# EasyPOS Intelligent Interface system to facilitate Customers, Employees and Cashier

**Final Year Project**

**Session 2018-2022**

A project submitted in partial fulfillment of the degree of

**BS** in Computer Science

Department of Computer Science

Faculty of Computer Science & Information Technology

Superior University, Lahore

Spring 2022

| | |
|---|---|
| Typse (Nature of project) | [ ✓ ] **D**evelopment [ ] **R**esearch [ ] **R&D** |
| Area of specialization | Web Development |
| FYP ID | **FYP-BCSM-F21-030** |

**Project Group Members**

| Sr.# | Reg. # | Student Name | Email ID | *Signature |
|---|---|---|---|---|
| (i) | BCSM-F18-342 | Muhammad Ali | ali.ikram714@gmail.com | |
| (ii) | BCSM-F18-160 | Nauman Shabbir | Bcsm-f18-160@superior.edu.pk | |

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

## Plagiarism Free Certificate

This is to certify that, I Muhammad Ali S/D of Ikram Ul Haq group leader of FYP under registration no BCSM-F18-342 at Computer Science Department, The Superior Univesrity, Lahore. I declare that my FYP report is checked by my supervisor.

Date: 21-07-2022 Name of Group Leader: Muhammad Ali Signature: _____

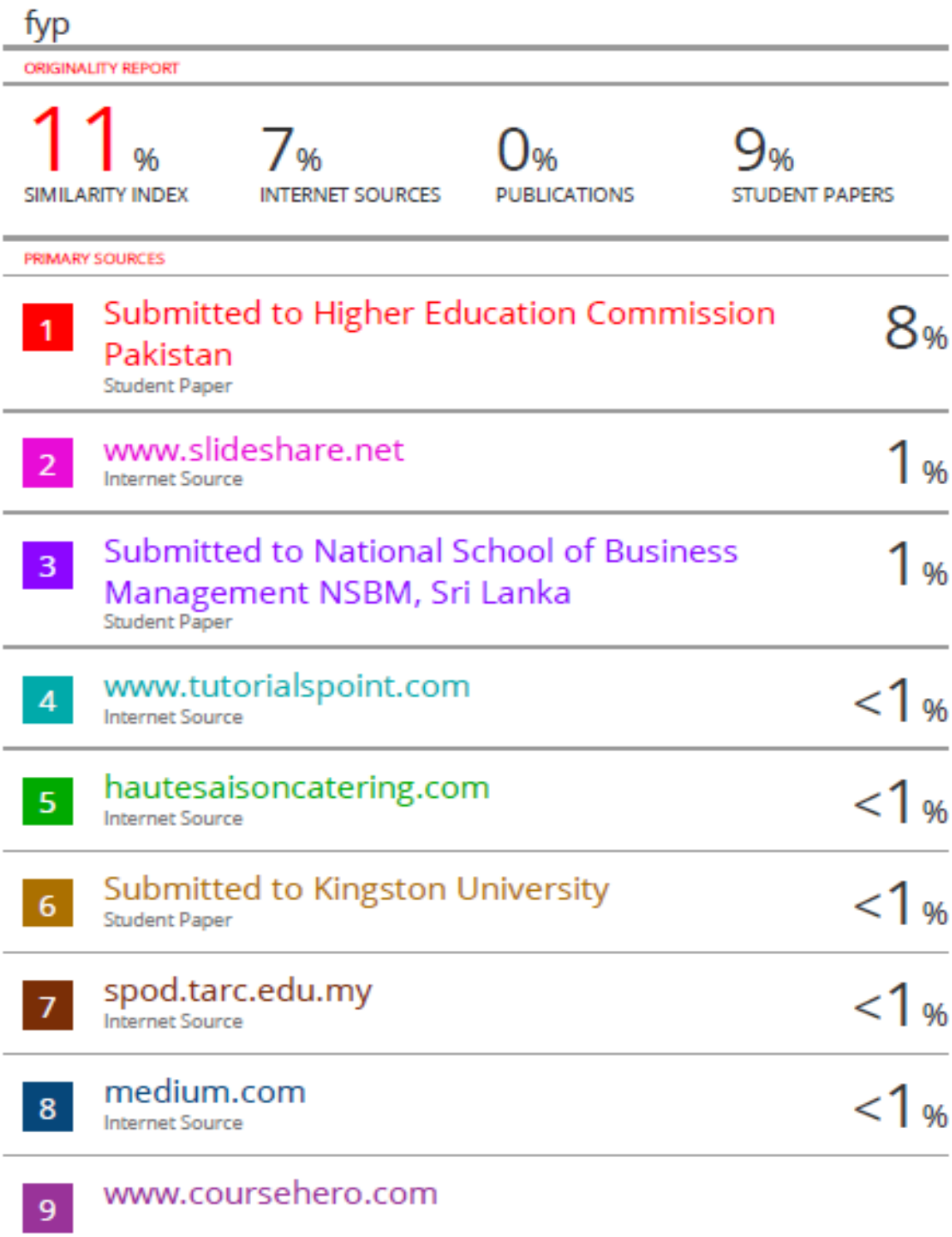
Name of Supervisor: Miss Arshia Naeem

Designation: Lecturer

Signature: _____

HoD: Dr. Irfan Ud Din

Signature: _____



fyp

ORIGINALITY REPORT

| 11% | 7% | 0% | 9% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Higher Education Commission Pakistan<br>Student Paper | 8% |
| 2 | www.slideshare.net<br>Internet Source | 1% |
| 3 | Submitted to National School of Business Management NSBM, Sri Lanka<br>Student Paper | 1% |
| 4 | www.tutorialspoint.com<br>Internet Source | <1% |
| 5 | hautesaisoncatering.com<br>Internet Source | <1% |
| 6 | Submitted to Kingston University<br>Student Paper | <1% |
| 7 | spod.tarc.edu.my<br>Internet Source | <1% |
| 8 | medium.com<br>Internet Source | <1% |
| 9 | www.coursehero.com | |

# Project Report

## EasyPOS Intelligent Interface system to facilitate Customers, Employees, and Cashiers

## Change Record

| Author(s) | Version | Date | Notes | Supervisor's Signature |
|---|---|---|---|---|
| Muhammad Ali<br>Nauman Shabbir | 1.0 | 13-10-2021 | Started working on the project | |
| Muhammad Ali<br>Nauman Shabbir | 1.0 | 01-11-2021 | Supervisor has approved the project and advised to initiate the project. | |
| Muhammad Ali<br>Nauman Shabbir | 1.0 | 15-11-2021 | Now working on approved project has been started. | |
| Muhammad Ali<br>Nauman Shabbir | 1.0 | 30-11-2021 | Firstly, we Working on project documentation. | |
| Muhammad Ali<br>Nauman Shabbir | 1.0 | 17-12-2021 | We Added and removed some content according to our superior visor advice. | |
| Muhammad Ali<br>Nauman Shabbir | 1.0 | 25-12-2021 | Now Documentation is completed. | |
| Muhammad Ali<br>Nauman Shabbir | 2.0 | 01-01-2022 | Now design the frontend according to supervisor instructions. | |
| Muhammad Ali<br>Nauman Shabbir | 2.0 | 30-01-2022 | Database Implementation | |
| Muhammad Ali<br>Nauman Shabbir | 2.0 | 09-02-2022 | Backend Coding | |
| Muhammad Ali<br>Nauman Shabbir | 2.0 | 18-03-2022 | CRUD Operations | |
| Muhammad Ali<br>Nauman Shabbir | 2.0 | 11-04-2022 | Data Entry | |
| Muhammad Ali<br>Nauman Shabbir | 2.0 | 28-04-2022 | Testing & Cross Checking | |
| Muhammad Ali<br>Nauman Shabbir | 2.0 | 09-05-2022 | Project is Completed | |

# APPROVAL

**PROJECT SUPERVISOR**

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

**PROJECT MANAGER**

Comments: _____

_____

Date:_____ Signature:_____

**HEAD OF THE DEPARTMENT**

Comments: _____

_____

Date:_____ Signature:_____

# Dedication

This project is dedicated to the All-Powerful God for his love, direction, and protection during my academic career.

# Acknowledgments

I give thanks to the Almighty God for his protection, wisdom, knowledge, and insight during my academic career. My supervisor, Miss Arshia Naeem, has my sincere gratitude for her unwavering support and meticulously coaching me throughout the time. Despite his busy schedule and obligations, he accommodated all of my requests for his time. I appreciate your diligence and skill.

# Executive Summary

Managing records keep an eye on everything is the biggest challenge for the Storeowners because they do not have any system to keep records of everything, manage documents, collect sales and check inventory for the stock. The problem is keeping records of the sales staff about their daily sales, cash, and inventory they sold out. Your consumer makes payments for goods or services at your business using EasyPOS. Simply put, every time a customer makes a purchase, they complete a POS transaction. For the benefit of shops, our POS software offers a card processing option along with inventory control, contactless payment methods, e-commerce integration features, and much more. EasyPOS system is the solution we suggested. It is a development over the cash register, manual inventory control, and traditional selling methods. But it is also a lot more than that. With our cutting-edge POS software, you can keep track of your inventory, generate order lists automatically, and learn everything there is to know about your clients and, consequently, your company. For proper inventory tracking and client billing, a POS system is required. Additionally, you are collecting consumer email addresses through your EasyPOS system in order to target them in the future.

# Table of Contents

Dedication ..... vi
Acknowledgments ..... vii
Executive Summary ..... viii
Table of Contents ..... ix
Chapter 1 ..... 2
Introduction ..... 2
1.1. Background ..... 3
1.2. Motivations and Challenges ..... 4
1.3. Goals and Objectives ..... 4
1.4. Existing Solutions ..... 4
1.5. Proposed Solution ..... 5
1.6. EasyPOS Project Plan ..... 6
1.6.1. Work Breakdown and Structure ..... 7
1.6.2. Roles & Responsibility Matrix ..... 8
1.6.3. Gantt Chart ..... 8
Chapter 2 ..... 9
Software Requirement Specifications ..... 9
2.1. Introduction ..... 10
2.1.1. Purpose ..... 10
2.1.2. Intended Audience and Reading Suggestions ..... 11
2.1.3. Product Scope ..... 11
2.1.4. References ..... 11
2.2. Overall Description ..... 12
2.2.1. Product Perspective ..... 12
2.2.2. User Classes and Characteristics ..... 12
2.2.3. Operating Environment ..... 13
2.2.4. Design and Implementation Constraints ..... 13
2.2.5. User Documentation ..... 13
2.2.6. Assumptions and Dependencies ..... 14
2.3. External Interface Requirements ..... 14
2.3.1. Hardware Interfaces ..... 14
2.3.2. Software Interfaces ..... 14
2.3.3. Communications Interfaces ..... 14
2.4. System Features ..... 14
2.4.1. System Feature 1 ..... 17
2.4.1.1. Description and Priority ..... 17
2.4.1.2. Stimulus/Response Sequences ..... 18
..... 18
2.4.1.3. Functional Requirements ..... 18
2.4.2. System Feature 2 ..... 18
2.4.2.1. Description and Priority ..... 18
2.4.2.2. Stimulus/Response Sequences ..... 19
2.4.2.3. Functional Requirements ..... 19
2.5. Other Nonfunctional Requirements ..... 20

2.5.1. Performance Requirements ..... 20
2.5.2. Safety Requirements ..... 20
2.5.3. Security Requirements ..... 21
2.5.4. Software Quality Attributes ..... 21
2.5.5. Business Rules ..... 21
2.6. Other Requirements ..... 22
Chapter 3 ..... 23
Use Case Analysis ..... 23
3.1. Use Case Model ..... 25
Figure 3. 1:Use Case Model ..... 25
3.2. Use Case Descriptions ..... 26
3.3. Use Case Descriptions ..... 28
Chapter 4 ..... 31
System Design ..... 31
4.1. Architecture Diagram ..... 32
4.1. Domain Model ..... 33
4.2. Entity Relationship Diagram with data dictionary ..... 34
4.3. Class Diagram ..... 35
4.4. Sequence / Collaboration Diagram ..... 36
4.5. Activity Diagram ..... 37
4.6. State Transition Diagram ..... 40
Chapter 5 ..... 42
Implementation ..... 42
5.1. Components, Libraries, Web Services and stubs ..... 43
5.2. Deployment Environment ..... 44
5.3. Tools and Techniques ..... 44
5.4. Best Practices / Coding Standards ..... 45
5.5. Version Control ..... 45
Chapter 6 ..... 46
Testing and Evaluation ..... 46
6.1. Use Case Testing ..... 47
6.2. Equivalence partitioning ..... 48
6.3. Boundary value analysis ..... 48
6.4. Data flow testing ..... 48
6.5. Unit testing ..... 48
6.6. Integration testing ..... 49
6.7. Performance testing ..... 49
6.8. Stress Testing ..... 49
Chapter 7 ..... 50
Summary, Conclusion and Future Enhancements ..... 50
7.1. Project Summary ..... 51
7.2. Achievements and Improvements ..... 51
7.3. Lessons Learnt ..... 52
7.4. Future Enhancements/Recommendations ..... 52
Reference and Bibliography ..... 53

# List of Figures

Figure 1. 1: Project Plan ..... 6
Figure 1. 2: Gantt Chart ..... 8
Figure 2. 1:Manager Response Sequence ..... 18
Figure 2. 2:Cashier Response Sequences ..... 19
Figure 3. 1:Use Case Model ..... 25
Figure 3. 2: Use case Cashier Panel ..... 26
Figure 3. 3:Use case Customer Panel ..... 27
Figure 3. 4:Use case Manager Panel ..... 27
Figure 3. 5:Use case Bank Panel ..... 28
Figure 4. 1: Architecture Diagram ..... 32
Figure 4. 2: Domain Model ..... 33
Figure 4. 3: ERD ..... 34
Figure 4. 4: Class Diagram ..... 35
Figure 4. 5: Sequence Diagram ..... 36
Figure 4. 6: Activity Diagram Admin Side ..... 37
Figure 4. 7:Activity Diagram Staff Side ..... 38
Figure 4. 8:Activity Diagram Manager Side ..... 39
Figure 4. 9:State Transition Diagram Admin Side ..... 40
Figure 4. 10:State Transition Diagram Manager Side ..... 41
Figure 4. 11:State Transition Diagram Staff Side ..... 41

# Chapter 1

# Introduction

# Chapter 1: Introduction

Your consumer makes payments for goods or services at your business using EasyPOS software helps businesses like *the* restaurants and stores integrate mobile POS features, contactless payment methods, e-commerce integration of different capabilities, and *much* more in addition to processing credit cards. Every business owner wants to know which products are selling the fastest, which clients are making the most purchases, how much inventory is on hand, what forms of payment customers prefer (cash, credit, check, or debit), and how much money has been made. sell on a particular day. A point-of-sale system is made to give the business owner access to this data. The biggest retailer has a unique point-of-sale setup. The (Walmart)Store is aware of the exact amount of sales made on any given day and at any given hour of the day. With the use of this data, they are able to accurately forecast the number of cashiers they will require each day of the week. Consider for a moment how it lowers labour expenses since they can open the appropriate number of registers and avoid hiring an excessive number of cashiers. Additionally, by having the right number of cashiers on duty at busy times, customers won't leave the shop because the lines are too lengthy.

## 1.1. Background

In past years, it is very hard to manage the shop inventory and sales for the store managers and owners. They use the old hand written system page register for the sales and inventory. In past years, it is very hard to manage the shop inventory and sales for the store managers and owners. They use the old hand written system page register for the sales and inventory. Cash registers were the first point-of-sale devices, developed in 1879 by Ohio saloon owner James Ritty. The cash register made it possible for users to accurately record transactions, improving bookkeeping and capital management. Ritty eventually sold the National Cash Register Corporation (NCR) his idea in 1884.

IBM has introduced in the term Point of sale by inventing first point of sale system to handle business for the businessperson.

The modern point of sale system allows managing

- Inventory
- Staff
- Sales
- Miscellaneous
- Product Batching
- Customer Management
- Employee Management

## 1.2. Motivations and Challenges

- The insecure point of sale system is the biggest challenge for the owners
- The Cyberattacks on the system is very harmful for the system
- The database of the system can be easily hacked by the sales staff Motivations
- The biggest motivation is the inventory management for the store through Point of sale system
- The customer data is collected through point of sale by collecting their card information and cell number

## 1.3. Goals and Objectives

- Customer can be easily managed
- Customer data can be saved
- Inventory record easily maintained
- Sales record
- Profit/loss record

## 1.4. Existing Solutions

Currently the point of sale software's are not well efficient and well managed in term of working and keep record of inventory sales and sales staff.

• POS system for desktop s.

• POS systems for tablets and mobile devices.

• POS systems for self-service kiosks.

## 1.5. Proposed Solution

We proposed a solution is EasyPOS system. It is a development over the cash register, manual inventory control, and traditional selling methods. But it is also a lot more than that. With our cutting-edge POS software, you can keep track of your inventory, generate order lists automatically, and learn everything there is to know about your clients and, consequently, your company. For proper inventory tracking and client billing, a POS system is required. Additionally, you are collecting consumer email addresses through your EasyPOS system in order to target them in the future. Having a POS system in place makes it simple to access precise data on product sales and inventory levels. With the click of a mouse or the push of a button, you can receive comprehensive information on any product you offer, including daily, weekly, and monthly sales volumes, present inventory levels, order status, and profit margins. You are can also view this data in the form of charts that make it easier for you to understand the data. We create a POS that provide you different languages and this POS software is use in different fields like Super stores, Grocery stores and Pharmacy stores etc.

## 1.6. EasyPOS Project Plan

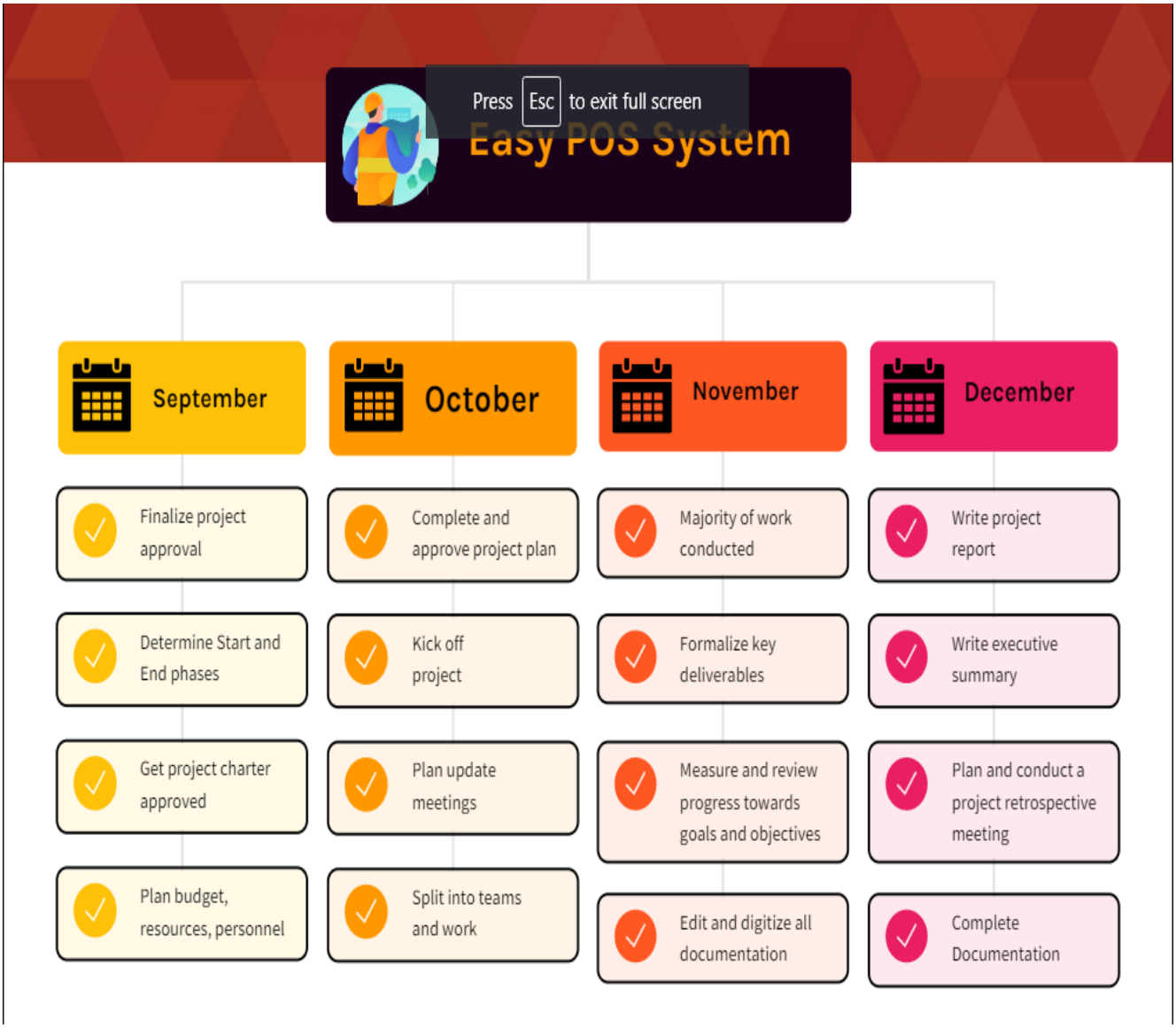



**Figure 1. 1: Project Plan**

### 1.6.1. Work Breakdown and Structure

| WBS # | WBS Deliverable | Activity # | Activity to Complete the Deliverable | Duration (# of Days) | Responsible Team Member(s) & Role(s) |
|---|---|---|---|---|---|
| 01 | **Requirement gathering** | | | 7 | All of us |
| 02 | System Design flow | | | 10 | Muhammad Ali, Nauman Shabbir |
| 03 | Database design | | | 7 | Muhammad Ali, Nauman Shabbir |
| 04 | Database implementation | | | 5 | Muhammad Ali, Nauman Shabbir |
| 05 | **GUI design** | | | 10 | Muhammad Ali, Nauman Shabbir |
| 06 | GUI & Database Integration | | | 5 | Muhammad Ali, Nauman Shabbir |
| 07 | **Coding** | | | 15 | Muhammad Ali, Nauman Shabbir |
| 08 | Bug Fixing | | | 3 | Muhammad Ali, Nauman Shabbir |
| 09 | **Final Testing** | | | 4 | Muhammad Ali, Nauman Shabbir |

### 1.6.2. Roles & Responsibility Matrix

The role and responsibility matrix is used to decide who will be responsible for what.

### 1.6.3. Gantt Chart

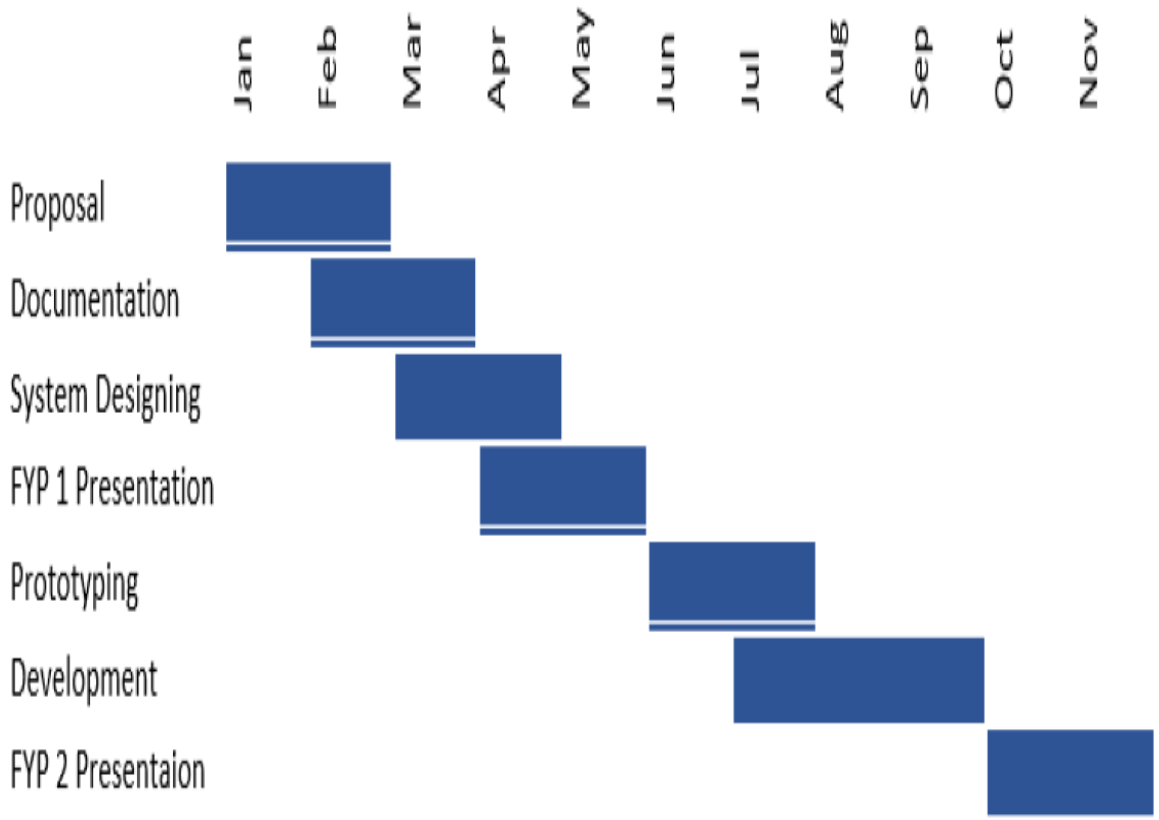



**Figure 1. 2: Gantt Chart**

# Chapter 2

# Software Requirement Specifications

# Chapter 2: Software Requirement Specifications

## 2.1. Introduction

Your consumer makes payments for goods or services at your business using EasyPOS. A point-of-sale system is made to give the business owner access to this data. The biggest retailer has a unique point-of-sale setup. The Store is aware of the exact amount of sales made on any given day and at any given hour of the day. With the use of this data, they are able to accurately forecast the number of cashiers they will require each day of the week. Consider for a moment how it lowers labor expenses since they can open the appropriate number of registers and avoid hiring an excessive number of cashiers. Additionally, by having the right number of cashiers on duty at busy times, customers won't leave the shop because the lines are too lengthy.

### 2.1.1. Purpose

A point of sale system is the computer based software system, which is used to manage the sale, inventory and staff of the store. The point of sale includes many software and hardware components like Computer, barcode reader, scanner, printer, credit card reader. When the customers done with their shopping they came to the sales vendor for billing. The vendor scan the product and generates the bill for the products the customer buy. The customer can pay through cash and credit card. The purpose of the Point of sale system is to keep records and eye on inventory, sales, staff, store profit or loss by storing data in the database of the POS.

### 2.1.2. Intended Audience and Reading Suggestions

The following individuals are expected to have the document completed:

- Project managers(Who manage it)
- Developers(Who develop the Software).
- Testers(Who test the software).
- Marketing personnel.
- Customers.
- Writing Documentation.

**This documentation is arranged into three Different parts.**

1. Introduction.

2. Software Requirements Specifications.

3. Use case Analysis.

Although each section is separate, reading the document in order helps the reader grasp the order.

### 2.1.3. Product Scope

The scope of a EasyPOS is to help the people to manage their store and manage their inventory. This software is used in many businesses. We can change a language of our software easily that's why it is used in different countries. This Projects is planned, designed and prepared as a web based used to provide ease for the Business Owners to manage his business. The system is designed specifically according to business owners and customer needs. The main working in this project is upon payments management and manage inventory for business owner in that way that he will be able to check the daily sale, monthly sale, profit and loss.

### 2.1.4. References

https://futuregenlabs.com/srs-for-pos-system/

## 2.2. Overall Description

Your consumer makes payments for goods or services at your business using EasyPOS. EasyPOS software helps businesses like the restaurants and stores integrate mobile POS features, contactless payment methods, e-commerce integration of different capabilities, and much more in addition to processing credit cards.

### 2.2.1. Product Perspective

The Point of sale system store the following information:

- Inventory Management
- Customer Data
- Sales record
- Employee records
- Batch record

### 2.2.2. User Classes and Characteristics

The sales men who is operating the point of sale system able to store data of their sale in the system. The owner can see the records of daily, monthly and yearly sale. Customer can check the bill, which is generate by the POS system.

Employee functions:

- Add / Edit/ Delete products in bill
- Generate bill
- Keep records
- Accept Cash / Credit card

Customer functions:

- View Bill
- Pay Cash / Credit card
- Get Slip

Administrative functions:

- Check records

- Check employee
- Check inventory
- Check sale of daily, weekly , monthly
- Update / delete record

### 2.2.3. Operating Environment

The point of sale works on Windows operating system. The POS is integrated with:

- Barcode scanner.
- Receipt printer (Thermal Printer).
- Google Location API.
- Email Integration.
- Cash drawer.

### 2.2.4. Design and Implementation Constraints

Admin can add employee, user, or seller to the system. User has unique ID and password for login into system. Admin has access to the system, inventory and employee. Manager has access to check out the whole sales of the store and the product, which is sell in the day, weekly or monthly.

### 2.2.5. User Documentation

The user guide contains:

- System specifications is to run the POS System
- Installation Guide
- User Manual
- Queries
- System Maintenance
- Database Queries

### 2.2.6. Assumptions and Dependencies

Assumptions:

- Point of sale can be easily hacked by anyone
- Point of sale database can be delete easily
- Point of sale server can be destroy.
- Point of sale system can be failed.

## 2.3. External Interface Requirements

### 2.3.1. Hardware Interfaces

The Hardware components of a POS system:

- Barcode scanner.
- Receipt printer (Thermal Printer).
- Computer
- Cash drawer.

### 2.3.2. Software Interfaces

It can install on all windows that come after the Window Xp.

### 2.3.3. Communications Interfaces

The client-server model must guide the communication architecture. A REST-compliant web service should be used for client-server communication, and HTTP Secure must be used (HTTPS). Stateless client-server communication is required. The client and server roles must be separated by a consistent interface.

## 2.4. System Features

1. Membership Management System.
    - Add a new member into the system (Monthly, Yearly, Lifetime).
    - View a list of members.
    - Delete a member from the system.
    - Block a members from the system.

- Update a member from the system.
- Auto-renewal the membership plan system.

2. Users/Cashier Management System.
   - Allow member to add a new user (Admin, sales, inventory users and etc.).
   - View a list of users.
   - Delete a user from the system.
   - Update a user from the system.
   - Block a user from the system.
   - Allow users a permission/rights for access the system.
3. Inventory or Stock Management System.
   - Add a new product manually/barcode into the system.
   - View a list of products.
   - Delete a product from the system.
   - Update a product from the system.
4. Supplier/Vendor Management System.
   - Add/Enter a new supplier into the system.
   - View a list of suppliers.
   - Delete a supplier from the system.
   - Update a supplier from the system.
5. Purchase & Purchase Return Management System.
   - Add/Enter a new purchase into the system.
   - View a list of purchases.
   - Delete a purchase from the system.
   - Update a purchase from the system.
   - Return a purchase from the system.
6. Sale & Sales Return Management System.
   - Add a new sale into the system.
   - View a list of sales.

- Delete a sale from the system.
- Update a sale from the system.
- Return a sales from the system.

7. Expense Management System.
   - Add a new expense into the system.
   - View a list of expenses.
   - Delete an expense from the system.
   - Update an expense from the system.
8. Payment Method Management System.
   - Allow cashier to choose a payment method during the sales (Direct cash or Card payment).
9. Discount & Offers Management System.
   - Allow cashier to add a discount during the sales.
   - Allow admin to add offers for the product.
10. Location Management System.
    - Add a new country/city into the system.
    - View a list of countries/cities.
    - Delete a country/city from the system.
    - Update a country/city from the system.
11. Branch Management System.
    - Add a new branch into the system.
    - View a list of branches.
    - Delete a branch from the system.
    - Update a branch from the system.
12. Language Management System.
    - Members should be able to switch language in both admin/sales dashboard.
13. Categories Management System.
    - Add a new category into the system.
    - View a list of categories.

- Delete a category from the system.
- Update a category from the system.

14. Bulk Import Management System.
    - Allow users to add a list of product through excel import.
    - Allow users to export all products from the system.
15. Email Management System.
    - Verify a member via email.
    - Email the admin when new user, seller, category, product, branch, country, city, expense added into the system.
    - 15 days auto-renewal membership email notification.
    - Notify the admin for the login alerts.

### 2.4.1. System Feature 1

**Inventory/ System for managing stocks**

#### 2.4.1.1. Description and Priority

Admin manage the inventory by add/update/delete the inventory items.

#### 2.4.1.2. Stimulus/Response Sequences

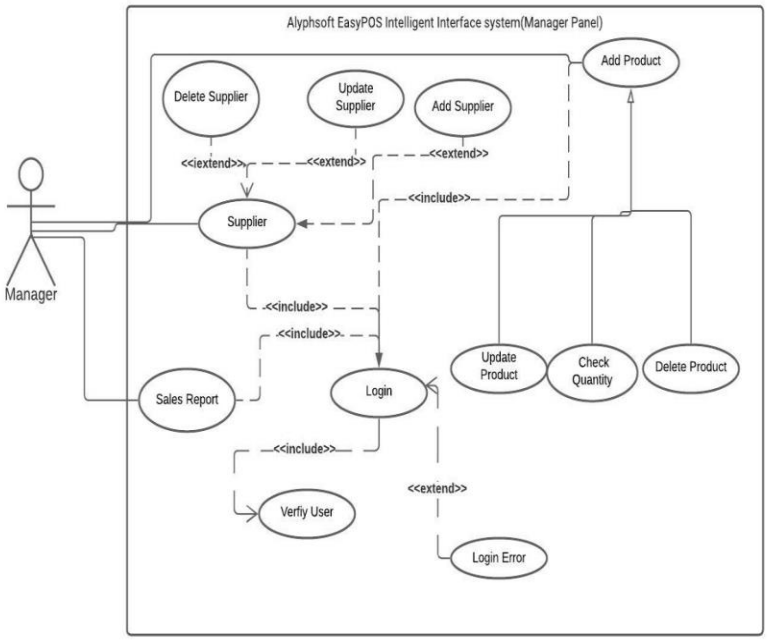



**Figure 2. 1:Manager Response Sequence**

#### 2.4.1.3. Functional Requirements

- Product Name
- Product id
- Product brand
- Price
- Quantity

### 2.4.2. System Feature 2

**User/mechanism for managing cash receipts**

#### 2.4.2.1. Description and Priority

User can be add into the system the admin or manager. User can be delete or user status can be update.

### 2.4.2.2. Stimulus/Response Sequences

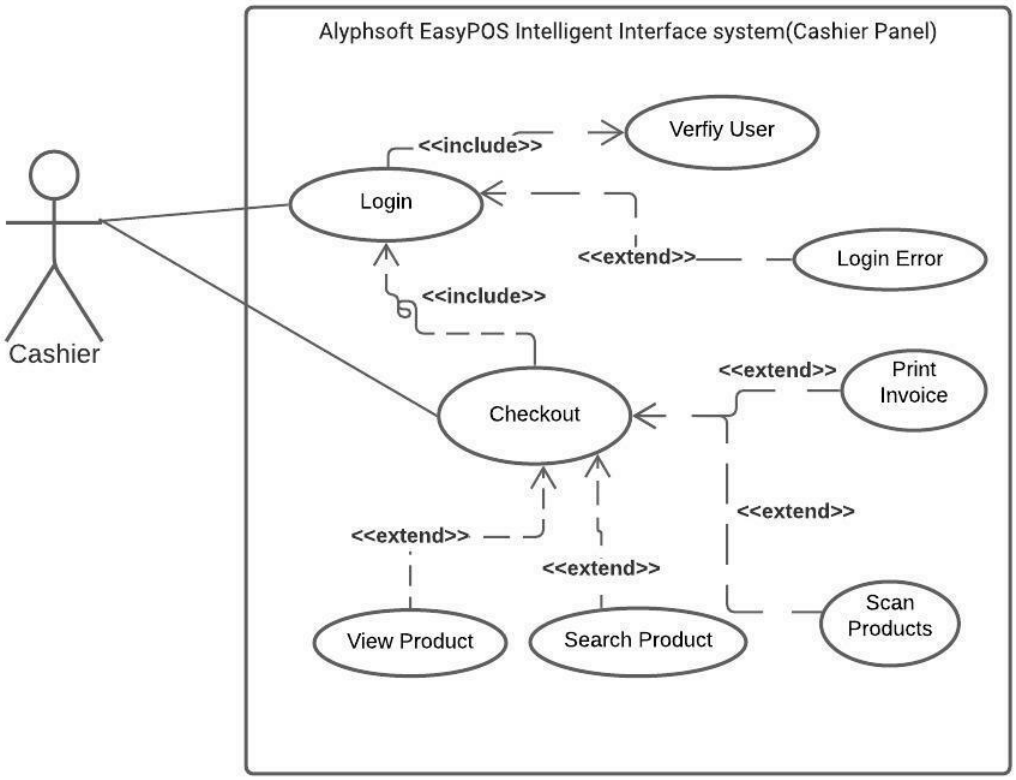



**Figure 2. 2:Cashier Response Sequences**

### 2.4.2.3. Functional Requirements

- User Name
- Email
- Address
- Password
- CNIC
- Contact No

## 2.5. Other Nonfunctional Requirements

- Performance
- Scalability
- Responsiveness
- Use-ability
- Reliability
- Security
- Documentation

### 2.5.1. Performance Requirements

The system has to be participatory and require fewer delays. There are no immediate delays in any of the system's action-response processes. There is a delay of less than 3 seconds when the Windows user interface is opened, when an error message is displayed, or when settings or sessions are saved. However, there are no delays while opening databases, sorting distinct queries, or doing assessments; for more than 95% of the files, the procedure is finished in under 3 seconds. There is a good probability that a successful connection will be created in under 20 seconds for the sake of clear communication because the time it takes to connect to the server relies on its setup and the distance between the two systems.

### 2.5.2. Safety Requirements

- Data can be safely store to the database without any loose in the data.
- Install antivirus software
- Encryption
- Network Security
- Enable to factor authentication
- Keep software updated

### 2.5.3. Security Requirements

The user account is the key area of security concerned, hence an appropriate login procedure should be used to guard against hacking. A spam prevention strategy to improve security is tablet ID registration. Thus, protection against untheorized usage of the POS software is provided.

- EMV
- Tokenization
- Point-to-point encryption

### 2.5.4. Software Quality Attributes

- Availability
- Correctness
- Maintainability
- Reliability
- Testability
- Usability

### 2.5.5. Business Rules

To access and interact with the system, at least the Super-User and User roles are required. If the business requirements for the administrator and user roles are satisfied, additional roles can be developed for the system. The Superuser role must, at the very least, correspond to the Data Explorer, Data Curator, and Administrator accounts. At the very least, the user role needs to match the data explorer account. The administrator and user roles are subject to the ensuing business regulations.

**Admin Business Rules**

- Admin/001 upholds all Database and all the system setups on the back end
- Admin/002 upholds all inventory
- Admin/003 Maintains all employee and user accounts
- Admin/004 Maintains all queries about POS

**Employee Role Business Rules**

Employee Role/001 connecting to, obtaining, and reading data from system and also maintained the system database.

## 2.6. Other Requirements

- Visual Studio
- Laravel
- Bootstrap
- GitHub
- AI
- SQL
- PHP

➢ AI based user interface

# Chapter 3

# Use Case Analysis

# **Chapter 3:** System Analysis

Use case diagrams are helpful in demonstrating how design choices and development priorities are influenced by the functional requirements of a system. Additionally, they aid in detecting any internal or external variables that may have an impact on the system and should be considered. The use case diagram need to be straightforward and have just a few shapes. Use cases establish expected behavior (what), not the precise procedure (how). Use cases can be labelled with both a textual and a visual representation after being specified.

**Use Case Model**

Cashier first login then go to main page after verification.It can Add products, Search Products, Print Invoice, scan products and payment. Manager add staff, supplier and products. Customer received the invoice and make payment for it.

## 3.1. Use Case Model

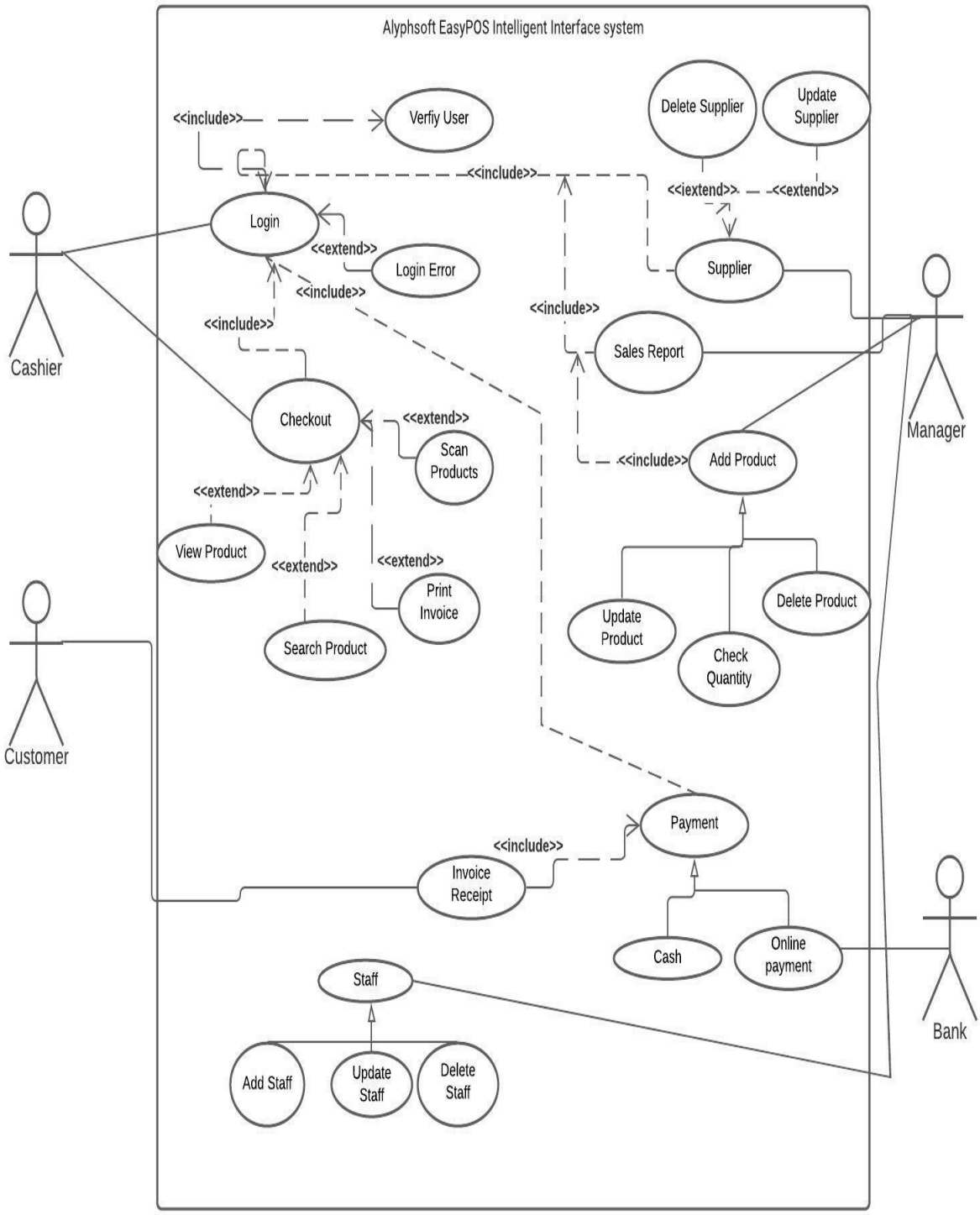



**Figure 3. 1:Use Case Model**

## 3.2. Use Case Descriptions

### Cashier Panel:

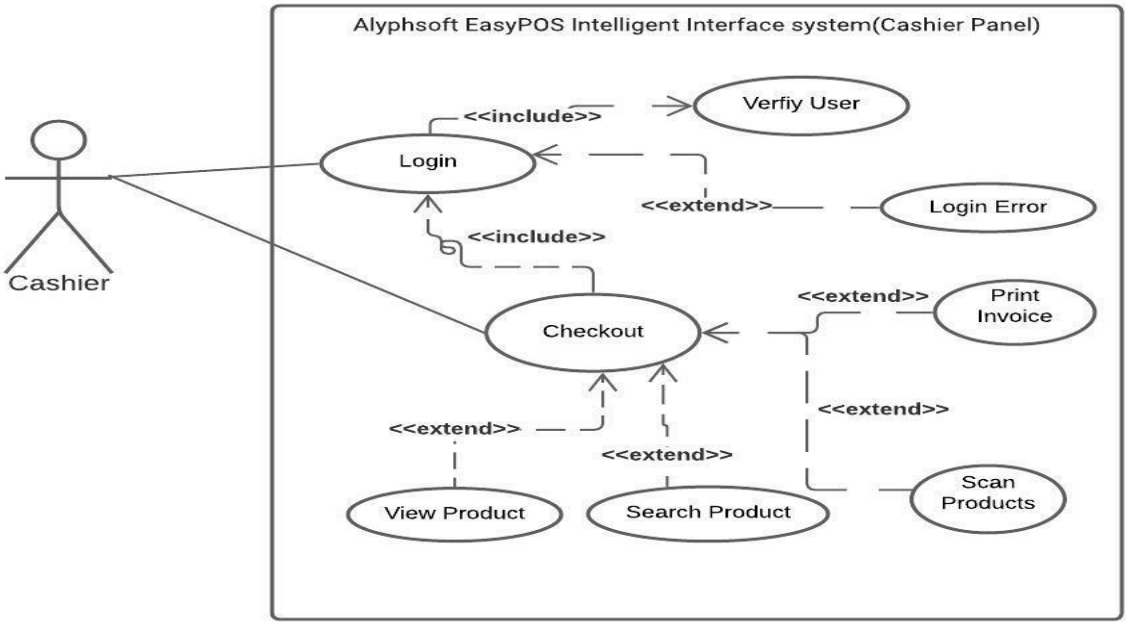



**Figure 3. 2: Use case Cashier Panel**

### Customer Panel:

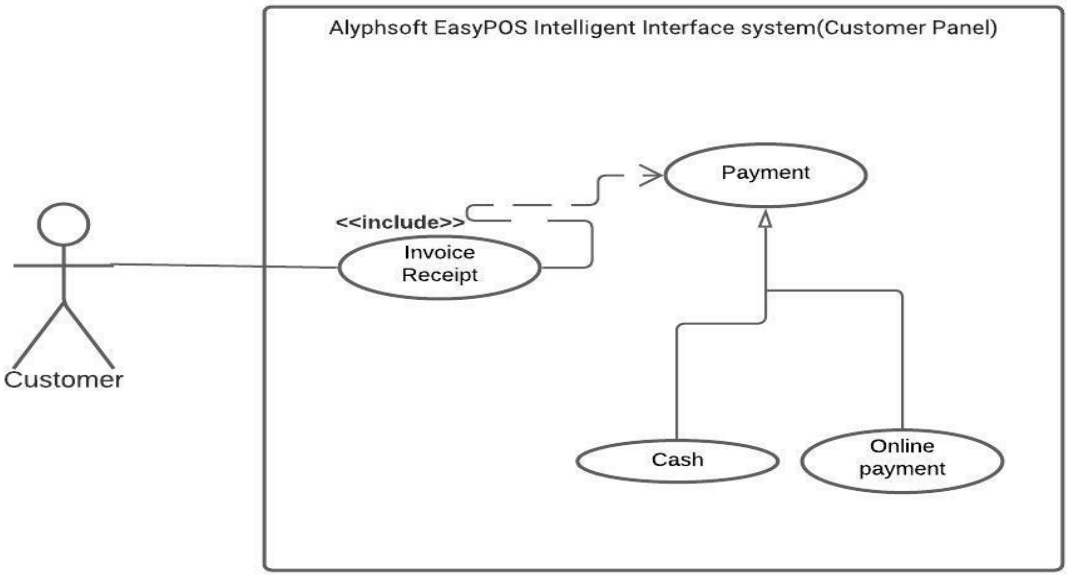



**Figure 3. 3:Use case Customer Panel**

## Manager Panel:

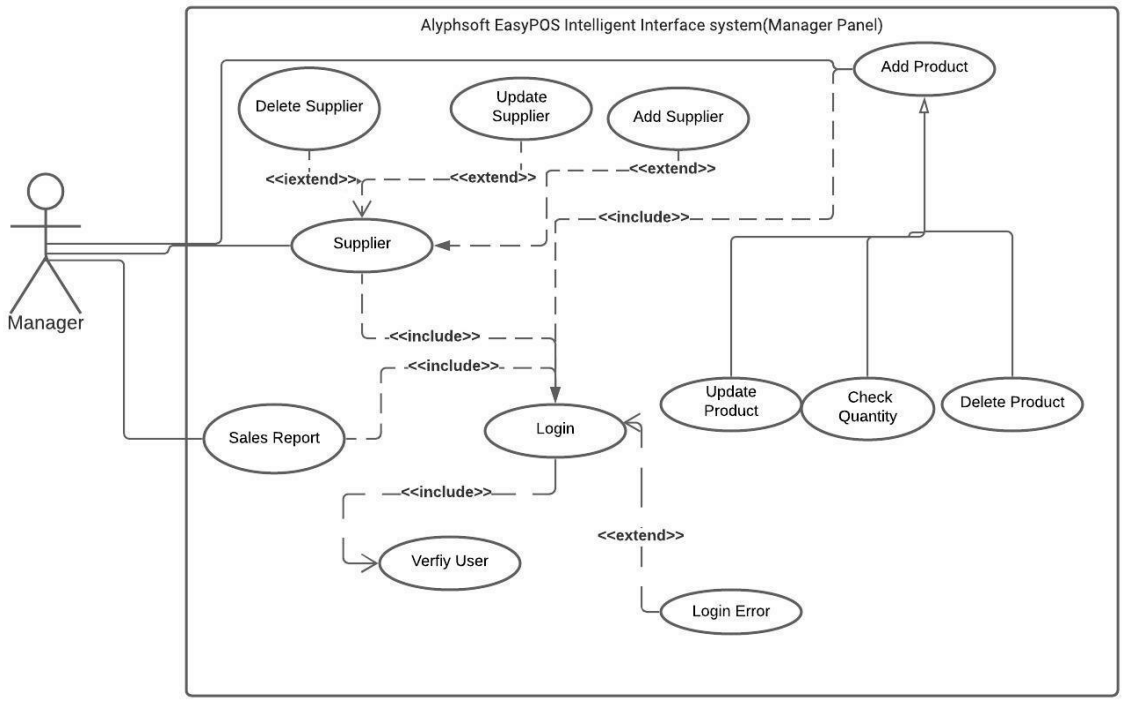



**Figure 3. 4:Use case Manager Panel**

**Bank Panel:**

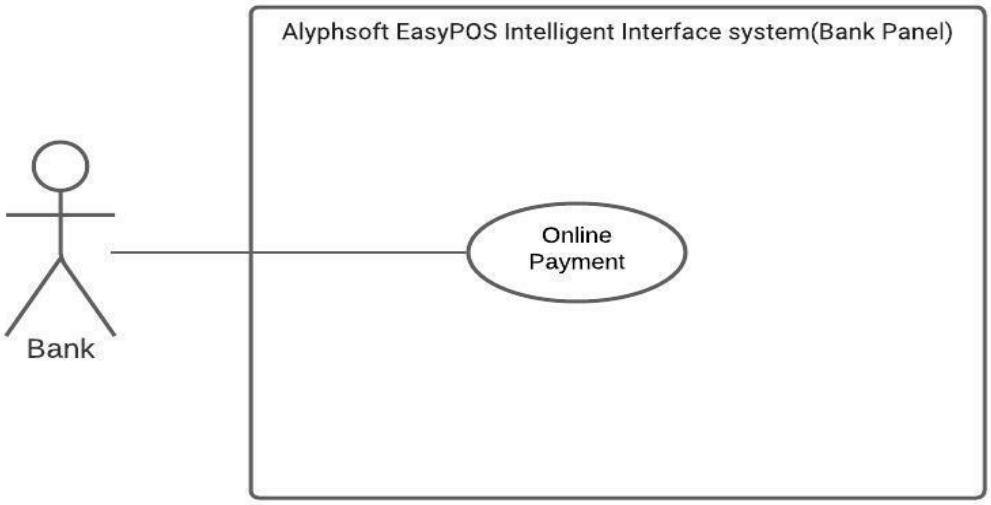



**Figure 3. 5:Use case Bank Panel**

## 3.3. Use Case Descriptions

**The Login Use Case Description:**

| | |
|---|---|
| Use case Name | Login |
| Scope | Log in for product selling |
| Primary Actors | Cashier |
| Stakeholders and Interests | Cashier: Cashier uses it to Login on the POS System. |
| Pre-Conditions | Cashier must have account on this site. |
| Post-Conditions | After Login Cashier do their own tasks. |
| Main Scenario | ➢ Cashier opens POS website.<br>➢ Cashier enter their login info. |

| | | |
|---|---|---|
| | ➢ Then Cashier Successfully login on website.<br>➢ | |
| Failure Case/ Alternative Case | Failure<br>In failure case it would be due to internet problem or Cashier cannot register on this website. | Alternative Case<br>Change your info that match into the system. |
| Frequency of Occurrence | Cashier can Login themselves Infinite times onto the POS Website. | |
| | | |

## Checkout Use Case Description:

| Use case Name | Checkout | |
|---|---|---|
| Scope | It is used to View products, Search products, Print Invoice, in website. | |
| Primary Actors | Cashier | |
| Stakeholders and Interests | Cashier: Cashier uses it to view Products, Search products, Print Invoice on the site. | |
| Pre-Conditions | Login must | |
| Post-Conditions | After Login Cashier do their own tasks. | |
| Main Scenario | ➢ Cashier opens POS website.<br>➢ Then Cashier can view products on website. | |
| Failure Case/ Alternative Case | Failure<br>In failure case it would be due to internet problem or website server down. | Alternative Case<br>Try again when internet is available. |
| Frequency of Occurrence | Cashier can view products, Search products, Print Invoice Infinite times on this website. | |

## Add New Staff Use Case Description:

| | | |
|---|---|---|
| Use case Name | Add new staff | |
| Scope | It is used to Add new staff. | |
| Primary Actors | Cashier | |
| Stakeholders and Interests | Manager: Manager uses it to add new staff. To maintain their business. | |
| Pre-Conditions | Manager must login. | |
| Post-Conditions | After adding the new staff Manager can delete and update staff. | |
| Main Scenario | ➢ Manager opens POS website.<br>➢ Manager login.<br>➢ Manager add new staff. | |
| Failure Case/ Alternative Case | Failure<br>In failure case it would be due to internet problem or Manager cannot successfully login. | Alternative Case<br>Try again with right credentials. |
| Frequency of Occurrence | Manager can add staff how much he want to added. | |

# Chapter 4

# System Design

# Chapter 4:

# System Design

The System Design give us the complete overview of our software. You can use different Diagrams to represent the data.

### Architecture Diagram

Architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. First we login to the system then perform POS tasks. Data store in database MySQL.

## 4.1. Architecture Diagram

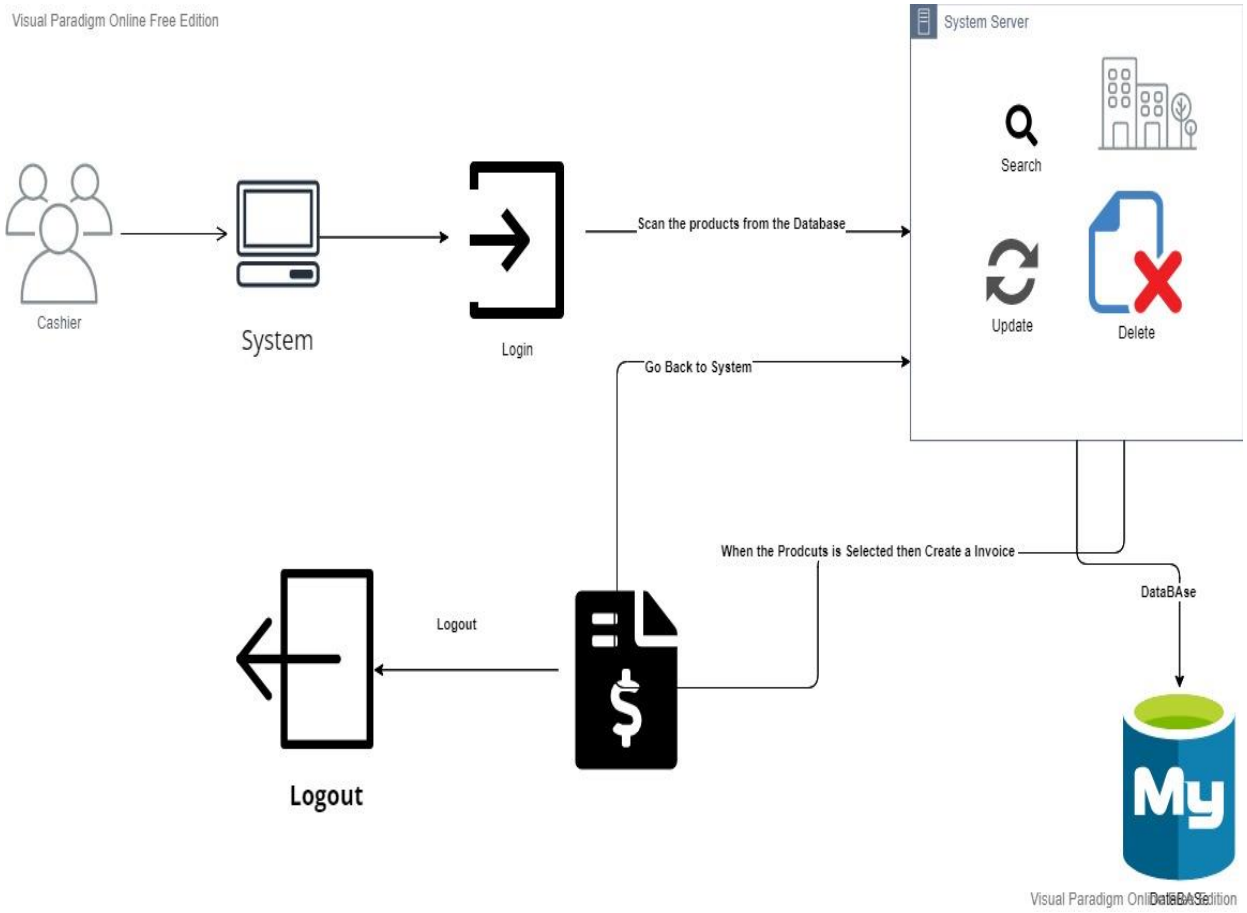



**Figure 4. 1: Architecture Diagram**

## 4.1. Domain Model

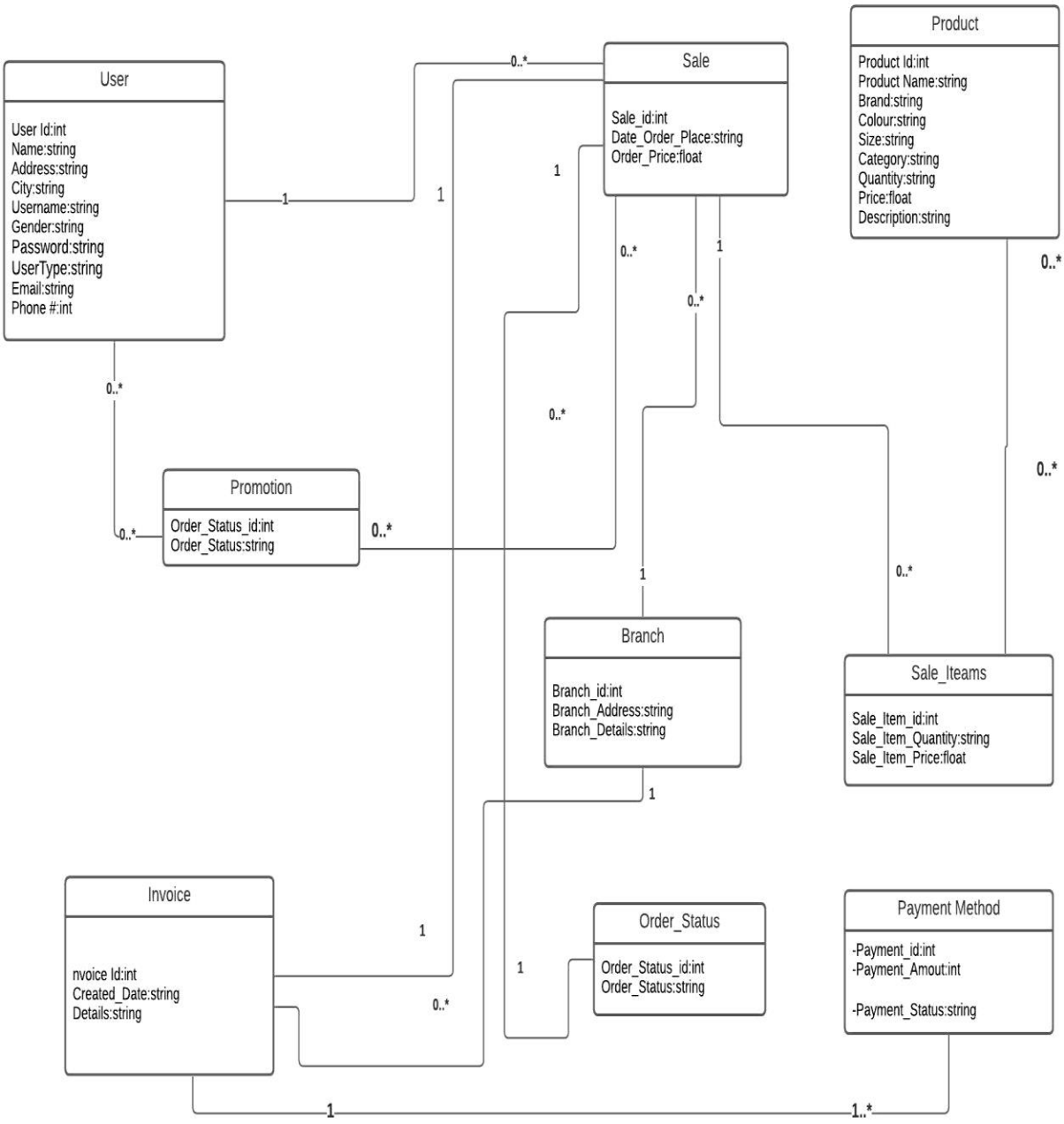



**Figure 4. 2: Domain Model**

## 4.2. Entity Relationship Diagram with data dictionary

This graphical representation depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. ERD explain our complete POS system.

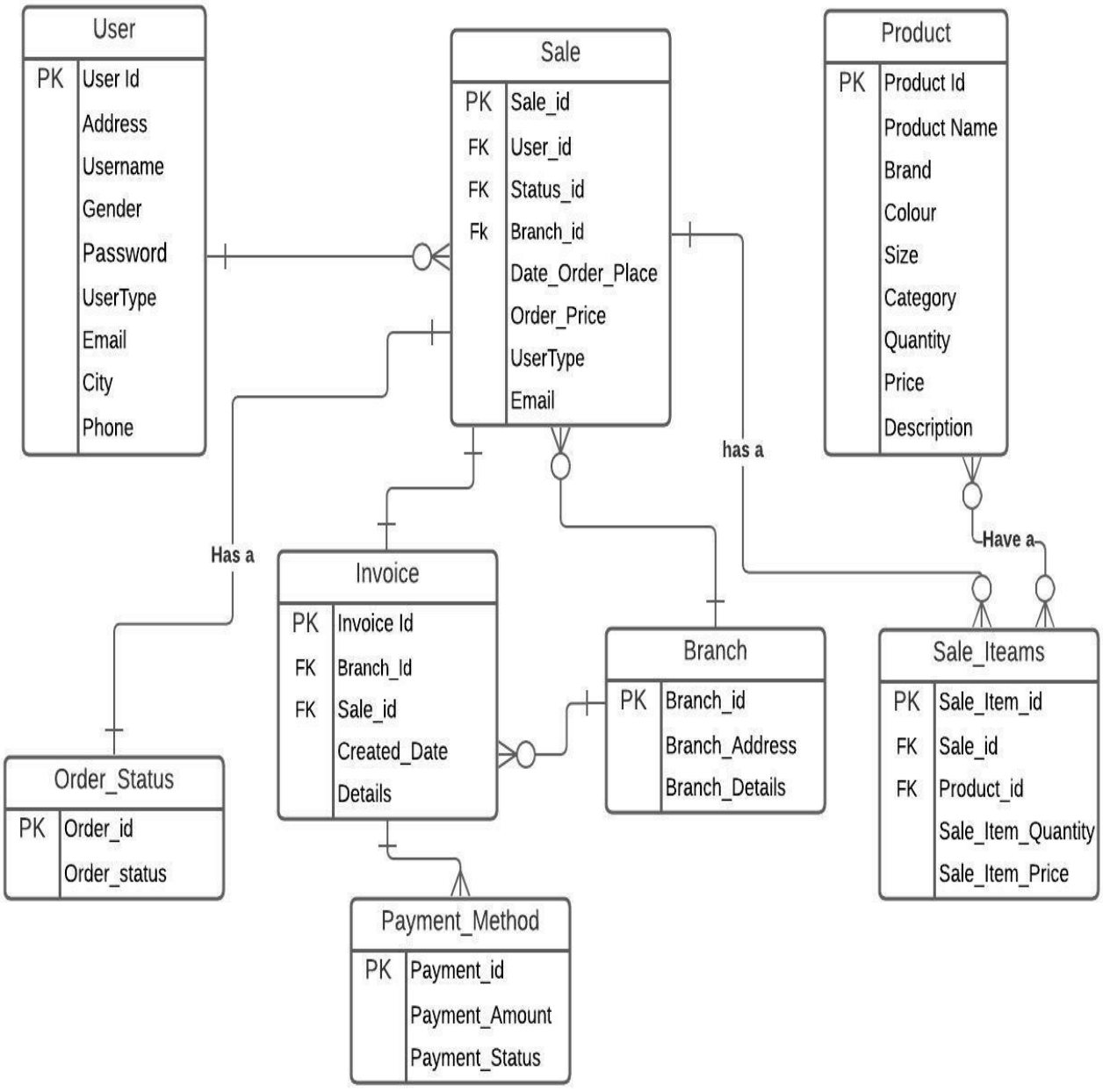



**Figure 4. 3: ERD**

## 4.3. Class Diagram

There are 11 classes and connected with different relations one to many, many to many, one to one etc then explain the attributes of each class as shown in **Figure 4.4**.

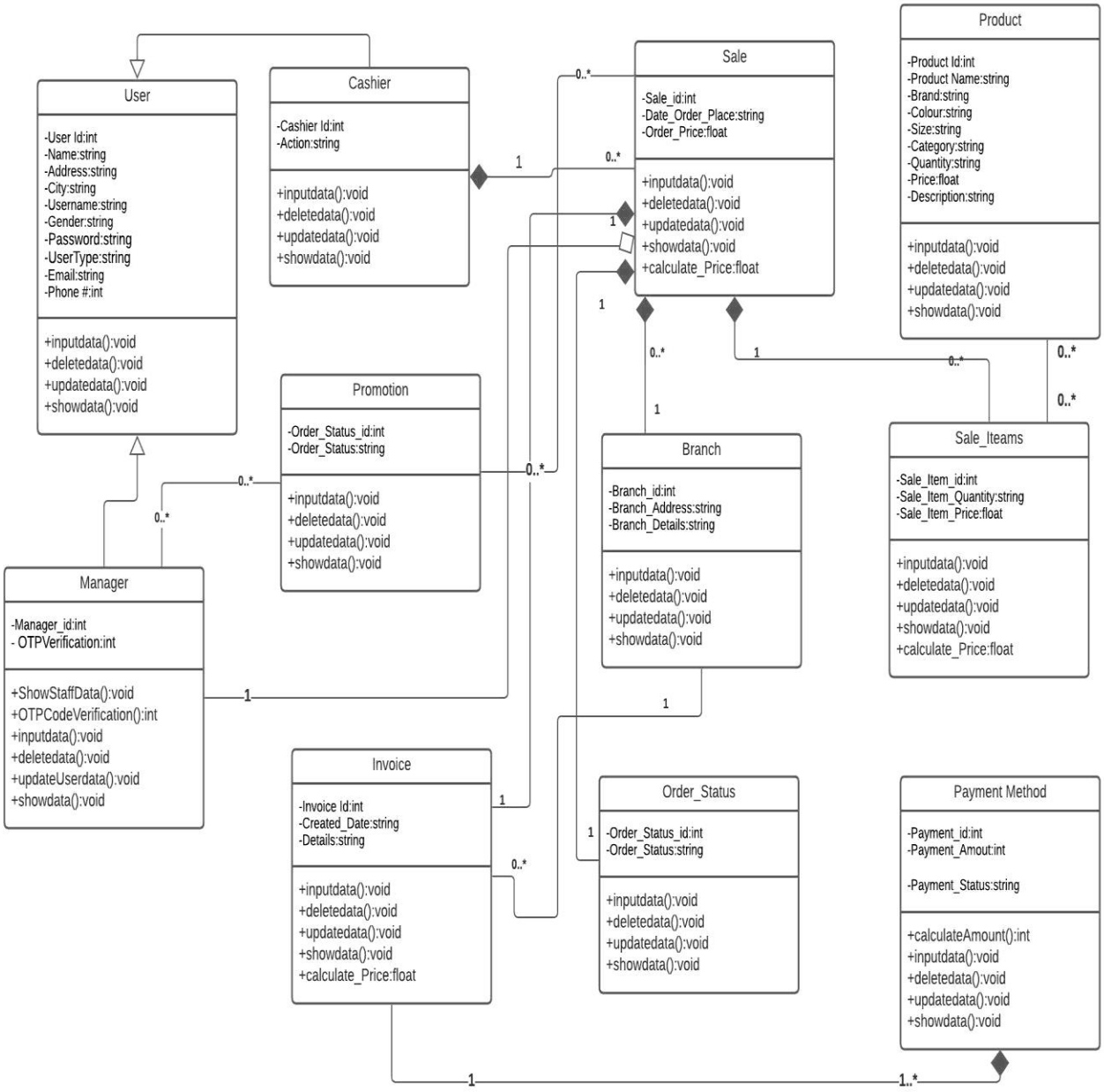



**Figure 4. 4: Class Diagram**

## 4.4. Sequence / Collaboration Diagram

This sequence diagram is a Unified Modeling Language diagram that illustrates the sequence of messages between objects in an interaction. It consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

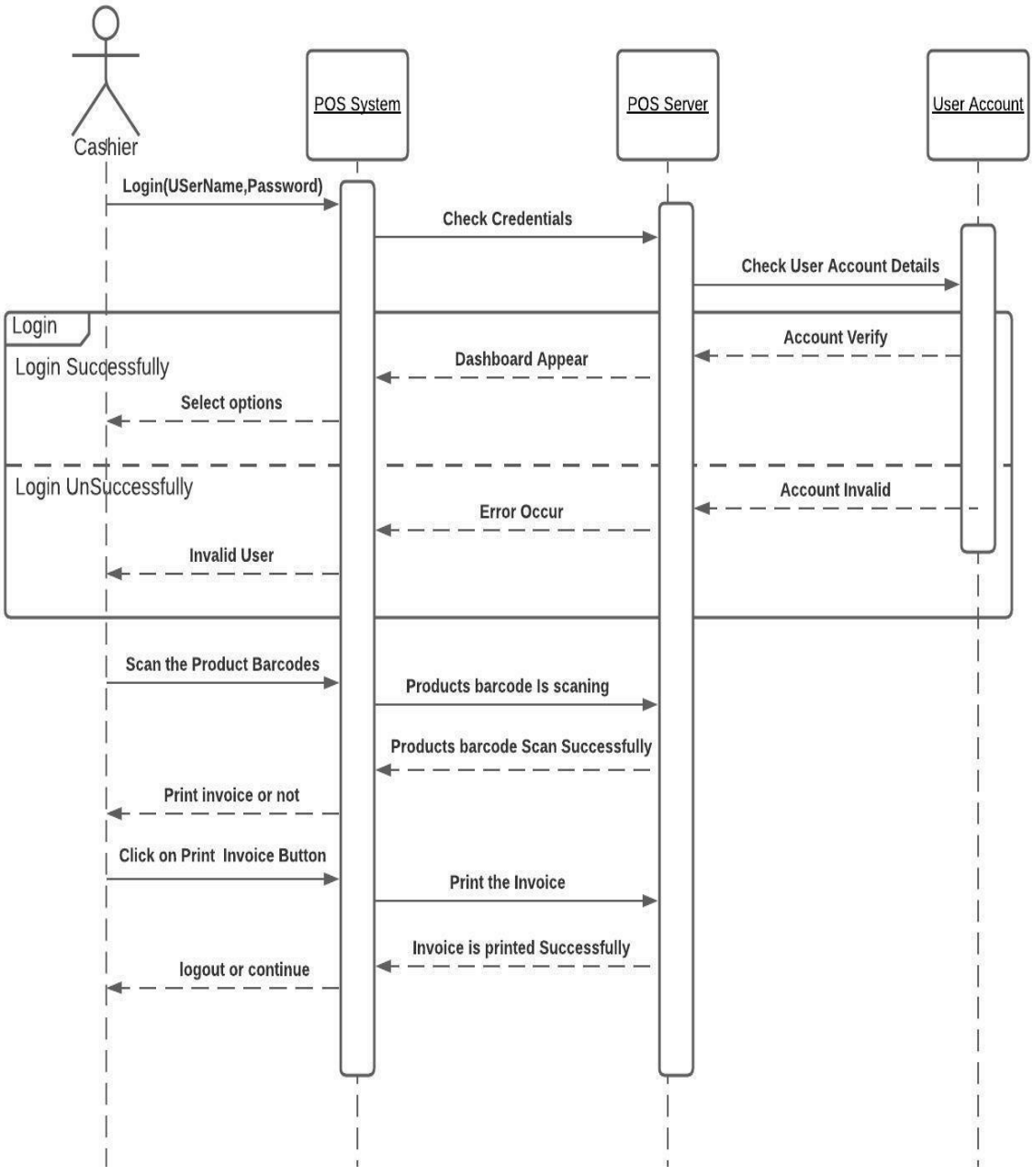



**Figure 4. 5: Sequence Diagram**

## 4.5. Activity Diagram

If the User is Login Successfully then it performs pos tasks. See in the activity diagram to understand that which kind of task user perform in it.

**Admin Side:**

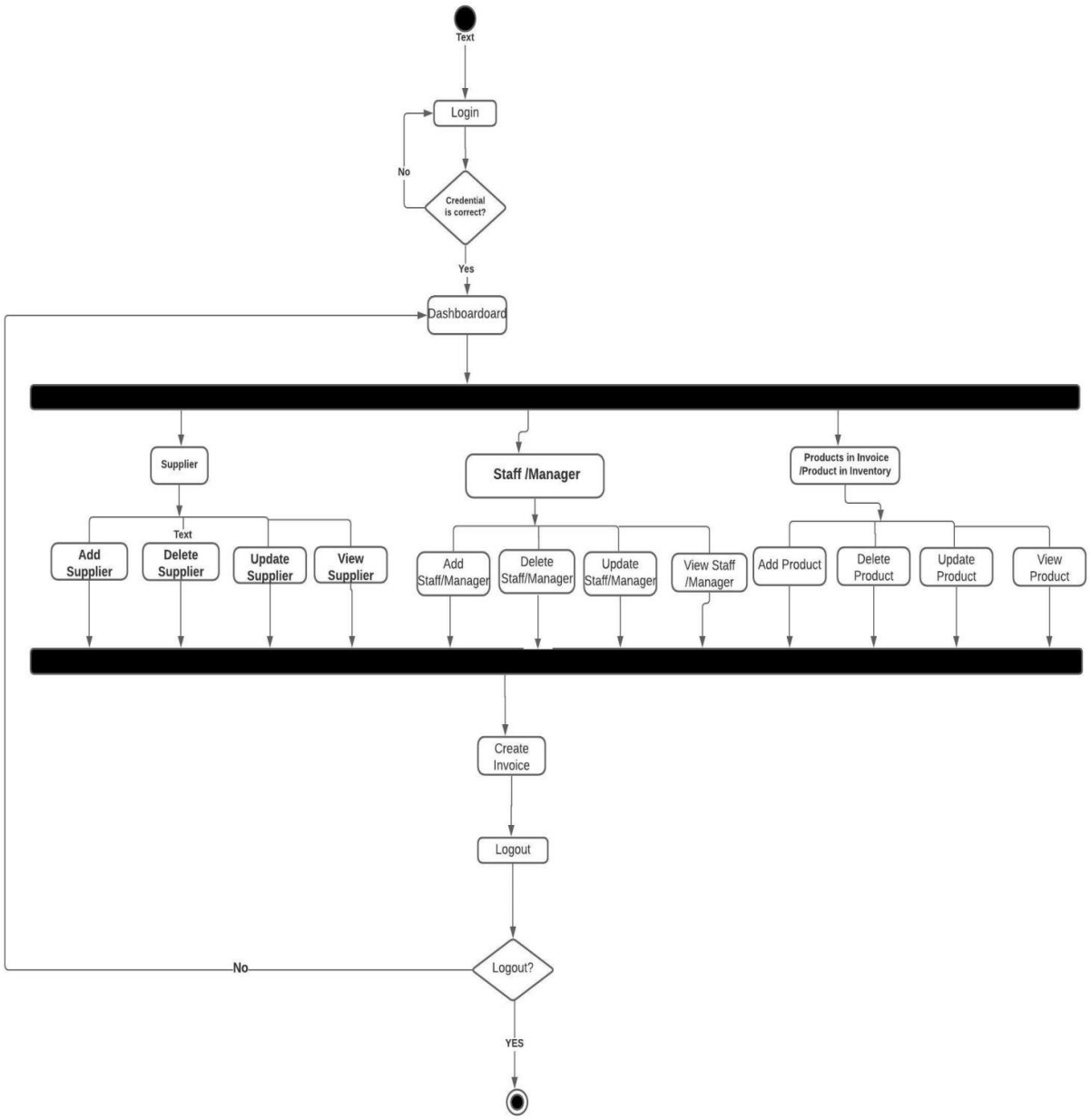



**Figure 4. 6: Activity Diagram Admin Side**

**Staff Side:**

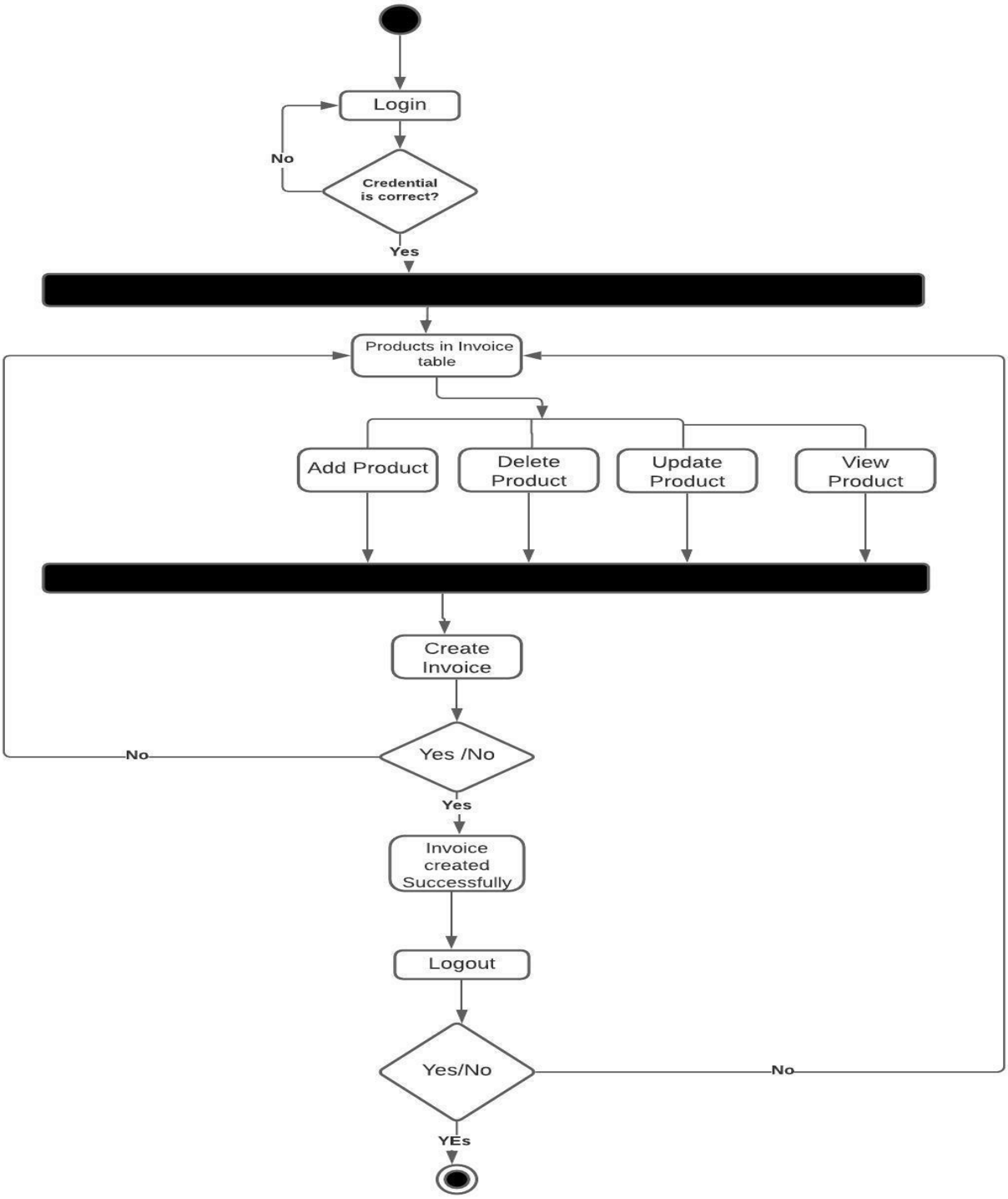



**Figure 4. 7:Activity Diagram Staff Side**

**Manager Side:**

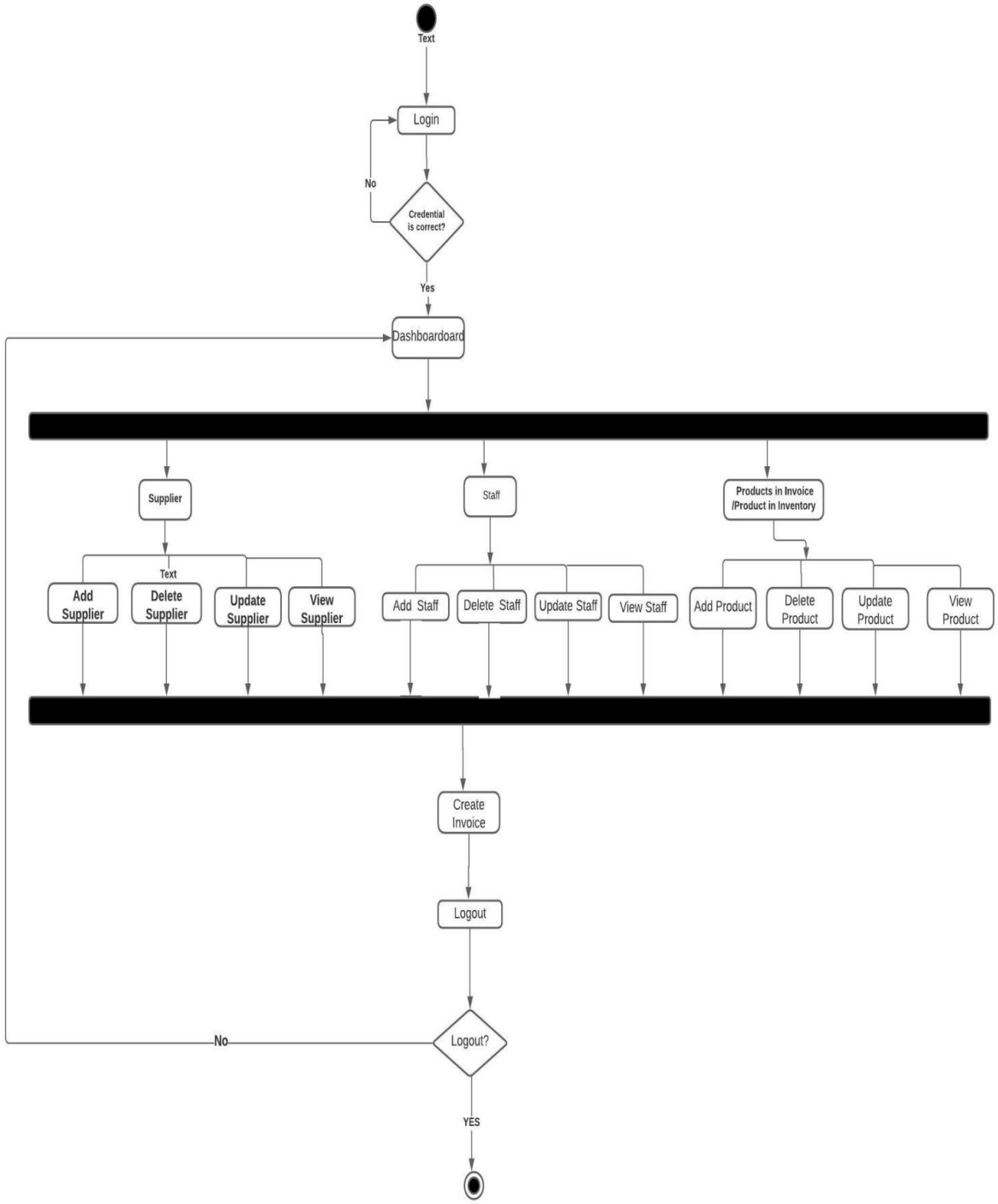



**Figure 4. 8:Activity Diagram Manager Side**

## 4.6. State Transition Diagram

EasyPOS can be a great way to control a inventory and this diagram can help you illustrate how they usually work. Arrows point to the next step which you can easily follow or repeat a step if that is necessary.

**Admin Side:**

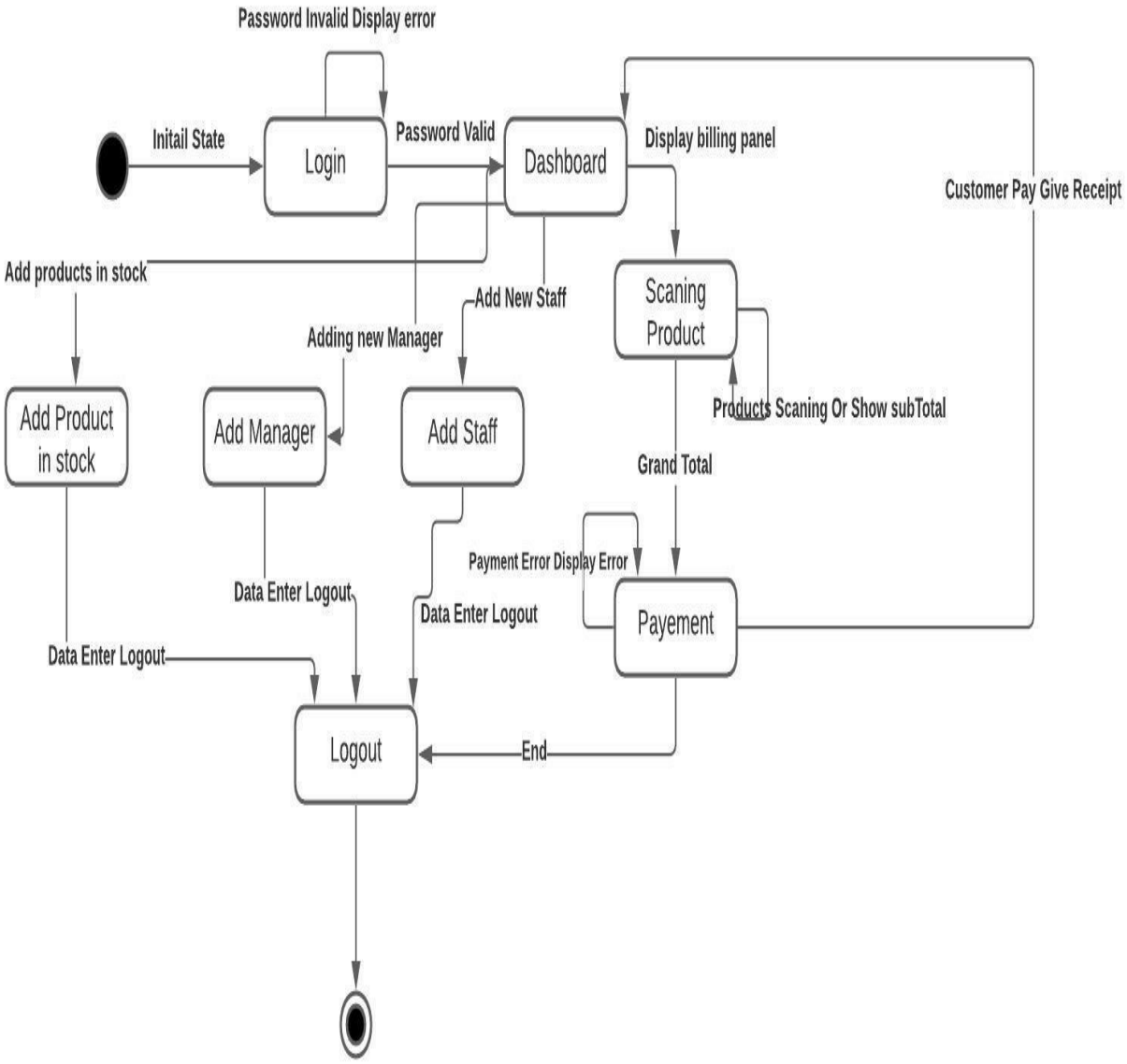



**Figure 4. 9:State Transition Diagram Admin Side**

## Manager Side:

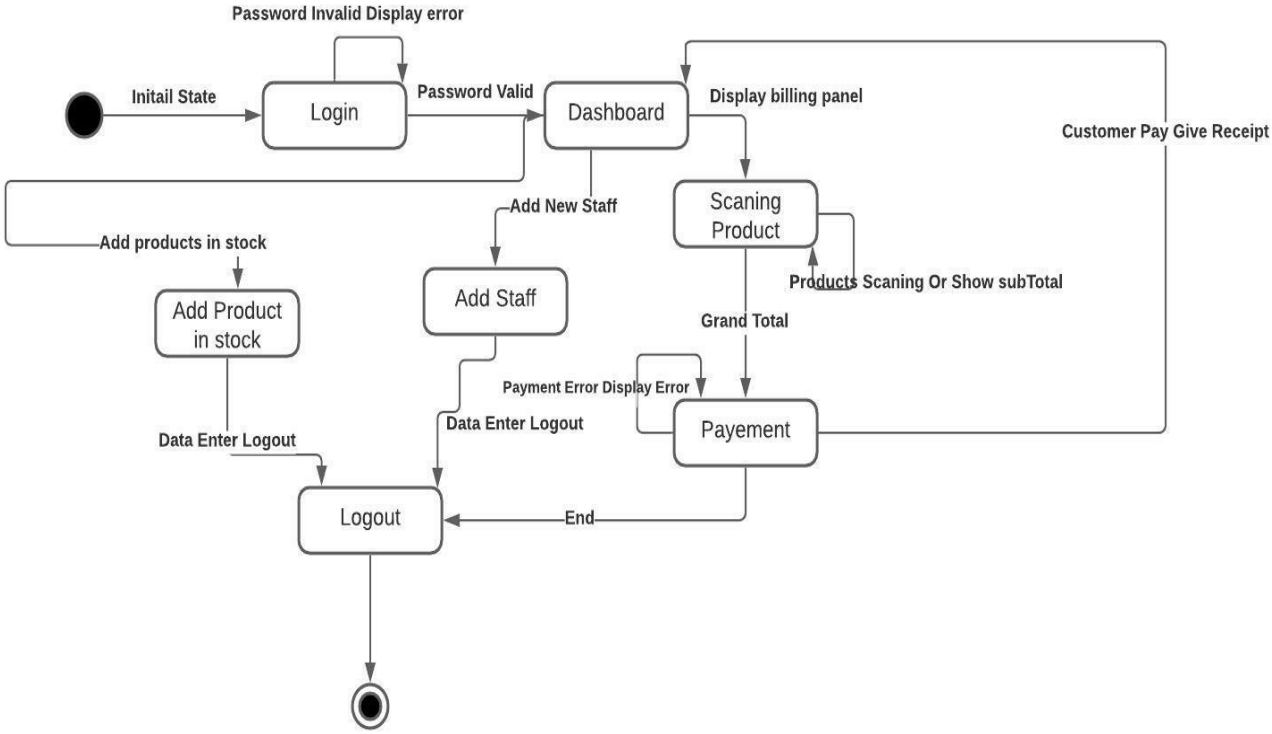



**Figure 4. 10:State Transition Diagram Manager Side**

## Staff Side:

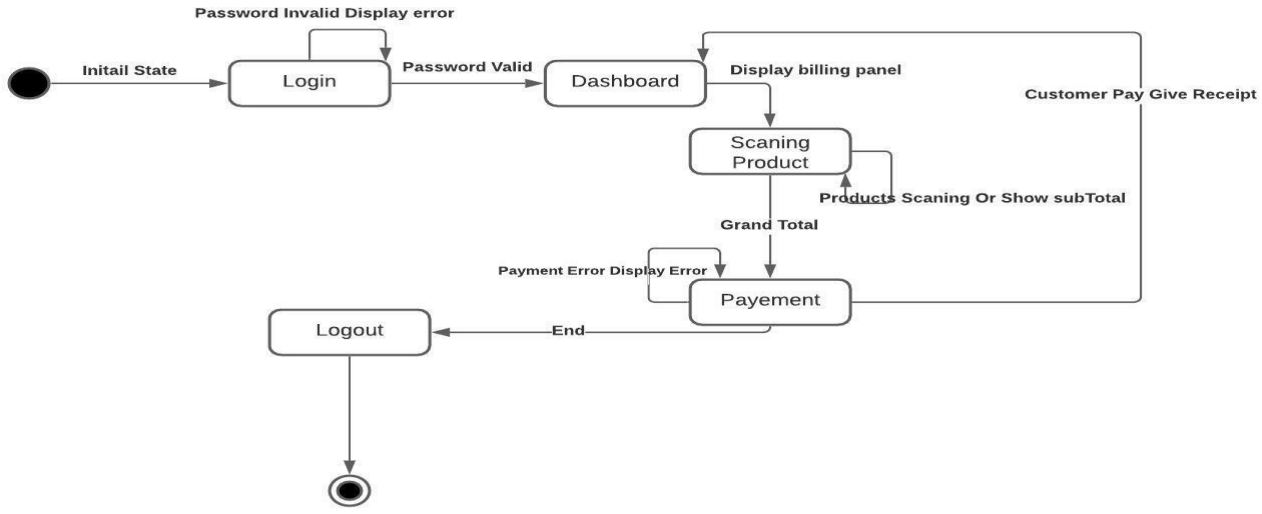



**Figure 4. 11:State Transition Diagram Staff Side**

# Chapter 5

# Implementation

# Chapter 5: Implementation

In Implementation, After the user has examined and accepted the system, it begins. During this process, the system is installed in order to support the mentioned business functions. The performance benchmarks set during the planning process are used to compare system performance. Users must be informed, trained, and the system must be installed on computers used for production as well as integrated into routine business operations. This stage lasts until the system is producing data in accordance with established user requirements.

- Better inventory management and supply ordering capabilities
- Quicker job management and scheduling
- Consolidated payment processing
- Reduced workload for staff
- Eradication of human error
- Thorough customer satisfaction tracking and evaluation

## 5.1. Components, Libraries, Web Services and stubs

- POS Software
- Computer System
- Barcode Scanner
- Card Reader
- Cash Drawer
- Display

Libraries

- laravel/ui
- laravel/framework
- twilio/sdk
- stripe/stripe-php
- Font Awesome
- Google Font

- Namecheap Web Services

During the process of Top-Down Integration Testing, stubs are employed to simulate the behaviour of the lower-level modules that are still being integrated. Stubs are modules that temporarily take the place of called modules and produce output identical to that of the final product. We have individually tested the system's many parts, such as CRUD operations for both users and administrators.

## 5.2. Deployment Environment

The Deployment Environment is where engineers, product owners, QA and automation teams can be used to get a picture of how well a new code revision behaves, works, and looks and feels from a user interface perspective. Our POS system uses an agile software development model.

- Windows (OS)
- Visual studio Code (Software)
- Laptop, Computer, Tablet (Hardware devices)

## 5.3. Tools and Techniques

- Visual Studio Code
- Namecheap Web Services
- Rest API
- MY SQL database
- HTML/CSS
- JavaScript
- Vue js
- MY SQL
- Laravel
- Bootstrap
- AI
- GitHub

## 5.4. Best Practices / Coding Standards

1. Working on reusable Components in Vue js
2. Testing API and integrations
3. Writing comments and documentation for another Users
4. Backups of the project
5. Readable Code

## 5.5. Version Control

By keeping track of changes, version control enables you to make sure that everyone is working with the most recent version of a file or document at all times. You can see what modifications were made, when they were made, and who made them. You can always tell which version of a document or file is the most recent iteration by using version control, which is a technique for tracking changes. In case you want to check what has changed or need to restore an earlier version, it also enables you to keep track of former versions.

# Chapter 6

# Testing and Evaluation

## 6.1. Use Case Testing

It helps us test the entire system before deployment. We have four actors that perform a function in the system. Cashier, customer, bank and manager. They are able to perform the following functions: login, checkout, product display, product search, invoice printing, payment and supplier addition.

### User Login Use case testing

| Main Success Scenario | Step | Description |
|---|---|---|
| **Login**<br>A:Actor<br>S:System | 1 | **A: Enter User Name & Password** |
| | 2 | **S: Validate User Name & Password** |
| | 3 | **S: Allow Account Access** |
| Extensions | 2a | **Password not valid**<br>S: Display Message and ask for re-try 5 times |
| | 2b | **Password not valid 4 times**<br>S: Close Application |

### Enter Products Details Use case testing

| Main Success Scenario | Step | Description |
|---|---|---|
| **Enter product Details**<br>A:Actor<br>S:System | 1 | **A: Enter User Product Details** |
| | 2 | **S: Validate Product Details** |
| | 3 | **S: Allow to Enter a Product details** |
| Extensions | 2a | **Product details is not valid**<br>S: Try again and enter the product Details. |
| | 2b | **Product data is enter successfully**<br>S: data is entered |

### Enter Supplier Details Use case testing

| Main Success Scenario | Step | Description |
|---|---|---|
| **Enter supplier details**<br>A:Actor<br>S:System | 1 | **A: Enter Supplier Details** |
| | 2 | **S: Validate Supplier Details** |
| | 3 | **S: Allow to enter supplier data** |
| Extensions | 2a | **Supplier details is not valid**<br>S: Try again and enter the Supplier Details. |
| | 2b | **Supplier data is enter successfully**<br>S: data is entered |

## 6.2. Equivalence partitioning

Equivalence Partitioning is a type of black box testing technique that may be applied to unit, integration, and system testing of software at all levels. By partitioning the input data units into equivalent groups, test cases can be derived using this technique, which cuts down on the time required for testing due to the sparse amount of test cases.

## 6.3. Boundary value analysis

Boundary testing is the process of evaluating the boundaries between input value segments or their extreme ends.

## 6.4. Data flow testing

1. creating a data flow chart.
2. Selection of test criteria.
3. Classifying the paths that meet the selection criteria in the data flow graph.
4. Develop path predicate expressions to derive test input.

## 6.5. Unit testing

To verify the code written during the coding phase and examine the internal workings of the modules, unit testing is done. It speaks of the isolation-based verification of a single programmed module. To detect bugs, unit testing initially looked at modules apart from one another. Each dialogue is tested and run separately after encoding. All unused code has been eliminated, guaranteeing that every module operates as a programmer would anticipate.

## 6.6. Integration testing

This form of POS automation testing is also referred to as "Integration and Testing," "String Testing," and "Thread Testing." Integrity testing essentially verifies that data is successfully transferred across POS software system elements. QA engineers execute these POS test cases to look for any problems with how the modules interact after integration.

## 6.7. Performance testing

- Check that transaction-based regulations (discounts, taxes, rebates, etc.) are in effect.
- Ensure that the right code is created for approved, held, or denied transactions.
- Check the speed or duration of receiving a response or sending a request.

## 6.8. Stress Testing

• Stress testing is carried out until the system fails and is designed to test the system's behaviour under adverse circumstances.

• Stress testing uses enormous amounts of data or resources to try and crash a system.

# Chapter 7

# Summary, Conclusion and Future Enhancements

# Chapter 7: Summary, Conclusion & Future Enhancements

## 7.1. Project Summary

Your consumer makes payments for goods or services at your business using EasyPOS. EasyPOS software helps businesses like *the* restaurants and stores integrate mobile POS features, contactless payment methods, e-commerce integration *of different* capabilities, and *much* more in addition to processing credit cards. For the benefit of shops, our POS software offers a card processing option along with inventory control, contactless payment methods, e-commerce integration features, and much more. EasyPOS system is the solution we suggested. It is a development over the cash register, manual inventory control, and traditional selling methods.

## 7.2. Achievements and Improvements

We provide web-based POS system that accept credit card payment. Also, we use this POS all over the world because we added different languages, Currencies and email service. For the benefit of shops, our POS software offers a card processing option along with inventory control, contactless payment methods, e-commerce integration features, and much more. Every business owner wants to know which products are selling the fastest, which clients are making the most purchases, how much inventory is on hand, what forms of payment customers prefer (cash, credit, check, or debit), and how much money has been made sell on a particular day.

## 7.3. Lessons Learnt

We learn that always gather the data about the project in advance then start working of it. Without knowing the need of the user, we cannot create a good software that hep our users in daily life.

## 7.4. Future Enhancements/Recommendations

For future we add home deliver option in our pos and delivery the product at home. Also, we add a online shopping where customer buy the product on the their mobile and order it. Also, we create a app that help customer to order the product easily. For future we add shopping club in our app. When we purchase the products then we scan the serial number and the purchasing points are added in our app. This point is used to purchase the items.

# Reference and Bibliography

# Reference and Bibliography

1. Gabriel Miguel T. Dela Cruz, Tovvy B. Dumaplin, Patrick Vincent T. Gaerlan, Oliver M. Junio*,* (2019). Inventory Management System*. Education for Information,* 6(4), 1-9.

2. Ashutosh Rajat Kerketta1, Gaurav Kumar2, (*2016).* Sales and inventory system*, Research Strategies, 3(2),153-170.*

3. Kim, S.-K., & David, C. (1999). Formalizing the UML class diagram using Object-Z*. Paper presented at the International Conference on the Unified Modelling Language*, 1, 1-80

4. Fantechi, A., Gnesi, S., Lami, G., & Maccari, (2003). Applications of linguistic techniques for use case analysis. *Requirements Engineering, 8*(3), 161-170.

5. Jacob, R. J. K. (1985). A state transition diagram language for visual programming. *Computer, 18*(08), 51-59.

6. Polack, F. (1992). Integrating formal notations and systems analysis: Using entity relationship diagrams. *Software Engineering Journal*, 7(5), 363-371.

7. Kaur, A., & Vig, V. (2018). Automatic test case generation through collaboration diagram: a case study. *International Journal of System Assurance Engineering and Management*, 9(2), 362-376.