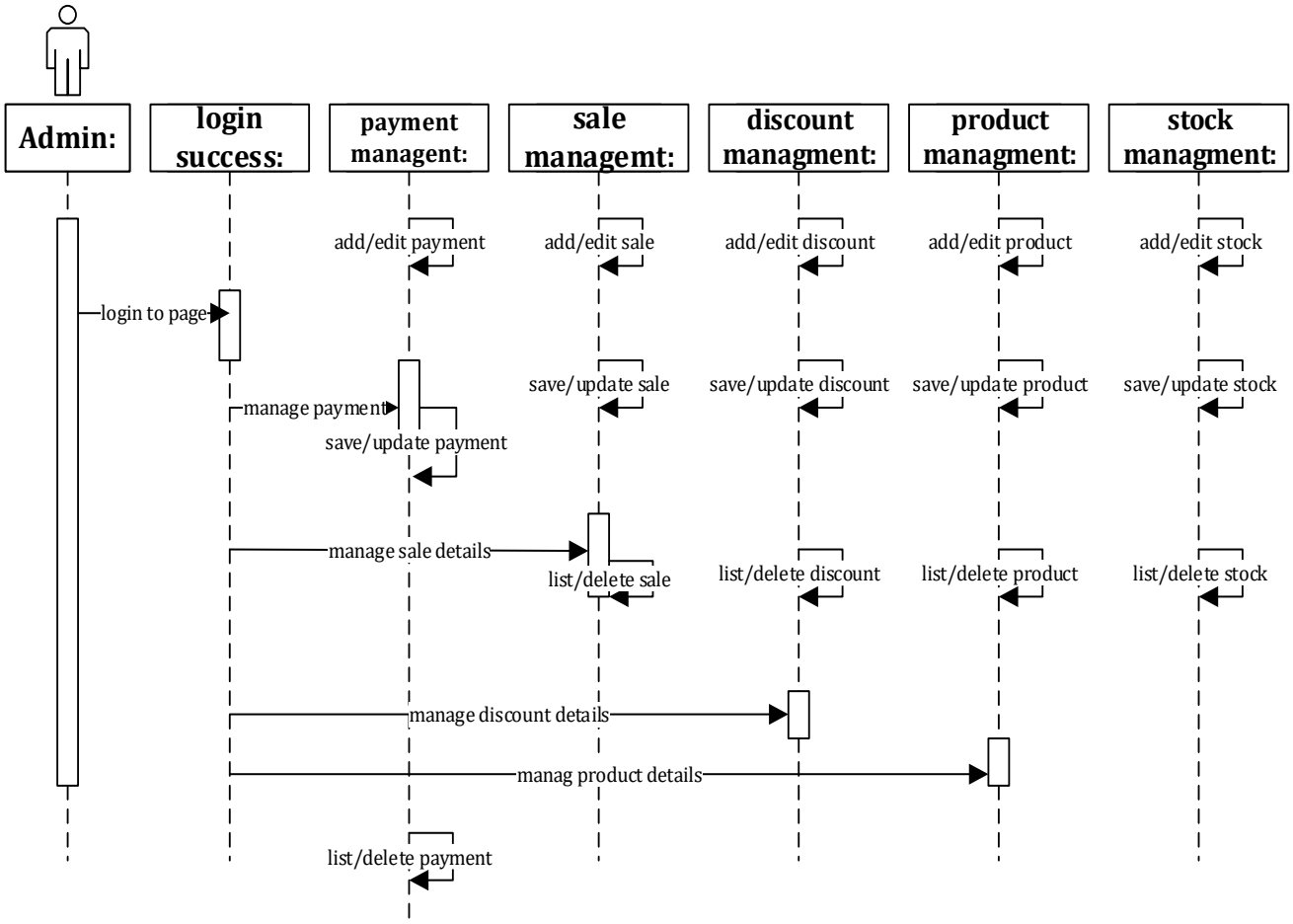


Figure 22 Sequence Diagram

4.5.1. Sequence Diagram Admin Login:

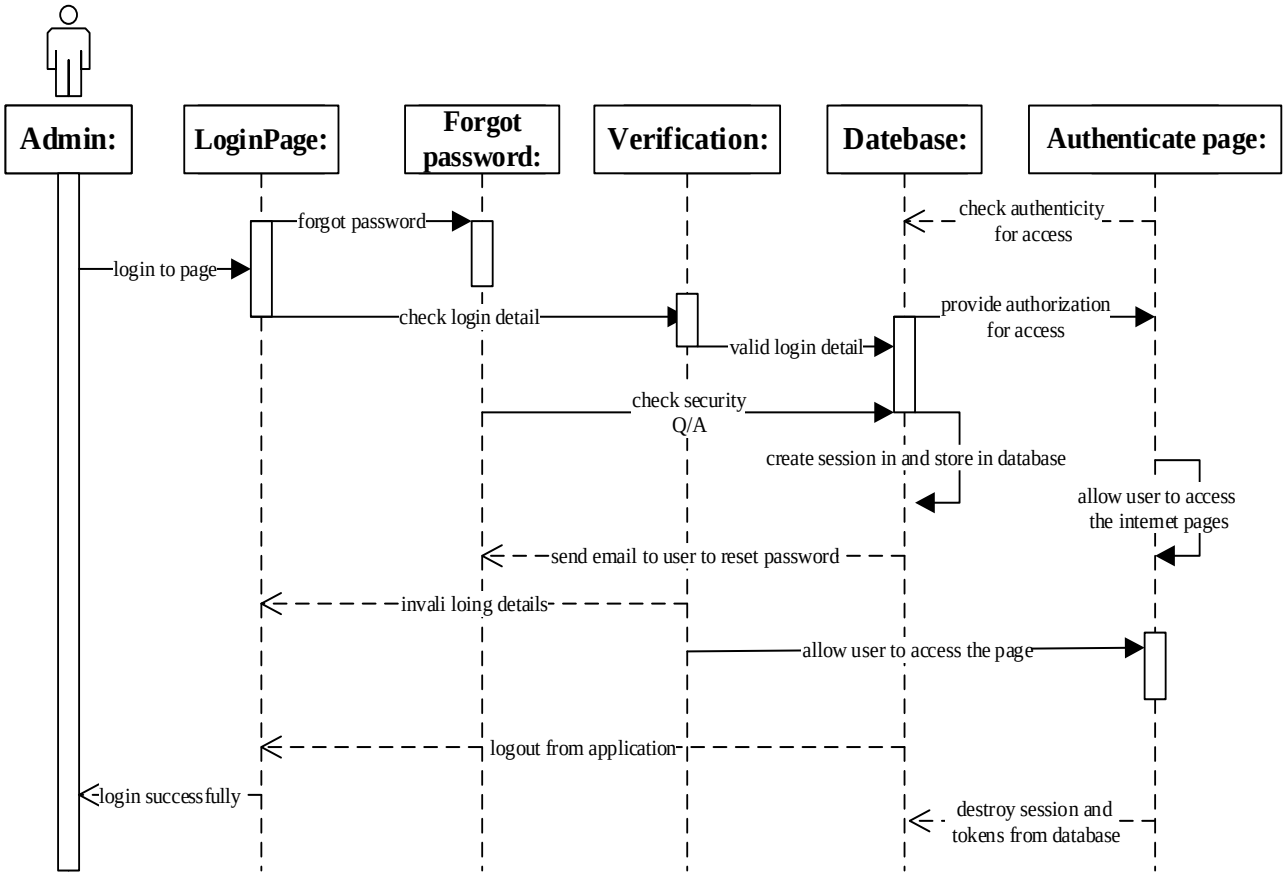


Figure 23 Sequence Diagram Admin Login

4.6. Operation contracts

4.6.1. Operation Contract For Admin Login Operation

4.6.1.1.Operation: Admin login (user-name:var password:var)

4.6.1.2. Cross References:

- Use Case: Login
- Scenario: Process Admin Login

4.6.1.3. Preconditions: System should ask to login when the system starts.

4.6.1.4. Post conditions: Login instance "LIP" has been created for login operation. LIP was

Associated with the admin (association formed).

4.6.2. Operation Contract For Admin Logout Operation

4.6.2.1. Operation: Admin Exit()

4.6.2.2. Cross References:

- Use Case: Exit Admin panel
- Scenario: Process Admin Exit

4.6.2.3. Preconditions: Admin should be logged in.

4.6.2.4. Post conditions: Exit instance has been created for Exit operation.

4.6.3. Operation Contract For User Login Operation

4.6.3.1. Operation: User login (user-name:var password:var)

4.6.3.2. Cross References:

- Use Case: Login User panel
- Scenario: Process User Login

4.6.3.3. Preconditions: System should ask to login when the system starts.

4.6.3.4. Post conditions: Login instance "LIA" has been created for User login operation. LIA was

Associated with the User (association formed).

4.6.4. Operation Contract For Admin Logout Operation

4.6.4.1. Operation: User Exit()

4.6.4.2. Cross References:

- Use Case: Exit User panel.
- Scenario: Process User Exit.

4.6.4.3. Preconditions: Admin should be logged in.

4.6.4.4. Post conditions Login instance "LOA" has been created for User logout operation. LOA

was associated with the admin (association formed).

4.6.5. Operation Contract For Manage User Operation

4.6.5.1. Operation: User()

4.6.5.2. Cross References:

- Use Case: Manage User.
- Scenario: Admin can manage user for access or not.

4.6.5.3. Preconditions: Admin should be authorized login.

4.6.5.4. Post conditions: a user instance "MUO" was created for manage user operation (instance creation).MUO was associated with the Admin (association formed).

4.6.6. Operation Contract For Manage Jewellery Operation

4.6.6.1. Operation: Jewellery(j-id j-no)

4.6.6.2. Cross References:

- Use Case: Manage Jewelries.
- Scenario: Admin will process record of jewelries.

4.6.6.3. Preconditions: Admin should be authorized login.

4.6.6.4. Post conditions: a user instance "MJO" was created for manage jewelries operation (instance creation).MJO was associated with the Admin (association formed).

4.6.7. Operation Contract For Generate Receipt Operation

4.6.7.1. Operation: Receipt(R-Id)

4.6.7.2. Cross References:

- Use Case: Generate Receipt.
- Scenario: User or retailer can generate receipt for customer record or return medicine.

4.6.7.3. Preconditions: Jewelries can be purchased by the customer.

4.6.7.4. Post conditions: a user instance "GR" was created for generate receipt operation (instance creation).GP was associated with the user (association formed).

4.7. Activity Diagram

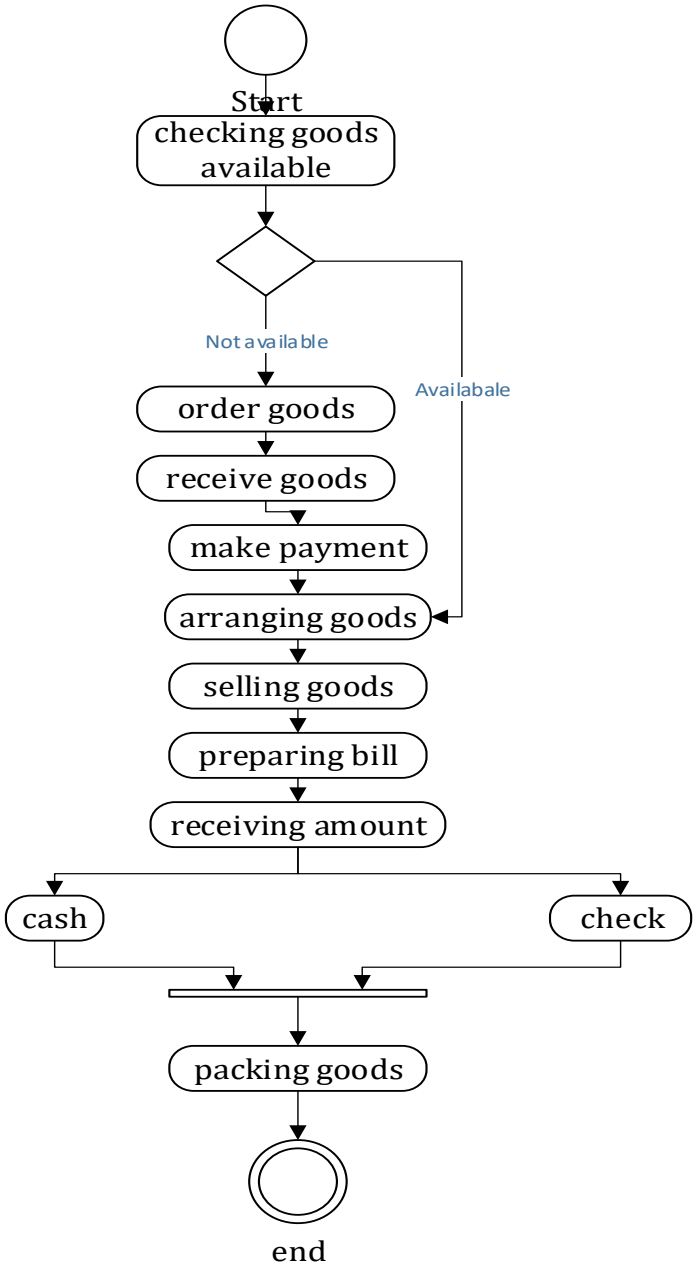


Figure 24 Activity Diagram

4.7.1. Admin login

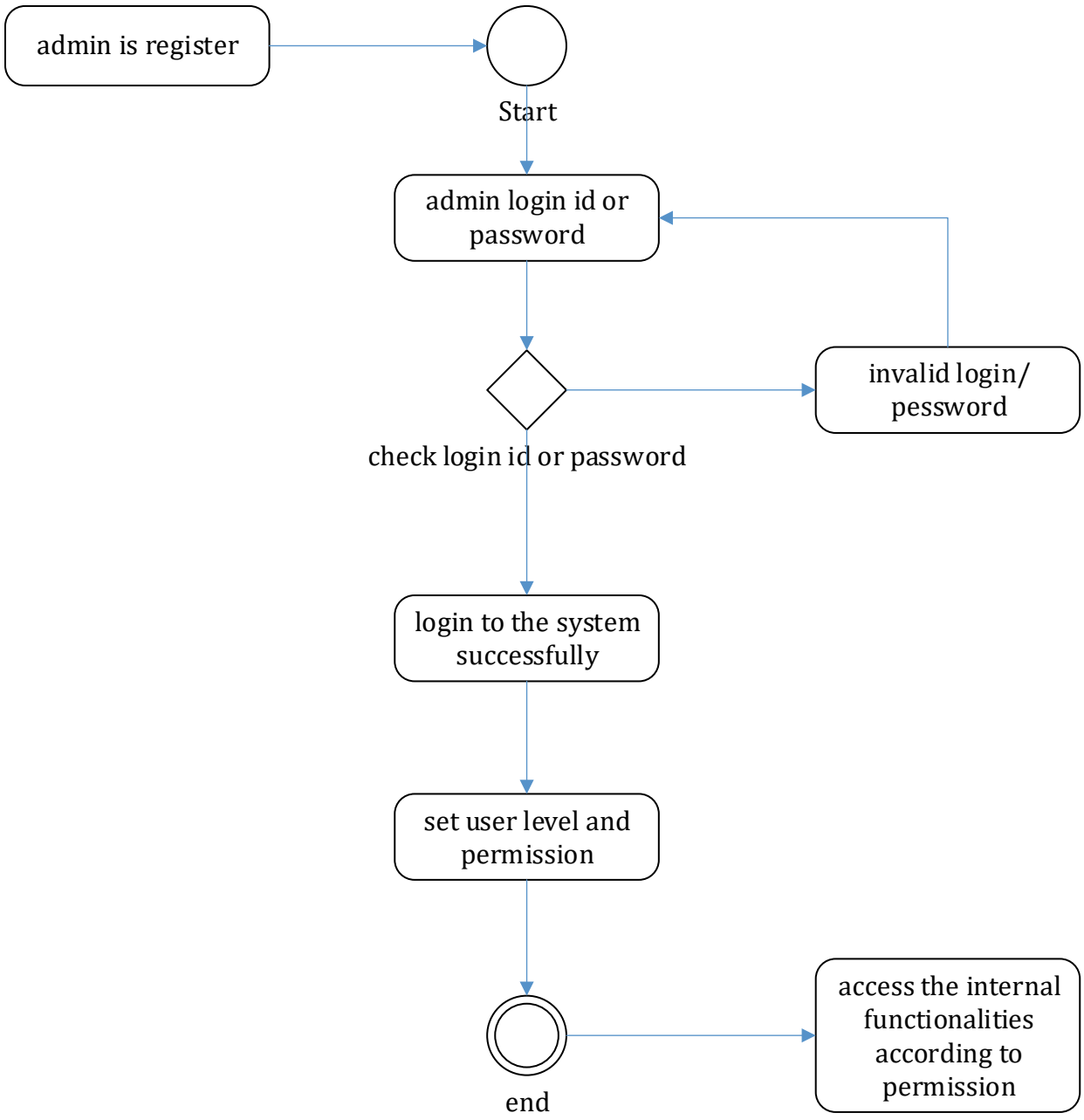


Figure 25 Activity Diagram Admin Login

4.8. State Transition Diagram:

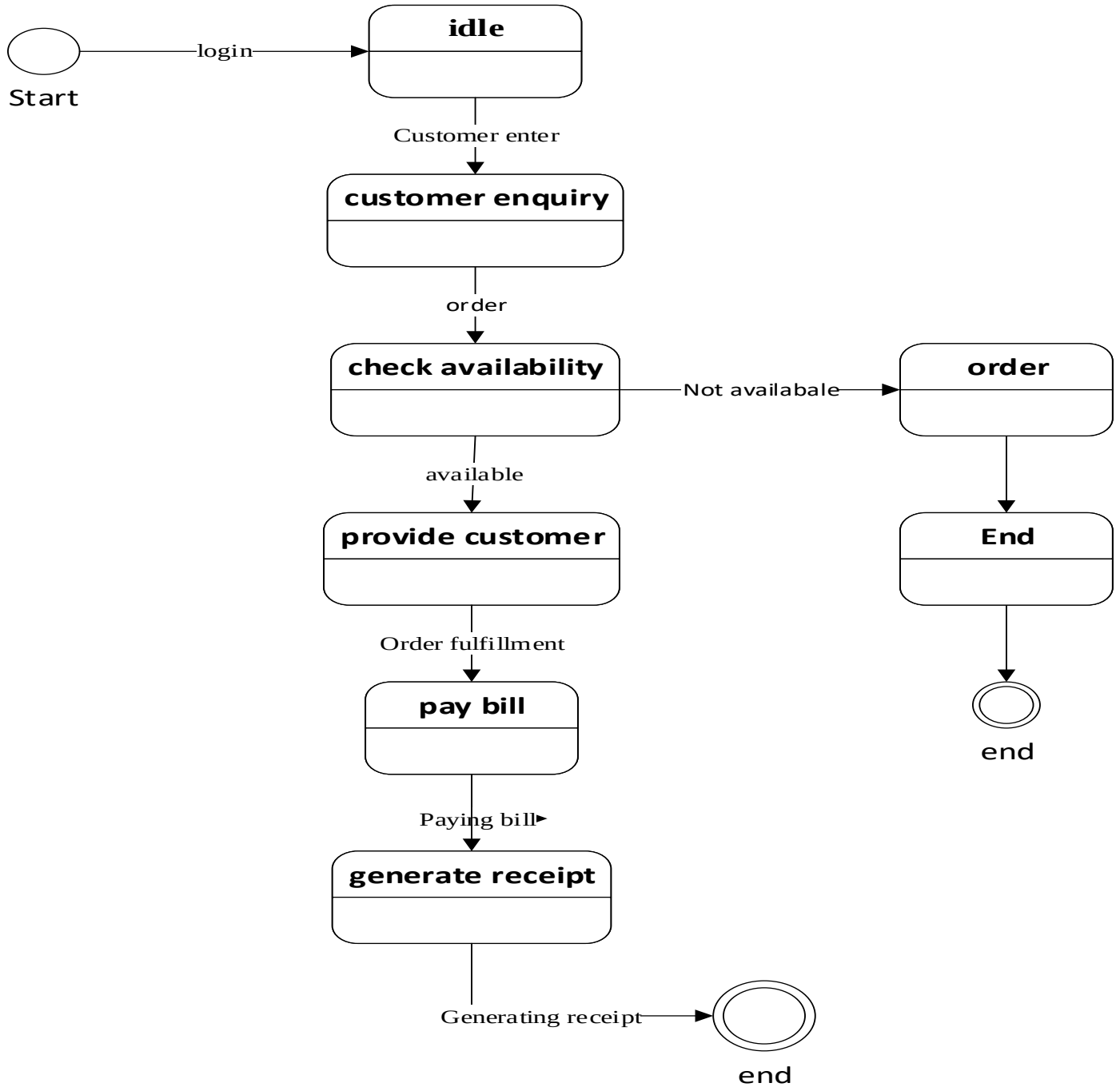


Figure 26 State Transition Diagram

4.9. Component Diagram

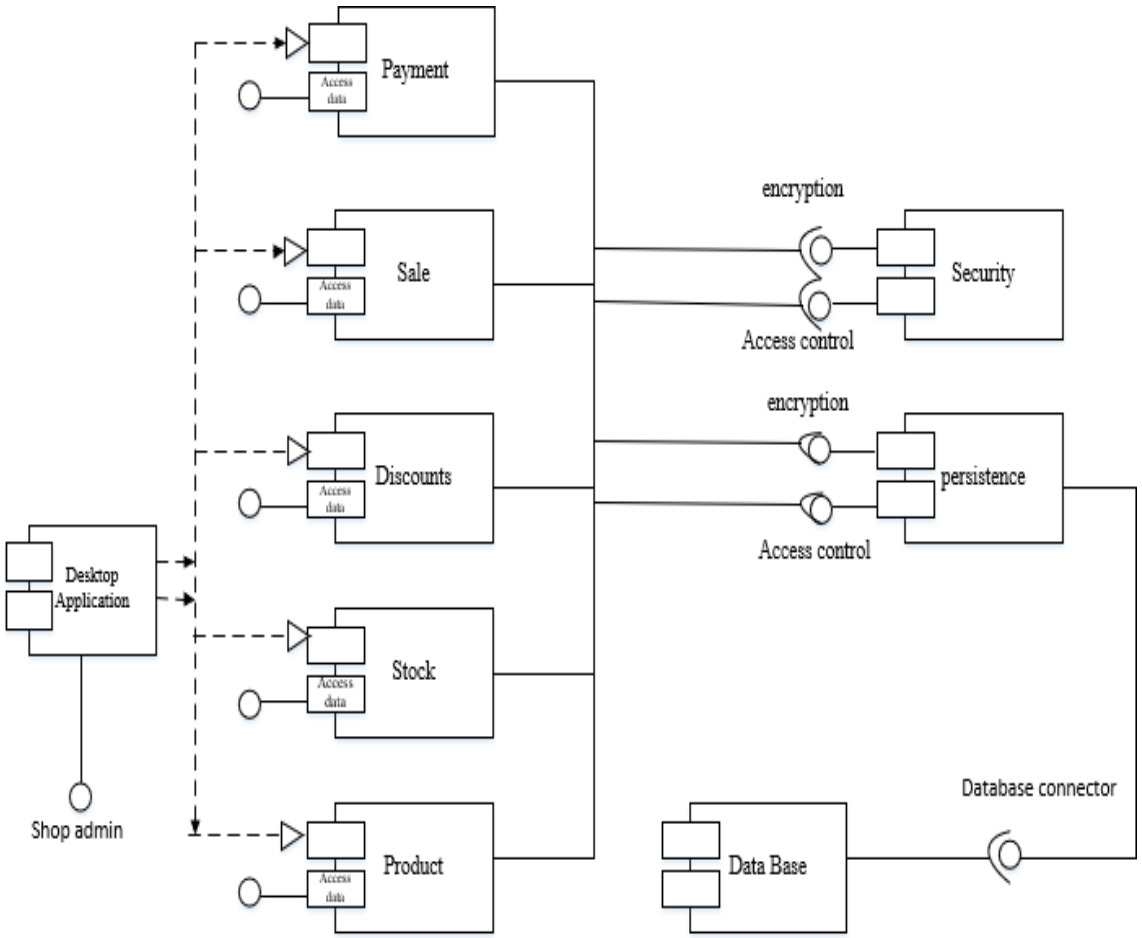


Figure 27 Component Diagram

4.10. Deployment Diagram

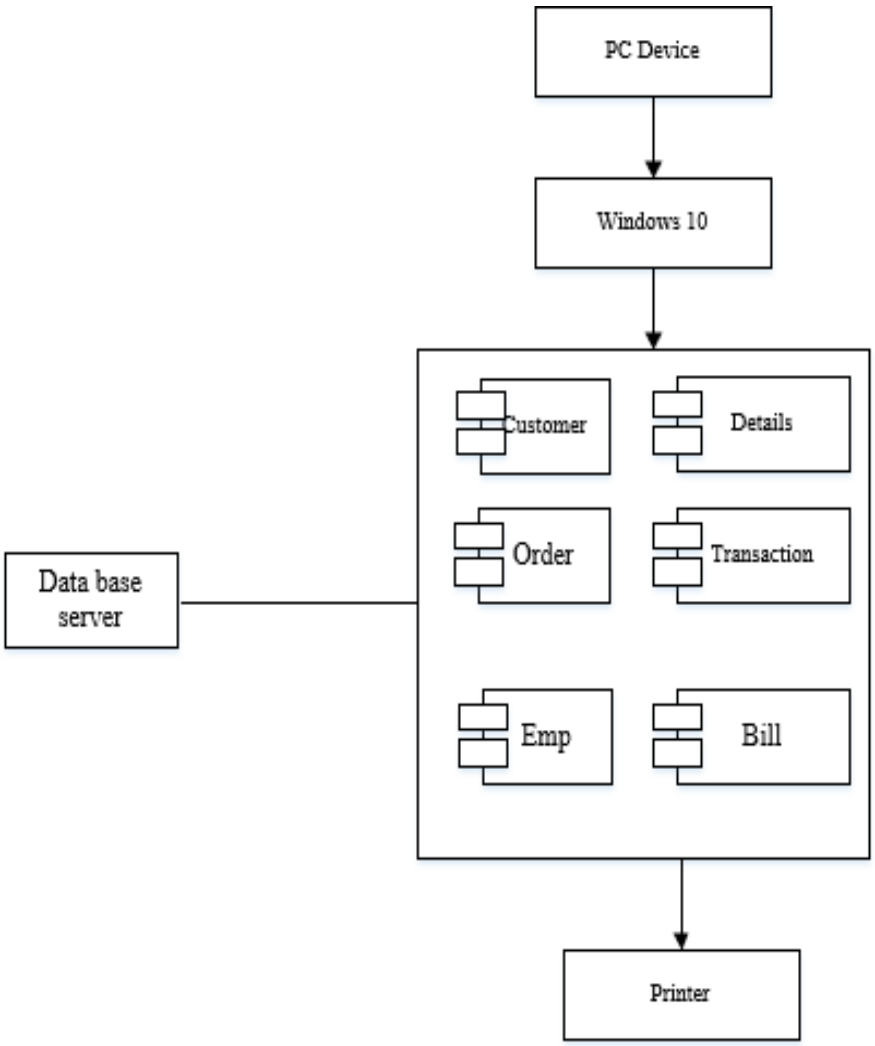


Figure 28 Deployment Diagram

4.11. Data Flow diagram [*only if structured approach is used - Level 0 and 1*]

Zero level data flow diagram

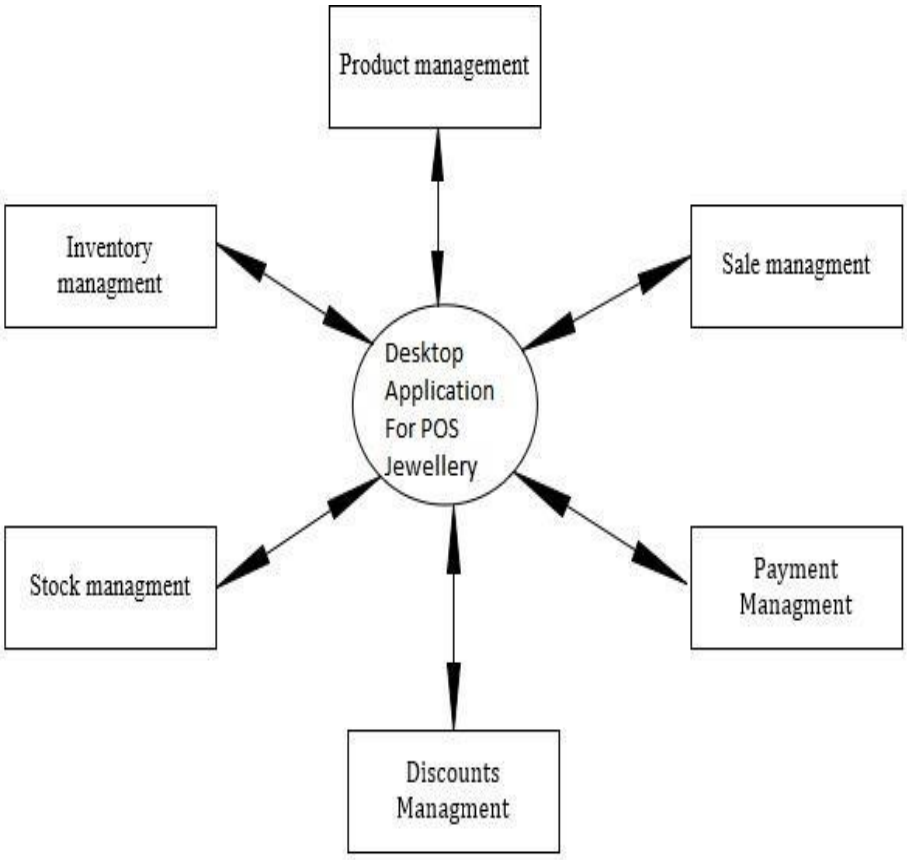


Figure 29 DFD 0 Level Diagram

4.11.1. First Level Data Flow Diagram

Figure 30 DFD Level 1 Diagram

Chapter 5
Implementation

Chapter 5: Implementation

In this chapter, we are going to define in detail the implementation of the system. We shall discuss the flow control, web servers, tools, and techniques used in the making of the system. Also, we shall tell where we are going to deploy the project.

5.1. Important Flow Control/Pseudo codes

Sign in Page

If

user authorize Login successfully

Else

Error message show

Employee Registration

If

Emp. Code, Emp. Name, Last Name, mobile no. CNIC, Address, Joining Date, and designation

Click save button

New Login Assign

Fill username , password and assign employee whose validate

On closing backup generate in local drive

5.2. Components, Libraries, Web Services and stubs

5.2.1. Components:

1. Login
2. Registration
3. Reset password
4. Order different Jewellery accessories,
5. Detail of product,
6. Purchase detail,
7. Sale detail,
8. Expense detail,
9. Stoke details.

5.2.2. Web Server:

- Local-host

5.3. Deployment Environment

In our project we use Local-host Deployment Environment for back-end and front-end.

5.4. Tools and Techniques

The technological requirements include the front-end programming tool and the back-end database system.

- Coding language: C#, .NET frameworks for WinForm Applications
- For front end we are using C# and .NET frameworks for WinForm Applications
- SQL for writing back-end
- Operating System: Window

5.5. Best Practices / Coding Standards

- Write readable code
- Write efficient code.
- Use helper methods.
- If avoidable, don't hard-code
- Confirm to the coding standards of your current project.

- Pair programming/code review
- Backup and save work often

5.6. Version Control

We will launch version 1 and after our customer review, we will test our product again and will make some changes to it, and will then release version 2 if needed. We will keep updating our software from time to time.

Chapter 6
Testing and Evaluation

Chapter 6: Testing and Evaluation

Testing is a process, which reveals errors in the program. To make sure the unavailability of any error or bug, we run a simulation test on the system. Different cases and different types of testing will run to check the system is running well or not. If there is any error during the testing, then provide a solution to outcomes the errors. In this Chapter we provide the whole System testing and evaluation that how to use the system and what work has been done by which feature.

It contains the results of tests, which were executed during the testing phases.

6.1. Use Case Testing

6.1.1. Login page testing (admin)

Table 7 Admin Testing

Test Case ID	TC001
Test Case Summary	When admin click on login button it must be login and show the Dashboard page.
Prerequisites	User must be on login page and have username and password
Test Procedure	Verify the user Email and password.
Actual Result	User successful login.
Status	Successful.
Add user/ admin	Add a new user to the database to allow him to use services of system
Remove user	Remove a user to the database
Payment proceeding	Manage payments from customers.
Test Steps	1. Navigate the login page 2. Enter the username 3. Enter the password 4. And click on Log in Button 5. Add new user or admin 6. Manage payment 7. Logout

6.1.2. Sign up testing

Table 8 Sing up testing

Test Case ID	TC002

Test Case Summary	User must be sign up after filling form.
Prerequisites	User must be on sign up page.
Test Procedure	Verify the inputs given in form.
Expected Result	User must be allotted code and username, password and employee Dashboard.
Actual Result	User successful sign Up
Status	Successful.
Test Steps	1. Navigate the sign-up page. 2. Fill up the form. 3. Select User Role (Add, Search, modify) 4. And click on Sign up button

6.1.3. Extensions (Errors)

Table 9 Extensions Error

No.	Step	Description
1-a	Login failed	Login failed if admin enter wrong username System will display an error message
1-b	Login failed	Login failed if admin enter wrong password. System will display an error message
1-c	Login failed	Login failed if admin enter wrong code. System will display an error message.
1-d	Login failed	Login failed if admin enter wrong employ. System will display an error message.
2	Registration failed	Registration will have failed if a new user doesn't follow the rules System will display an error message
3-a	Login failed	Login failed if Customer enter wrong username System will display an error message
3-b	Login failed	Login failed if Customer enter wrong password. System will display an error message.

6.2. Boundary value analysis

Boundary testing in the process of testing that is between start point and extreme ends or boundaries between partials of input values.

6.2.1. Login username

- Username that a person enter in the text box at the time of login must be correct.
- If the username is not correct, the system must restrict the user at the current page.

6.2.2. Login password

- Username that a person enter in the text box at the time of login must be correct.
- If the username is not correct, the system must restrict the user at the current page.

6.2.3. Text fields:

- All text fields system has must accept only correct text
- User must add that information which is needed and to the point
- Text fields are name, city, country etc.

6.2.4. Numeric fields:

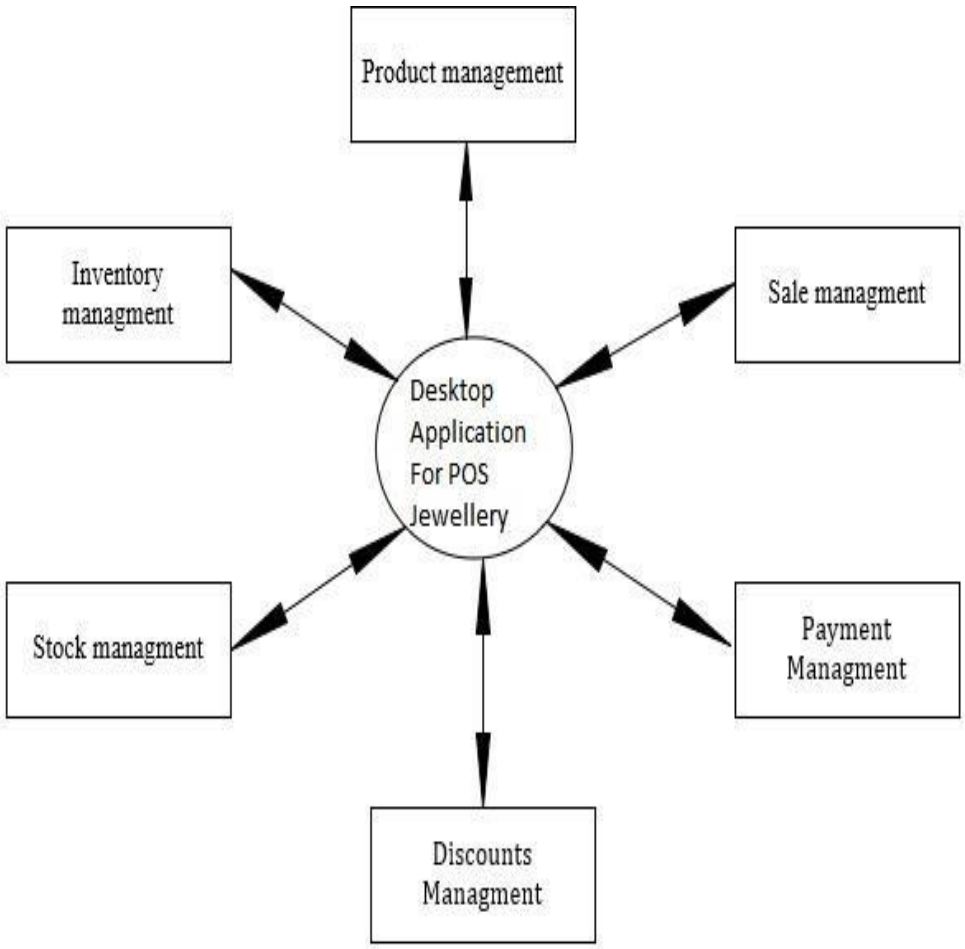
- The numeric fields must accept only numbers
- Numeric fields are mobile number, CNIC number, quantity of products etc.

6.2.5. Email fields:

- Email fields are also available in the software system
- Email fields must accept email formats.

6.3. Data flow testing:

Data flow testing refers to how database react to something you have in your database so as you know my project is POS Retail Jewellery Software and the database is connected for saving the records about jewellery available. Database is not just used to store data it also used to store a large amount of previous records which you have been in a collecting from some time like the jewellery that you have in your stock and also the jewellery which is going out of the stock. Here database is not only used to store data it also used to give you an alert when something is going out of the stock so we use database and the data flow testing is it due to that database working properly and like you see that the storage of jewellery and alerting of system is ok for our data flow testing is positive.



6.4. Unit testing

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

Point 01:

In unit testing we check all button working correctly or not. Login button working correctly

New Login Assign button working correctly Login list button working correctly

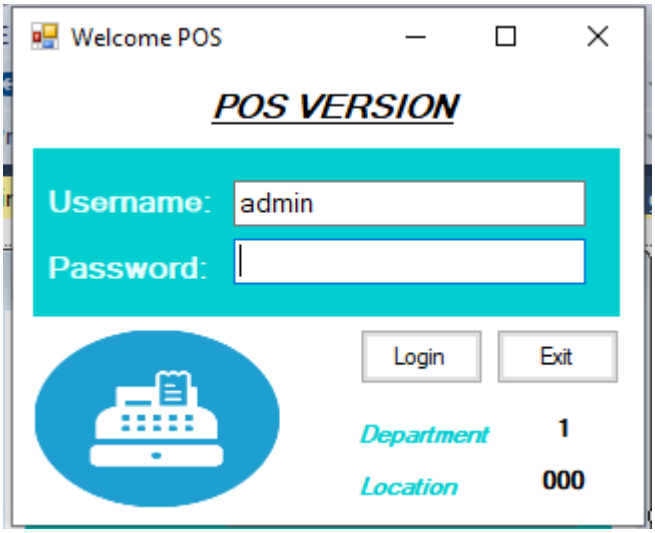


Figure 31 Login

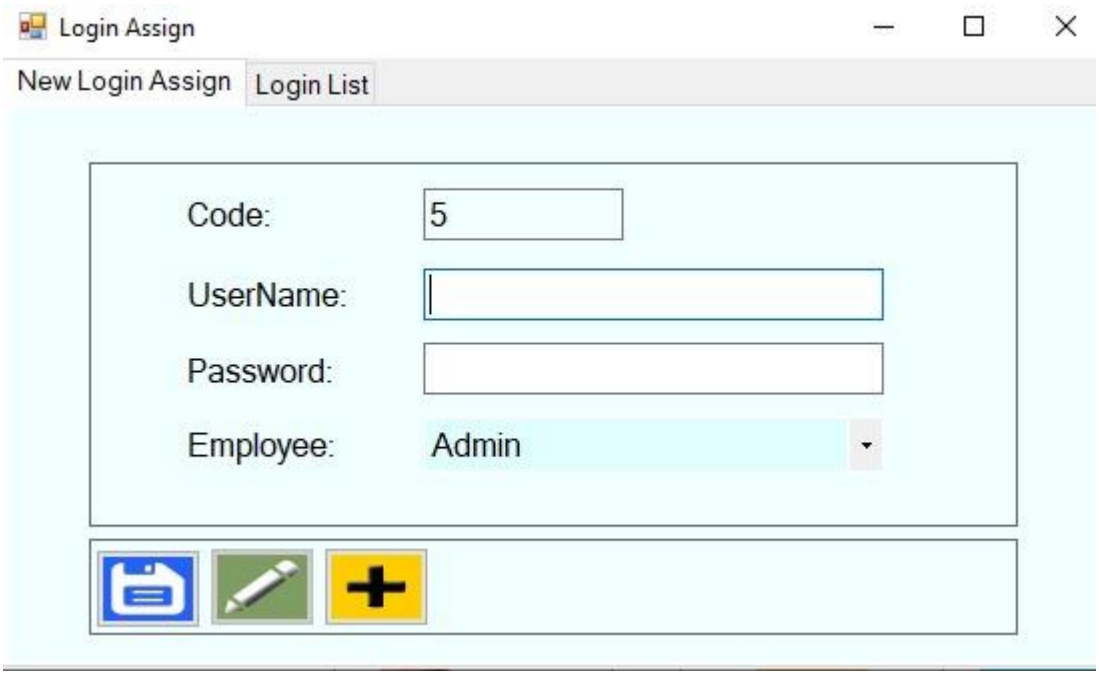


Figure 32 New Login Assign

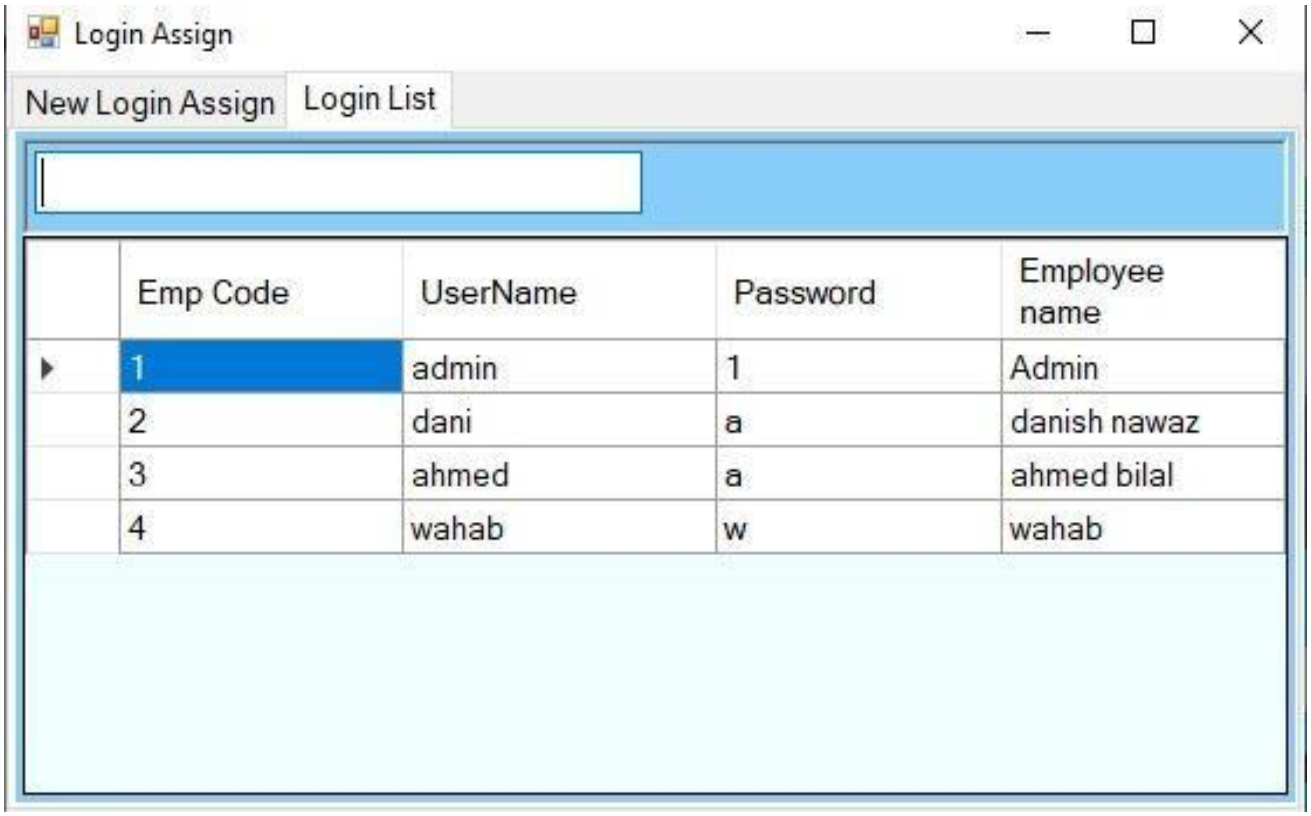


Figure 33 Login List

Point 02:

Add new items button working correctly Create new customer button working correctly Salary voucher button working correctly Product list button working correctly

Bar-code or search bar button working correctly Updating of employ right button working correctly Sales button working correctly

Add detail button working correctly Apply button working correctly Search detail button working correctly

Update employ rights button working correctly Expense Detail button working correctly Purchase detail button working correctly

Sale button working correctly

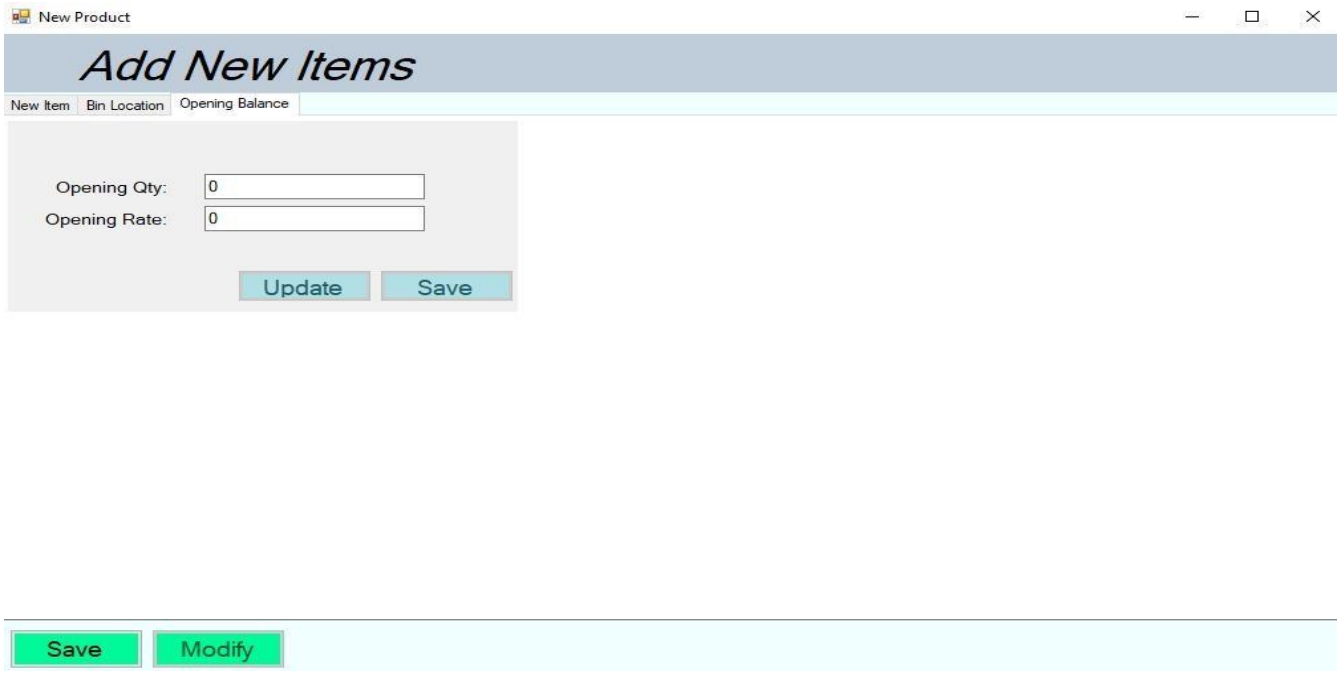


Figure 34 Add New Products

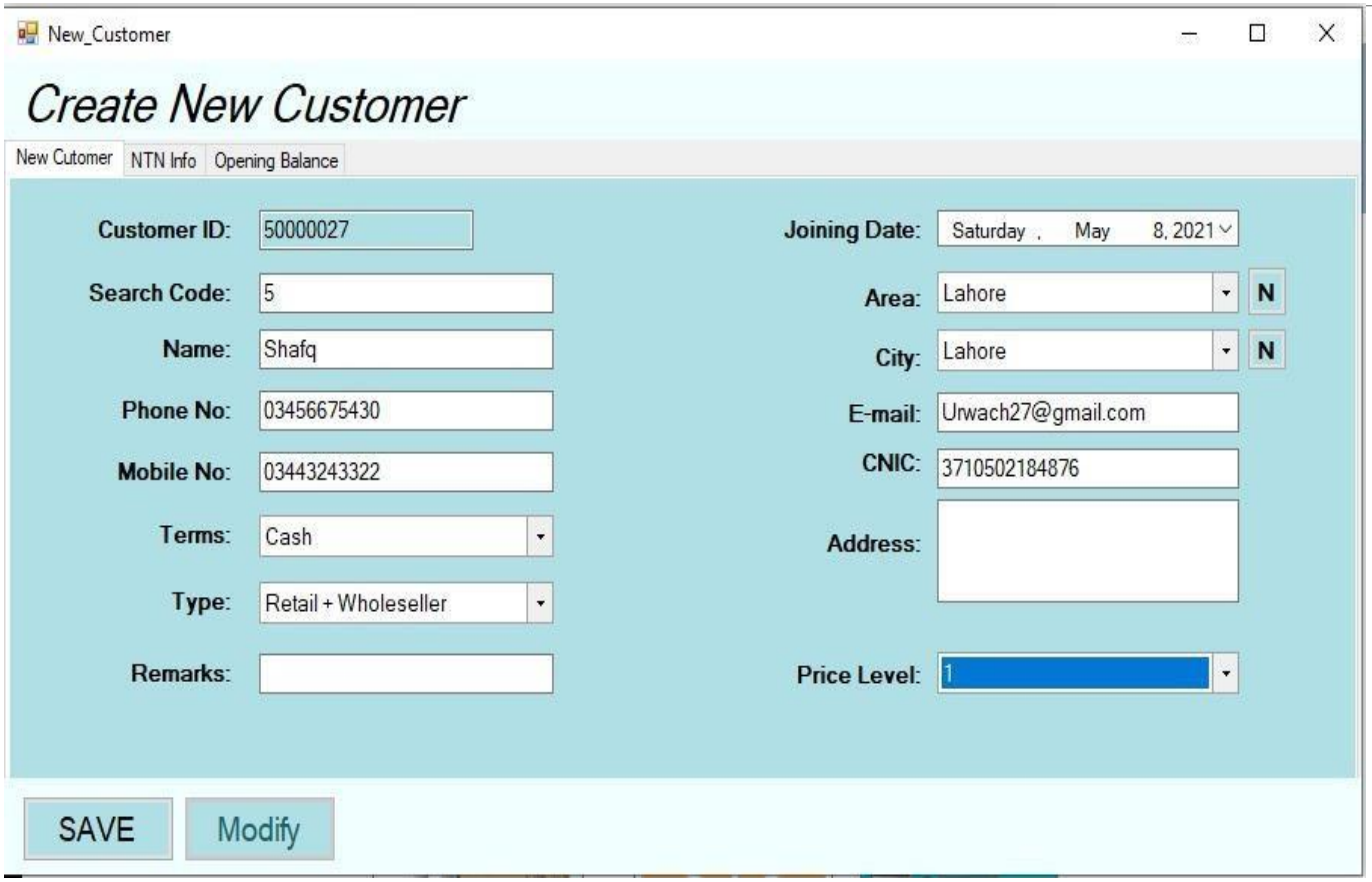


Figure 35 Create New Customer

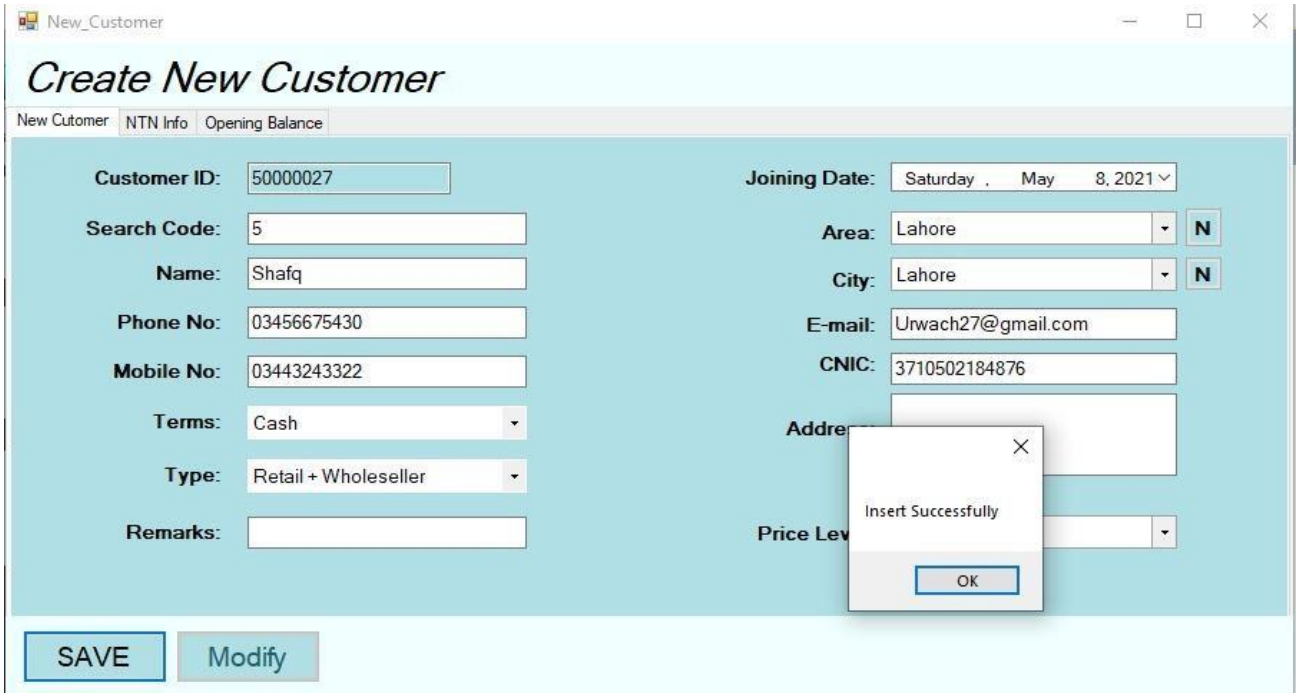


Figure 36 Add New Customer Successfully

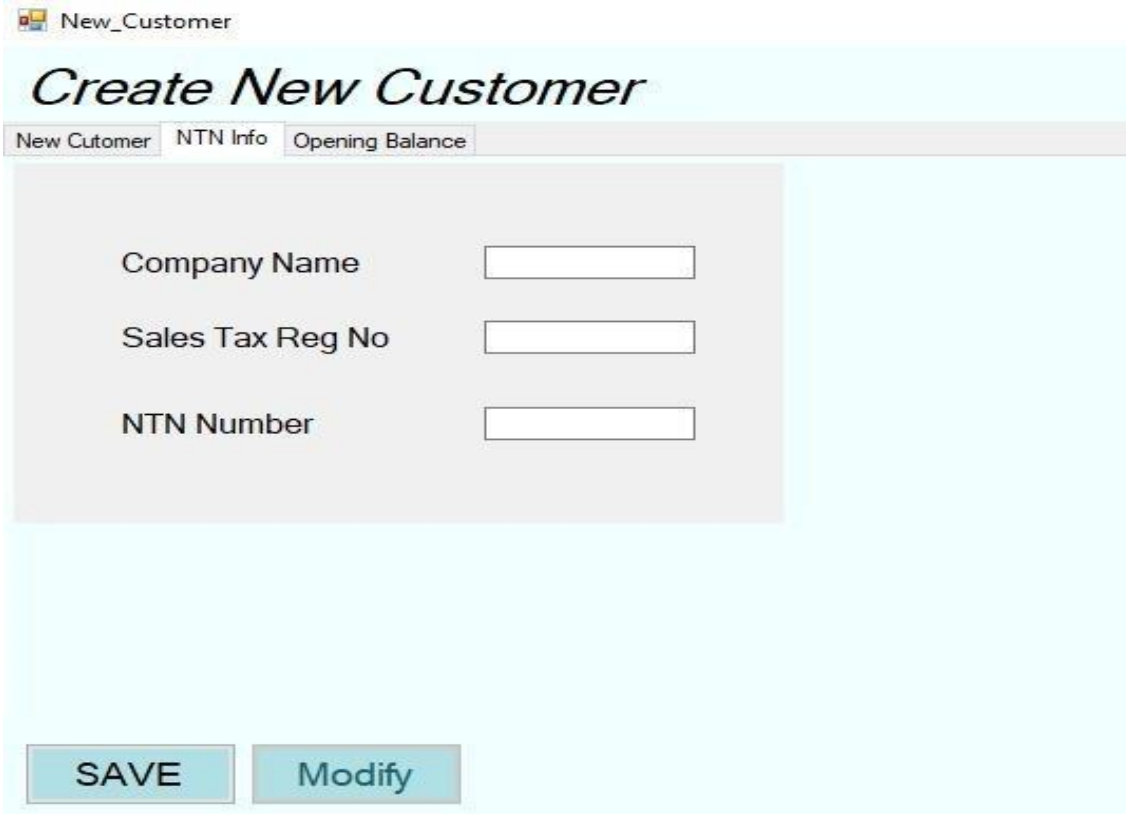


Figure 37 Enter NTN Number

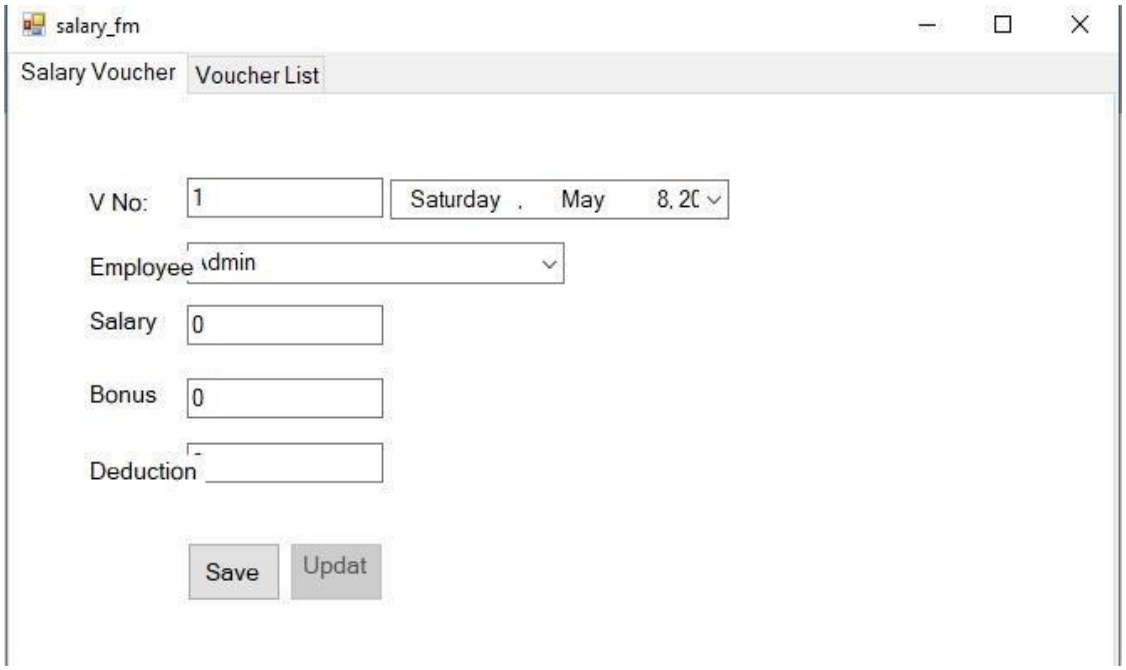


Figure 38 Salary Vouche

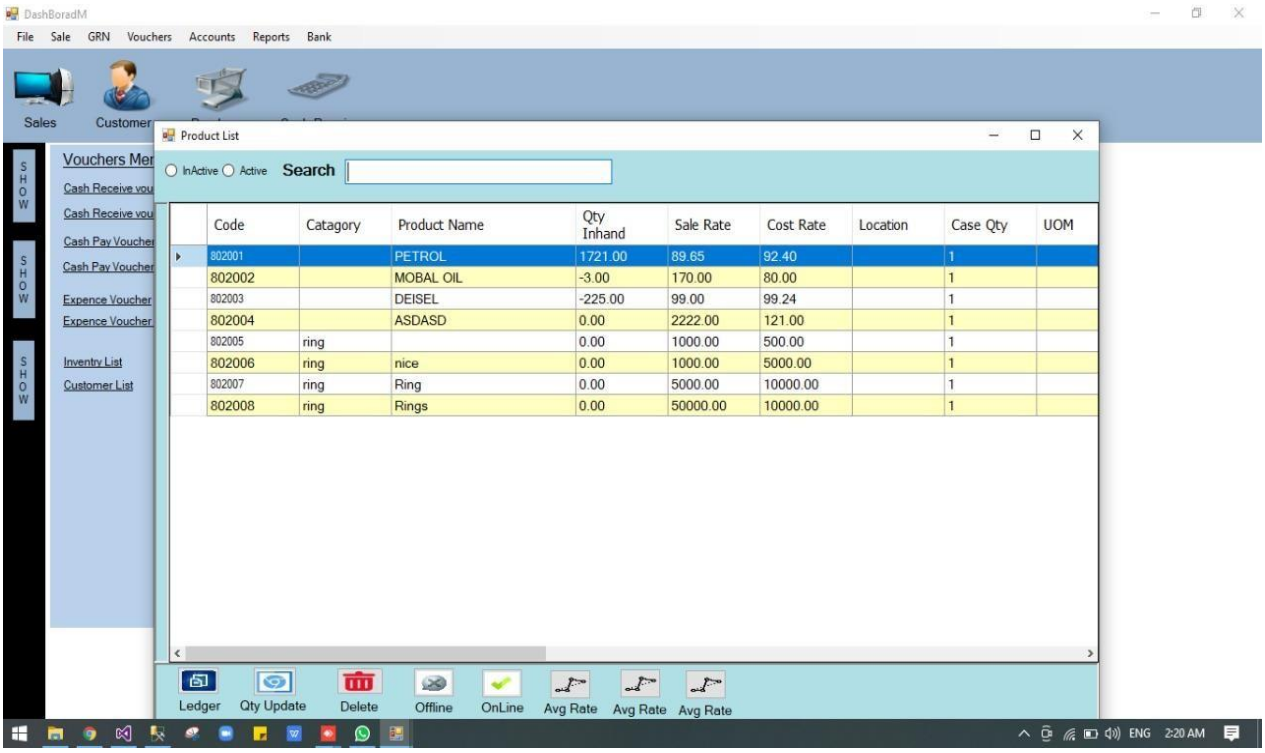


Figure 39 Product List

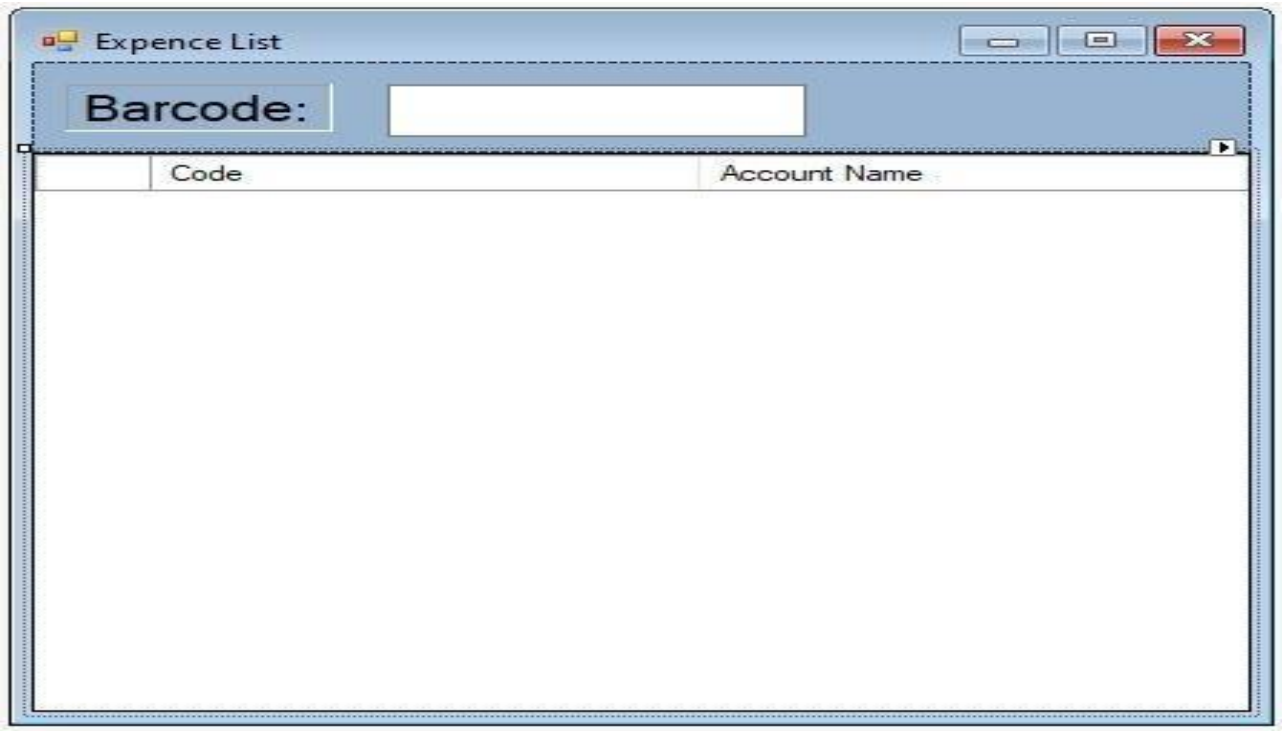


Figure 40 Bar-cod

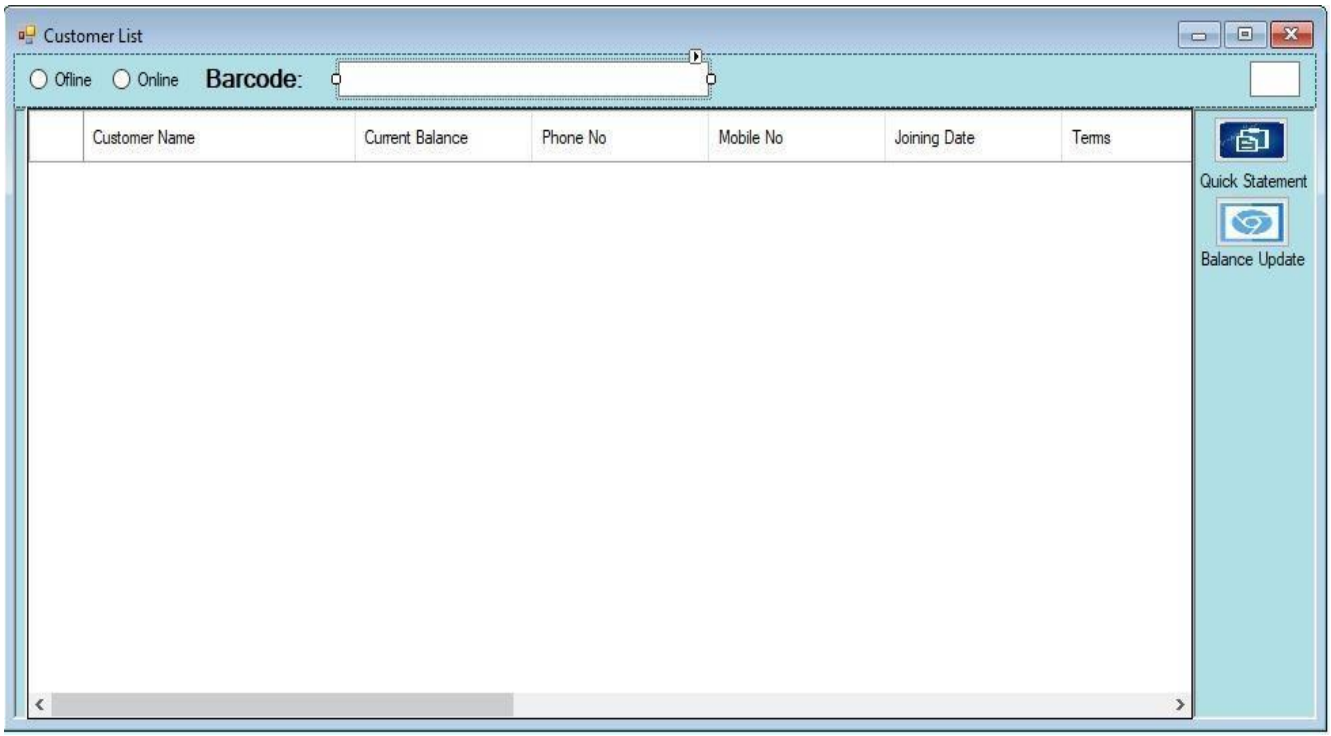


Figure 41 Search Bar-code

Figure 42 User Rights

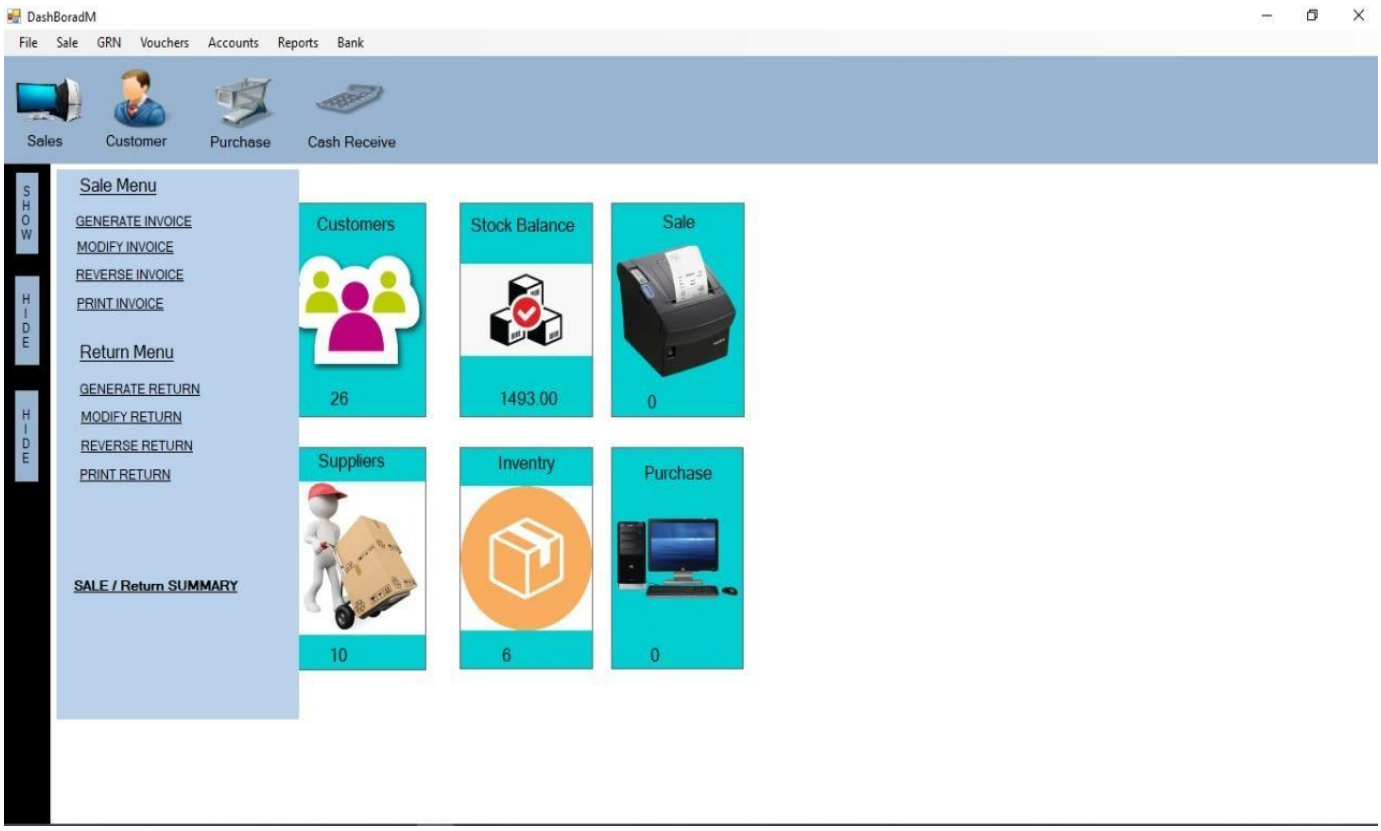


Figure 43 Sales Menu

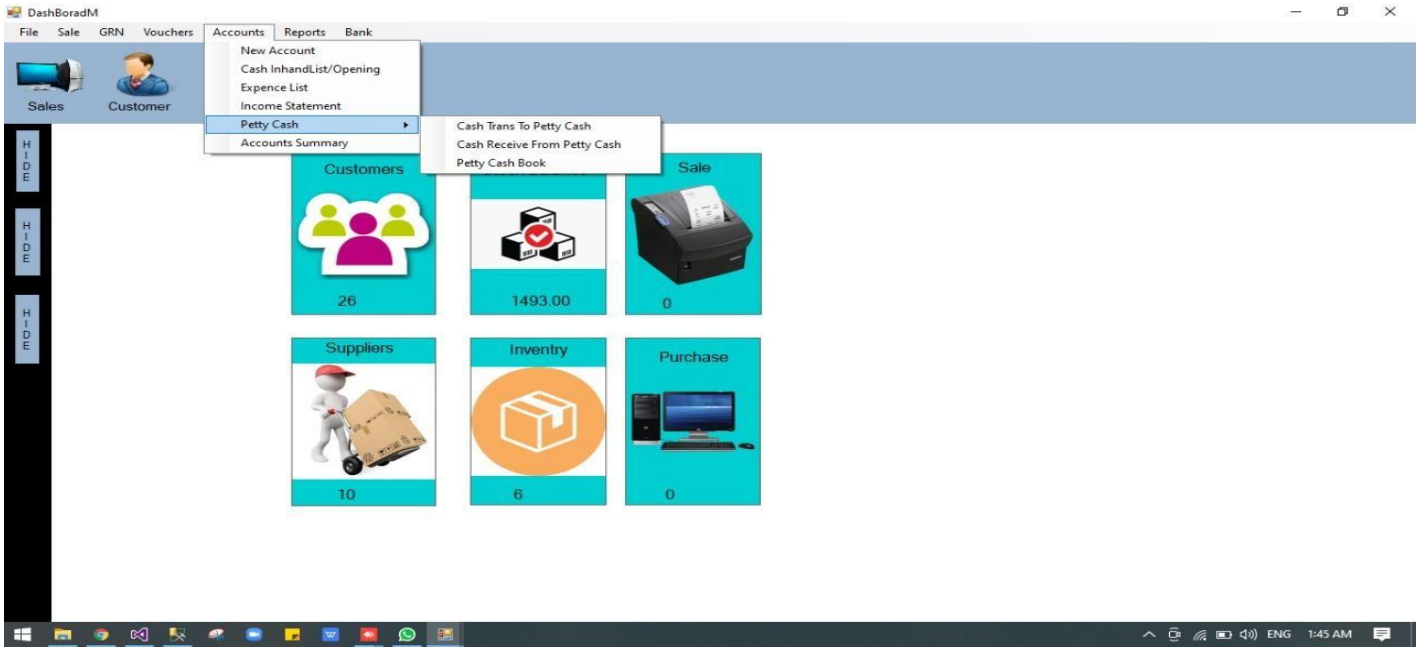


Figure 44 Accounts Menu

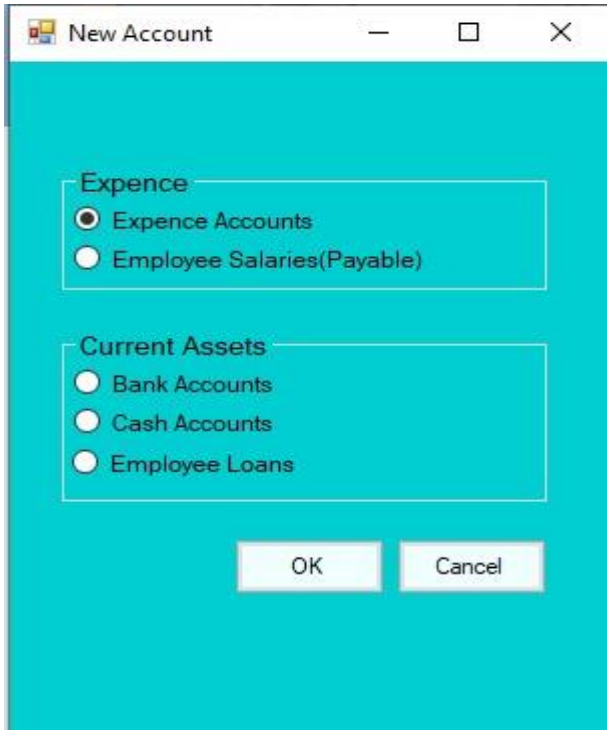


Figure 45 New Account

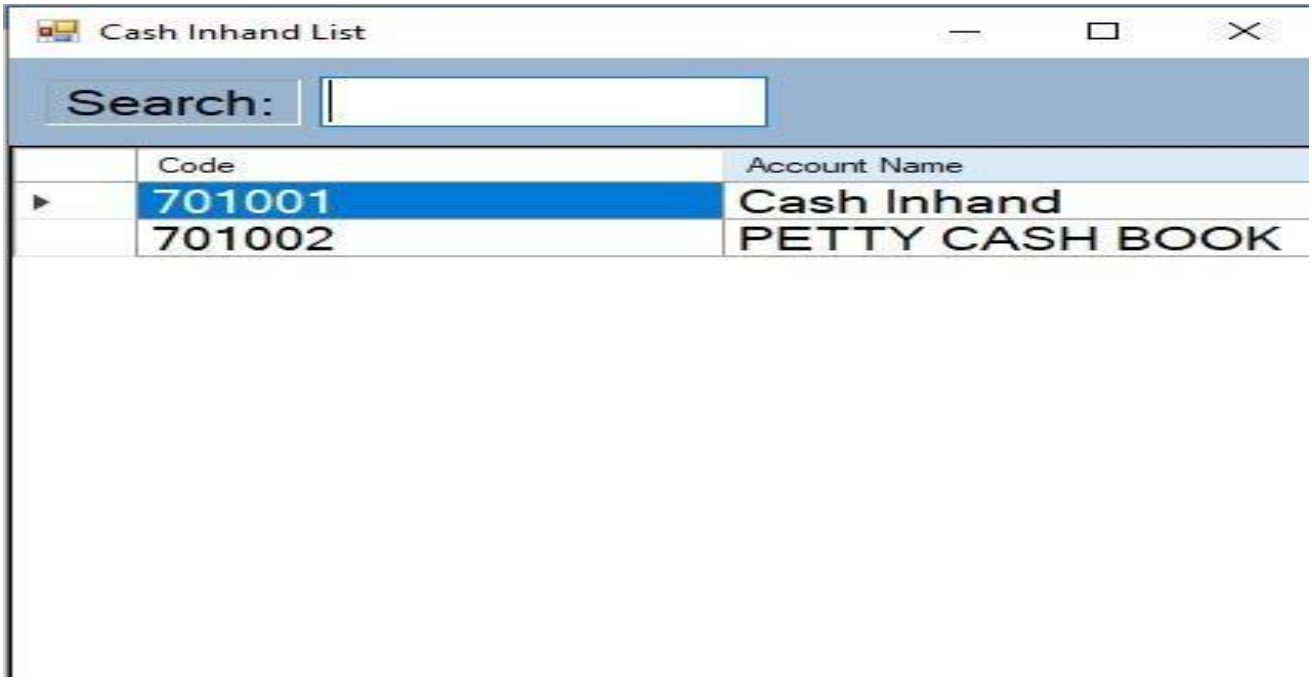


Figure 46 Cash Inhand List

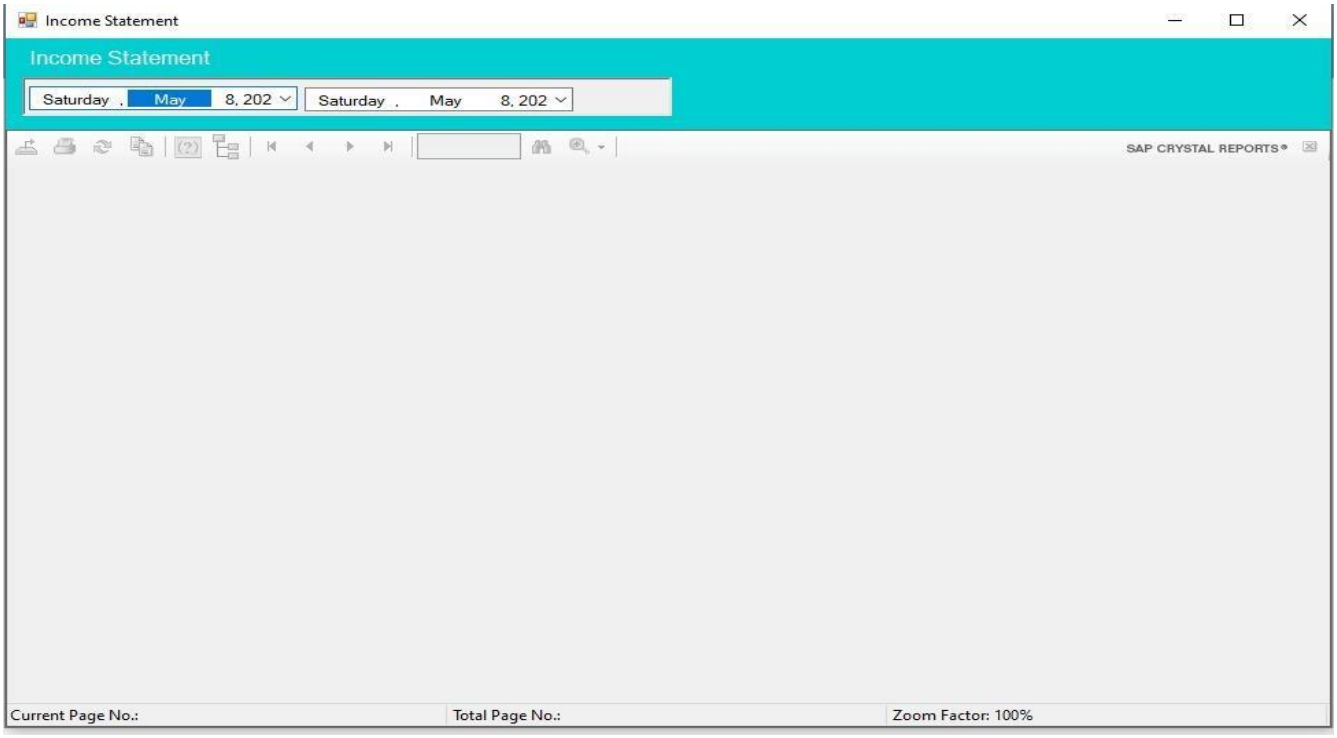


Figure 47 Income Statement

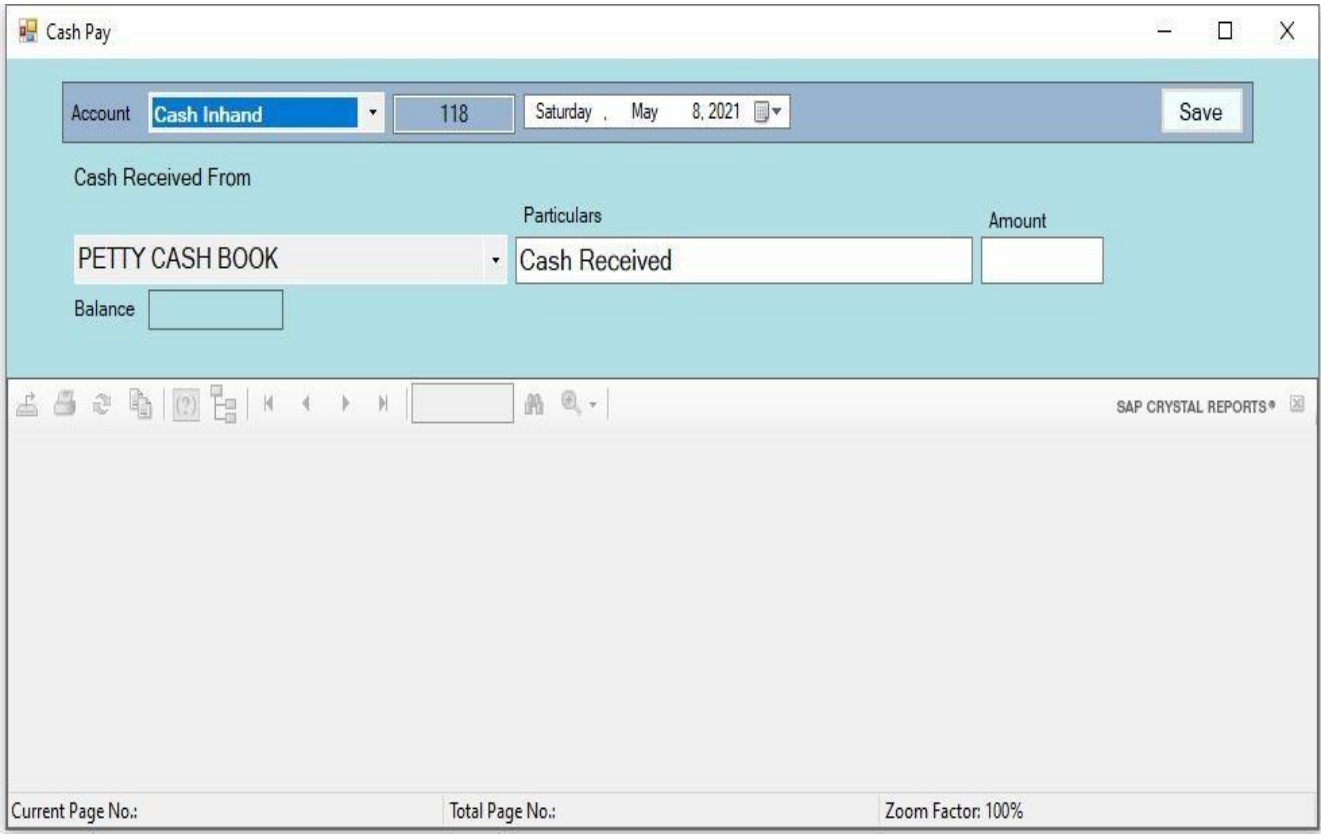


Figure 48 Cash Pay

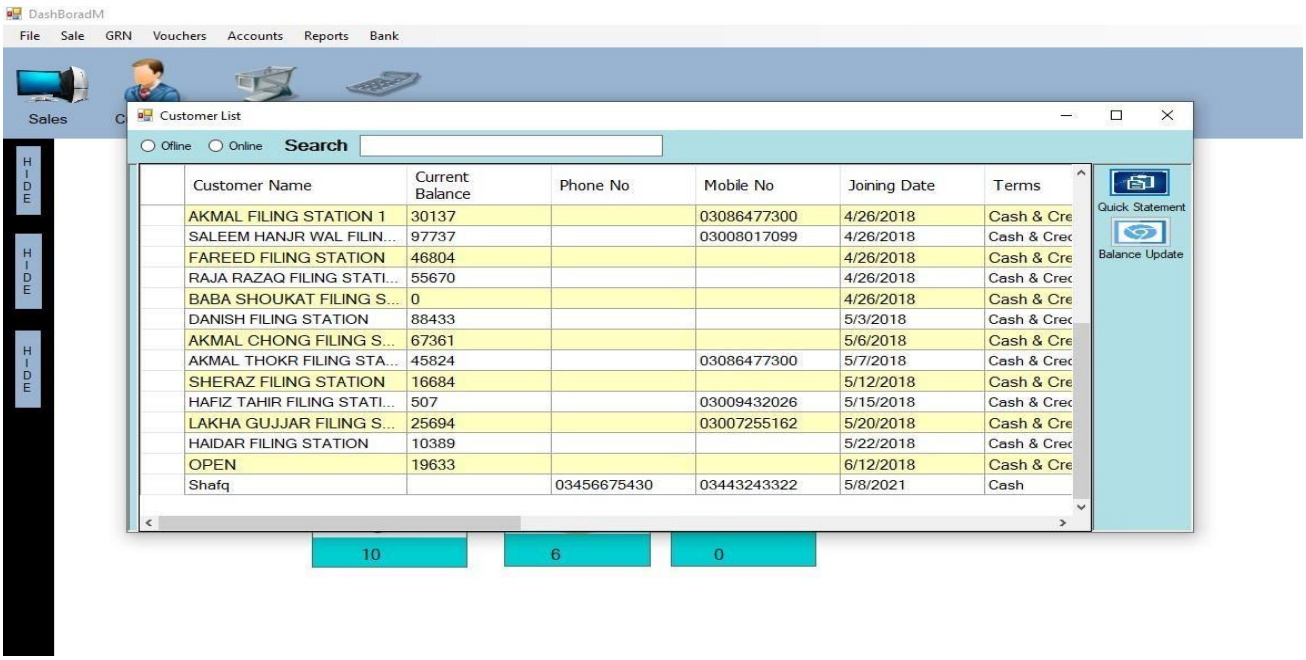


Figure 49 Customers List

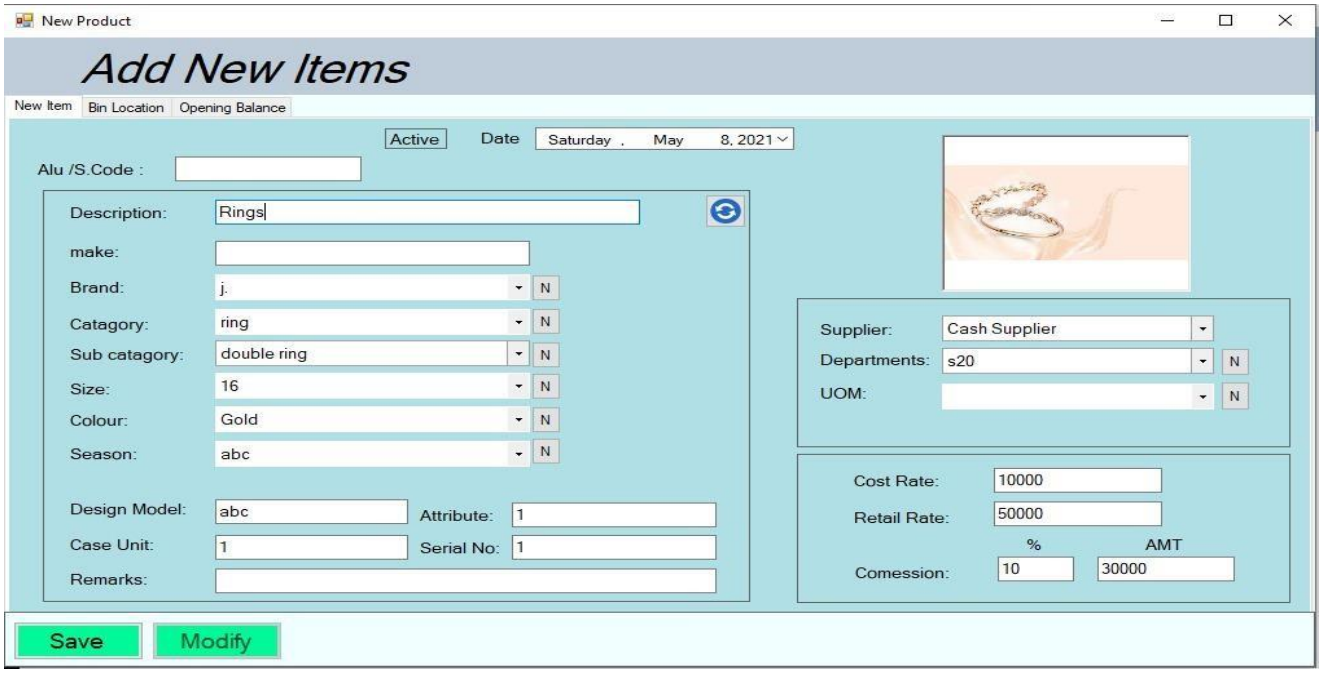


Figure 50 Add New Product Successfully

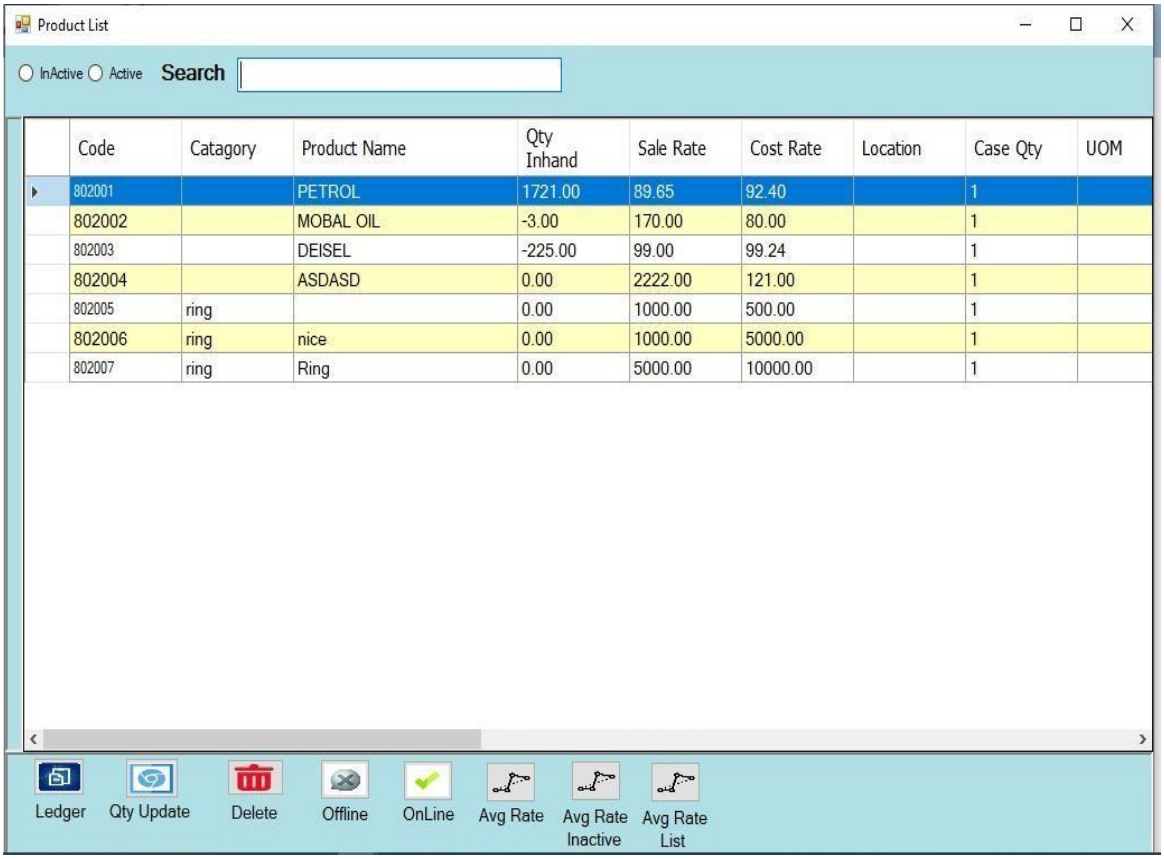


Figure 51 Product List

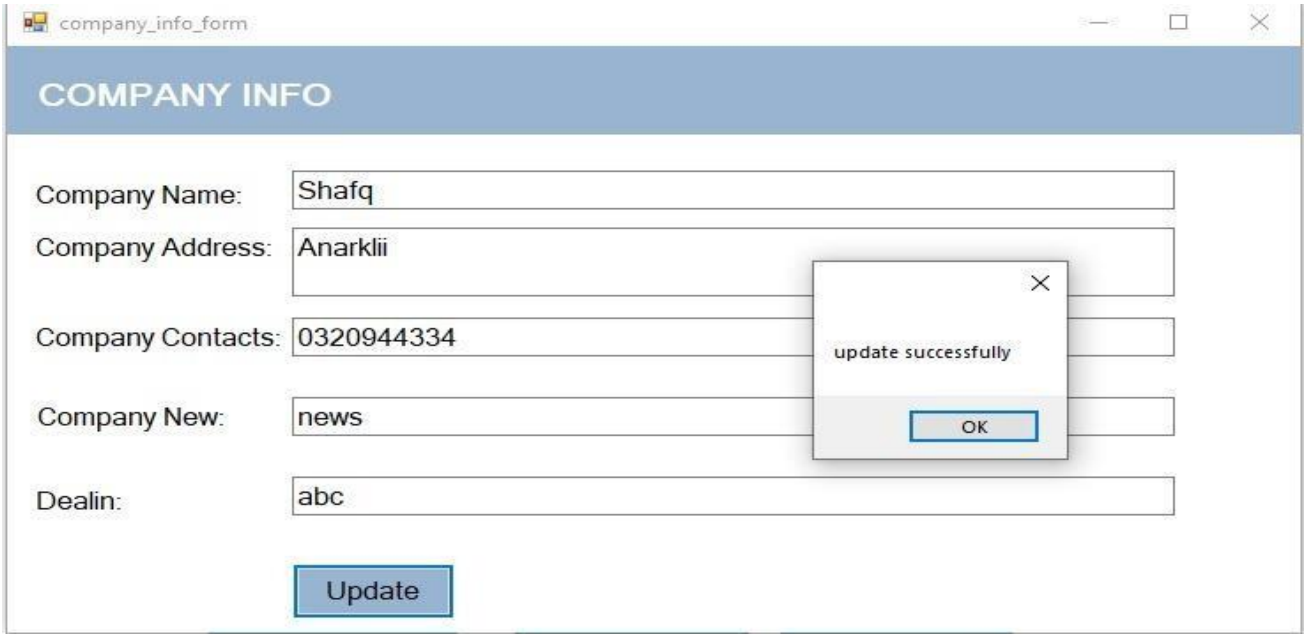


Figure 52 Company's Info

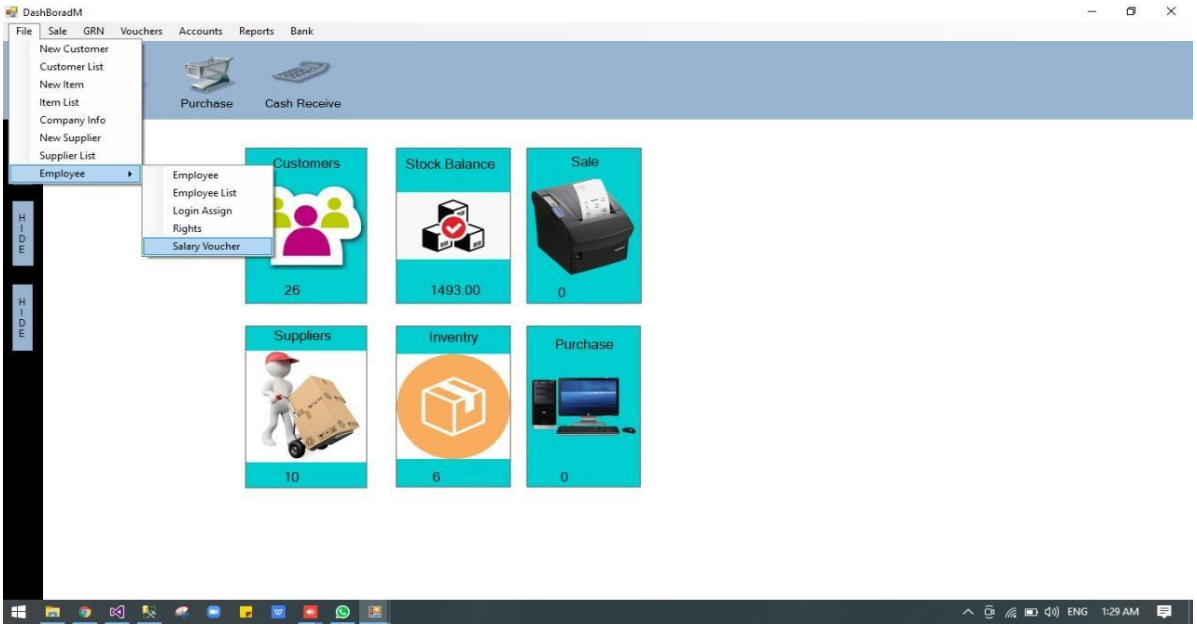


Figure 53

Figure 54

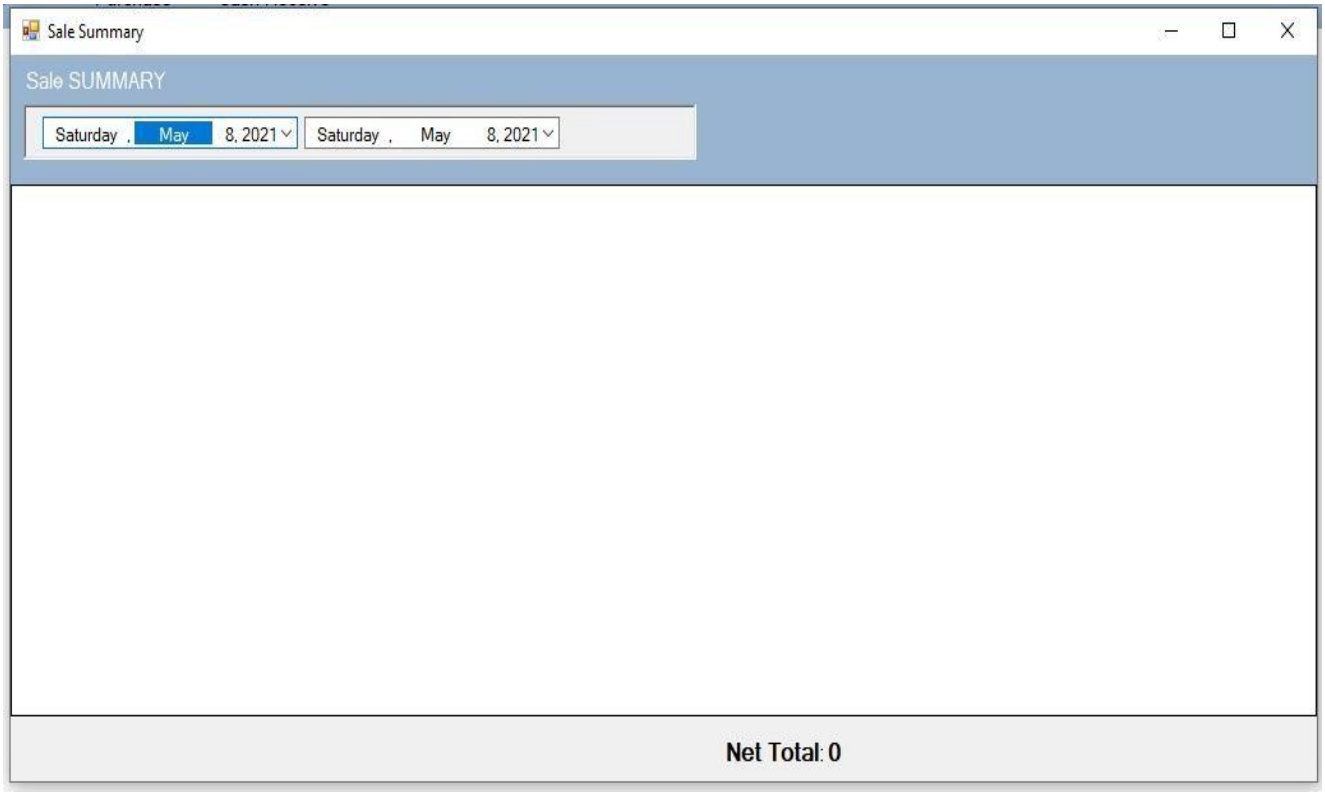


Figure 55 Sale Summary

Figure 56 GRN

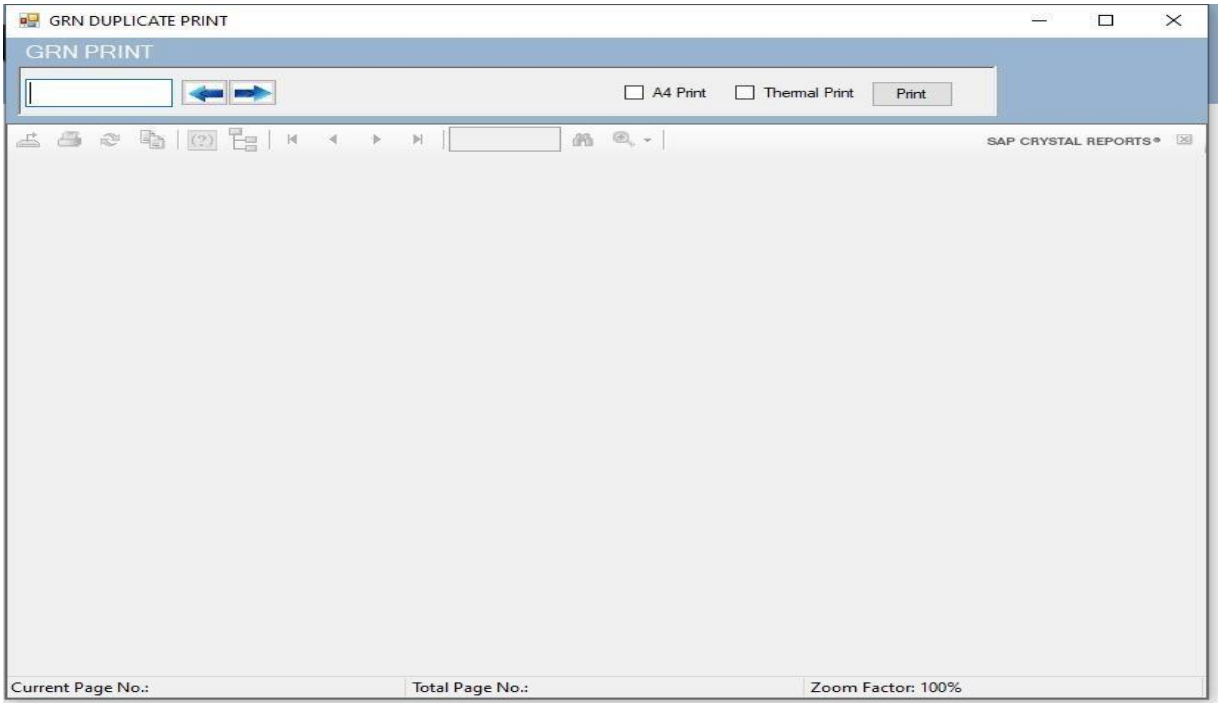


Figure 57 GRN Duplicate Print

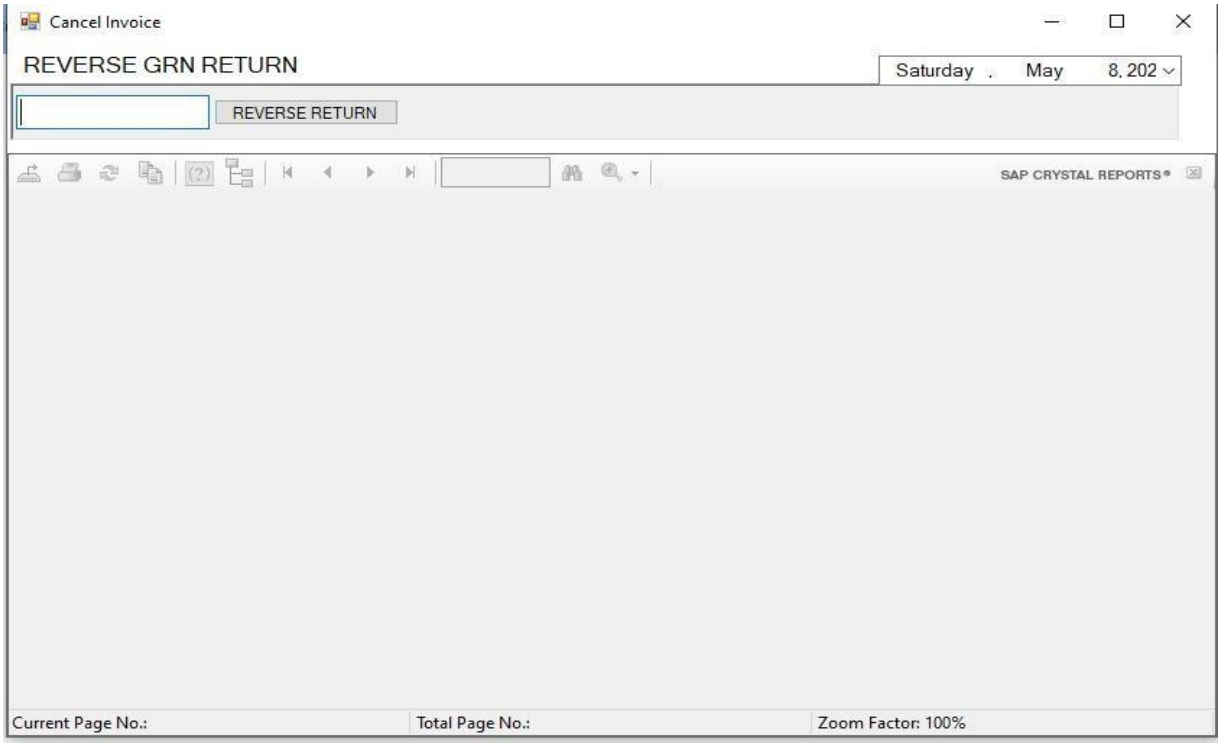


Figure 58 Cancel Invoice

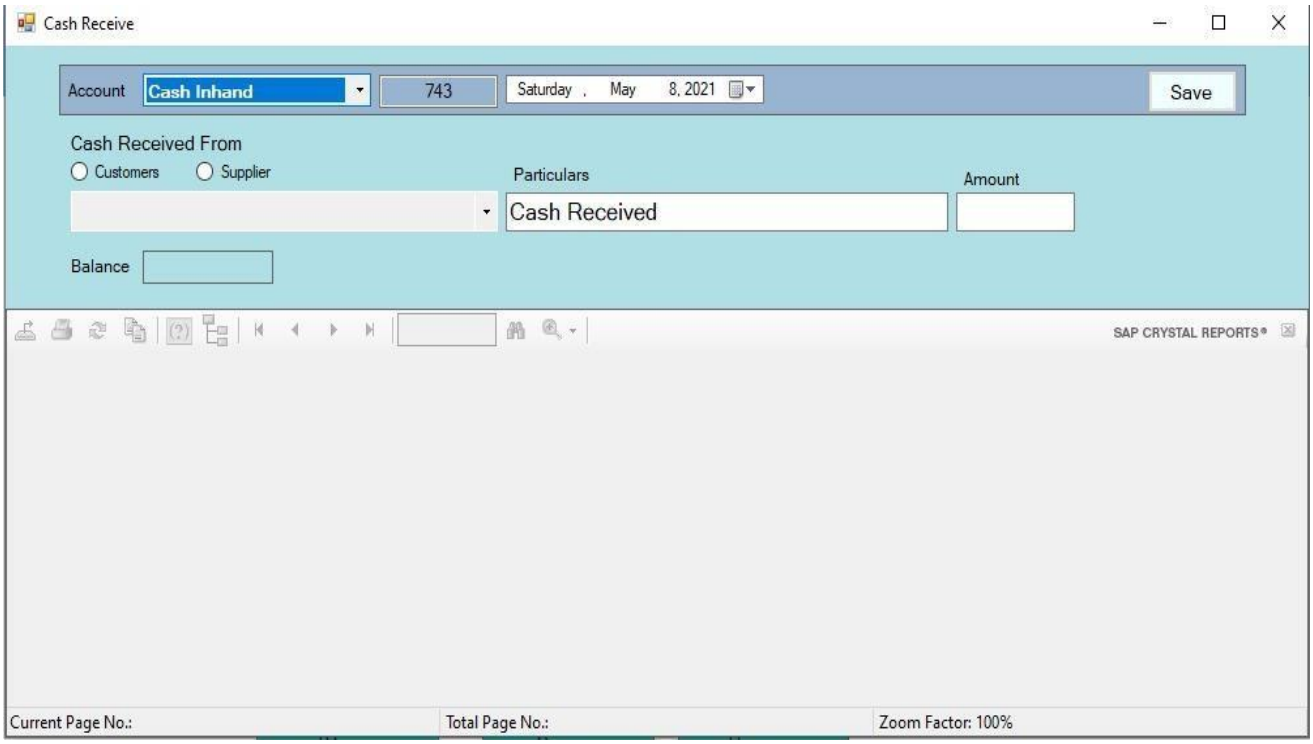


Figure 59 Cash Receive

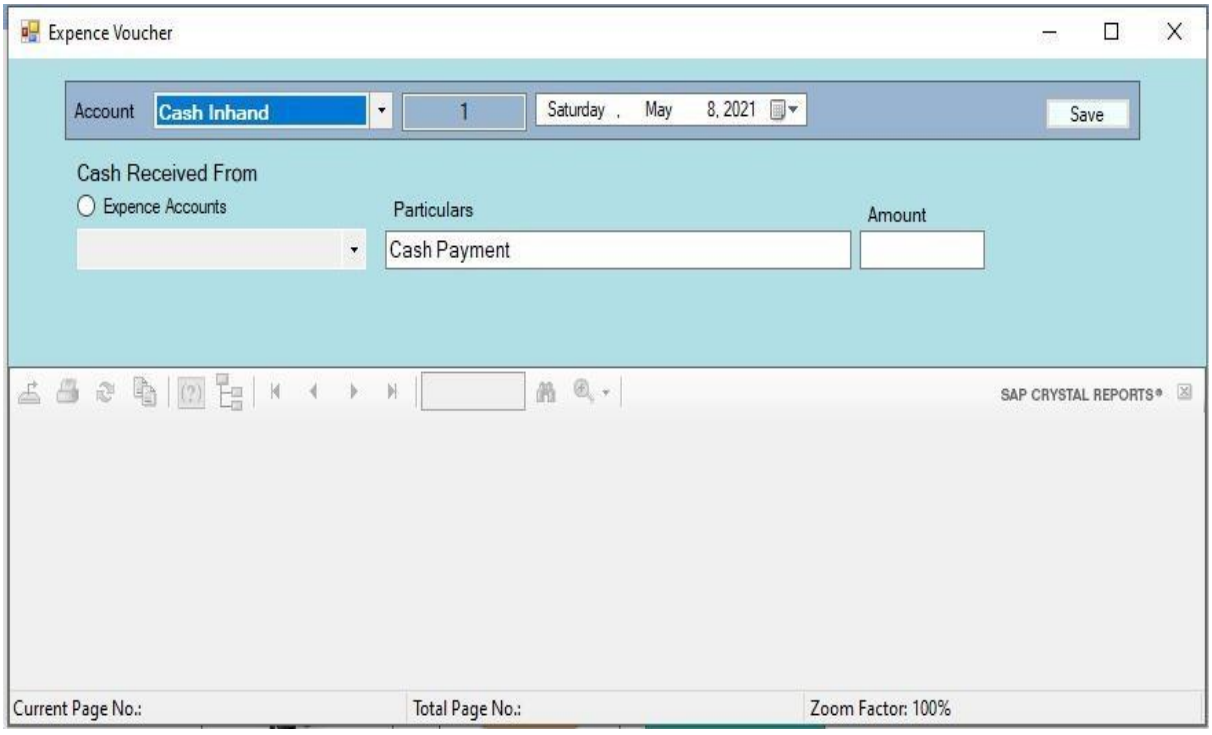


Figure 60 Expense Voucher

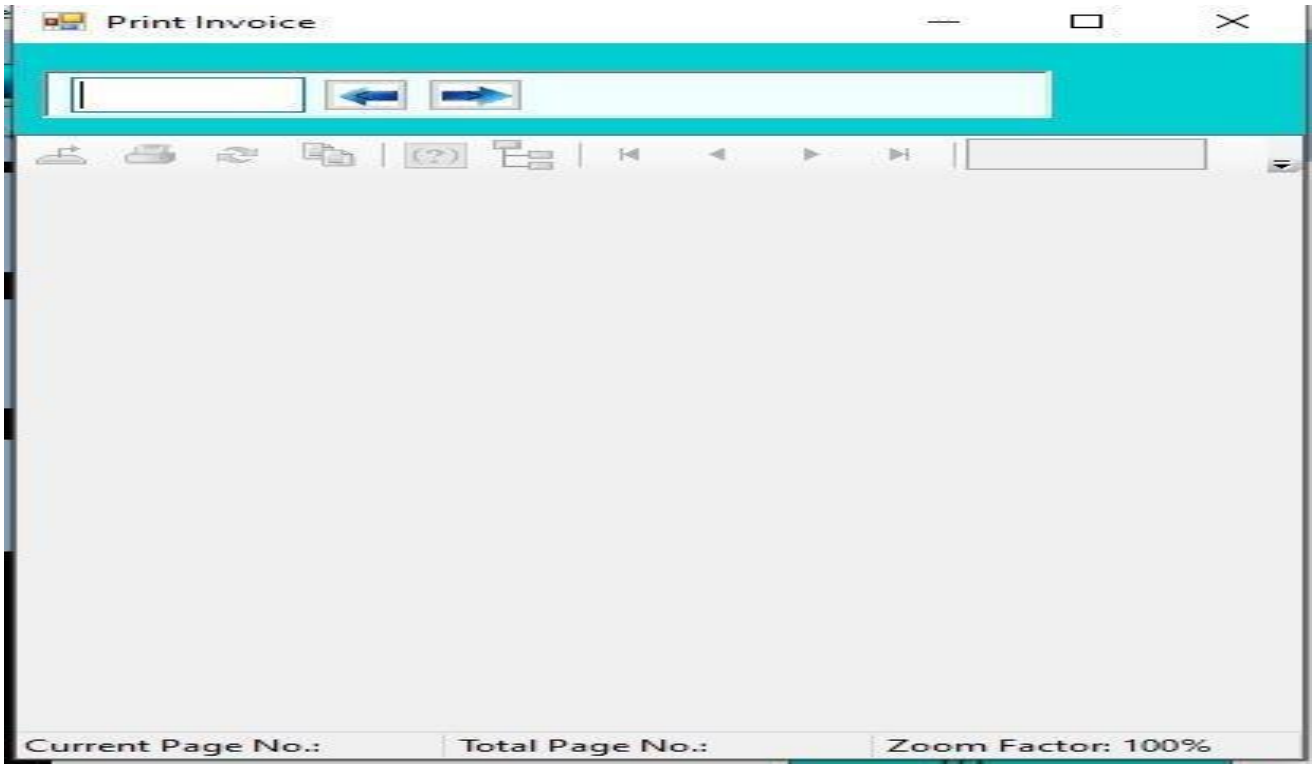


Figure 61 Print Invoice

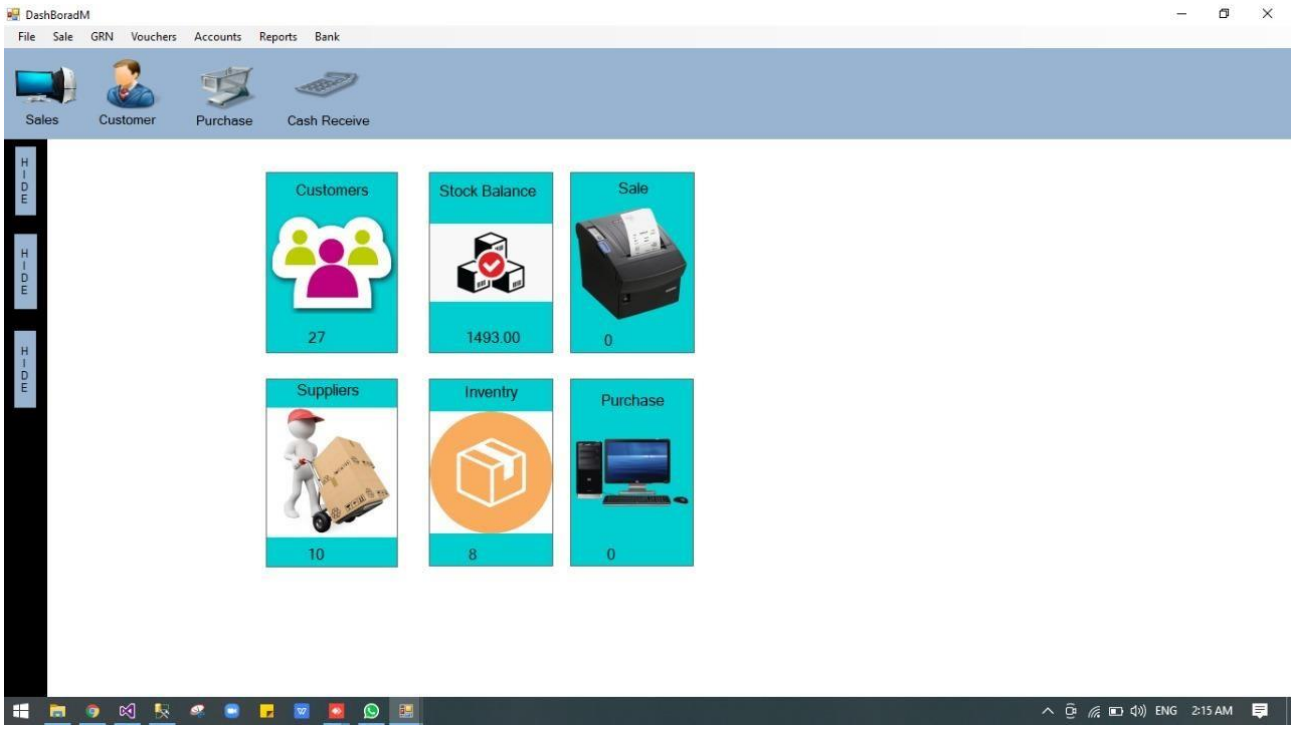


Figure 62 Dashboar

Figure 63 Product

Figure 64 Product

Figure 65 Product

Figure 66 Product

Figure 67 Product

6.5. Integration testing

This testing strategies combines all the modules involved in the system. After the independent modules are tested, dependent modules that use the independent modules are tested. This sequence of testing layers of dependent modules continues until the entire system is constructed. Though each module individually, they should work after linking them together. Data may be lost across interface and one module can have adverse effect on another. Subroutines, after linking, may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time, conducting test to uncover errors associated with the interface. In the testing the programs are constructed and tested in the small segments

The integration testing of this project is done scene-wise. The components that were previously individually tested are now integrated into one scene and tested to see if they still perform as they should.

Table 10 integration testing Each Button

Test Case ID	IT_SC1_002
Test case Summary	To verify that all buttons behave as expected
Prerequisites	All button must be rendered.
Test Procedure	Click on each button to see if it performs correctly.
Expected Result	1. Login button should provide you a dashboard. 2. Detail button provide a detail of Product. 3. Exit button redirect you on login page.
Actual Result	1. Login button provide user to enter in software. 2. Detail button provide you a detail. 3. Exit button redirect you on login page.
Status	Pass
Test Environment	C#

Table 11 integration testing for Update Button

Test Case ID	IT_SC2_002
Test case Summary	To verify that clicking on update button update the data.
Prerequisites	All Update sprites must be rendered.
Test Procedure	Click on each button to see if it performs correctly.
Expected Result	1. Unrecorded data not allow to update data.
Actual Result	1. Unrecorded data not allow to update data. 2. Nothing happens when user is not login.
Status	Pass
Test Environment	C#

Table 12 : integration testing for interaction component

Test Case ID	IT_SC3_002
Testcase Summary	To verify that all interaction components behave correctly.
Prerequisites	None.
Test Procedure	Manually check that each interaction component is behaving correctly.
Expected Result	1. Try should be login page. 2. Upon clicking login button, it should be displayed the main page. 3. When they enter in the software, it applies where the user full fill requirement. 4. Update work using update button. 5. Modify Customer or product details using delete button. 6. User should be send request again for same thing. 7. User should be taken to main screen on pressing Exit button
Actual Result	1. Login button are behaving correctly 2. update on pressing update button. 3. Exit data on pressing Exit button. 4. User is taken to main screen on pressing Exit button
Status	Pass
Test Environment	C#

6.6. Performance testing

6.6.1. Test case number: PT 01

Test case title: System response time testing

Table 13 : System response time testing

Test type	Performance testing	Test result
Required performance	The required response time of the system is less than 2 seconds	
System performance	The actual performance response time is 2 seconds.	Test passed

6.6.2. Test case number: PT 02

Test case title: Activity load time testing

Table 14 Activity load time testing

Test type	Performance testing	Test result
Required performance	The required page load time of the system is less than 3 seconds	
System performance	The actual page load time is 2 seconds.	Test passed

6.6.3. Test case number: PT 03

Test case title: Huge number of users

Table 15 Huge number of users

Test type	Performance testing	Test result
Required performance	System should perform normally and efficiently with a large number of users	
System performance	System performed all activities normally with a large number of users.	Test passed

6.6.4. Test case number: PT 04

Test case title: Waiting time testing

Table 16 Waiting time testing

Test type	Performance testing	Test result
Required performance	The average waiting time of the system is less than 3 seconds	
System performance	The actual performance response time is 2 seconds in some circumstances.	Test passed

6.6.5. Test case number: PT 05

Test case title: Connection establish time testing

Table 17 Connection establish time testing

Test type	Performance testing	Test result
Required performance	The connection must be established in less than 2 seconds	
System performance	The actual connection establish time is about 1 second.	Test passed

6.6.6. Test case number: PT 06

Test case title: Database send/get information time testing

Table 18 : Database send/get information time testing

Test type	Performance testing	Test result
Required performance	The system should save or retrieve data from database in less than 2 seconds.	
System performance	The actual time system sends or gets information to database in 1 second.	Test passed

6.7. Stress Testing

Stress testing is a process of measuring the stability and reliability of the system. Stress testing used to check the ability of system to handling the bugs and errors. Stress testing used to test the system's ability of error handling during extremely heavy load. To ensure that our system behaves normal during the stress we overload our system and examine the system's behavior in extreme circumstances. The (career recommended forum) behaves normally in unfavorable conditions and all of its features and functions perform their functionality.

Through this testing, we monitor our system and make sure our system

- System should save all data before crashing
- Make sure the connection from server
- If server no response, wait on login current page
- Make sure the saving of data and then show success message after operation done
- System should display of modifying and updating of records correctly.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

We are providing the Point-of-Sale(POS) Retail Jewellery software to create Cash Receipts, Invoices, and receive A/R Payments. In addition, you can create returns for Receipts and Invoices. The POS system replaces the cash register in a retail store. The POS system can track sales by various payment methods like cash, cheque and credit cards.

The POS system requires that you have the Accounts Receivable and the Inventory modules. For example, POS will update inventory sold figures in Inventory, or the salesperson's commission in Accounts Receivable. In addition, if you want to create Sales Orders, you will need the Order Entry (O/E) module. Other modules are optional.

The POS system replaces your manual cash drawer with a personal-computer. In addition, you can attach accessories like the POS Cash-drawer and the Column Receipt Printer.

We are providing IT based solution to different Jewellery shop holders so that their manual system can be automated. The system will provide all accessories and expense details. We are targeting a shop as a pilot and the whole market for overall business.

Today people prefer automated system overall manual system. We have also a noticed problem that normally Jewellery shop holders are still using manual systems through which they face lots of problems as data loss, payments issue, etc. There are very few shop holders who are using software-based solution. We are making desktop Application for Jewellery shop holders and its customers. We are also going to add a feature of add all detail of product, purchase detail, sale detail, expense detail, and how much is the monthly profit have we made this month and also the loss detail. The project objective is to convert the manual system into automated system.

7.2. Achievements and Improvements

During this complete project we face many challenges but the side of achievements is greater. Our project needs zero percent (0%) investment. Following are our biggest achievements:

7.2.1. Admin Module

The administrator panel is managing by administrators.it is responsible for accepting and rejecting the Customer requests and update the data.

7.2.2. Transaction Module

Transaction Module facilities the entries for programmer of work and Customer registration from the from-end of the desktop application. When a user if first enrolled with this software and registration is done at the beginning after filling the form user can check his status.

7.2.3. Improvements

As we know that there are always some accommodations of improvement in any system. Following are some basic improvements we need to change in future:

7.2.4. System quality

The quality is very important for the online system. Always keep an eye on the standard define quality ‘’the totality

of features and characteristics of product or service that bears its ability to satisfy stated or implied needs.

7.3. Lessons Learnt

When you come to practical life you should learn multiple lessons. Our team also learns multiple lessons while promoting this idea. Following are some basic lessons which we learn during this complete project:

7.3.1. Choose the rightsoftware:

Firstly, in technically side, you should choose the right software which is more beneficially for your system. You should not waste your time on pirated software. Selecting most appropriate software for your company will make your company healthier and profitable.

7.3.2. Always consult with an expert:

Before taking a critical and crucial decision, always consult with an expert because their advice can be more effective for you at that stage. In our project we consult all the time with our supervisor.

7.3.3. Keep an eye on your customers:

For running Career Recommended Forum keep an eye on Customers. Admin should always keep an eye on records all the time to avoid misusing any features.

7.4. Future Enhancements/Recommendations

- Detailed information gathering has to be done. Without that the purpose for using the
- software won't be satisfied properly.
- However it can give good profits in the long run.
- Implementing the software requires change in the business practices.
- Efficient organization of all knowledge is the analysis company and easy analysis access and retrieval of information is possible.
- In this project we can also include BAR CODE facility using the bar code reader, which will detect the expiry date and the other information about the related jewelries.
- Company using this software will always be able to plan in future and always be aware of their financial position in the market.
- It leads to streamline of business processes.
- The implementation and maintenance costs run very high (about 2 to 3 % of the company's revenue

Reference and Bibliography

Reference and Bibliography

https://www.atlassian.com/software/jira.pk

https://www.telemart.pk/jewellery-and-rings.html