BSc (Hons) in Software Engineering

**Application Development - CS6004ES**

**XYZ – Point of Sale System**

**Student Name:**

**Student Reg No:**

# Acknowledgements

I would like to thank all the personalities who are involved in the creation of this project report. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project.

My grateful thanks to my parents, my friends and lectures who despite being extraordinarily busy with their duties, took time out to hear, guide and kept me on the correct path. I do not know where I would have been without them.

Last but not the least, there were so many who shared valuable information that helped in the successful completion of this project and I am very much delighted to thank all the personalities involved

# Introduction

Application Development Course work 1 requires developing a Point of Sale desktop application in C# using object-oriented approach and Visual Studio, and the course work1 requires a documentation in MS Word file format. Visual Studio 2017 or any higher version should be used for development.

This Point of Sale application support for XYZ Company to manage their daily sales, and the purchase. Therefore, this desktop application will have two main portions to manage the Purchase (Good Received Note) and Sale (Point of Sales). So, this will help the client to track the daily and monthly sales and purchase this application as well as can generates several reports such as purchase order, sales order and top selling products

Two types of users can be work in this system which are Administrator and Normal User (Cashier), depend on the user login type, users can access the form (features). Especially some GRN approve features and sales income reports are available only for Admin user and Normal users cannot be accessed above forms.

Additionally, I have added Use case diagram, class diagram, ER diagram, source code and this will help to understand any third parties.

Table of Contents

[Acknowledgements 2](#_Toc44753009)

[Introduction 3](#_Toc44753010)

[1. The Architectures of the Applications 7](#_Toc44753011)

[2.1 Use Case Diagram - XYZ Inventory 7](#_Toc44753012)

[1.2 Class Diagram - XYZ Inventory 8](#_Toc44753013)

[1. Master 8](#_Toc44753014)

[1.1 frmLogin: 8](#_Toc44753015)

[1.2 frmMain: 8](#_Toc44753016)

[1.3 frmCustomer: 8](#_Toc44753017)

[1.4 frmEmployee: 9](#_Toc44753018)

[1.5 frmProduct: 9](#_Toc44753019)

[1.6 frmSupplier: 9](#_Toc44753020)

[2. Inventory 9](#_Toc44753021)

[2.1 frmGRN: 9](#_Toc44753022)

[3. POS 9](#_Toc44753023)

[3.1 frmPOS: 9](#_Toc44753024)

[4. Search 9](#_Toc44753025)

[4.1 frmGRNSearch: 9](#_Toc44753026)

[4.2 frmPOSSearch: 9](#_Toc44753027)

[4.3 frmProductSearch: 9](#_Toc44753028)

[4.3 frmSupplierSearch: 10](#_Toc44753029)

[Class Diagram - 1 10](#_Toc44753030)

[5. C sharp Classes 11](#_Toc44753031)

[5.1 DBConnection: 11](#_Toc44753032)

[5.2 Employee: 11](#_Toc44753033)

[5.3 GRN: 11](#_Toc44753034)

[5.4 GRNDetails: 11](#_Toc44753035)

[5.5 Location: 11](#_Toc44753036)

[5.6 POSDetails: 11](#_Toc44753037)

[5.7 POSHeader: 11](#_Toc44753038)

[Class diagram – 2 12](#_Toc44753039)

[1.3 Entity Relationship Diagram 13](#_Toc44753040)

[1.4 DFD Diagram for XYZ Inventory 14](#_Toc44753041)

[1.5 Used Algorithms 15](#_Toc44753042)

[ GRN detail part grid view cell value change calculation 15](#_Toc44753043)

[ GRN Header part discount values change calculation 15](#_Toc44753044)

[ Loyalty calculation 16](#_Toc44753045)

[ POS Detail part Calculation 17](#_Toc44753046)

[ Loyalty customer sorting 18](#_Toc44753047)

[ Daily and Monthly income summary 18](#_Toc44753048)

[1.6 Data Structure 19](#_Toc44753049)

[2. Implementation and Testing 20](#_Toc44753050)

[3.1. Objectives and Scope of Software Testing 20](#_Toc44753051)

[3.2. Software Testing Approach 20](#_Toc44753052)

[3.3. Test Plan 21](#_Toc44753053)

[3.4. Test Cases 21](#_Toc44753054)

[Test cases 01 21](#_Toc44753055)

[Test cases 02 22](#_Toc44753056)

[Test cases 03, Test cases 06 22](#_Toc44753057)

[Test cases 04 23](#_Toc44753058)

[Test cases 05 23](#_Toc44753059)

[Test cases 07, Test Cases 10 24](#_Toc44753060)

[Test cases 08 24](#_Toc44753061)

[Test cases 09 25](#_Toc44753062)

[Test cases 11 25](#_Toc44753063)

[Test cases 12 26](#_Toc44753064)

[Test cases 13 26](#_Toc44753065)

[Test cases 14 27](#_Toc44753066)

[3. My Reflection of Own Experience of Using C Sharp and Visual Studio 2015 28](#_Toc44753067)

[4.1 Advantages of Visual Studio 28](#_Toc44753068)

[ Initializers 28](#_Toc44753069)

[ Particle Classes and methods 28](#_Toc44753070)

[ Easy to Navigate Classes and Forms 28](#_Toc44753071)

[ Debugging 28](#_Toc44753072)

[ Intelligences 28](#_Toc44753073)

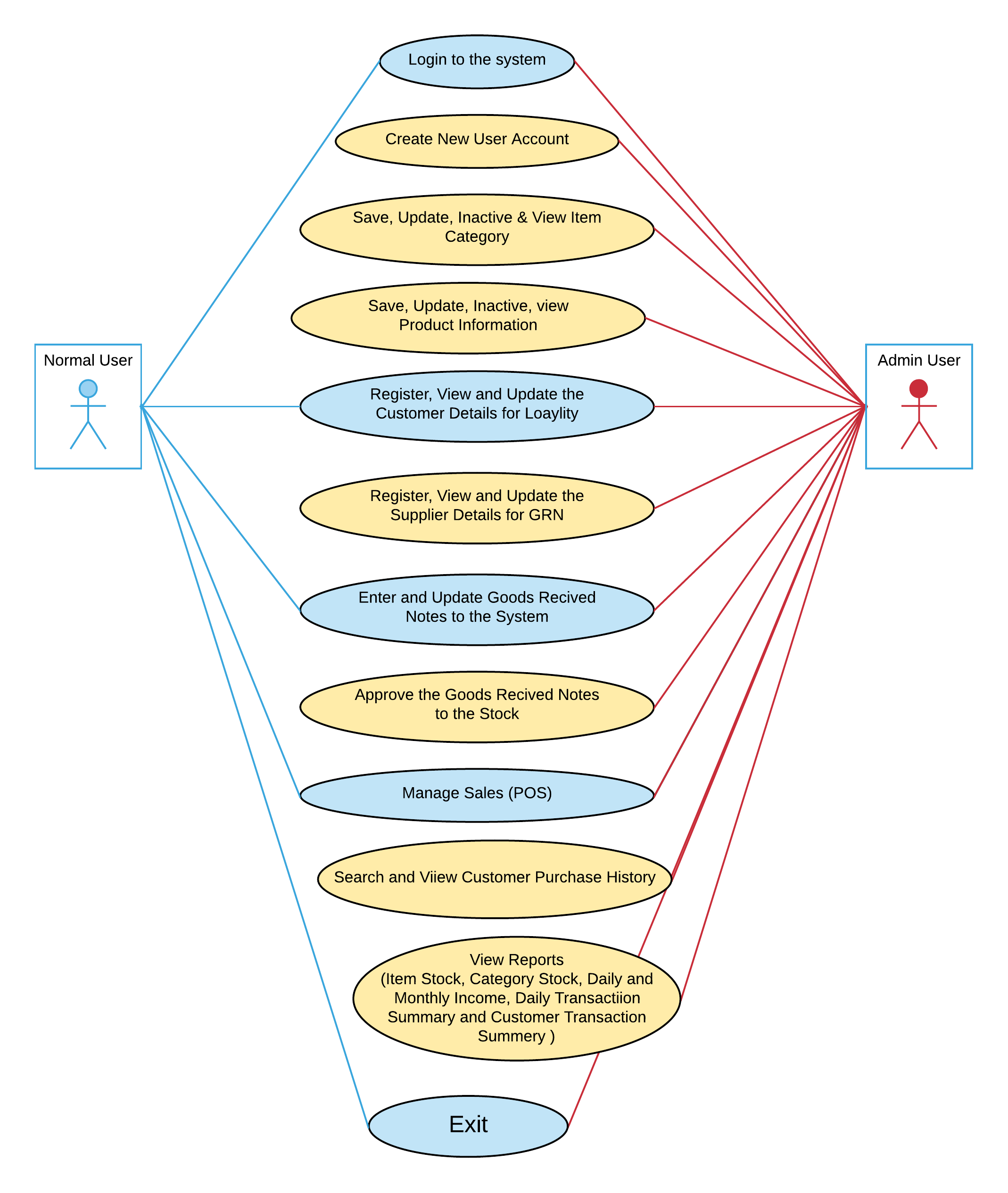
[4.2 Challenges and Self Criticism 28](#_Toc44753074)

[Challenges 28](#_Toc44753075)

[Self-Criticism 29](#_Toc44753076)

# The Architectures of the Applications

## 2.1 Use Case Diagram - XYZ Inventory



## 

Figure 19: Use Case Diagram

## Class Diagram - XYZ Inventory

### Class Diagram - 1

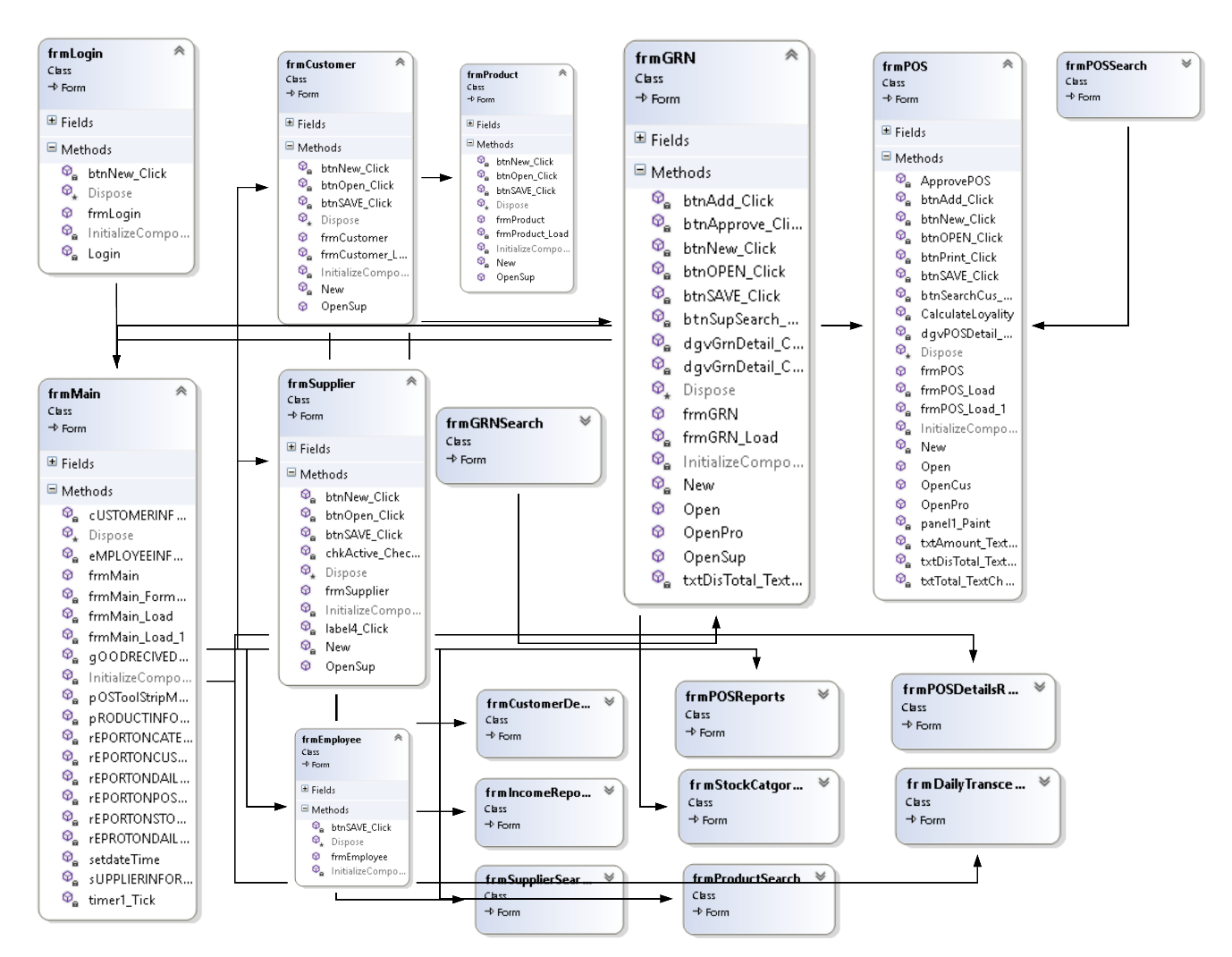


Figure 20: Forms

### Class diagram – 2

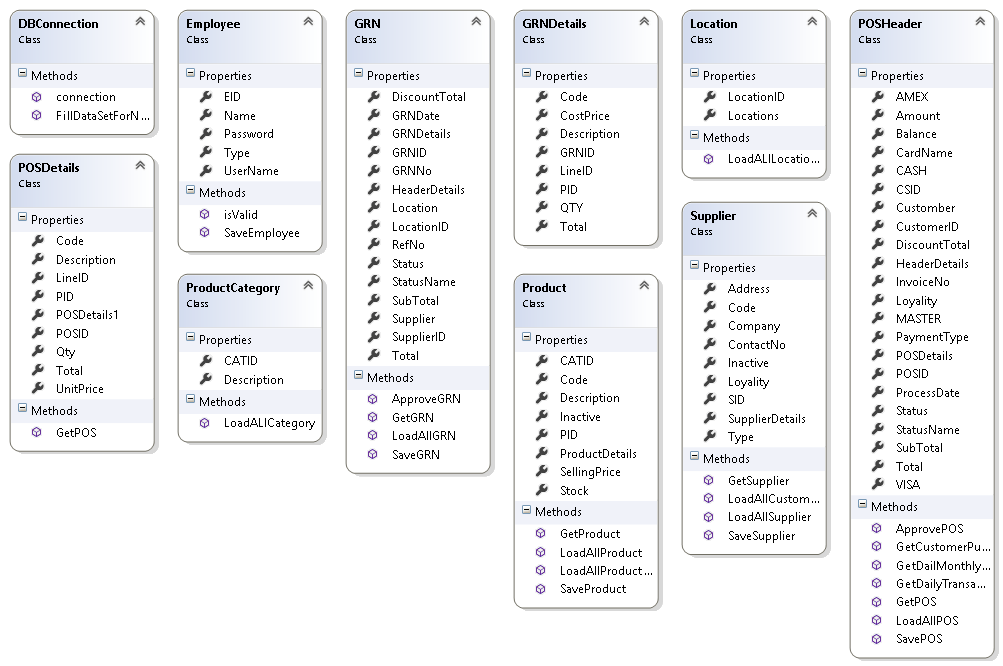


Figure 21: Classes

## Entity Relationship Diagram

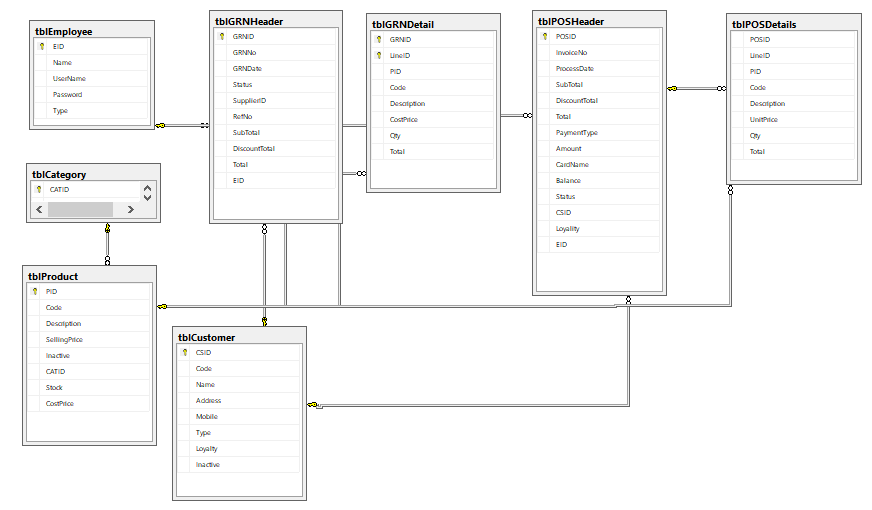


Figure 22:ER Diagram

## DFD Diagram for XYZ Inventory

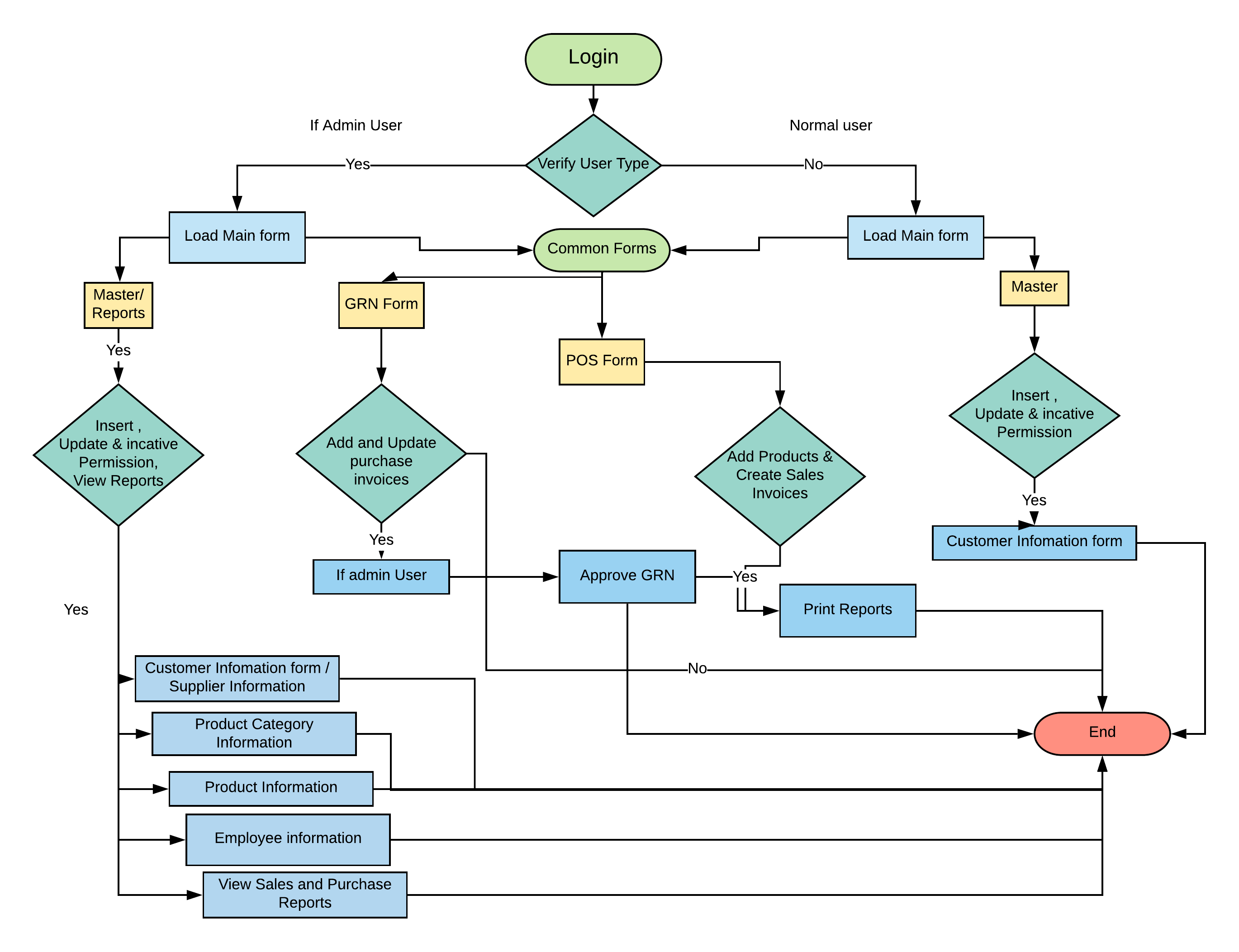


Figure 23: DFD Diagram for AGRO POS

## Used Algorithms

I have used few algorithms to solve the calculation problems in this application

Here are some sample coding portions:

### GRN detail part grid view cell value change calculation

This function will automatically calculate the row value of the product by entering quantity and unit price.

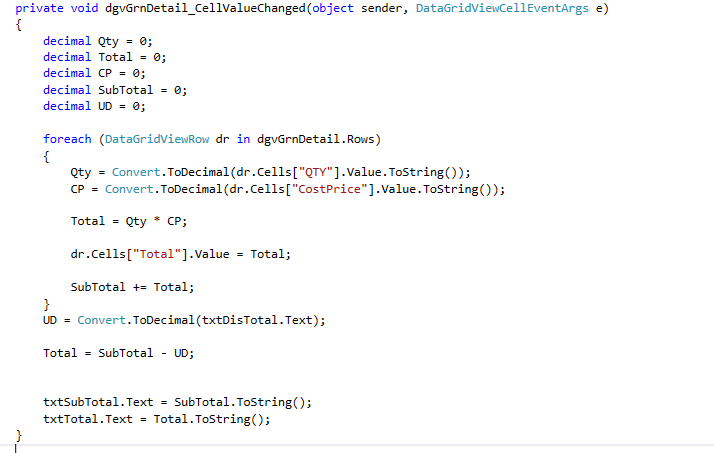


Figure 24: POS Details grid view value change

### GRN Header part discount values change calculation

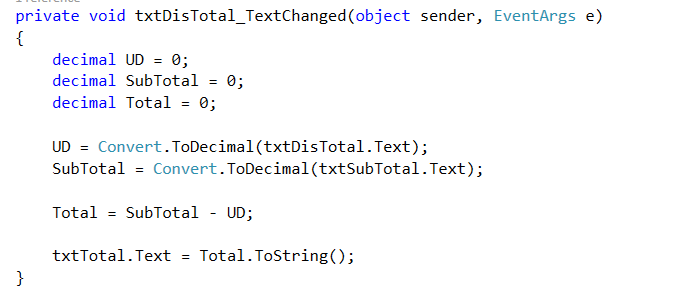


Figure 25: Discount value calculation

### Loyalty calculation

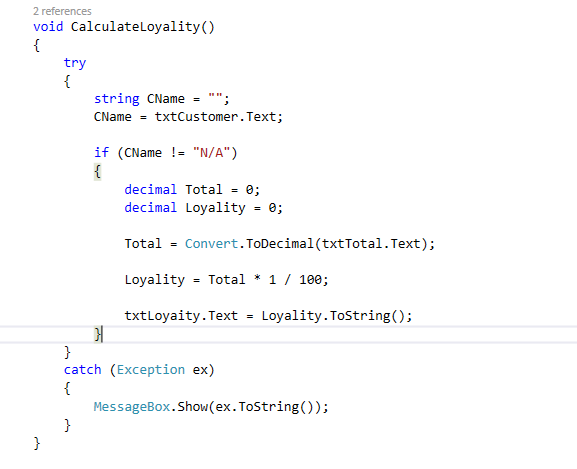


Figure 26: Loyalty Calculation

### POS Detail part Calculation

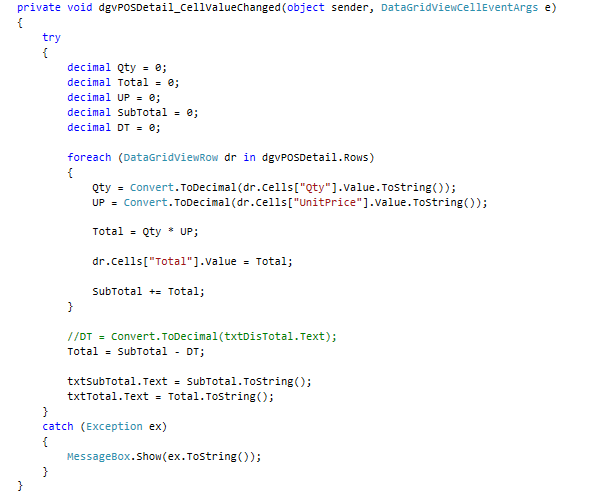


Figure 27: POS Detail part Calculation

### Loyalty customer sorting

Here I have use MS SQL Stored Procedure query to sort the Loyalty Customer Details

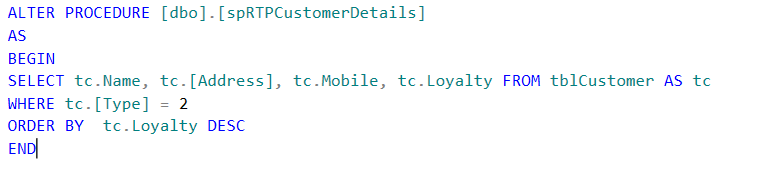


Figure 28: Loyalty Customer Shorting

### Daily and Monthly income summary

Here I have use MS SQL Stored Procedure select query to view daily and monthly income summary

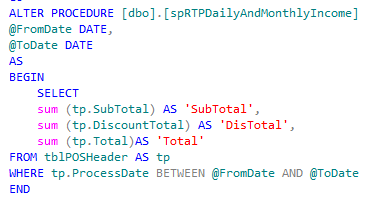


Figure 29: Daily and monthly income 1

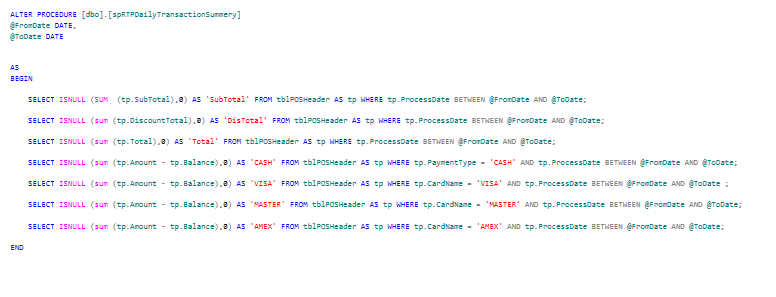


Figure 30: Daily and Monthly Income 2

## Data Structure

Here I have used three tier architectures for the XYZ Inventory Development. because this is very straight forward method and very easy to handle during the development process, and It makes logical separation between business layer and presentation layer.

We can have all the insert and calling functions in sperate class as well lines of coding can be reduced in front end.

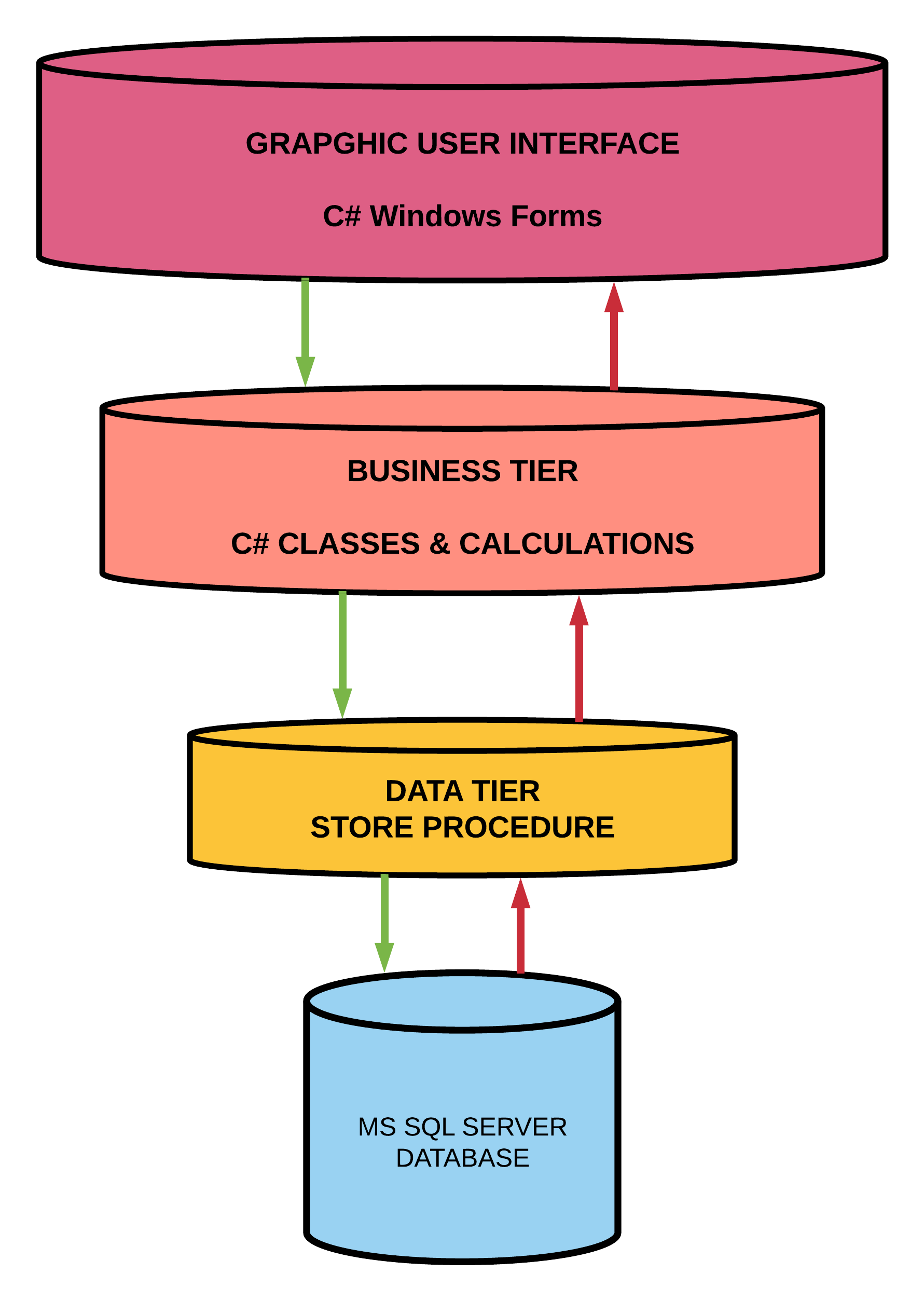


Figure 31: 3 Tier Layer

# Implementation and Testing

## 3.1. Objectives and Scope of Software Testing

Generally, software testing has different types of goals and objectives, but there are some major objectives in software testing; such as,

* To ensure the **Business Requirement Specification.** (BRS)
* Gaining the confidence in and providing information about the level of **Quality.**
* **Finding Defect –** which may get while developing the software by programmers/developers.
* To ensure that the end product or result meets the business and user requirements.
* To Cover 100% of functional requirements

## 3.2. Software Testing Approach

Software Testing approach is the test strategy implementation of a project, and it define how testing will be carried out and what ate testing techniques will be used. To carry out the software testing many types can be used, but Black Box testing is going to be used for this project.

**Black Box Testing**

Black box testing is Software testing techniques in which functionality of the software under test is tested without looking at the internal code structure, implementation details and knowledge of internal path of the software.

This type of testing is based entirely on the software requirements and specifications.



Figure 32: Black box Testing

## 3.3. Test Plan

|  |  |
| --- | --- |
| **Test Cases** | **Objectives** |
| TC\_1 | Users able to be logging to the system using “XYZ Inventory” login form |
| TC\_2 | Main form successfully loading |
| TC\_3 | User able to register, view and update customer information to System |
| TC\_4 | User able to insert product information, view and update product information to System |
| TC\_5 | User able to register, view and update employee details to the system |
| TC\_6 | User able to register, view and update supplier information to System |
| TC\_7 | Main form can navigate Goods Received Note (GRN) form |
| TC\_8 | User able to create GRNs and update the product information with supplier details |
| TC\_9 | Admin can open and approve the GRNs for Stock update |
| TC\_10 | Only admin will have the Approve privilege |
| TC\_11 | Main form able to navigate POS form |
| TC\_12 | User able create sales invoice using POS application for each sales transaction |
| TC\_13 | System is generating 1% loyalty point for the customer in total amount of the invoice |
| TC\_14 | Admin can access all the purchase and sales reports |

## 3.4. Test Cases

### Test cases 01

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | User able to login to the system |
| **Test Data** | Username = Jerom Password = 123 |
| **Expected Result** | Allowing to enter to the main form |
| **Actual Result** | Main form is loading successfully – Result is ok |

### Test cases 02

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | Main form is successfully loading |
| **Test** | Showing all the navigations to other forms depend on user types |
| **Expected Result** | 1. Admin can access all the forms 2. Normal user can access only selected forms |
| **Actual Result** | 1. All the forms navigations are visible and accessible for admin – Result is ok  2. Selected forms are only visible and accessible for Normal User - Result is ok |

### Test cases 03, Test cases 06

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | User able to register, view and update customer and supplier information to the System |
| **Test Data** | 1. CSID (Customer ID, Supplier ID) – Auto generate  2. Name, address and mobile number  3. Active or inactive status |
| **Expected Result** | All data should be saved to database and able view after saving |
| **Actual Result** | All data saved to database and able to view – Result is ok |

### Test cases 04

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | User able to insert product information, view and update product information to System |
| **Test Data** | 1. Code (Product Code) – Auto generate  2. Description, Category and selling price  3. Active or inactive status |
| **Expected Result** | All data should be saved to database and able view after saving |
| **Actual Result** | All data saved to database and able to view – Result is ok |

### Test cases 05

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | User able to register, view and update employee details to the system |
| **Test Data** | 1. EID (Employee ID) – Auto generate  2. Username  3. Password – Default password 123 auto generate  4. Active or inactive status |
| **Expected Result** | All data should be saved to database and able view after saving |
| **Actual Result** | All data saved to database and able to view – Result is ok |

### Test cases 07, Test Cases 10

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | Main form can navigate Goods Received Note (GRN) form |
| **Test** | GRN form able to open for any user type |
| **Condition 1** | Approve button only visible for Admin |
| **Expected Result** | GRN form should open Admin button only visible for Admin |
| **Actual Result** | GRN form is opening – Result is ok Admin button is not visible for Normal user and Admin can do the approve process – Result is ok |

### Test cases 08

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | User able to create GRNs and update the product information with supplier details |
| **Condition 1** | 1. Add the products to the grid view 2. Able select supplier 3. Able to enter Supplier invoice number 4. Able to enter supplier invoice date 5. Total value of the supplier invoice calculates automatically 6. Able to enter Supplier invoice discount |
| **Expected Result** | Able to save all above data to the database header and detail parts with unique GRN no. |
| **Actual Result** | Header and detail parts data is saving to database – Result is ok |

### Test cases 09

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | Admin can open and approve the GRNs for Stock update |
| **Condition 1** | 1. should able to load all the entered GRNS 2. After approving the GRNs the stock should be added to the system as well product table |
| **Expected Result** | Stock should be updated to the product table |
| **Actual Result** | Product table is updated – Result is ok |

### Test cases 11

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | Main form able to navigate POS form |
| **Test** | Point of sale form able to open for any user type |
| **Expected Result** | POS form should open |
| **Actual Result** | POS form is loaded successfully – Result is ok |

### Test cases 12

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | User able create sales invoice using POS application for each sales transaction |
| **Test** | 1. Add the products to the grid view 2. Able select Customer for loyalty 3. Able to generate auto invoice no 5. Total value of the Sales invoice calculates automatically 6. Able to enter discount value  7. Different payment type – Cash and Card (VISA, MASTER &   AMEX) |
| **Expected Result** | 1. Able to save all above data to the database header and detail parts with unique POSID 2. Able to print sales Invoice |
| **Actual Result** | 1. POS Header and POS detail parts data is saving to database – Result is ok 2. Sales Invoice can be print – Result is ok |

### Test cases 13

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | System is generating 1% loyalty point for the customer in total amount of the invoice |
| **Condition 1** | 1 % loyalty points should be added to the customer database for each sale invoice amount |
| **Expected Result** | 1% loyalty amount is successfully added to customer database |
| **Actual Result** | 1% loyalty amount is successfully added to customer database – Result is ok |

### Test cases 14

|  |  |
| --- | --- |
| **Test Case** | **Test** |
| **Test Objective** | Admin can access all the purchase and sales reports |
| **Test** | Only admin can view all the reports |
| **Expected Result** | All the reports are available for admin user |
| **Actual Result** | Reports are available – Result is ok |

# My Reflection of Own Experience of Using C Sharp and Visual Studio 2015

## Advantages of Visual Studio

## Initializers

Easy to form Objects and Collections while not business constructors.  
It will use with anonymous sort. I even have used this in several elements of the code wherever I initialized details fetched from alternative forms.

## Particle Classes and methods

In Visual Studio development kit, we can create the classes and code files separately and during the compilation process it will combine files.

## Easy to Navigate Classes and Forms

During the coding process users can access the related classes with fn + F12 function, it will more help to locate the exact class while having many classes.

## Debugging

It is very easy to find the coding error or other errors by using debugging feature and error notifications are very useful to find error coding line.

## Intelligences

While writing the code user can get suggestion, so novice users can easily code.

## Challenges and Self Criticism

### Challenges

* Designing the GRN form and writing methods to get save the GRN Header and Detail Part
* GRN Approve methods and Stock (Product Count Update)
* Writing Stored Procedure to add customer loyalty to customer table and it should add with current loyalty points.
* Creating Crystal Report to print sales invoice and Good Received Note

### Self-Criticism

* Little bit poor in Time Management
* Unable to give responsive user interface with more attractive views
* Main form should be more informative
* Need to add more elements to product information