



Daffodil
International
University

TITLE OF THE PROJECT

Point Of Sale (POS)

Submitted By

Israt Jahan

ID:181-16-240

Supervised By

Ms. Nayeema Rahman

Sr. Lecturer (DIU)

Department of Computing and Information System (CIS)

DAFFODIL INTERNATIONAL UNIVERSITY

Submission Date- 13 February 2022

Declaration

I hereby declare that; this project has been done by me under supervision of Ms. Nayeema Rahman, Senior Lecturer, department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By



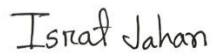
Ms. Nayeema Rahman

Designation

Department of CIS

Daffodil International University

Submitted By



Name:

ID: 182-16-329

Department of CIS

Daffodil International University

APPROVAL

This Project titled “Point of sale”, Submitted by Israt Jahan, ID No:181-16-240 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on- 13-02-2022.

BOARD OF EXAMINERS




Mr. Md Sarwar Hossain Mollah
Associate Professor and Head
 Department of Computing & Information Systems
 Faculty of Science & Information Technology
 Daffodil International University

Chairman



Ms. Nayeema Rahman
Sr. Lecturer
 Department of Computing & Information Systems
 Faculty of Science & Information Technology
 Daffodil International University

Internal Examiner



Mr. Abdullah Bin Kasem Bhuiyan
Lecturer
 Department of Computing & Information Systems
 Faculty of Science & Information Technology
 Daffodil International University

Internal Examiner



Dr. Mohammad Shorif Uddin
Professor
 Department of Computer Science and Engineering
 Jahangirnagar University, Savar, Dhaka

External Examiner

Acknowledgment

It's my pleasure to take this occasion to thank a few people, who have assisted, encouraged, directed, and supported me throughout my practicum program. First of all, I want to thank my parents, who have endowed their immeasurable-innumerable support and encouragement to attain this exquisite event of my life.

I would like to pay my gratitude to my project supervisor MS. **Nayeema Rahman**. I am grateful to her because kind support and guidance during the course work.

Dedication

I like to dedicate this project to my late father, my late brother. Without my mother and brother's courage, support and inspiration I am nothing. My father and brother encouraged me to study computer science.

Abstract

This report is for practicum defense. The primary objective of this report is to learn how to conduct a project and work in real field and write it down in a formal and specific way. The secondary objective of this report is to learn about how this Point of Sale System can manage ordering process for customer more effective. The Point of Sale System is developed for providing the ordering and manages service to the staff of the Shop without facing any problem. Currently the shop handles all process manually which a very time is consuming and complex. This software not only deals with ordering products but also deals with the internal operation of the shop also. There are two types of user in this system including Owner and Staff. Owner can handle user access to the system by giving their permission to a specific module and see the sales report. Also, owner can manage expense, purchase, products, categories and set settings to run the system smoothly. Owner also able to see various reports. Staff able to create new order suspends a sale, print invoice for the sale and add payment to a sale. This system reduces time consumption to perform a task like get report or calculate total of an order. The main objective of this system is to automate the ordering process and calculate total and generate report. I have plans to implement other features in future. I could not add all of them just because of the time limitation. At the end of the day, what I can say is I put my honest effort and hard work to implement the system as efficient as possible. I wish to make it flawless in near future.

Contents

Acknowledgment	iii
Dedication	iii
Abstract	iv
Chapter 1 – Introduction	1
Chapter 2 – Initial Study	3
2.1 Project Proposal	3
Chapter 3 – Literature Review	4
3.1 Discussion on the Problem Solution	4
3.2 Comparison of three leading solutions	4
3.2.1 Best Features	4
3.2.2 Limitations of the Project.....	4
3.2.3 Recommended approach.....	5
Chapter 4 – Methodology	6
4.1 Data Sources	6
(Database Management System, 2018).....	6
4.2 Sections of methodology.....	6
4.3 Reason why I choose the Incremental process model	7
Chapter 5 – Planning.....	8
5.1 Functions of Proposed System.....	8
5.2 Function Specification	8
5.3 Function Point Estimation.....	10
(Overflow, How to calculate function points, 2018)	10
5.4 Effort Distribution.....	14
5.5 Gantt Chart.....	15
5.6 Risk Management	15
5.6.1 Risk Identification.....	16
5.6.2 Risk Analysis	16
5.6.3 Risk Planning.....	17
5.6.4 Risk Monitoring.....	18
5.7 Quality Management.....	18

5.7.1 System Quality Management	18
5.7.2 Software Quality	18
5.7.3 Quality Planning	18
5.7.4 Quality Assurance	19
5.7.5 Quality Control	19
Chapter 6 – Feasibility	20
6.1 Feasibility Study	20
6.1.1 Technical feasibility	20
6.1.2 Economic Feasibility	20
6.1.3 Operational feasibility.....	20
6.2 Project Estimation.....	21
6.2.1 Cost	21
6.4 Hardware cost	22
6.5 Software cost.....	22
6.7 Other Cost	22
6.8 Accounts Table	22
Chapter 7 – Foundation.....	24
7.1 The Problem Area Identification	24
7.1.1 Interview	24
7.1.2 Observations	24
7.1.3 Questionnaires.....	24
7.2 Overall Requirement List.....	25
7.2.1 Functional Requirements	25
7.2.2 Non- Functional Requirements	25
Chapter 8 – Exploration	26
8.1 Old System Use Case.....	26
8.2 Activity Diagram	27
8.2.1 Activity Diagram for owner.....	27
8.2.2 Activity Diagram for Staff.....	28
Chapter 9 – Engineering	33
9.1 New System Modules	33

9.1.1 User Requirements of the system Modules	33
9.1.2 The New system Modules.....	33
9.1.3 Functional Requirements of the New System.....	38
9.1.4 Non- Functional Requirements of the New System	39
9.2 Use Case Diagram.....	39
9.2.1 Use Case Description.....	41
Figure 12: Class Diagram	42
9.4 Entity Relationship Diagram.....	43
9.6 System Interface Design	46
Chapter 10 – Deployment	53
10.1 Core Module Coding Sample.....	53
10.2 Possible Problem Breakdown	60
10.3 Prioritization while Developing the Solution	60
Chapter 11 – Testing.....	61
11.1 Software Testing Strategy.....	61
11.2 System Testing Methodology	62
11.2.1 Black-box Testing.....	62
11.2.2 White-box Testing	62
11.3 Testing.....	63
Chapter 12 – Implementation.....	67
12.1 Training.....	67
12.2 Implementation Scheme.....	67
12.3 Scaling.....	67
12.4 Load Balancing	67
Chapter 13 – Critical Appraisal and Evaluation	68
13.1 Objective could be met	68
Chapter 14 Lessons Learned.....	69
14.1 Pre-project – review – closing	69
14.2 What I have learned	69
14.3 The problems I have faced.....	69
14.4 What Solution Occurred	69

Chapter 15 – Conclusion.....	70
15.1 Summary of the Project	70
15.2 Goal of the project	70
15.3 Success of the project.....	70
15.4 What I have done in the documentation	71
15.5 Value of the Project	71
15.6 My experience.....	71
Appendix.....	72
References.....	73

List Of Figure

FIGURE 1: INCREMENTAL PROCESS MODEL	7
FIGURE 2: EFFORT BASED ESTIMATION	14
FIGURE 3: GANTT CHART.....	15
FIGURE 4: OLD SYSTEM USE CASE	26
FIGURE 5: ACTIVITY DIAGRAM FOR OWNER.....	27
FIGURE 6: ACTIVITY DIAGRAM FOR STAFF	28
FIGURE 11: USE CASE DIAGRAM.....	40
FIGURE 13: ENTITY RELATIONSHIP DIAGRAM	44
FIGURE 14: SEQUENCE DIAGRAM	45
FIGURE 15: DASHBOARD INTERFACE DESIGN.....	46
FIGURE 16:POINT OF SALE INTERFACE DESIGN	47
FIGURE 16: CUSTOMER SALES INTERFACE DESIGN	48
FIGURE 17: PRODUCT SALES HISTORY	48
FIGURE 18: PRODUCT PURCHASES INTERFACE DESIGN.....	49
FIGURE 19: CATEGORY INTERFACE DESIGN	49
FIGURE 20: EMPLOYEE INTERFACE DESIGN.....	50
FIGURE 21: CUSTOMER INTERFACE DESIGN.....	50
FIGURE 22: VENDORS INTERFACE PAGE DESIGN.....	51
FIGURE 23: SETTINGS INTERFACE PAGE DESIGN.....	51
FIGURE 24: LOGIN PAGE	53
FIGURE 25: DASHBOARD CODING.....	54
FIGURE 26: USER CODE	56
FIGURE 27: CUSTOMER CODE	57
FIGURE 28: VENDOR CODE	58
FIGURE 29: CATEGORY CODE.....	59
FIGURE 30: BLACK BOCK AND WHITE BOX TESTING.....	62

List Of Table

TABLE 1: FUNCTION SPECIFICATION	9
TABLE 2: TRANSACTION FUNCTION	11
TABLE 3: DATA FUNCTIONS.....	11
TABLE 4: UNADJUSTED FUNCTION POINT CONTRIBUTION (TRANSITION FUNCTION).....	11
TABLE 5: UNADJUSTED FUNCTION POINT CONTRIBUTION (DATA FUNCTION).....	11
TABLE 6: PERFORMANCE AND ENVIRONMENT IMPACT	12
TABLE 7: FUNCTION POINT ESTIMATION	13
TABLE 8: PROCESS BASED ESTIMATION.....	13
TABLE 9: RISK IDENTIFICATION	16
TABLE 10: RISK ANALYSIS	17
TABLE 11: RISK PLANNING	18
TABLE 12: TECHNICAL FEASIBILITY STUDY	20
TABLE 13: SALARY RANGE	21
TABLE 14: HARDWARE COST.....	22
TABLE 15: SOFTWARE COST.....	22
TABLE 16: OTHER COST.....	22
TABLE 18: ACCOUNTS TABLE	23
TABLE 19: PRIORITIZED REQUIREMENT LIST (PRL)	29
TABLE 20: DAILY SALE INFORMATION MODULE TESTING.....	63
TABLE21: POS TERMINAL TESTING	63
TABLE 22: MANAGE EXPENSES MODULE TESTING.....	63
TABLE 23: MANAGE PRODUCTS MODULE TESTING.....	63
TABLE 24: MANAGE CATEGORY LIST MODULE TESTING.....	64
TABLE 25: MANAGE PURCHASES MODULE TESTING.....	64
TABLE 26: SALE REPORT MANAGE MODULE TESTING.....	64
TABLE 27: MANAGE CUSTOMERS MODULE TESTING	64
TABLE 28: MANAGE USERS MODULE TESTING	65
TABLE 29: MANAGE DUE MODULE TESTING	66
TABLE 30: MANAGE NOTIFICATION MODULE TESTING	66
TABLE 31: MANAGE SETTING MODULE TESTING	66
TABLE 32: MANAGE INVOICE MODULE TESTING	66
TABLE 32: REPORT MODULE TESTING	66
TABLE 33: TRAINING	67

Chapter 1 – Introduction

The internship is one kind of practical experience of conceptually acquired knowledge that is quantifiable as a groundwork trial to be aware of any organization and to make oneself confident enough to enter service life and start building a career. And, a better and more effective way to connect academic experience with the professional work area. It allows gaining valuable experience in the workplace, provides the opportunity for skill development, and gives a competitive edge in the job search. This chapter attempts to describe the objectives, scope, and all topics of the initialization period of this project.

Document Contents

This document will cover the following chapters to document the project work.

Chapter 1: Introduction

Introduction about the proposed system.

Chapter 2: Initial Phase

This chapter contains the initial study details about the project proposal, goal, aim, objective, problem area, alternative solution.

Chapter 3: Literature Review

This chapter contains the scope of the project, limitations of the Project, Recommended approach.

Chapter 4: Methodology

Here using the methodology and its implementation will be discussed.

Chapter 5: Planning

This chapter is about the project plan, test plan, risk, and management.

Chapter 6: Feasibility

This chapter contains a feasibility study report and a cost-benefit analysis will be documented here.

Chapter 7: Foundation

Details about problem area identification, requirement list.

Chapter 8: Exploration

It contains some old system and new system basic UML diagrams and requirement catalog with the prototype.

Chapter 9: Engineering

In this chapter the logical and behavioral modeling of the proposed system

Chapter 10: Deployment

Here coding samples and development problems break down with development priority

Chapter 11: Testing

The test plan and result are attached here.

Chapter 12: Implementation

The implementation approach, training model, and related things are discussed here.

Chapter 13: Critical Appraisal and Evaluation

Discusses about the initial objectives that were met and not met.

Chapter 14: Lessons Learned

The pre-project-closing review contains the learning and difficulties faced during the project

Chapter 15: Conclusion

The summary of the project along with the goal and success and experience will be listed here.

Chapter 2 – Initial Study

2.1 Project Proposal

Point of Sale is a Desktop based application that works within a private network area. Point of Sale software is a system, that handles the regular activity of a café. Using the point of sale software users can perform selling and manage products, manage expenses, purchases, staff, and permission of users. Point of Sale software not only deals with sales but also deals with the order process. The software stores all sales date to date records with quantity, cost, selling price, discount, and other information. In this Point of sale software, the system auto-calculates the total price, discount, and other charges of a product and order. The software also keeps records of orders, bills, and payment receipts of an order. The software can suspend/hold a sale while need to add another order. Using the software user can process payment using only hand cash now. Hopefully, bKash and DBBL services will be added afterward. The store can manage its expenses through the system. It is also possible to generate sales reports, loss/profit reports, and many more. Some reports are visualized by the chart. The owner and Staff login systems are separated to make the system more secure with proper permission.

Chapter 3 – Literature Review

3.1 Discussion on the Problem Solution:

- **Insecure POS System:** Cyber-attacks are one of the most difficult ways of operating a POS system. From the terminal to the endpoints where data is kept, POS systems present a huge attack.
- **Data Can Be Hacked by a POS Vendor:** Vendors or third parties are routinely entrusted with the management of many aspects of a retailer's business, including POS systems. A single POS device or terminal, or a complete network, maybe under the control of a vendor.
- **Picking the Wrong POS System for Your Business:** Every retail business is unique and needs unique POS support. It might be costly to select the wrong POS system or vendor, especially if your company's charge limits are crossed.
- **POS System Isn't Correctly Installed:** Because- Wi-Fi connectivity issues can also be caused by improperly installed. Devices and software are incompatible, and the system contains errors.
- **Insufficient Troubleshooting Support:** Proper training for the customer for using pos. Establish a process for the pos problem in collaboration with the pos partner.

3.2 Comparison of three leading solutions

3.2.1 Best Features

This project aims to automate the ordering system of the shop. The owner will add staff to perform their task. Owners can manage products, sales, expenses, and purchases. Staff can create a new order and print the receipt for the order and edit or add the payment to the order. The owner and staff can generate various reports to track down profit/loss and sales performance of the shop. All the processes will be done in the following proper software development process.

3.2.2 Limitations of the Project

This project is a real-life implementation of the knowledge regarding software development gathered throughout the graduation program. But it was difficult for me to implement all the terms and conditions of the software development process as a fresh graduate. Initially, it was a big issue but, gradually with the support of some senior experts and honorable supervisors, I adapt myself to the environment.

3.2.3 Recommended approach

For the new system, this objective should be provided. Here-

- To develop a user-friendly application
- To develop a system which helps to find and order products easily
- To develop a system to secure payment transaction
- To develop an admin panel to manage all the business activity digitally
- To develop a system to automate the calculation of an order
- To develop a system to track down sales.

Chapter 4 – Methodology

The methodologies are derived from the collected data of the early requirement gathering phase. There are many methodologies but I use here Incremental Process Model in this system.

4.1 Data Sources

(Database Management System, 2018)

The sources of data for this project are:

- Primary
- Secondary

The primary data are collected based on user requirements of the software, suggestions of intern organization, and the practical experience of senior experts.

The Secondary data are gathered by studying different books, newspapers, journals, and information collected via the internet. I studied many other open sources management projects documentation and data on the internet to understand the project better.

4.2 Sections of methodology

In software development, the software process model is an essential part that designs a strong structure of the development phases. It is noticed that in many software developments the requirements are specified well initially. But when the developer tries to import their best effort to fulfill the maximum scope of the development, the development process is precluded. Then they will be bound to provide limited functionality to the users. After the release, they will be able to refine and expand the functionality according to the feedback. In such a situation I would like to follow the Incremental process model to develop the software that I have described above.

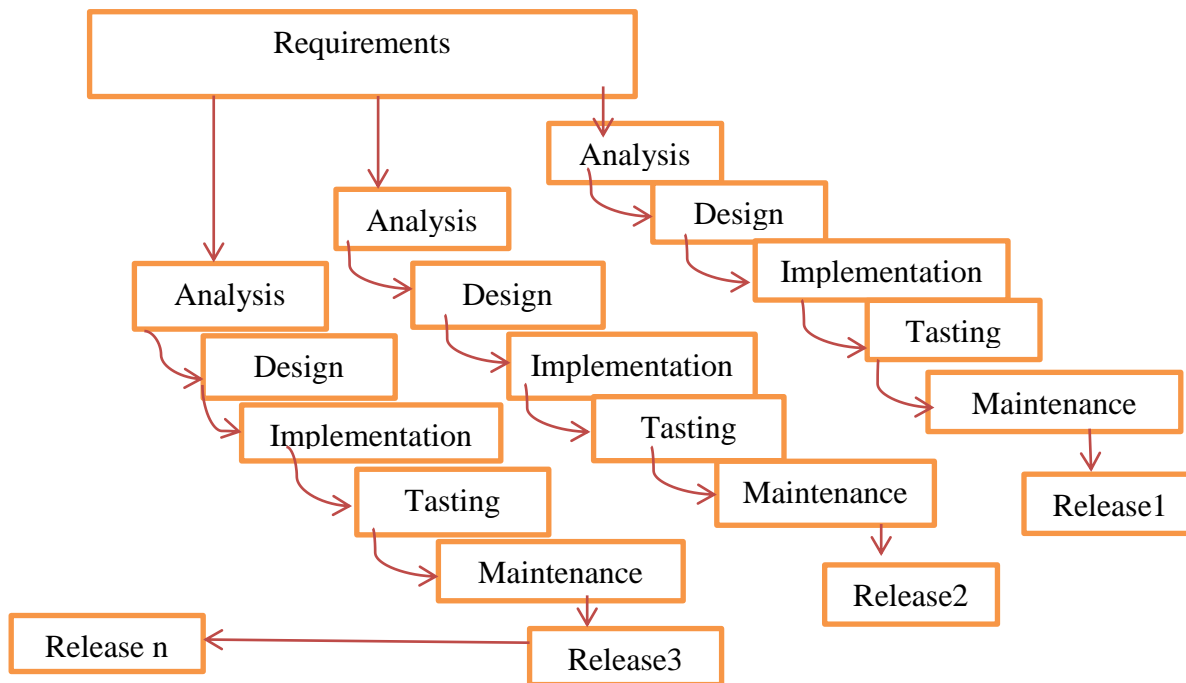


Figure 1: Incremental Process Model

4.3 Reason why I choose the Incremental process model

- This project is small.
- This model is basic and easy to explain and apply.
- Its process activities are separated and organized.
- The incremental technique works best for smaller projects with well-defined requirements.
- Requirements for this project changed a lot during development and after testing.
- Technology is understood.

Chapter 5 – Planning

5.1 Functions of Proposed System

1. Today's sales	[F01]
2. Point of sale management	[F02]
3. Add products in cart	[F03]
4. Rewrite products in cart	[F04]
5. Create payment	[F05]
6. Make sale	[F06]
7. Make Invoice	[F07]
8. Product management	[F08]
9. Expenses management	[F09]
10. Purchase management	[F10]
11. Category management	[F11]
12. Customer management	[F12]
13. User management	[F13]
14. Vendors management	[F14]
15. Due management	[F15]
16. Sales report	[F16]
17. Store settings	[F17]
18. Reports	[F18]

5.2 Function Specification

Function description descriptive the function in details. It concerns on three factors:

- What is the possible input,
- Possible output for a function
- Which database table that function uses.

Functionality	Input	Output
The owner can view today's sales details	Click on the sale reports from the top bar dropdown	View today's sales, revenue, due, purchase, expense as default
The owner can view all	Click on the add/update/delete	Add, modify and delete the

categories and add, edit, search and delete the category	button, fill up all required fields	category
The owner can view the product list and add, edit, search and delete products	Click on the add/update/delete button, fill up all required fields	Add, modify and delete products
Owner and Staff can view expense and add, edit, search and delete expense	Click on the add/update/delete button, fill up all required fields	Add, modify and delete Expense
The owner can add, edit, search and delete purchases and vendor	Click on the add/update/delete button, fill up all required fields	Add, modify and delete purchases and vendor
The owner can view, add, edit or delete customers	Click on the add/update/delete button, fill up all required fields	Add, modify and delete customers
The owner can add, edit, search and delete vendor	Click on the add/update/delete button, fill up all required fields	Add, modify and delete vendor
The owner can see all types of report	Select types of report	See report
The owner can set store settings (Image, name, etc)	Click on setting from the sidebar	Modify store setting
Owner add, edit, search and delete user/employees	Click on the add/update/delete button, fill up all required fields	Add, modify and delete users
Owner and Staff can make sales and create invoice	Select items, create bills, receive payment, cancel orders	Make invoice
Staff can see details of products, categories	Click on products/ categories	See details
Owner and staff can manage due payment	Click on the question mark icon from the sales details table and fill up repay the amount	Add dye repayment and make an invoice
Owner and Staff can view all sales details	Click on the sales report button	See sales report

Table 1: Function Specification

5.3 Function Point Estimation

(Overflow, How to calculate function points, 2018)

Transaction function	Fields/File involvement	FTR	DET
Manage Category (2*EI)	Fields - Code, image, details Files - categories	1	3
Manage products (2*EI)	Fields – code, name, image, quantity, category, type, cost, price Files – category, products	2	8
Manage expenses (2*EI)	Fields – date, reference, amount, note Files- expenses	1	4
Manage purchases (2*EO)	Fields – date, reference, products, total, totalAmount, vendor, received Files – purchase, products, vendor	3	7
Manage Customer (2*EI)	Fields – name, email, contact, address Files - customers	1	4
Manage vendor (2*EI)	Fields – name, email, contact, address Files - vendors	1	4
Manage users (2*EI)	Fields – name, email, password, type Files - users	1	4
Make sales (EO)	Fields – customerID, paymentStatus, reference, saleNote, total, orderedDiscount, grandtotal, item, itemType, staffNote, saleID, saleDate, productID, quantity, cost, salePrice, subtotal, discount, saleID, customerID, addedDate, amount, due, note Files – sales, saleitems, products, payments, customers	5	24
Repay due (EO)	Fields – saleID, customerID, amount, due, status Files – payment, sales	2	5
See sales /customer report (EQ)	Fields – date, sales, purchase, expense, due, productName, customerName, quantity, revenue, paymentStatus, discount Files – sales, saleitems, payments, products, purchase, expenses, customers	7	11
Manage store setting (EQ)	Fields – name, address, image, favicon Files – setting	1	4
Manage theme (EQ)	Fields – customerID, name, contact, email, password, sidebar, topbar, brandName, footer, borderedLine Files – theme, customers	2	10

Table 2: Transaction function

Data functions	Fields/ File involvement	RETs	RETs
Add to cart (ILF)	Fields- productName, price, quantity, subtotal, discount, saleNote	3	6
Calculate Grantotal (ILF)	Fields – salePrice, subtotal, discount	2	3

Table 3: Data Functions

Transaction function	FTR	DET	Complexity	UFP
Manage category (2*EI)	1	3	2*Low	6
Manage products (2*EI)	2	8	2*Average	8
Manage expenses (2*EI)	1	4	2*Low	6
Manage purchases (2*EO)	3	7	2*Average	10
Manage Customer (2*EI)	1	4	2*Low	6
Manage vendor (2*EI)	1	4	2*Low	6
Manage users (2*EI)	1	4	2*Low	6
Make sales (EO)	5	24	High	7
Repay due (EO)	2	5	Low	4
See sales /customer report (EQ)	7	11	High	6
Manage store setting (EQ)	1	4	Low	3
Manage theme (EQ)	2	10	Average	4
Total				72

Table 4: Unadjusted Function Point Contribution (Transition Function)

Data functions	RETs	DETs	Complexity	UFP
Add to cart (ILF)	3	6	Average	10
Calculate Grantotal (ILF)	2	3	Low	7
Total				17

Table 5: Unadjusted Function Point Contribution (Data Function)

General System Characteristics	Brief Description	DI
Data communications	How many communication facilities are available to help with data transfer or exchange with the application or system?	3
Distributed data processing	What is the process for managing distributed data and processing functions?	0
Performance	Was response time or throughput required by the client?	3
Heavily used configuration	How popular is the current hardware platform on which the program will run?	1
Transaction rate	How often are transactions carried out on a daily, weekly, monthly, or annual basis?	4
On-Line data entry	What percentage of the data is entered digitally?	5
End-user efficiency	Was the app created with the user in mind?	4
On-Line update	How many ILFs do On-Line transactions update?	2
Complex processing	Is there a lot of logical or mathematical processing in the app?	3
Reusability	Was the app developed to satisfy the demands of a single user or a group of users?	2
Installation ease	How tough is it to convert and install?	1
Operational ease	Start-up, backup, and recovery procedures: how effective and automated are they?	1
Multiple sites	Was the program developed, built, and supported with the intent of being installed at various locations for multiple organizations?	3
Facilitate change	Was the app developed, developed, and supported with the goal of facilitating change?	0
The total degree of influence (TDI)		32

Table 6: Performance and Environment Impact

Value adjustment factor (VAF) = $(.65 + (.01 * TDI)) = (.65 + (0.01 * 32)) = 0.97$

Unadjusted Function point count (UFP) = UFP (Data function) + UFP (Transaction Function)
 $= 17 + 72 = 89$

Adjusted Function point count (AFP) = UFP * VAF = $89 * 0.97 = 86.33$

Value adjustment factor (VAF)	0.97	
Unadjusted Function point count (UFP)	89	
Adjusted Function point count (AFP)	86.33	
Effect for Codeigniter	AFP * Productivity for Php	
	86.33 * 15.5	Productivity for Php = 15.5
	1338.115	Person Hours
	167.26	Person Days (8 Hours Per Day)
	7.27	Person Months (23 Days Per Month)
	3.64	Months (2 people for developing the project)

Table 7: Function Point Estimation

Function	Pre- requisite		Engineering		Construction & release		CE	Total
	CC	Planning	Analysis	Design	Code	Test		
F01	0.3	0.1	0.3	1.5	2	1	N/A	5%
F02	0.2	0.3	1.5	2	2	2	N/A	8%
F03	0.1	0.15	0.55	0.55	5	0.75	N/A	7%
F04	0.2	0.4	2	2	4	2	N/A	11%
F05	0.01	0.15	2	2	3	2	N/A	9%
F06	0.01	0.15	0.6	0.6	2	0.8	N/A	4%
F07	0.15	0.14	0.5	0.5	1	0.6	N/A	3%
F08	0.15	0.2	0.8	0.8	2	0.6	N/A	5%
F09	0.1	0.15	0.8	0.7	1	0.5	N/A	3%
F10	0.2	0.25	0.9	0.9	2	0.7	N/A	5%
F11	0.15	0.15	0.8	0.9	1	0.9	N/A	4%
F12	0.1	0.15	0.8	0.9	4	1	N/A	7%
F13	0.02	0.02	0.8	0.7	3	1	N/A	6%
F14	0.02	0.03	1.5	1	2	0.8	N/A	5%
F15	0.15	0.2	0.6	0.8	2	1	N/A	5%
F16	0.15	0.2	0.4	0.7	2	1	N/A	4%
F17	0.02	0.03	0.4	0.5	1	1	N/A	3%
F18	0.02	0.03	0.3	0.4	3	2	N/A	6%
Total	2%	3%	16%	17%	42%	20%	0%	100%

Table 8: Process Based Estimation

5.4 Effort Distribution

The project estimation technique leads to estimates of work units required to complete the software development. The 40-20-40 rule is a recommended distribution of effort between the definition and development phases. Frontend analysis and design accounted for 40% of the total effort twenty percent allocated to coding, and the remaining forty percent allocated to back-end testing. This rule is used as a guideline only. In this project, 38% of full software development has been allocated to communication, Planning, analysis, and design, 42% has been allocated to coding and the remaining 20% is allocated to software testing and support.

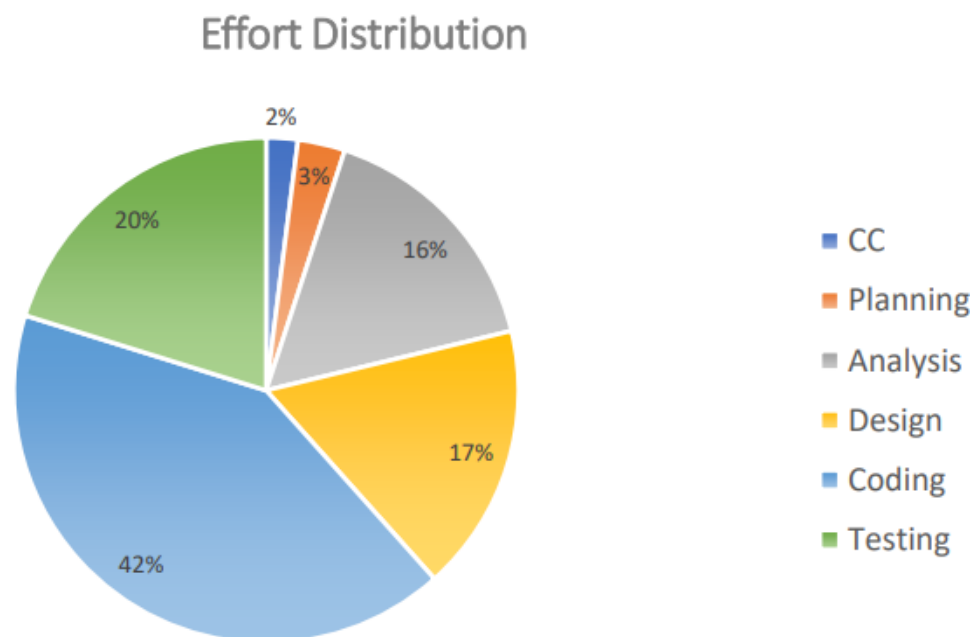


Figure 2: Effort Based Estimation

5.5 Gantt Chart

Total system development is a combination of a set of tasks. This set of tasks should do sequentially and timely. The project schedule works as the guideline of the system developer.

The following is the Gantt chart of the project:

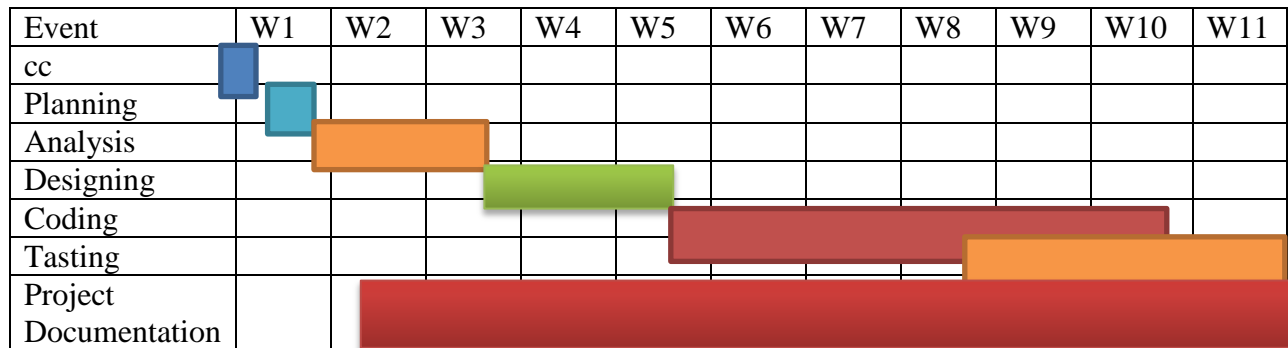


Figure 3: Gantt Chart

5.6 Risk Management

Risk analysis and management refer to a set of activities that assist a system development team in comprehending and managing uncertainty. Many problems can arise while developing a system. A risk is a possible problem that may or may not occur. Risk analysis and management can be broken down into numerous parts. The first step is to identify the risks. Then each risk is assessed to see how likely it is to occur and how much harm it will cause if it occurs. Once this information has been gathered, dangers are noted. Finally, a strategy for dealing with high-probability and high-impact threats is developed.

5.6.1 Risk Identification

Risk type	Possible risks
Technology	1. The system's database can't handle as many transactions per second as it should. 2. Reusable software components have flaws that prevent them from being used as intended.
People	3. It is impossible to find people with the necessary skills. 4. Important personnel are sick and unavailable at critical periods. 5. Staff training is not available as required.
Organizational	6. The organization is reformed so that the project is overseen by distinct management. 7. Project budget cuts are required by organizational financial issues.
Tools	8. The software code creation techniques generate wasteful code. 9. Software tools are incompatible with one another.
Requirement	10. Requirement changes that necessitate significant design revision are requested. 11. Customers are unaware of the consequences of changing requirements.
Estimation	12. The time necessary for software development is underestimated. 13. The rate at which defects are repaired is undervalued. 14. The software's size is underestimated.

Table 9: Risk Identification

5.6.2 Risk Analysis

Risk analysis assesses the potential impact of risk patterns or scenarios, the potential magnitude of the loss, and the direct and indirect recovery costs. This step analyzes vulnerabilities, considers the organization's willingness to tolerate risk in light of the consequences, and develops mitigation strategies.

Risk	Probability	Effects
Budget cuts are necessary due to organizational financial issues.	Low	Catastrophic
It is impossible to find people with the necessary skills for the project.	High	Catastrophic
At important points in the project, key personnel become unwell.	Moderate	Serious
Reusable software components with faults must be corrected before they may be used again.	Moderate	Serious
Changes in requirements that require elaborate design	Moderate	Serious

rework are proposed.		
The organization is reformed so that the project is overseen by various management.	High	Serious
The system's database can't handle as many transactions per second as it should.	Moderate	Serious
The amount of time it takes to develop software is underestimated.	High	Serious
Integration of software tools is not possible..	High	Tolerable
Customers are unaware of the consequences of changing requirements.	Moderate	Tolerable
Customers are oblivious to the consequences of changes in needs.	Moderate	Tolerable
The rate of fault repair is significantly undervalued.	Moderate	Tolerable
The software's size is seriously underestimated.	High	Tolerable
The code produced by code generators is inefficient.	Moderate	Insignificant

Table 10: Risk analysis

5.6.3 Risk Planning

Risk	Strategy
Organizational financial problems	Prepare a briefing document for senior management that demonstrates how the project contributes significantly to the company's goals and explains why cutting the project budget would be ineffective.
Recruitment problems	Inform consumers of possible problems and delays; look into buying-in components.
Staff illness	Inform customers to potential difficulties and the possibility of delays; investigate buying-in components.
Defective components	Replace possibly problematic components with known-to-be-reliable bought-in components.
Requirements changes	Obtain traceability information in order to gain access to the impact of requirements changes; maximize the amount of information hidden in the design.
Organizational restructuring	To gain access to the impact of requirements modifications, obtain traceability information; maximize the amount of information hidden

	in the design
Database performance	Examine the possibility of purchasing a more capable database..
Underestimated development time	Investigate the utilization of program produces and buying-in components.

Table 11: Risk Planning

5.6.4 Risk Monitoring

- Assess each identified risk regularly to decide whether it is becoming less or more probable.
- Also, assess whether the effects of the risk have changed.
- Each key risk should be discussed at management progress meetings.

5.7 Quality Management

The system quality management and the planning for the software quality management of the project 'Point of Sale' is given in the chapter planning.

5.7.1 System Quality Management

By submitting changes to the product development process, software quality management for the project "Point of Sale" maintains that the desired level of quality is met. The quality of this software is measured, and the process is improved until the proper quality level is achieved. It is measured by several variables. This software quality management aims to manage the quality of this 'Point of Sale' software and its development process

5.7.2 Software Quality

Processes of Management The application of particular quality processes and monitoring that these planned processes have been followed are both part of the software quality management for the project 'Point of Sale. The processes involved in the software quality management of this software. They are:

5.7.3 Quality Planning

Quality planning is performed by first selecting the applicable procedures and standards for this project 'Point of Sale' and then modifying these as required. All the important factors that contribute to the customer requirements are explained in this development.

5.7.4 Quality Assurance

Quality assurance established organizational procedures and standards for the quality of the project 'Point of Sale'. It included assessing the overall project's performance to ensure that it met the required quality standards.

5.7.5 Quality Control

The 'Point of Sale' project's quality control verified that the software development team followed procedures and met standards. The project's result is tracked to verify that it improves, as well as to identify ways to improve overall quality in line with quality standards.

Chapter 6 – Feasibility

6.1 Feasibility Study

The feasibility study determines whether that solution is feasible or achievable for the organization. There are three major areas of the feasibility study.

- Technical feasibility
- Economic feasibility
- Operational feasibility

6.1.1 Technical feasibility

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system

Hardware Requirement	Software Requirement
Computer (Desktop/Laptop/Equivalent)	Operating System (Windows 7/10 or equivalent) with Visual Studio code.
Proper electricity Support	PHP(Codeigniter3)
Adequate system memory and secondary Memory	MySQL

Table 12: Technical Feasibility Study

6.1.2 Economic Feasibility

Generally, it means whether a business or a project feasible cost-wise and logistically. We consider whether the company will be able to pay the cost for redesigning and whether the project will be cost-effective or not

6.1.3 Operational feasibility

Concerns about user acceptance, management support, and the requirements of entities and factors in the organization's external environment are addressed by operational feasibility. The proposed system is created with the client in mind. As a result, all of the features are included only for the benefit of the users.

6.2 Project Estimation

The accuracy of a software project estimate is predicated based on several things:

- Accurately calculated the size of the product to construct.
- The ability to convert size estimates into human effort, calendar time, and financial value.
- The extent to which the project plan represents the software team's or engineer's competence.
- The product requirements' stability, as well as the environment that supports the software engineering process.

Software size estimation is the most important matter that I have to consider during the software project. If the software size is not calculated properly, then this will cause various problems such as scheduling problems, budget problems, etc. As the project goes on before estimating the software size, I have to confirm that the software scope is bounded.

6.2.1 Cost

Number of days in a year = 365

Number of government holidays in a year =24

Number of weekly holidays in a year =52

Total number of working days to develop the project = $365-(52+24) =289$ days

Total number of working days per month to develop the project = $289/12 =24.083$ days

Organization working hours per day = 8 hours

Organization working hours per month= $24.083*8= 192.66$ hours

Position	Salary/Month	Salary/Hour
System Analyst, Project Manager	25000	129.76
System designer, Coder, Tester	15000	77.86

Table 13: Salary range

6.4 Hardware cost

Particulars	Quantity	Depreciation calculation	Depreciation Expense
Computer	1	93,000*4.16	3875
Peinter	1	8,000*25.00%	2000

Table 14: Hardware Cost

6.5 Software cost

Particulars	Cost
Visual Studio Code	Free (Student pack)
Google Chorme	Free
Xampp	Free

Table 15: Software cost

6.7 Other Cost

Particulars	Cost
Furniture	500
Electricity bill	300
Extra	1000

Table 16: Other Cost

6.8 Accounts Table

Particulars	Cost	Cost
Salary-		
<ul style="list-style-type: none"> System Analyst, Project Manager 	129.76*8*30	31142.7
<ul style="list-style-type: none"> System designer, Coder, Tester 	77.86*8*65	40487.2

		Total- 71629.6
Hardware cost <ul style="list-style-type: none"> • Computer • Printer 	3500 1800	3500 1800 Total-5300
Software cost <ul style="list-style-type: none"> • Visual Studio Code • Google Chrome • Xampp 	Free Free Free	Total-0
Other Cost- <ul style="list-style-type: none"> • Furniture • Electricity Bill • Extra 	500 300 1000	Total-1800
Total		78729.6

Table 18: Accounts Table

Chapter 7 – Foundation

7.1 The Problem Area Identification

7.1.1 Interview

For Owner:

- Owner gets what kind of problem face in the manual system
- Can get the notification in time when product less than 25.
- What kind of face problem in the sales report and price.

7.1.2 Observations

Observations are very helpful for collecting requirements for the user. It is very helpful to find the solutions and also fulfill the user requirement. Some main points of observation are given below-

- For the owner, check the notification system in the existing system. Also, add product price.
- Know the actual problem they face.
- Also can add sales reports and keep sales product history in this system.
- The owner or staff can keep customer and vendor contact information.
- The owner can add products with pictures and prices. In this system keep product cost and also will show a profit.

7.1.3 Questionnaires

Question for problem Identification

For the owner

Question-1

What requirements would be in this system?

Question-2

What kind of sales report is needed daily or monthly?

Question-3

What kind of information should keep the vendor?

Question-4

Will the notification come when the product goes down?

7.2 Overall Requirement List**7.2.1 Functional Requirements**

- a. All products should show first as default.
- b. 4 rows 4 column products should be shown.
- c. Image should not be more than 1 MB
- d. Every table should have a search, pagination, order filter.
- e. Time zone will be based on the company.
- f. Staff have no access to delete or edit or show any reports.
- g. Staff have no access to add, delete or edit products, purchases, categories.

7.2.2 Non- Functional Requirements

- a. Easy user interface
- b. Responsive design
- c. No internet connection needed
- d. Don't need other software to run
- e. High security

7.3 Recommendation and Justification

This point of sale system should be available internet connection like laptop desktop. Clint can use this system easily and can make invoices for the customers. So this web application is the recommendation for the proposed system.

Chapter 8 – Exploration

8.1 Old System Use Case

I am going to draw an old system use case.

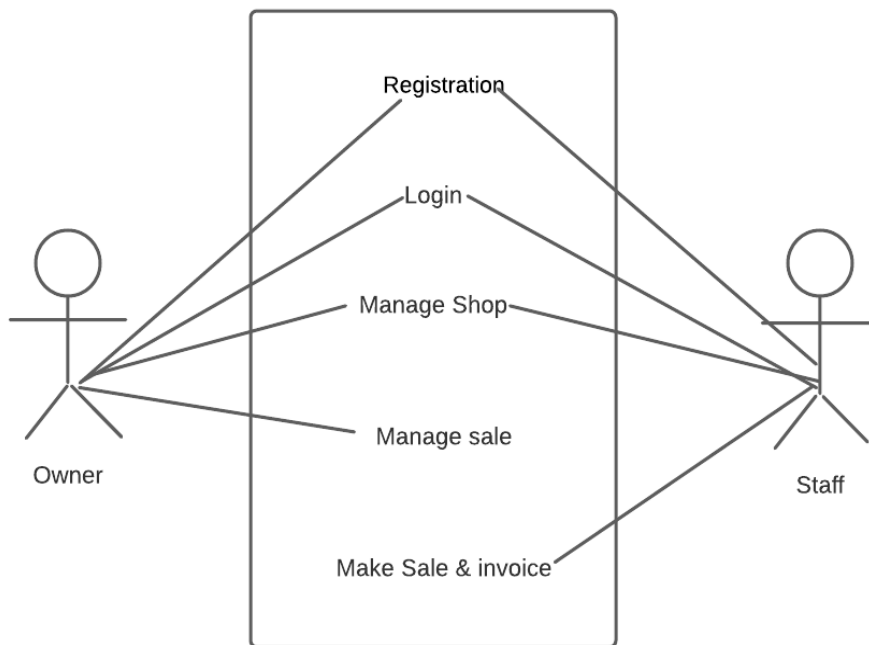


Figure 4: old system use case

8.2 Activity Diagram

8.2.1 Activity Diagram for owner

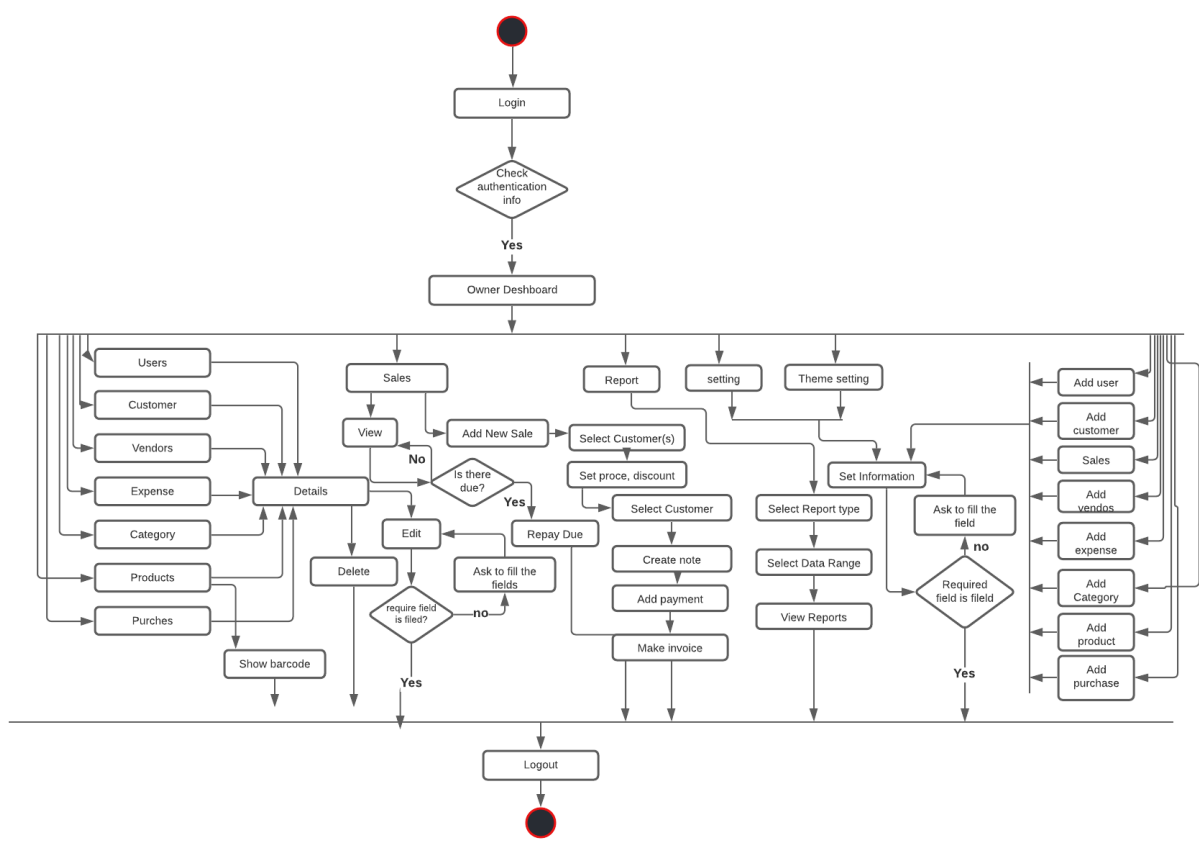


Figure 5: Activity Diagram for owner

8.2.2 Activity Diagram for Staff

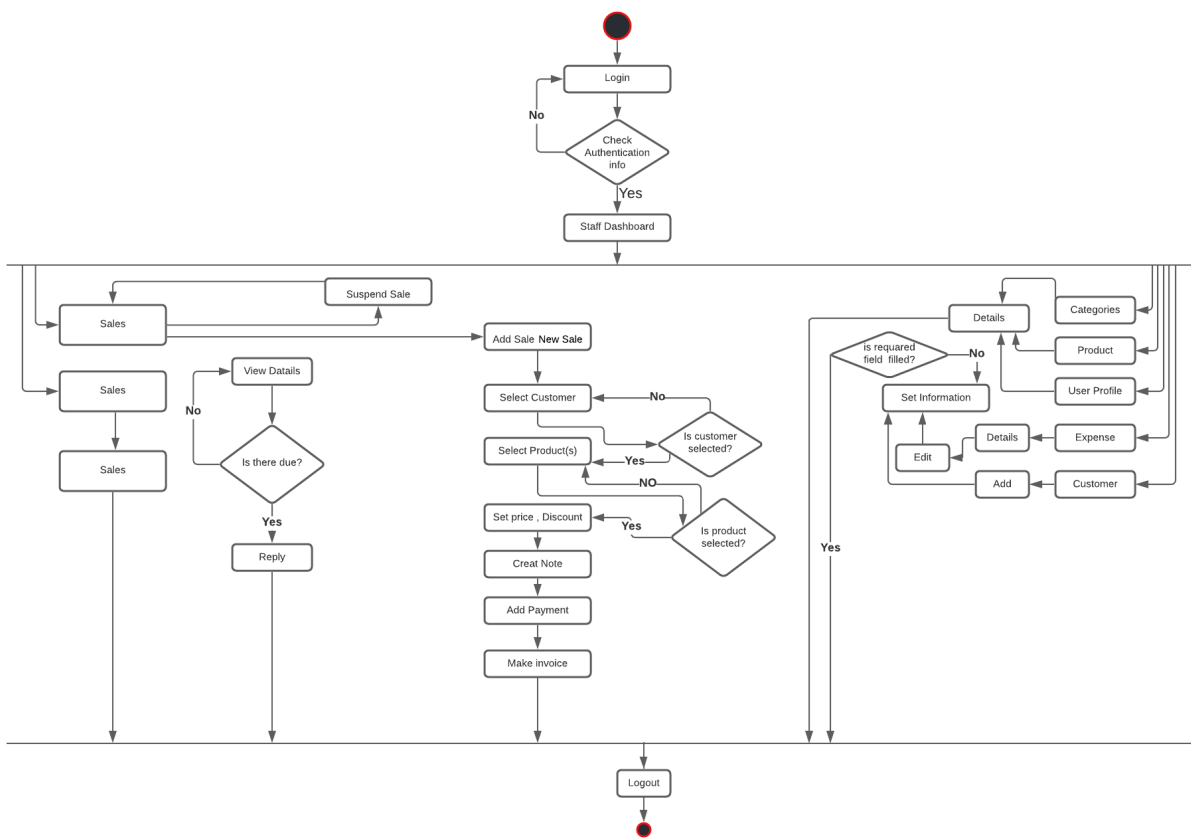


Figure 6: Activity Diagram for Staff

User Requirements of the system Modules

- a. The owner can add, edit and delete users (another owner or staff) and permission.
- b. The owner can view various report
- c. The owner can view today's sales details
- d. The owner can view the top 5 most selling items as the chart
- e. The owner can manage a sale (edit, delete or void a sale)
- f. The owner can view all categories and add, edit and delete the category
- g. The owner can view the product list and add, edit, and delete products
- h. The owner can view, add, edit and delete purchases
- i. The owner can view expense and add, edit, search and delete Expense
- j. The owner can view, add, edit and delete vendor
- k. The owner can view, add, edit and delete customer
- l. Staff can manage a sale (edit, delete or void a sale)
- m. Owner and Staff can view all sales details
- n. Staff can view categories
- o. Staff can view products
- p. Staff can view, add, update, delete expense

Prioritized Requirement List (PRL)

Number	Requirement	Priority
F01	Today's sales	Must
F02	Point of sale management	Must
F03	Add products in cart	Could
F04	Rewrite products in cart	Could
F05	Create payment	Must
F06	Make sale	Must
F07	Make Invoice	Must
F08	Product management	Must
F09	Expenses management	Should
F10	Purchase management	Should
F11	Category management	Should
F12	Customer management	Could
F13	User management	Should
F14	Vendors management	Should
F15	Due management	Should
F16	Sales report	Must
F17	Store settings	Must
F18	Reports	Must

Table 19: Prioritized Requirement List (PRL)

Prototype of the new system

Some prototypes are given here of point of sales.

Login page prototype of point of sales:

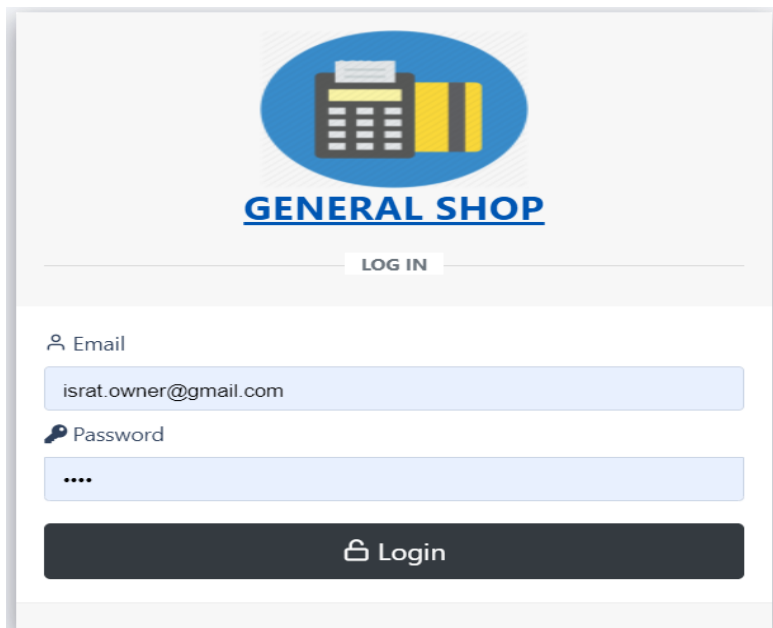


Figure 7: Login page prototype

Home page prototype of point of sales with daily sales report:

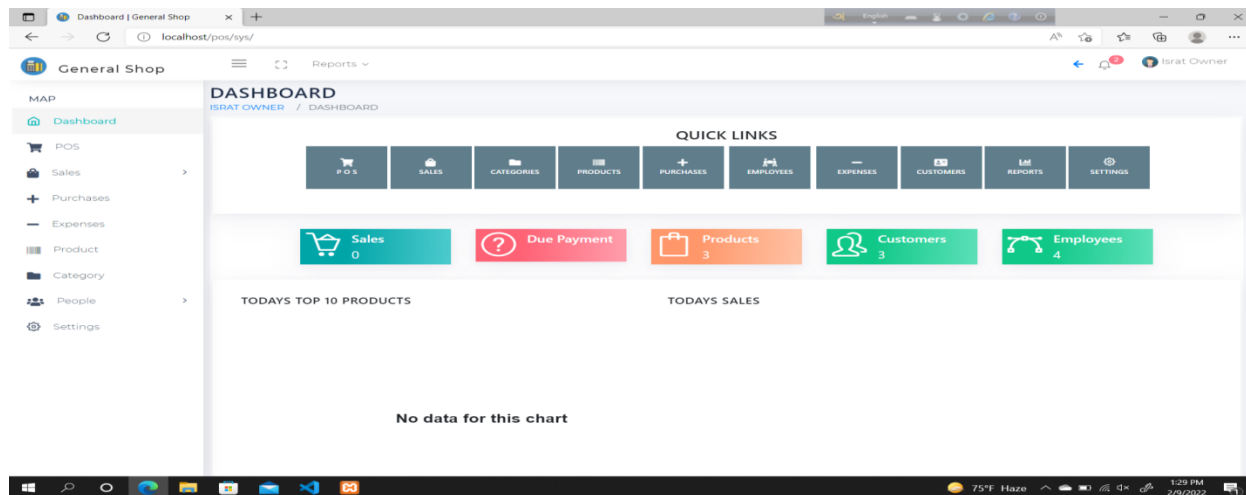


Figure 8: Home page prototype

Point of sales Prototype of point of sales:

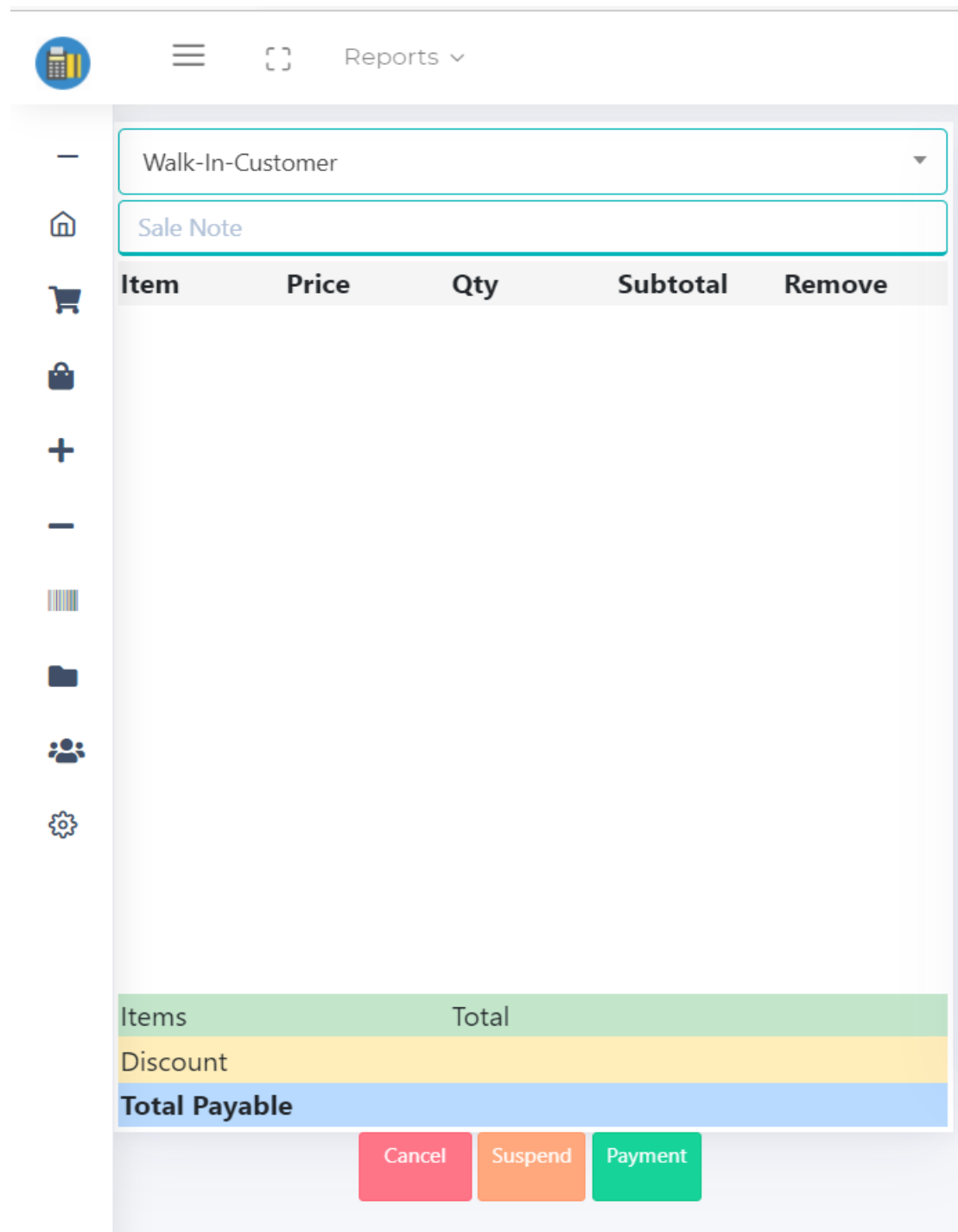


Figure-9: Point of sales Prototype

Settings Prototype of point of sales:

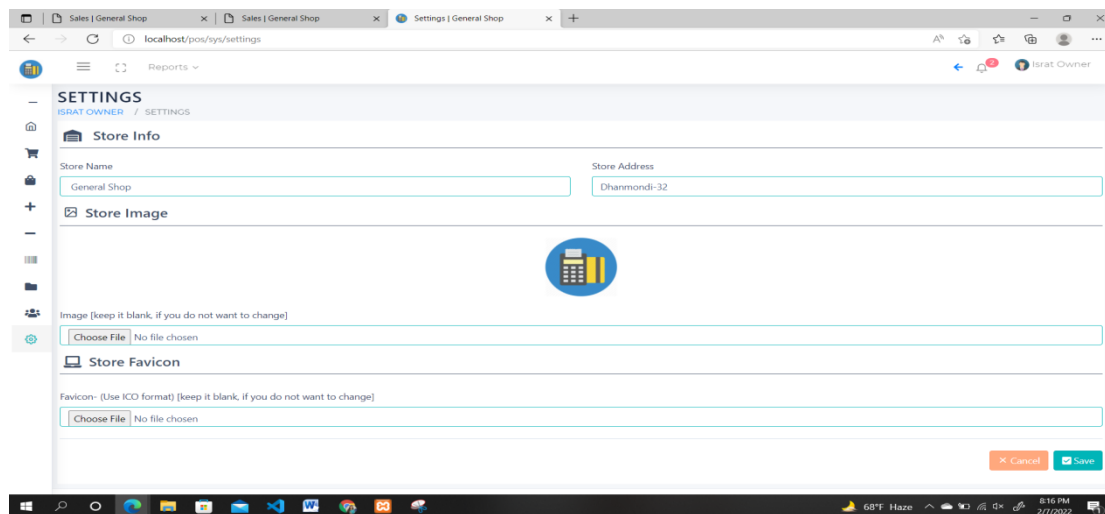


Figure 10: Settings Prototype

Chapter 9 – Engineering

9.1 New System Modules

(Scott Tilley & Harry J. Rosenblatt, February 5, 2016)

9.1.1 User Requirements of the system Modules

- q. The owner can add, edit and delete users (another owner or staff) and permission.
- r. The owner can view various report
- s. The owner can view today's sales details
- t. The owner can view the top 5 most selling items as the chart
- u. The owner can manage a sale (edit, delete or void a sale)
- v. The owner can view all categories and add, edit and delete the category
- w. The owner can view the product list and add, edit, and delete products
- x. The owner can view, add, edit and delete purchases
- y. The owner can view expense and add, edit, search and delete Expense
- z. The owner can view, add, edit and delete vendor
- aa. The owner can view, add, edit and delete customer
- bb. Staff can manage a sale (edit, delete or void a sale)
- cc. Owner and Staff can view all sales details
- dd. Staff can view categories
- ee. Staff can view products
- ff. Staff can view, add, update, delete expense

9.1.2 The New system Modules

A. The owner can add, edit and delete user

- a. Owner login to the system with proper login credentials
- b. Then click on the (People - Employees) button from the left menu bar.
- c. A list will show all the users their information.
- d. By clicking on the add user button, a new popup will appear and after filling up all required fields owner will click on the Save button to add a new user.
- e. By clicking on the edit button from the user list new popup window will appear to update user information.
- f. By clicking on the delete button user will be deleted.

B. The owner can view various report

- a. The owner clicks on the sales report/customer reports from the top bar dropdown.
- b. By clicking on an a- date range to see all the information of the selected date range.

C. Owner can view today's sales details.

- a. The owner clicks on sales report/customer reports from the top bar.
- b. Show daily report as default.

D. Owner can view the top 10 most selling items as the chart

- a. The manager clicks on the dashboard button from the left bar or navigates to the dashboard manually.
- b. The system will show the top 10 sales product in a donut chart with their name and quantity stored in the warehouse.

E. Owner can manage a sale (edit, delete or void a sale)

- a. By clicking the POS button from the left bar menu owner navigate to the pos main screen.
- b. From the dropdown select customer, enter the new reference.
- c. From the right side select the item (product), then it will show in the left table with its price and other details. d. Set manual price if the owner wants to release price a little from the cell of that product.
- d. Set quantity from cell or quantity can be increased by clicking on the same product from the right side again and again.
- e. The owner can set a discount from the subtotal price if needed. This is a hidden field, can be active by clicking on that area.
- f. The owner can add a note for this sale by clicking on the comment icon button near the Total payable area.
- g. The owner can suspend a sale if multiple customers come by clicking on the suspend button. This will lead the user to another pos display.
- h. The owner can only access the pos display once for each customer. That means no session is included.
- i. By clicking on the cancel button user will cancel the current sale.
- j. By clicking on the payment button data about the sale will be saved into the corresponding table and navigated to invoice.
- k. Invoice can be printed by clicking on the print button.

F. Owner can view all categories and add, edit and delete the category

- a. By clicking on the delete icon owner can delete the specific category.
- b. By clicking on the category button from the left bar, the owner can navigate to the category list.
- c. The category list will show all the information about each category in a table.
- d. The owner can search individual fields under the table header.
- e. By clicking on Add Category button at the top of the category table, the owner will navigate to the new page.
- f. By filling in the necessary information, a new category can be added with an image(optional) and the owner will be redirected to the category list page.
- g. By clicking the edit icon, a popup will appear, and the owner can edit and update the existing category.

G. Owner can view product list and add, edit, and delete products

- a. By clicking on the product button from the left bar, the owner can navigate to the product list.
- b. The product list will show all the information about each product in a table.
- c. The owner can search individual fields under the table header.
- d. By clicking on Add Product button at the top of the product table, the owner will navigate to the new page.
- e. By filling in the necessary information, a new product can be added with the image(optional) and the owner will be redirected to the product list page.
- f. By clicking the edit icon, a popup will appear, and the owner can edit and update the existing product.
- g. By clicking on the delete icon owner can delete the specific product.

H. Owner can view, add, edit and delete purchases

- a. By clicking on the purchases button from the left bar, the owner can navigate to the purchase button from the left bar, the owner can navigate to the purchases list.
- b. The purchases list will show all the information about each purchase in a table.
- c. The owner can search individual fields under the table header.

- d. By clicking on Add Purchase button at the top of the purchases table, the owner will navigate to the new page.
- e. The owner can select for which product purchase will be made and from which vendor.
- f. By filling in the necessary information, a new purchase can be added, and product warehouse will be updated, and the owner will be redirected to the purchase list page.
- g. By clicking the edit icon, a popup will appear, and the owner can edit and update existing purchases.
- h. By clicking on the delete icon owner can delete a specific purchase and it will affect the product table as well.

I. Owner can view expense and add, edit, search and delete expense

- a. By clicking on the expense button from the left bar, the owner can navigate to the expense list.
- b. The expense list will show all the information about each expense in a table.
- c. The owner can search individual fields under the table header.
- d. By clicking on Add Expense button at the top of the expense table, the owner will navigate to the new page.
- e. By filling in the necessary information, a new expense can be added, and the owner will be redirected to the expense list page.
- f. By clicking the edit icon, a popup will appear, and the owner can edit and update existing expenses.
- g. By clicking on the delete icon owner can delete specific expenses.

J. Owner can view, add, edit and delete vendor

- a. By clicking on the vendor button from the left bar, the owner can navigate to the vendor list.
- b. The vendor list will show all the information about each vendor in a table.
- c. The owner can search individual fields under the table header.
- d. By clicking on Add Vendor button at the top of the vendor table, the owner will navigate to the new page.
- e. By filling in the necessary information, a new vendor can be added, and the owner will be redirected to the vendor list page.
- f. By clicking the edit icon, a popup will appear, and the owner can edit and update the existing vendor.
- g. By clicking on the delete icon owner can delete a specific vendor.

K. Owner can view, add, edit and delete customer

- a. By clicking on the customer button from the left bar, the owner can navigate to the customer list.
- b. The customer list will show all the information about each customer on a table.
- c. The owner can search individual fields under the table header.
- d. By clicking on Add Customer button at the top of the customer table, the owner will navigate to the new page.
- e. By filling in the necessary information, a new customer can be added, and the owner will be redirected to the customer list page.
- f. By clicking the edit icon, a popup will appear, and the owner can edit and update the existing customer.
- g. By clicking on the delete icon owner can delete the specific customer.

L. Staff can manage a sale (edit, delete or void a sale)

- a. By clicking the POS button from the left bar menu staff navigate to the pos main screen.
- b. From the dropdown select customer, enter the new reference. c. From the right side select the item (product), then it will show in the left table with its price and other details.
- c. Set manual price if staff wants to release price a little from the cell of that product.
- d. Set quantity from cell or quantity can be increased by clicking on the same product from the right side again and again.
- e. Staff can set discounts from the subtotal price if needed. This is a hidden field, can be active by clicking on that area.
- f. Staff can add a note for this sale by clicking on the comment icon button near the Total payable area.
- g. Staff can suspend a sale if multiple customers come by clicking on the suspend button. This will lead the user to another pos display.
- h. Staff can only access the pos display once for each customer. That means no session is included.
- i. By clicking on the cancel button user will cancel the current sale.
- j. By clicking on the payment button data about the sale will be saved into the corresponding table and navigated to invoice.
- k. Invoice can be printed by clicking on the print button.

M. Owner and Staff can view all sales details

- a. By clicking on the Sales button from the left bar user will navigate to the sales page where all the sales information is shown in a table.
- b. Users can search individual fields under the table header.

N. Staff can view categories

- a. By clicking on the Categories button from the left bar user will navigate to the categories page where all the categories information are shown in a table.
- b. Users can search individual fields under the table header.

O. Staff can view products

- a. By clicking on the Products button from the left bar user will navigate to the products page where all the products information is shown in a table.
- b. Users can search individual fields under the table header.

P. Staff can view, add, update, delete expense

- a. By clicking on the expense button from the left bar, staff can navigate to the expense list.
- b. The expense list will show all the information about each expense in a table.
- c. Staff can search individual fields under the table header.
- d. By clicking on Add Expense button at the top of the expense table, staff will navigate to the new page.
- e. By filling in the necessary information, a new expense can be added, and staff will be redirected to the expense list page.
- f. By clicking the edit icon, a popup will appear, and staff can edit and update existing expenses.
- g. By clicking on the delete icon staff can delete specific expenses.

9.1.3 Functional Requirements of the New System

- a. All products should show first as default.
- b. 4 row 4 column products should be shown.
- c. An image should not be more than 1 MB
- d. Every table should have a search, pagination, order filter.
- e. The Time zone will be based on the company.
- f. The staff has no access to delete or edit or show any reports.
- g. The staff has no access to add, delete or edit products, purchases, categories.

9.1.4 Non- Functional Requirements of the New System

- a. Easy user interface
- b. Responsive design
- c. No internet connection needed
- d. Don't need other software to run
- e. High security

9.2 Use Case Diagram

A use case diagram is a visual representation of how the parts of a system interact. A use case is a system analysis methodology for identifying, clarifying, and organizing system needs.

Actor: An actor is a user of the system who takes on a certain role.

Use case: A use case is a specific task that a user can do on the system.

Relationship: Relationships are depicted easily by drawing a line between actors and use cases.

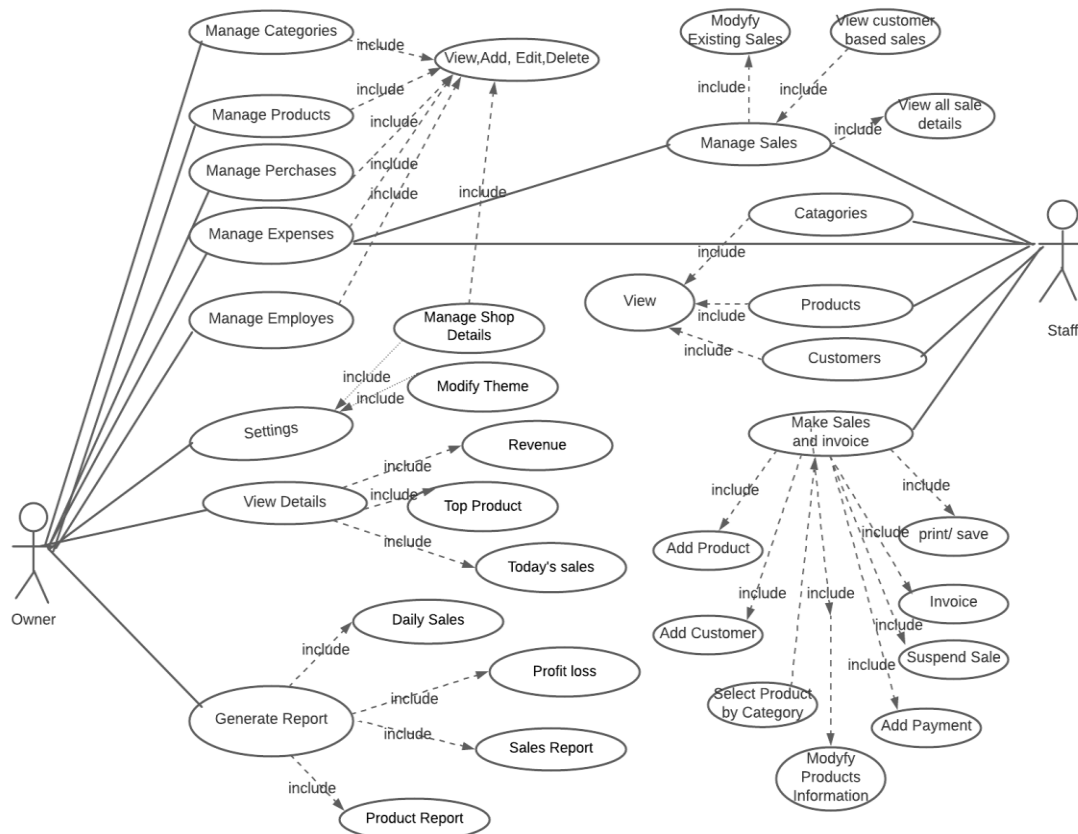


Figure 11: Use case diagram

9.2.1 Use Case Description

Use case Title: Manage User
Actor: Owner
Description: The owner will create and modify the user for the system.

Use case Title: Manage User Permission
Actor: Owner and Staff
Description: The user will set up the theme and another setting with the given permission.

Use case Title: View Details
Actor: Owner
Description: The owner will be able to see daily sales information with revenue charts and top-selling items.

Use case Title: Generate Report
Actor: Owner
Description: The owner can see sales, product, profit, and daily sales reports.

Use case Title: Manage Product
Actor: Owner
Description: The owner can view and modify products.

Use case Title: Manage Purchases
Actor: Owner
Description: The owner can view and modify purchases.

Use case Title: Manage Customers
Actor: Owner
Description: The owner can view and modify customers.

Use case Title: Manage Vendors
Actor: Owner
Description: The owner can view and modify vendors.

Use case Title: Manage Vendors
Actor: Owner
Description: The owner can view and modify vendors.

Use case Title: Settings
Actor: Owner
Description: Owner can manage and shop setting and theme

Use case Title: Manage Category
Actor: Owner
Description: The owner can view and modify categories.

Use case Title: Manage Sale
Actor: Staff and Owner
Description: Staff and Owner can see/modify sales, see selected customer details and add payment to selected payment.

Use case Title: Make sales and invoice
Actor: Owner and Staff
Description: Owner and Staff can add/ search/ modify products, suspend the sale, add payment, and print bills.

9.3 Class Diagram

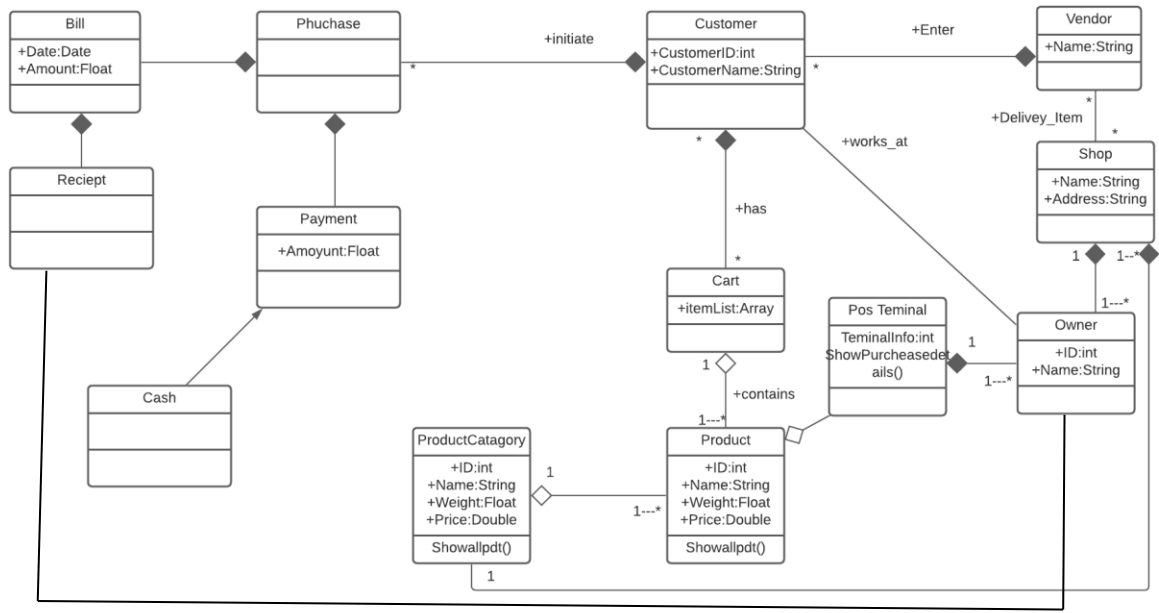


Figure 12: Class Diagram

9.4 Entity Relationship Diagram

An entity-relationship diagram (ERD) is a data modeling technique that current step the entities of an information system as well as their relationships. The entity framework infrastructure is represented by an ERD, which is a conceptual and representational model of data.

(ERD)

An ER Diagram contains three properties

- ❖ Entities: Within a system, an entity is a definite thing or idea.
- ❖ Attributes: A property or characteristic of the entity that owns it is called an attribute.
- ❖ Relationships: The term "relationship" refers to the fact that two entities are linked in some way.

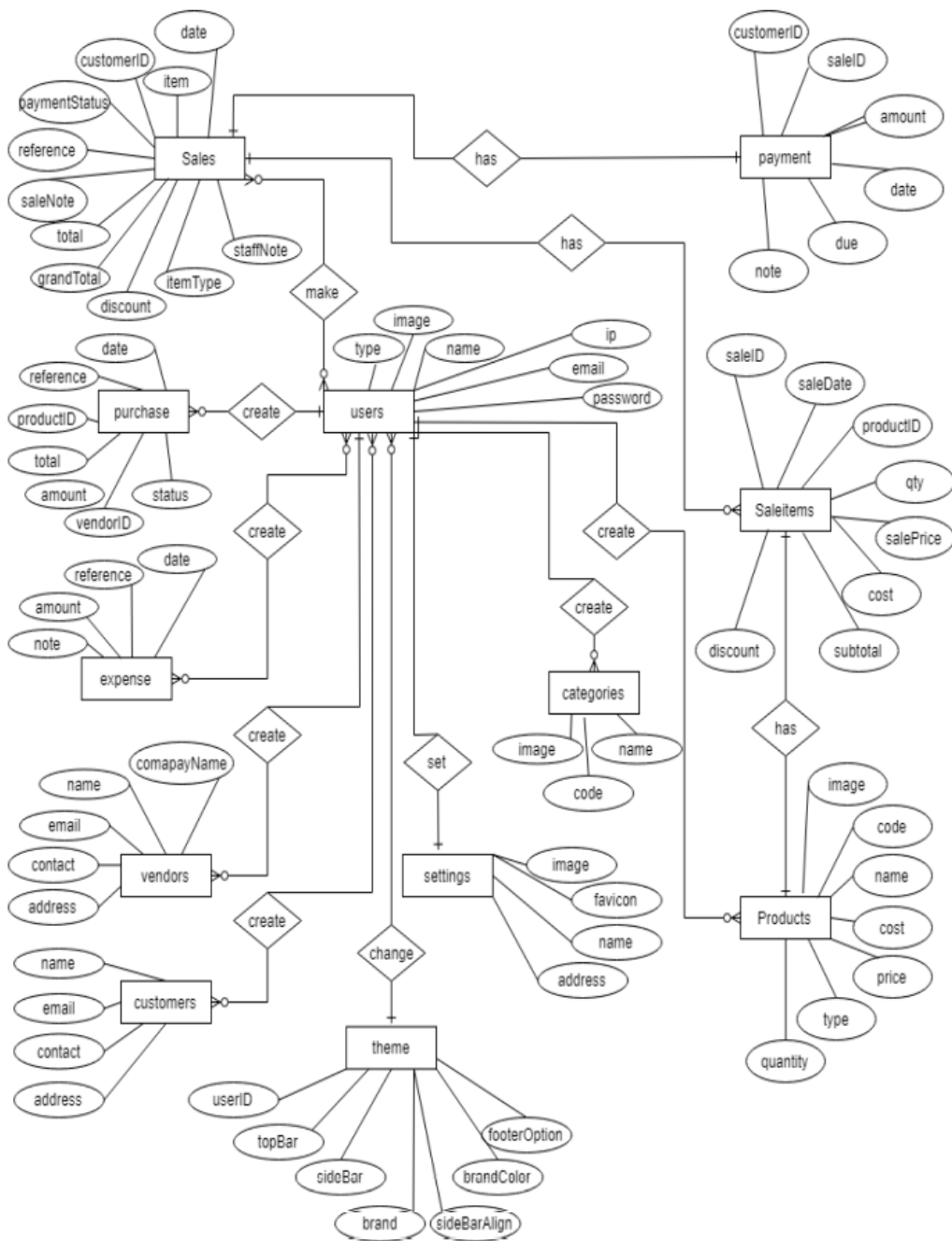


Figure 13: Entity Relationship Diagram

9.5 Sequence Diagram

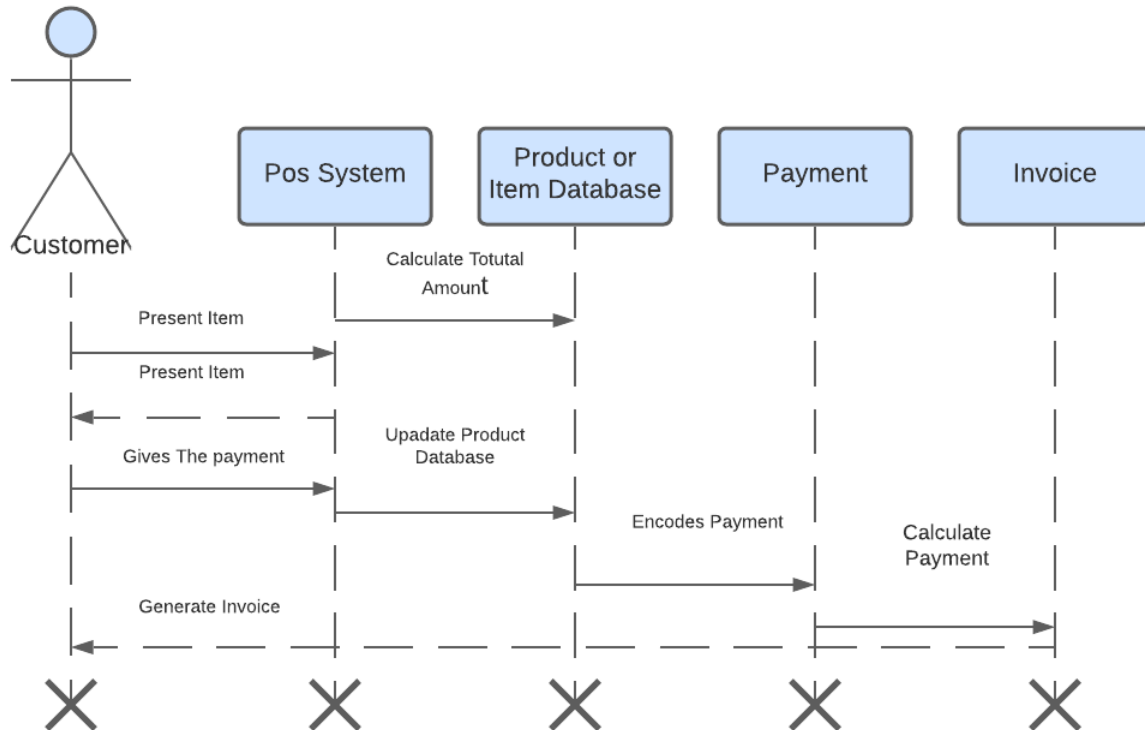


Figure 14: Sequence Diagram

9.6 System Interface Design

Dashboard Interface design

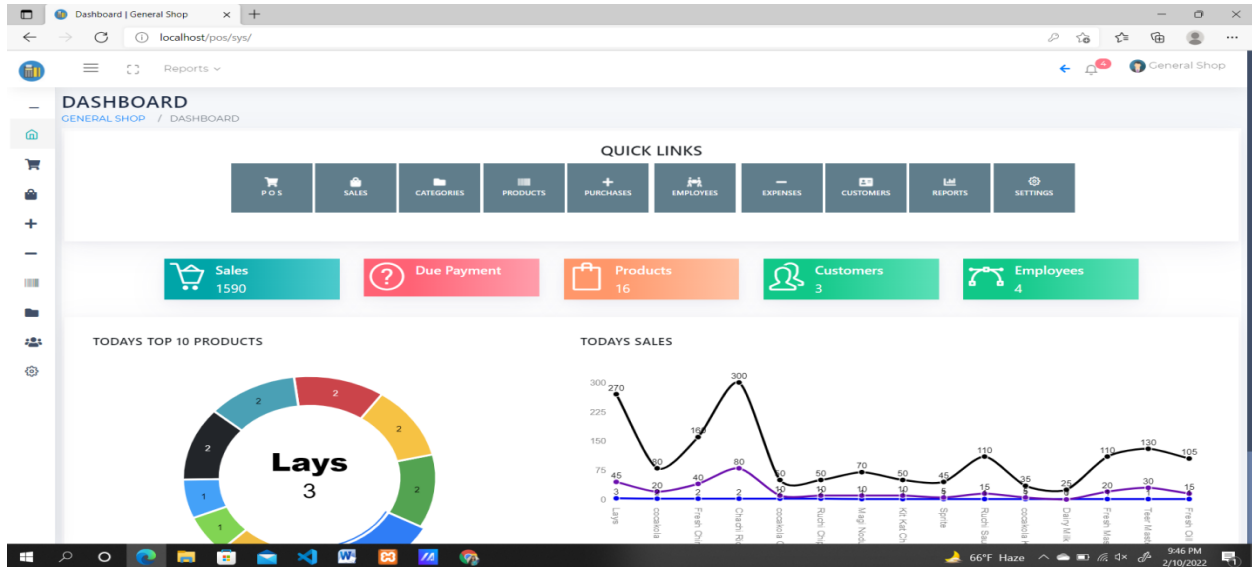
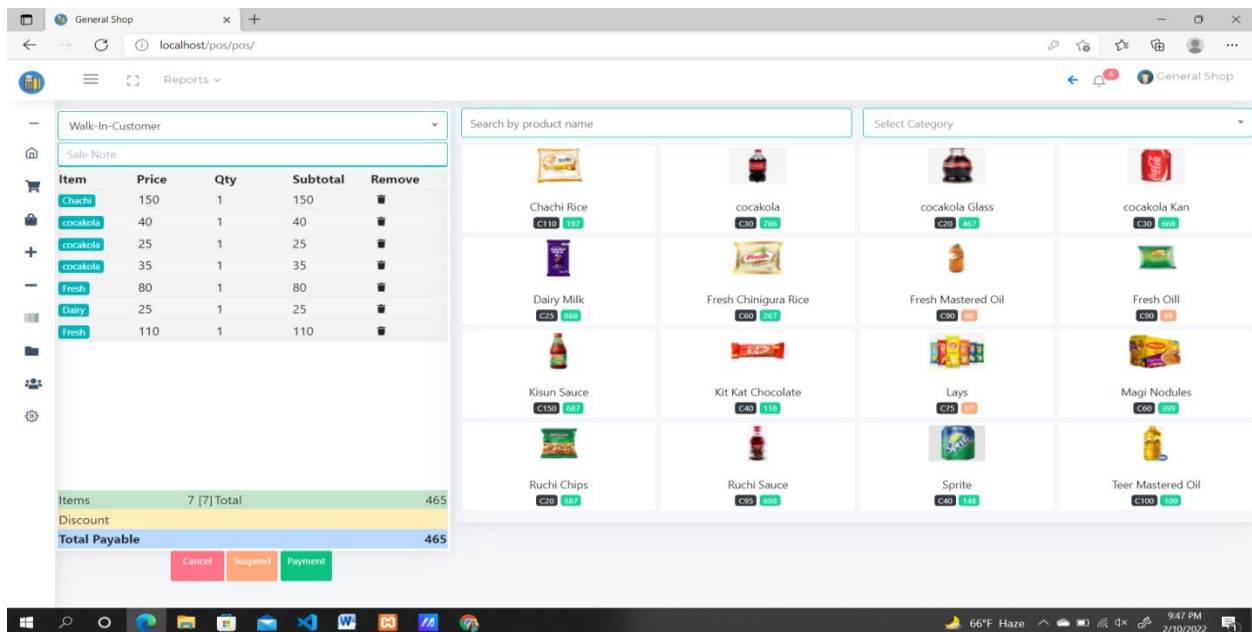


Figure 15: Dashboard Interface design

Point of sale Interface Design



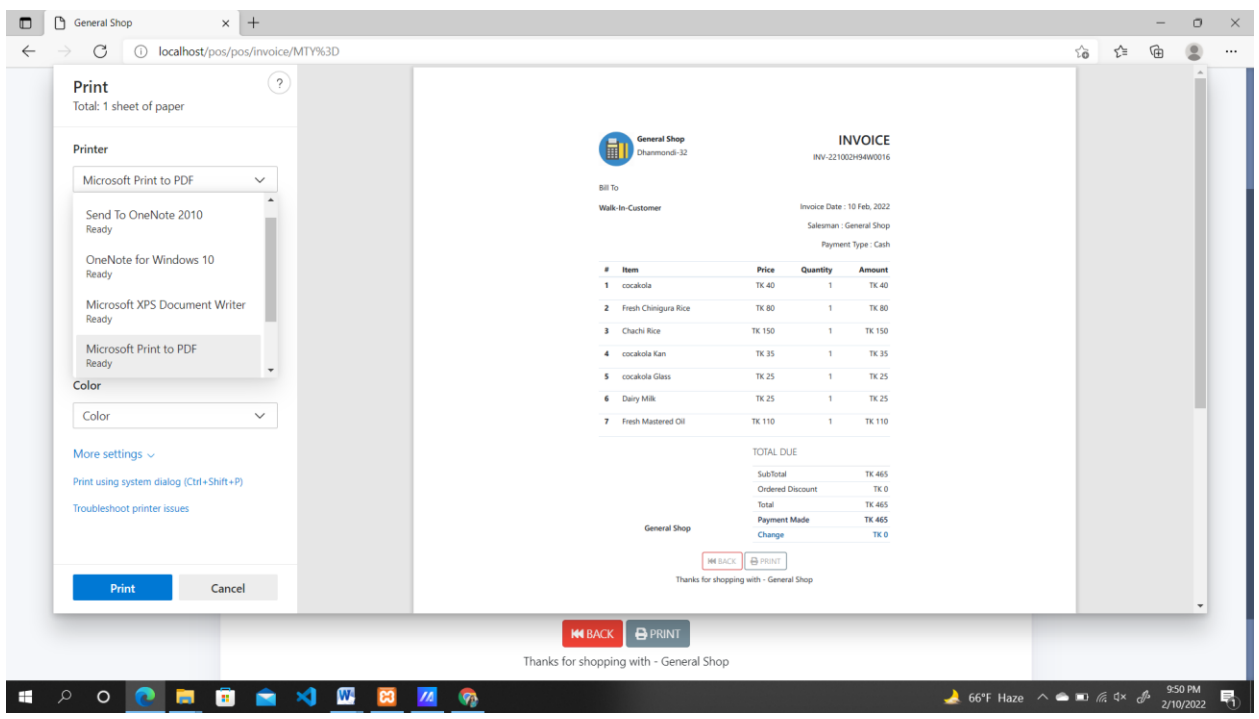
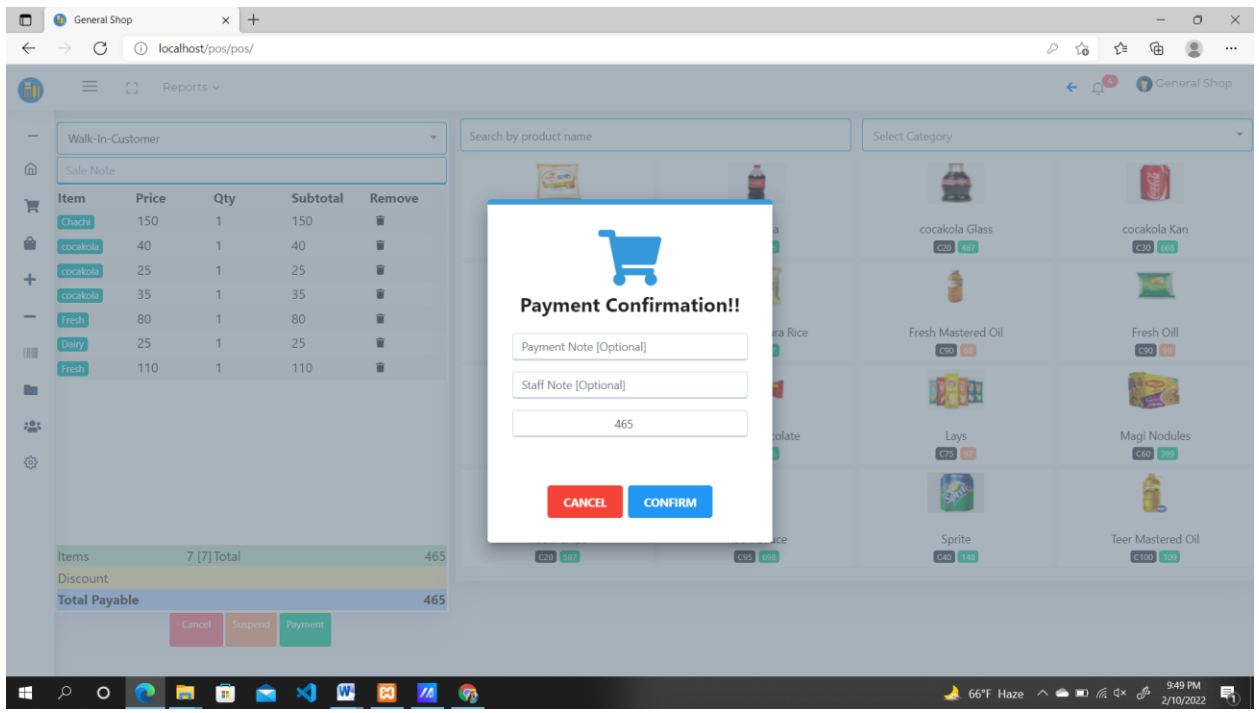


Figure 16: Point of sale Interface Design

Customer sales interface design:

Date	Reference	Customer	Discount	Grand Total	Status	Action
Thu, Feb 10, 2022-09:49 PM	INV-221002H94W0016	Walk-In-Customer	TK 0	TK 465	Paid	
Thu, Feb 10, 2022-07:04 PM	INV-221002UY5V0015	Walk-In-Customer	TK 0	TK 1020	Paid	
Thu, Feb 10, 2022-07:03 PM	INV-221002O1ZV0014	Walk-In-Customer	TK 0	TK 570	Paid	
Wed, Feb 09, 2022-03:49 PM	INV-220902TH9N0010	Walk-In-Customer	TK 0	TK 645	Paid	
Wed, Feb 09, 2022-02:53 PM	INV-22090285LR0012	Walk-In-Customer	TK 10	TK 160	Paid	
Wed, Feb 09, 2022-02:41 PM	INV-220902HHW00011	Walk-In-Customer	TK 0	TK 80	Paid	
Wed, Feb 09, 2022-02:40 PM	INV-220902L5SF0010	Walk-In-Customer	TK 0	TK 490	Paid	
Wed, Feb 09, 2022-02:13 PM	INV-2209026L0Y0009	Walk-In-Customer	TK 0	TK 10360	Paid	
Tue, Feb 08, 2022-02:23 AM	INV-2208029V380008	Walk-In-Customer	TK 0	TK 90	Paid	
Tue, Feb 08, 2022-02:22 AM	INV-220802ZQD50007	Walk-In-Customer	TK 0	TK 540	Paid	
			Total: 10	Total: 14400		

Figure 16: Customer sales interface design

Product Sales history interface design:

Date	Product Name	Quantity	Subtotal
Thu, Feb 10, 2022-09:49 PM	Fresh Mastered Oil	1	TK 110
Thu, Feb 10, 2022-09:49 PM	cocakola	1	TK 40
Thu, Feb 10, 2022-09:49 PM	Fresh Chinigura Rice	1	TK 80
Thu, Feb 10, 2022-09:49 PM	Chachi Rice	1	TK 150
Thu, Feb 10, 2022-09:49 PM	cocakola Kan	1	TK 35
Thu, Feb 10, 2022-09:49 PM	cocakola Glass	1	TK 25
Thu, Feb 10, 2022-09:49 PM	Dairy Milk	1	TK 25
Thu, Feb 10, 2022-07:04 PM	Ruchi Sauce	1	TK 110
Thu, Feb 10, 2022-07:04 PM	Sprite	1	TK 45
Thu, Feb 10, 2022-07:04 PM	Lays	1	TK 90
		Total: 10	Total: 710

Figure 17: Product Sales history

Product Purchases interface design

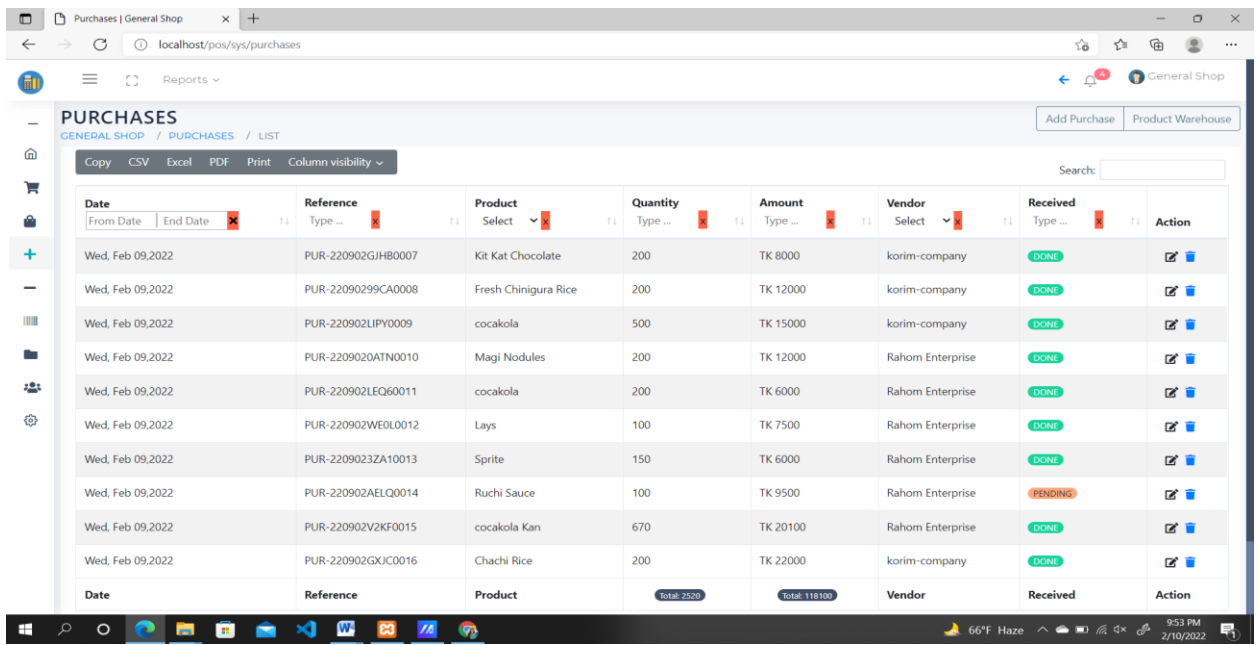


Figure 18: Product Purchases interface design

Category Interface design:

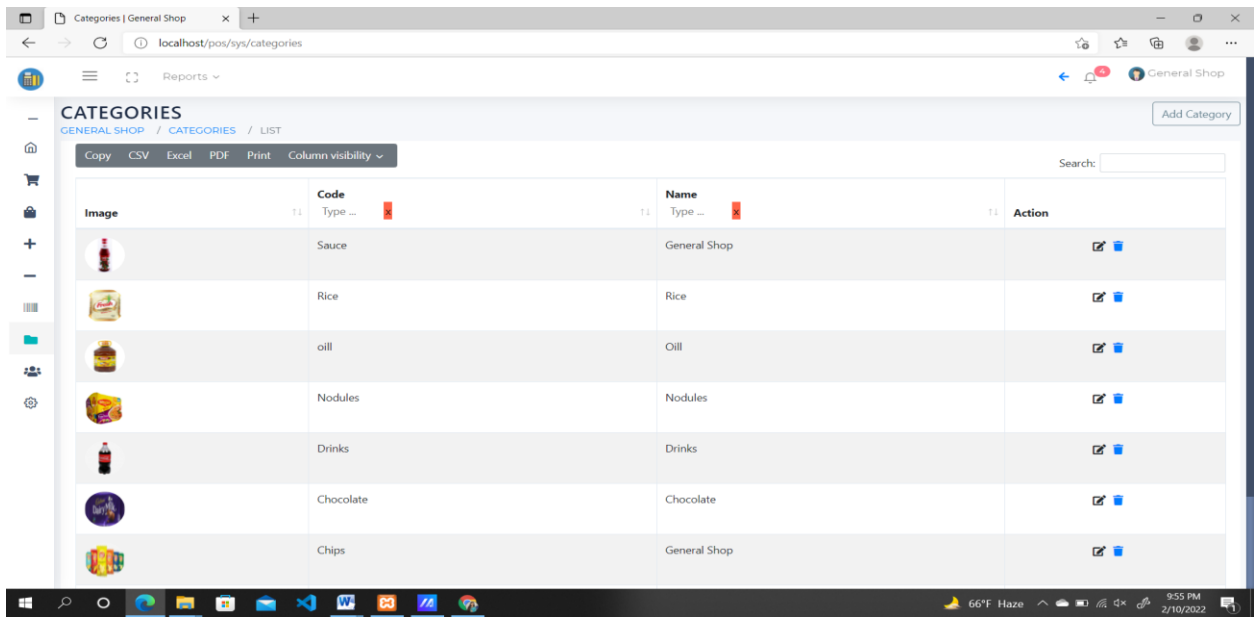


Figure 19: Category Interface design

Employee Interface design:

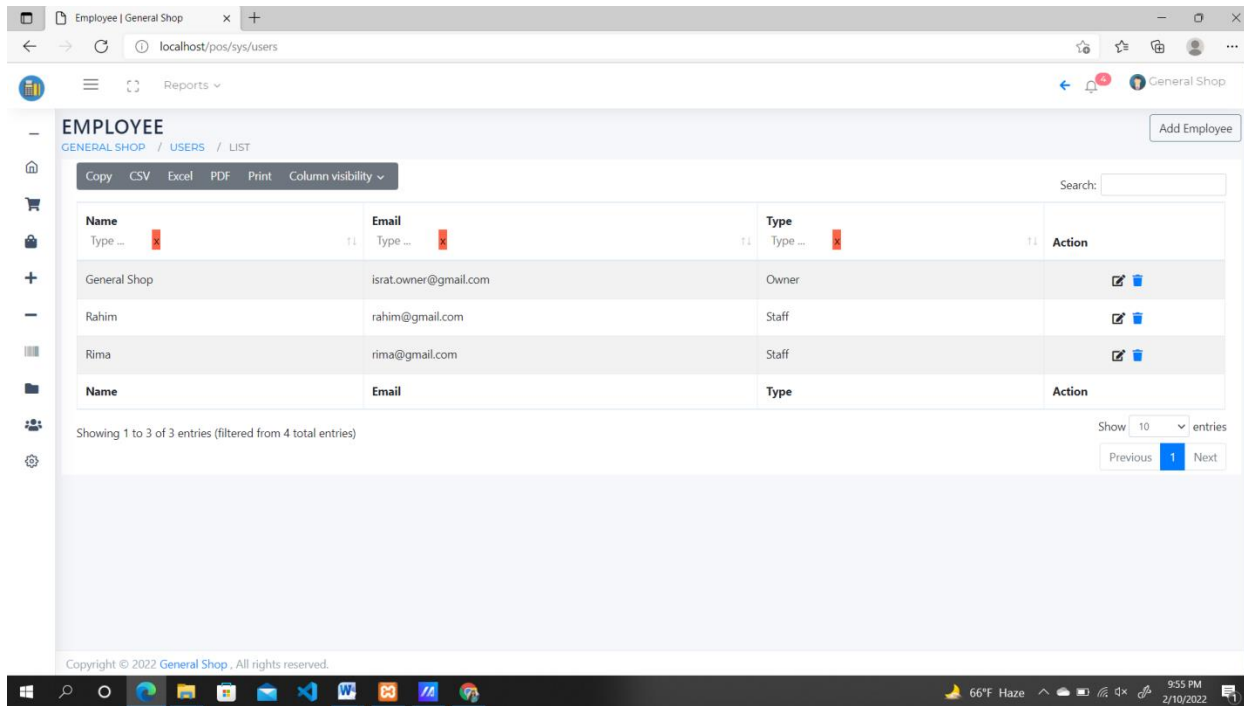


Figure 20: Employee Interface design

Customer interface design:

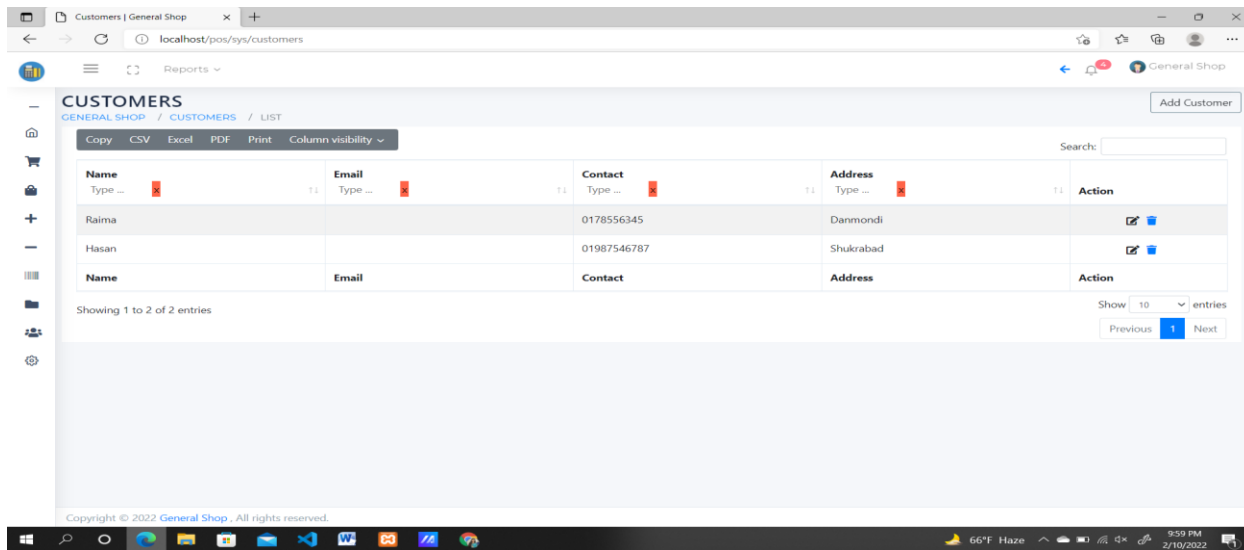


Figure 21: Customer interface design

Vendors interface page design

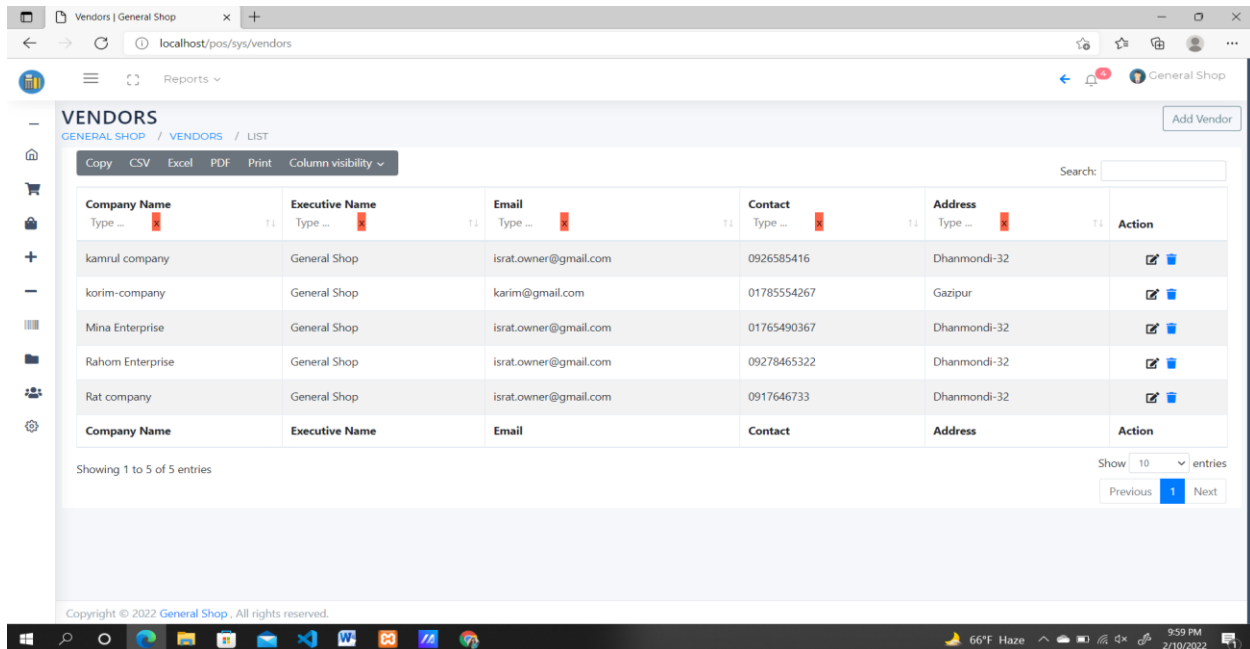


Figure 22: Vendors interface page design

Settings Interface page design

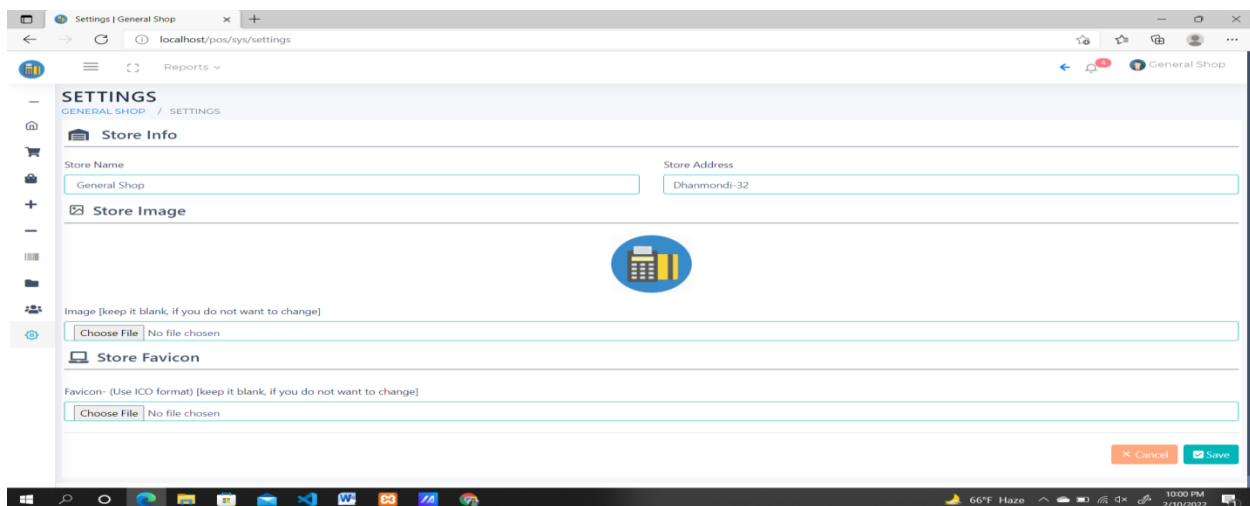
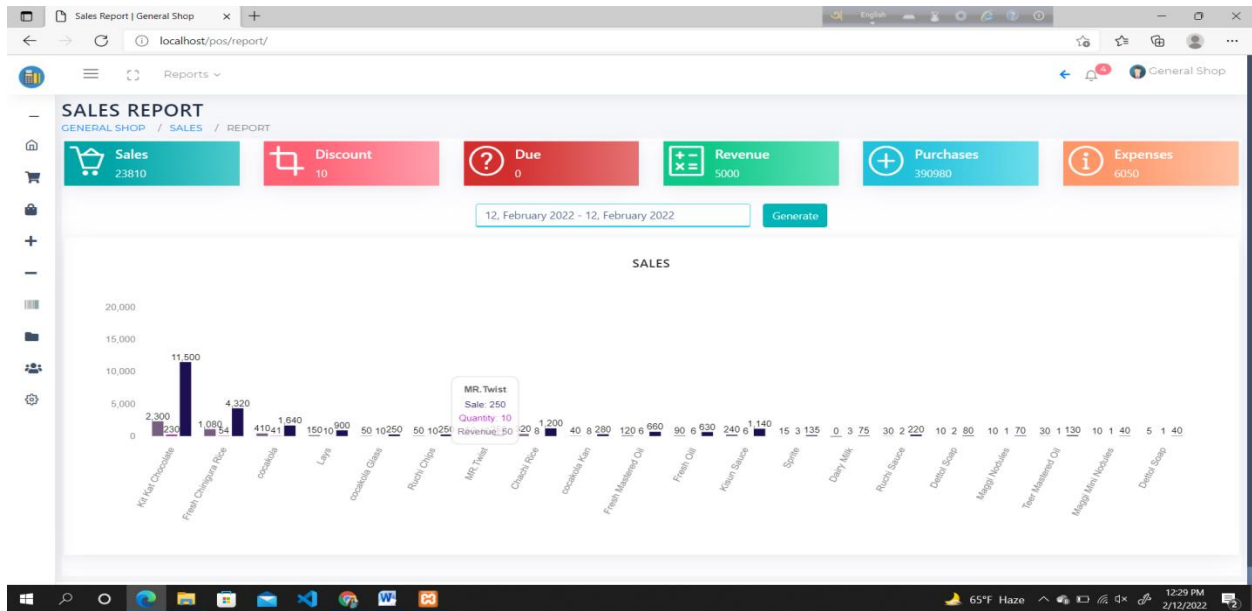
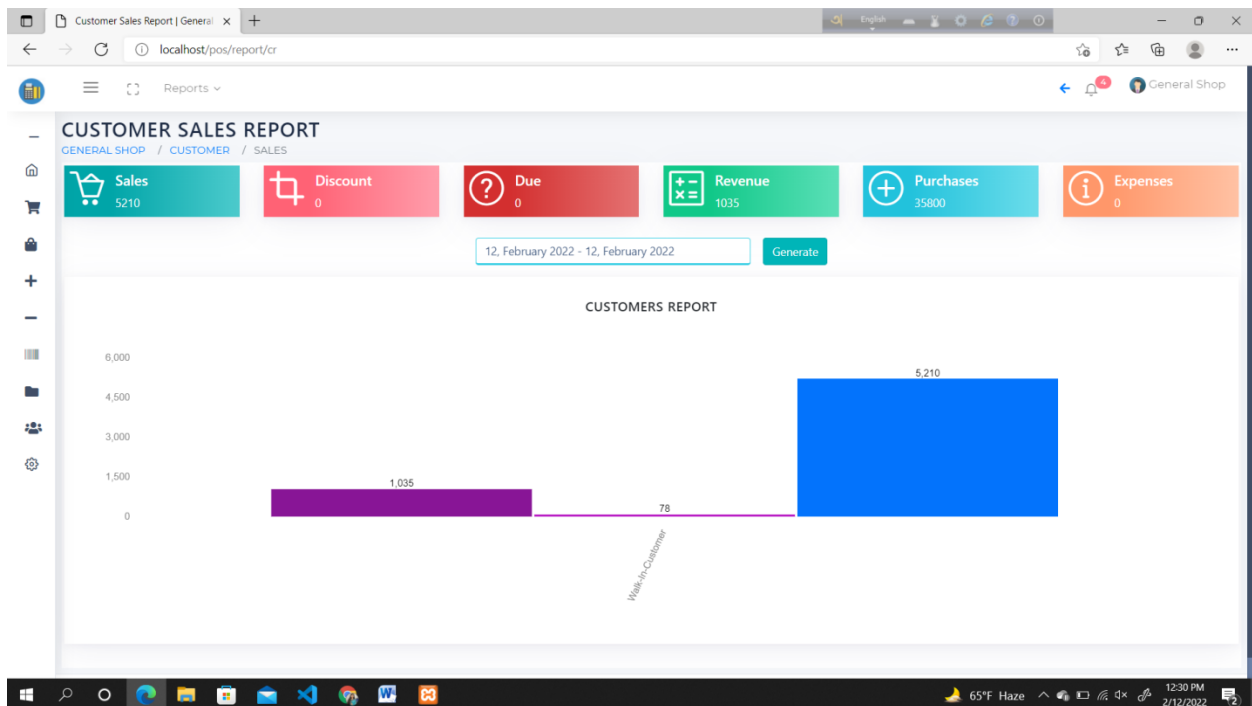


Figure 23: Settings Interface page design

Sales Report:



Customer Report:



Chapter 10 – Deployment

10.1 Core Module Coding Sample

I have to use the PHP(Codeigniter3) framework and used MySQL as a database movement system. There some important codes are given below:

Login Page for the user:

```

1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5
6  /**
7   * @property AuthModel $authModel Description
8   */
9  class Auth extends TQ_Controller {
10
11     public $viewPath = "auth/", $modalPath = "auth/modal/";
12
13     public function __construct() {
14         parent::__construct();
15         $this->load->model("authModel");
16     }
17
18
19     function password() {
20         dnp(getEncryptedText('2021'));
21     }
22
23     function index() {
24         $this->ifLogin();
25         $this->login();
26     }
27
28     function login() {
29         $this->navBarSettings(0, 0, 1,1);
30         $this->form_validation->set_rules('email', 'Email', 'required');
31         $this->form_validation->set_rules('password', 'Password', 'required|min_length[4]');
32
33         if ($this->form_validation->run()) {
34             $email = $this->input->post("email");
35             $pass = $this->input->post("password");
36             if ($user = $this->authModel->loginAuth($email)) {
37                 if (getDecryptedText($user->password) == $pass) {
38                     $user = (array)$user;
39                     unset($user["password"]);
40                     //dnp($user);
41                     $this->session->sess_expiration = '60';
42                     $this->session->sess_expire_on_close = 'true';
43                     $this->session->set_userdata("user", (object)$user);
44                     $this->goToUrl(sysURL(), "Login Succeed! Welcome " . currentUser() . " | Your are logged in as " . currentUserType() . " |", SUCCESS);
45                 } else {
46                     $this->goToReference("<br>Password not matched", DANGER);
47                 }
48             } else {
49                 $this->goToReference("<br>User not exists", DANGER);
50             }
51         }
52         $this->viewPath(__FUNCTION__);
53     }
54 }
55

```

Figure 24: Login Page

Dashboard coding:

```
public function dashboard() {
    $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => "Dashboard", "bc" => array(
        ["url" => sysUrl(), "page" => currentUser(), ["url" => "", "page" => __FUNCTION__]
    ));
    $this->data["totalCustomers"] = $this->sysModel->countTotal(TABLE_CUSTOMERS, withoutDeleted());
    $this->data["totalProducts"] = $this->sysModel->countTotal(TABLE_PRODUCTS, withoutDeleted());
    $this->data["totalDue"] = $this->sysModel->getData(TABLE_PAYMENTS, ("DATE(addedDate) = " . date("Y-m-d", strtotime(changeDateFormat("now"))) . " AND deleted = 0"));
    $this->data["totalUsers"] = $this->sysModel->countTotal(TABLE_USERS, withoutDeleted());
    $this->data["totalSales"] = $this->sysModel->getTotalSum(TABLE_SALES, "grandTotal", ("DATE(date) = " . date("Y-m-d", strtotime(changeDateFormat("now"))) . " AND deleted = 0"));
    $this->data["todaysSales"] = $this->todaysSales();
    $this->data["todaysTopTenProducts"] = $this->todaysTopTenProducts();
    $this->viewPath(__FUNCTION__);
}

public function todaysSales() {
    $sales = [];
    $date = changeDateFormat("now");
    $dt = $this->sysModel->getDataJoin(TABLE_SALEITEMS, ("DATE(saleDate) BETWEEN " . date("Y-m-d", strtotime($date)) . " and " . date("Y-m-d", strtotime($date)) . " AND deleted = 0"),
        TABLE_PRODUCTS, ("products.id = saleitems.productId"), [{"sum(saleitems.qty)" => "DESC"}], 10,
        ("sum(saleitems.orgSubTotal) as orgSubTotal, sum(saleitems.subTotal) as subTotal, sum(saleitems.qty) as qty, products.name as name"),
        0, ("saleitems.productId"));
    if ($dt) {
        foreach ($dt as $pro) {
            array_push($sales, (object)["label" => $pro->name, "sale" => $pro->subTotal, "qty" => $pro->qty, "profit" => ($pro->subTotal - $pro->orgSubTotal)]);
        }
    }
    return $sales;
}

public function todaysTopTenProducts() {
    $topTen = [];
    $date = changeDateFormat("now");
    $dt = $this->sysModel->getDataJoin(TABLE_SALEITEMS, ("DATE(saleDate) BETWEEN " . date("Y-m-d", strtotime($date)) . " and " . date("Y-m-d", strtotime($date)) . " AND deleted = 0"),
        TABLE_PRODUCTS, ("products.id = saleitems.productId"), [{"sum(saleitems.qty)" => "DESC"}], 10,
        ("sum(saleitems.qty) as qty, products.name as name"), 0, ("saleitems.productId"));
    if ($dt) {
        foreach ($dt as $pro) {
            array_push($topTen, (object)["label" => $pro->name, "value" => $pro->qty]);
        }
    }
    return $topTen;
}

/*
 * Dashboard end
 */
```

```
public function todaysTopTenProducts() {
    $topTen = [];
    $date = changeDateFormat("now");
    $dt = $this->sysModel->getDataJoin(TABLE_SALEITEMS, ("DATE(saleDate) BETWEEN " . date("Y-m-d", strtotime($date)) . " and " . date("Y-m-d", strtotime($date)) . " AND deleted = 0"),
        TABLE_PRODUCTS, ("products.id = saleitems.productId"), [{"sum(saleitems.qty)" => "DESC"}], 10,
        ("sum(saleitems.qty) as qty, products.name as name"), 0, ("saleitems.productId"));
    if ($dt) {
        foreach ($dt as $pro) {
            array_push($topTen, (object)["label" => $pro->name, "value" => $pro->qty]);
        }
    }
    return $topTen;
}

/*
 * Dashboard end
 */
```

Figure 25: Dashboard coding

User role's functional program snippet:

```

84  * user start
85  */
86
87  public function users() {
88      $this->ifNotOwner();
89      $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => "Employee", "bc" => array(
90          ["url" => sysUrl(), "page" => currentUser(), ["url" => sysUrl(__FUNCTION__), "page" => __FUNCTION_], ["url" => "", "page" => "LIST"]
91      )];
92      $this->viewPath("people/users/" . __FUNCTION__);
93  }
94
95  public function getUsers() {
96      $this->ifNotOwner();
97      $filt = "<span>Are you sure?</span><a class='btn btn-danger po-delete btn-sm p-1 rounded-0' href='' . sysUrl('removeUser/$1') . ''>I am sure</a> <button class='btn pop-c
98      $extra = '<a class = "btn btn-link p-0 px-1" modal-toggle="true" data-target="#remotemodal" data-remote="" . sysUrl('updateUser/$1') . ''><i class="fa fa-edit">
99      $extra .= '<button type="button" class="btn btn-link p-0 px-1" data-container="body" data-toggle="popover" data-placement="left" data-html="true" data-content=""
100      $action = "<div class = \"text-center\">
101      . $extra
102      . "</div>";
103      $this->datatables->select('id, name, email, type, addedBy, addedTime, ip, deleted')
104          ->from(TABLE_USERS)
105          ->addColumn('actions', $action, 'id')
106          ->where(['id !=> 1])
107          ->where(withoutDeleted())
108          ->generate();
109      return true;
110  }
111
112  public function checkUserAvailability() {
113      $checkUserName = $this->input->post('name');
114      if ($this->sysModel->is_available(TABLE_USERS, ['name' => $checkUserName, 'deleted' => 0])) {
115          echo '<label class="text-danger"><span class="glyphicon glyphicon-ok"></span>Email taken. Choose another</label>';
116      } else {
117          echo '<label class="text-success"><span class="glyphicon glyphicon-remove"></span>Available</label>';
118      }
119  }

```

```

121  public function checkEmailAvailability() {
122      $checkUserEmail = $this->input->post('email');
123      if ($this->sysModel->is_available(TABLE_USERS, ['email' => $checkUserEmail, 'deleted' => 0])) {
124          echo '<label class="text-danger"><span class="glyphicon glyphicon-ok"></span>Email taken. Choose another</label>';
125      } else {
126          echo '<label class="text-success"><span class="glyphicon glyphicon-remove"></span>Available</label>';
127      }
128  }
129
130  public function addUser() {
131      $this->ifNotOwner();
132      $this->form_validation->set_rules('email', 'Email', 'required');
133      if ($this->form_validation->run()) {
134          $array = ['name', 'email', 'type'];
135          $data = [];
136          $themeData = [];
137          foreach ($array as $a) {
138              $data[$a] = $this->input->post($a);
139          }
140          $data['password'] = getEncryptedText($this->input->post("password"));
141          $data['addedTime'] = getCurrentTime();
142          $data['ip'] = getIPAddress();
143          $data['addedBy'] = currentUser();
144
145          $insertUser = $this->sysModel->insertData(TABLE_USERS, $data);
146
147          if ($insertUser) {
148              $themeData['userID'] = $insertUser;
149              $themeData['topBar'] = "navbar-light bg-white";
150              $themeData['sideBar'] = "menu-light";
151              $themeData['centerBrand'] = "";
152              $themeData['borderMenu'] = "";
153              $themeData['footerOption'] = "light";
154              $this->sysModel->insertData(TABLE_THEME, $themeData);
155              return $this->goToUrl(sysUrl('users'), $data["name"] . " is being registered as " . $data["type"], SUCCESS);
156          } else {
157              return $this->goToReference("Failed to add to server!!!", DANGER);

```

```

158  }
159  }
160  $this->modalPath("people/users/modal/" . __FUNCTION__);
161  }
162
163  public function updateUser($id) {
164      $this->ifNotOwner();
165      $this->form_validation->set_rules('email', 'Email', 'required');
166      if ($this->form_validation->run()) {
167          $array = ['name', 'email', 'type'];
168          $data = [];
169          foreach ($array as $a) {
170              $data[$a] = $this->input->post($a);
171          }
172          $data['ip'] = getIPAddress();
173          if ($this->sysModel->updateData(TABLE_USERS, $data, ['id' => $id])) {
174              return $this->goToUrl(sysUrl('users'), $data["name"] . " is being updated as " . $data["type"], SUCCESS);
175          } else {
176              return $this->goToReference("Failed to add to server!!!", DANGER);
177          }
178      }
179      $this->data["auth"] = $this->sysModel->getId(TABLE_USERS, ["id" => $id]);
180      $this->modalPath("people/users/modal/" . __FUNCTION__);
181  }
182
183  public function removeUser($id) {
184      $data['deleted'] = 1;
185      if ($this->sysModel->softRemoveData(TABLE_USERS, $data, ["id" => $id])) {
186          return $this->goToUrl(sysUrl("users"), "Successfully deleted !!!", SUCCESS);
187      } else {
188          return $this->goToReference("Failed to delete!!!", DANGER);
189      }
190  }
191
192  public function profile() {
193      $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => currentUser(), "bc" => array(
194          ["url" => sysUrl(), "page" => currentUser(), ["url" => "", "page" => __FUNCTION_]]

```

```

192 public function profile() {
193     $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => currentUserName(), "bc" => array(
194         ["url" => sysUrl(), "page" => currentUserName()], ["url" => "", "page" => __FUNCTION_]
195     )];
196
197     $this->data["profile"] = $this->sysModel->getId(TABLE_USERS, ["id" => currentUserID()]);
198     $this->data["theme"] = $this->sysModel->getId(TABLE_THEME, ["userID" => currentUserID()]);
199
200     $this->form_validation->set_rules('email', 'Email can not be blank', 'required');
201     $this->form_validation->set_rules('name', 'Name can not be blank', 'required');
202     if ($this->form_validation->run()) {
203         $array = ["name", "email"];
204         $themeArray = ["topBar", "sideBar", "centerBrand", "colorBrand", "borderMenu", "flippedSideBar", "footerOption"];
205         $data = [];
206         $themeData = [];
207         foreach ($array as $a) {
208             $data[$a] = $this->input->post($a);
209         }
210         foreach ($themeArray as $aa) {
211             $themeData[$aa] = $this->input->post($aa);
212         }
213         if ($this->upload->do_upload('image') != null) {
214             $img = $this->upload->data();
215             $data["image"] = $img["file_name"];
216             if ($this->data["profile"]->image) {
217                 unlink($img["file_path"] . $this->data["profile"]->image);
218             }
219         }
220         if ($this->input->post('password') != NULL) {
221             $data["password"] = getEncryptedText($this->input->post('password'));
222         }
223         if ($this->sysModel->updateData(TABLE_USERS, $data, ['id' => currentUserID()]) {
224             $this->sysModel->updateData(TABLE_THEME, $themeData, ['userID' => currentUserID()]);
225             $this->goToUrl(sysUrl("profile"), "Successfully updated profile!", SUCCESS);
226         } else {
227             return $this->goToReference("Failed to add to server!!!", DANGER);
228         }
229     }
230     $this->viewPath("people/users/" . __FUNCTION__);
231 }
232
233 /*
234 * User end
235 */

```

Figure 26: User Code

Customer Code:

```

237 /* Customer start
238 */
239
240 public function customers() {
241     $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => "Customers", "bc" => array(
242         ["url" => sysUrl(), "page" => currentUserName()], ["url" => sysUrl(__FUNCTION__), "page" => __FUNCTION__], ["url" => "", "page" => "LIST"]
243     )];
244     $this->viewPath("people/customers/" . __FUNCTION__);
245 }
246
247 public function getCustomers() {
248     $dit = "<p>Are you sure?</p><a class='btn btn-danger po-delete btn-sm p-1 rounded-0' href='\" . sysUrl('removeCustomer/$1') . \"'>I am sure</a> <button class='btn p-
249 $extra = <a class = \"btn btn-link p-0 px-1 modal-toggler=true data-target=#remoteModal1 data-remote='\" . sysUrl('updateCustomer/$1') . \"'>xl class='fa fa-ed
250 $extra .= <button types='button' class='btn btn-link p-0 px-1' data-container='body' data-toggle='popover' data-placement='left' data-html='true' data-content='\"
251 $action = <div class = \"text-center\">
252     . $extra
253     . </div>";
254     $this->dataTable->select('id, name, email, contact, address, (select name from `users` where users.id = customers.addedBy) as addedBy, addedTime, deleted')
255     ->from(TABLE_CUSTOMERS)
256     ->addColumn('actions', $action, 'id')
257     ->where(['id !=' => 1, 'deleted' => 0])
258     ->generate();
259     return true;
260 }
261
262 public function addCustomer() {
263     $this->form_validation->set_rules('name', 'Name', 'required');
264     if ($this->form_validation->run()) {
265         $array = ["name", "email", "contact", "address"];
266         $data = [];
267         foreach ($array as $a) {
268             $data[$a] = $this->input->post($a);
269         }
270         $data["addedBy"] = currentUserID();
271         $data["addedTime"] = getCurrentTime();
272         if ($this->sysModel->insertData(TABLE_CUSTOMERS, $data)) {
273             return $this->goToUrl(sysUrl('customers'), $data["name"] . " [" . $data["contact"] . "] is being registered as customer", SUCCESS);

```

```

274     } else {
275         return $this->goToReference("Failed to add to server!!!", DANGER);
276     }
277 }
278 $this->modalPath("people/customers/modal/" . __FUNCTION__);
279 }
280
281 public function checkContactAvailability() {
282     $checkContactName = $this->input->post('contact');
283     if ($this->sysModel->is_available(TABLE_CUSTOMERS, ['contact' => $checkContactName, 'deleted' => 0])) {
284         echo '<label class="text-danger"><span class="glyphicon glyphicon-ok"></span>contact taken. Choose another</label>';
285     } else {
286         echo '<label class="text-success"><span class="glyphicon glyphicon-remove"></span>available</label>';
287     }
288 }
289
290 public function updateCustomer($id) {
291     $this->form_validation->set_rules('name', 'Name', 'required');
292     if ($this->form_validation->run()) {
293         $array = ['name', 'email', 'contact', 'address'];
294         $data = [];
295         foreach ($array as $a) {
296             $data[$a] = $this->input->post($a);
297         }
298         $data['addedby'] = currentUserID();
299         if ($this->sysModel->updateData(TABLE_CUSTOMERS, $data, ['id' => $id])) {
300             return $this->goToUrl(sysurl('customers'), $data['name'] . " is being updated!", SUCCESS);
301         } else {
302             return $this->goToReference("Failed to add to server!!!", DANGER);
303         }
304     }
305     $this->data["updateCustomer"] = $this->sysModel->getId(TABLE_CUSTOMERS, ['id' => $id]);
306     $this->modalPath("people/customers/modal/" . __FUNCTION__);
307 }

```

```

308
309 public function removeCustomer($id) {
310     $data['deleted'] = 1;
311     if ($this->sysModel->softRemoveData(TABLE_CUSTOMERS, $data, ['id' => $id])) {
312         return $this->goToUrl(sysurl("customers"), "Successfully deleted !!!", SUCCESS);
313     } else {
314         return $this->goToReference("Failed to delete!!!", DANGER);
315     }
316 }
317
318 /*
319 * Customer end
320 */

```

Figure 27: Customer Code

Vendor Code:

```

321 * Vendors start
322 */
323
324 public function vendors() {
325     $this->ifNotOwner();
326     $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => "Vendors", "bc" => array(
327         ["url" => sysurl(), "page" => currentUser(), ["url" => sysurl(__FUNCTION__), "page" => __FUNCTION__, ["url" => "", "page" => "LIST"]
328     )]);
329     $this->viewPath("people/vendors/" . __FUNCTION__);
330 }
331
332 public function getVendors() {
333     $this->ifNotOwner();
334     $dt = "<table border="1" class="table table-bordered"><thead><tr><th>id</th><th>companyName</th><th>name</th><th>email</th><th>contact</th><th>address</th><th>(select name from `users` where users.id = vendors.addedBy) as addedBy</th><th>addedTime</th><th>delete</th></tr></thead><tbody><tr><td><input type="checkbox"></td><td><input type="checkbox"></td><td><input type="checkbox"></td><td><input type="checkbox"></td><td><input type="checkbox"></td><td><input type="checkbox"></td><td><input type="checkbox"></td><td><input type="checkbox"></td></tr></tbody></table>";
335     $extra = "<a class="btn btn-link p-0 px-1" modal-toggle="true" data-target="#remoteModal" data-remote=" " . sysurl('updateVendor/$1') . "><i class="fa fa-edit"
336     $extra .= "<button type="button" class="btn btn-link p-0 px-1" data-container="body" data-toggle="popover" data-placement="left" data-html="true" data-content=" "
337     $action = "<div class="text-center">"
338     . $extra
339     . "</div>";
340     $this->datatables->select('id, companyName, name, email, contact, address, (select name from `users` where users.id = vendors.addedBy) as addedBy, addedTime, delete
341     ->from(TABLE_VENDORS)
342     ->addColumn('actions', $action, 'id')
343     ->where(withoutDeleted())
344     ->generate();
345     return true;
346 }
347
348 public function addVendor() {
349     $this->ifNotOwner();
350     $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => "Vendors", "bc" => array(
351         ["url" => sysurl(), "page" => currentUser(), ["url" => sysurl(__FUNCTION__), "page" => "vendors", ["url" => "", "page" => "ADD"]
352     )]);
353     $this->form_validation->set_rules('name', 'Name', 'required');
354     if ($this->form_validation->run()) {
355         $array = ['name', 'companyName', 'email', 'contact', 'address'];
356         $data = [];
357     }

```

```

358     foreach ($array as $a) {
359         $data[$a] = $this->input->post($a);
360     }
361     $data['addedBy'] = currentUserID();
362     $data['addedTime'] = getCurrentTime();
363     if ($this->sysModel->insertData(TABLE_VENDORS, $data)) {
364         return $this->goUrl(sysUrl('vendors'), $data["name"] . " is being registered as vendor provider!", SUCCESS);
365     } else {
366         return $this->goReference("Failed to add to server!!!", DANGER);
367     }
368 }
369 $this->viewPath("people/vendors/" . __FUNCTION__);
370 }
371
372 public function updateVendor($id) {
373     $this->ifNotOwner();
374     $this->form validation->set_rules('name', 'Name', 'required');
375     if ($this->form validation->run()) {
376         $array = ["name", "companyName", "email", "contact", "address"];
377         $data = [];
378         foreach ($array as $a) {
379             $data[$a] = $this->input->post($a);
380         }
381         $data['addedBy'] = currentUserID();
382         if ($this->sysModel->updateData(TABLE_VENDORS, $data, ["id" => $id])) {
383             return $this->goUrl(sysUrl('vendors'), $data["name"] . " is being updated as vendor provider!", SUCCESS);
384         } else {
385             return $this->goReference("Failed to add to server!!!", DANGER);
386         }
387     }
388
389     $this->data["updateVendor"] = $this->sysModel->getById(TABLE_VENDORS, ["id" => $id]);
390     $this->modalPath("people/vendors/modal/" . __FUNCTION__);
391 }
392
393 public function checkVendorContactAvailability() {
394     $checkContactName = $this->input->post('contact');
395
396     public function checkVendorContactAvailability() {
397         $checkContactName = $this->input->post('contact');
398         if ($this->sysModel->is_available(TABLE_VENDORS, ["contact" => $checkContactName, 'deleted' => 0])) {
399             echo "<label class='text-danger'><span class='glyphicon glyphicon-ok'></span>contact taken. Choose another</label>";
400         } else {
401             echo "<label class='text-success'><span class='glyphicon glyphicon-remove'></span>available</label>";
402         }
403     }
404
405     public function removeVendor($id) {
406         $this->ifNotOwner();
407         $data['deleted'] = 1;
408         if ($this->sysModel->softRemoveData(TABLE_VENDORS, $data, ["id" => $id])) {
409             return $this->goUrl(sysUrl('vendors'), "Successfully deleted !!!", SUCCESS);
410         } else {
411             return $this->goReference("Failed to delete!!!", DANGER);
412         }
413     }
414
415     /*
416     * Vendors end
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451

```

Figure 28: Vendor Code

Category code:

```

415     * Categories start
416     */
417
418     public function categories() {
419         $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => "Categories", "bc" => array(
420             ["url" => sysUrl('index'), "page" => currentUserID()], ["url" => sysUrl(__FUNCTION__), "page" => "", "page" => "LIST"]
421         )];
422         $this->viewPath("categories/" . __FUNCTION__);
423     }
424
425     public function getCategories() {
426         $dlt = "<p>Are you sure?</p><a class='btn btn-danger po-delete btn-sm p-1 rounded-0' href='\" . sysUrl('removeCategory/$1') . \"'>I am sure</a> <button class='btn po
427         $extra = "<a class = 'btn btn-link p-0 px-1' modal-toggler='true' data-target='#remoteModal1' data-remote='\" . sysUrl('updatecategory/$1') . \"'>x i class='fa fa-edi
428         $extra = "<button type='button' class='btn btn-link p-0 px-1' data-container='body' data-toggle='popover' data-placement='left' data-html='true' data-content='\"
429         $action = "<div class = '\\text-center\\'>
430             . $extra
431             . "</div>";
432         if (isOwner()) {
433             $this->datatables->select('id, image, code, name, (select name from users where users.id = categories.addedBy) as addedBy, addedTime, deleted')
434                 ->from(TABLE_CATEGORIES)
435                 ->addColumn("actions", $action, 'id')
436                 ->where(withoutDeleted());
437             ->generate();
438         } else {
439             $this->datatables->select('id, image, code, name, (select name from users where users.id = categories.addedBy) as addedBy, addedTime, deleted')
440                 ->from(TABLE_CATEGORIES)
441                 ->where(withoutDeleted());
442             ->generate();
443         }
444         return true;
445     }
446
447     public function addCategory() {
448         $this->ifNotOwner();
449         $this->navMeta = ["active" => __FUNCTION__, "pageTitle" => "Categories", "bc" => array(
450             ["url" => sysUrl(), "page" => currentUserID()], ["url" => sysUrl(__FUNCTION__), "page" => "Categories"], ["url" => "", "page" => "ADD"]
451         )];

```

```

453     $this->form_validation->set_rules('code', 'code', 'required');
454     if ($this->form_validation->run()) {
455         $array = ['code', 'name'];
456         $data = [];
457         foreach ($array as $a) {
458             $data[$a] = $this->input->post($a);
459         }
460         $data['addedBy'] = currentUserID();
461         $data['addedTime'] = getcurrentTime();
462         if ($this->upload->do_upload('image') != null) {
463             $img = $this->upload->data();
464             $data['image'] = $img['file_name'];
465         }
466         if ($this->sysModel->insertData(TABLE_CATEGORIES, $data)) {
467             return $this->goToUrl(sysurl('categories'), "[" . $data['code'] . "] is being registered as category", SUCCESS);
468         } else {
469             return $this->goToReference("Failed to add to server!!!", DANGER);
470         }
471     }
472     $this->viewPath("categories/" . __FUNCTION__);
473 }
474
475 public function checkCodeAvailability() {
476     $checkCode = $this->input->post('code');
477     if ($this->sysModel->is_available(TABLE_CATEGORIES, ['code' => $checkCode, 'deleted' => 0])) {
478         echo '<label class="text-danger"><span class="glyphicon glyphicon-ok"></span>code taken. Choose another</label>';
479     } else {
480         echo '<label class="text-success"><span class="glyphicon glyphicon-remove"></span>Available</label>';
481     }
482 }
483
484 public function updateCategory($id) {
485     $this->ifNotOwner();
486     $this->form_validation->set_rules('code', 'code', 'required');
487     $this->data["updateCategory"] = $this->sysModel->getById(TABLE_CATEGORIES, ["id" => $id]);
488 }
489
490 $this->data["updateCategory"] = $this->sysModel->getById(TABLE_CATEGORIES, ["id" => $id]);
491
492 if ($this->form_validation->run()) {
493     $array = ['code', 'name'];
494     $data = [];
495     foreach ($array as $a) {
496         $data[$a] = $this->input->post($a);
497     }
498     if ($this->upload->do_upload('image') != null) {
499         $img = $this->upload->data();
500         $data['image'] = $img['file_name'];
501         if ($this->data["updateCategory"]->image) {
502             unlink($img["file_path"] . $this->data["updateCategory"]->image);
503         }
504     }
505     if ($this->sysModel->updateData(TABLE_CATEGORIES, $data, ["id" => $id])) {
506         return $this->goToUrl(sysurl('categories'), "Category [" . $data['code'] . "] is being updated!!!", SUCCESS);
507     } else {
508         return $this->goToReference("Failed to add to server!!!", DANGER);
509     }
510 }
511 $this->modalPath("categories/modal/" . __FUNCTION__);
512 }
513
514 public function removeCategory($id) {
515     $this->ifNotOwner();
516     $data['deleted'] = 1;
517     if ($this->sysModel->softRemoveData(TABLE_CATEGORIES, $data, ["id" => $id])) {
518         return $this->goToUrl(sysurl('categories'), "Successfully deleted !!!", SUCCESS);
519     } else {
520         return $this->goToReference("Failed to delete!!!", DANGER);
521     }
522 }
523
524 /*
525 * Categories end
526 */

```

Figure 29: Category code

10.2 Possible Problem Breakdown

To make a project easier, at first need to divide it into smaller parts. There are some possible breakdown problems given below:

- Analysis project database.
- Owner management development.
- Product list management development.

Analysis project database

- ❖ Gather information to problem solve.
- ❖ Identify must all requirements to finalize.
- ❖ Gather all data and information.

Owner management development

- ❖ Design requirements all the page.
- ❖ Develop expense, products page, customer information.
- ❖ Develop invoice page.
- ❖ Vendor management page development
- ❖ Add vendor page development.

Product list management development

- ❖ Product price and picture page development
- ❖ Add product page development.
- ❖ Develop category page.

10.3 Prioritization while Developing the Solution

Important requirements already identified before the chapter. Now we develop more prioritizing requirements at first. Because it is very important to random development. The prioritized task is given below:

- ❖ Analysis project database and design.
- ❖ Database development.
- ❖ Registration and login for the owner.
- ❖ Add product list and add price.
- ❖ Add vendor for the product purchases.
- ❖ Add category for the product.
- ❖ Products sell list and highest product sell chart.

Chapter 11 – Testing

11.1 Software Testing Strategy

A software testing method includes software test case creation approaches into a well-planned series of stages that result in effective software development. The approach includes a road map that outlines the actions that must be followed throughout testing.

The testing strategy that will be followed in this software project –

- Unit testing
- Integration testing
- Validation testing

The first step in software testing is unit testing. Unit testing concentrates on each unit of the software as implemented in the source code. Unit testing focuses on each component individually. The unit test is white-box oriented. Thus, unit testing of this library software will be done after the completion of every module or component.

Integration testing is the next phase. Integration testing is a method of building a program's structure while also doing tests to find interface issues. Integration testing's goal is to take unit-tested components and use them to create a program structure that is defined by design.

The integration testing strategy that has been chosen for this project is top-down testing. The Black-box testing method is the most prevalent for integration testing. A top-down integration strategy will be used to perform integration testing. Top-down integration will be done in a breadth-first manner. All components directly subordinate at each level are included in breadth-first integration, which moves horizontally throughout the hierarchy. A series of high-order tests are run after the software has been incorporated. As a result, the validation criteria described in requirements engineering should be put to the test. Validation testing ensures that software complies with all functional, behavioral, and performance specifications. Invalidation is only tested using the black-box method.

11.2 System Testing Methodology

11.2.1 Black-box Testing

Black-box testing, also known as behavioral testing, focuses on the software's functional requirements. It allows a software engineer to create sets of input conditions that will completely exercise all of a program's functional requirements. The black-box testing method will be applied to test the modules of LMS.

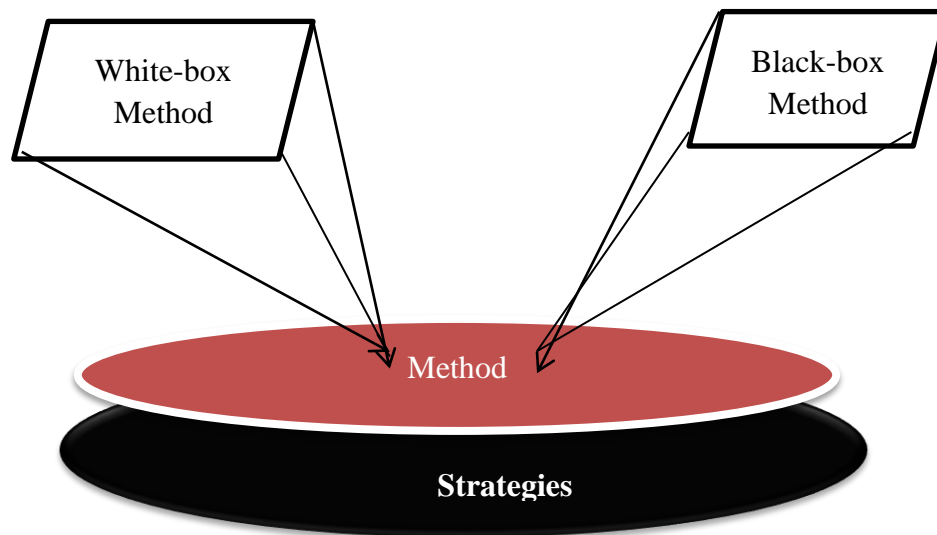


Figure 30: Black bock and white box testing

11.2.2 White-box Testing

White-box testing, also called glass-box testing, is a test case design method that generates test cases from the procedural design's control structure. Using white-box testing methods, a software engineer can derive test cases that,

- a) Ensure that each module's separate pathways have been tested at least once
- b) Consider both the true and false sides of all logical conclusions.
- c) Complete all loops inside their operational constraints and at their boundaries.
- d) Test internal data structures to confirm that they are valid.

The modules that contain some complex calculations or decision-making code such as checking the availability of the library item will be tested using the white-box method.

11.3 Testing

Input's	No user inputs
Desired output	View today's sale, due, revenue, purchase, expense, and products/customers individual report
Actual output	Successfully showed those items
Verdict	Getting all the desired output from the system.

Table 20: Daily sale information module testing

Input's	Customer data and items with quantity
Desired output	Can select product and customer to create an invoice for the customer also can suspend order and edit item from the cart.
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table21: POS terminal testing

Input's	Add new product with name, image, category, cost, and price
Desired output	Can add, delete and view products
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 22: Manage expenses module testing

Input's	Add new product with a name, image, category, cost, and price
Desired output	Can add, delete and view products
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 23: Manage products module testing

Input's	Add new category with category name and category image
Desired output	Can add, delete and view category
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 24: Manage category list module testing

Input's	Add new purchases with product name, vendor, and amount
Desired output	Can add, delete and view purchases
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 25: Manage Purchases module testing

Input's	Search with keyword to find the specific record
Desired output	View all sale reports and can be able to manage them
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 26: Sale report manage module testing

Input's	Add new customer with name, contact, image, and address
Desired output	Can add, delete and view customers
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system

Table 27: Manage Customers module testing

Input's	Add a new user with name, contact, image, type, and address
Desired output	Can add, delete and view users
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 28: Manage Users module testing

Input's	No input
Desired output	Can add, update due payment
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 29: Manage Due module testing

Input's	No input
Desired output	Can see the product warehouse storage
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 30: Manage notification module testing

Input's	No input
Desired output	Can edit store setting (name, address, and others)
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 31: Manage setting module testing

Input's	Make pos order
Desired output	Can see list products and their sale price, subtotal, grand total, due/change
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 32: Manage invoice module testing

Input's	No input
Desired output	Can view all kinds of report like profit loss report, daily sales report, chart report and date range report
Actual output	All the functions are working well
Verdict	Getting all the desired output from the system.

Table 32: Report module testing

Chapter 12 – Implementation

12.1 Training

Training is important to make the system user-friendly for the customer. It is very useful to ensure to operate the system perfectly. A chart is given below the training procedure for the customer-

SL	User	Training Scope	Time
1	Employ/ Owner	Log in and register as an owner. I will explain how can he/she add product photos, price, and print invoices. Also, I will explain how can he/she add the shop name and shop logo, add vendor.	1 hour

Table 33: Training

12.2 Implementation Scheme

Big Bang : Big bang theory means to move on the new system from the old system within one day.

12.3 Scaling

There was no plan of scaling. In the academic project, there was no use in scaling.

12.4 Load Balancing

Load balancing means the distribution network traffic across multiple servers. Load balancing provides the flexibility to add the server as you demand and also ensure high availability and reliability requests to online servers

Chapter 13 – Critical Appraisal and Evaluation

13.1 Objective could be met

- The objectives are listed below.
- Add products to the cart
- Rewrite products in the cart
- Customer management

Objective-1

Achievement rate and others

When a customer purchases a product then the customer can add it to the cart.

Objective-2

Achievement rate and others:

In the cart system, the owner can give a reward for the particular product. Customers will use the cart more for rewarding.

Objective-3

Achievement rate and others:

The owner or staff can keep information customers give a discount for the regular customer.

The reasons why it could not be touch

The point of sale is my internship project. In this project, the client was given their requirement. That's why I could not develop anything from mine.

Chapter 14 Lessons Learned

14.1 Pre-project – review – closing

Point of sale is mainly a web-based application. This system mainly makes for shop management, reporting daily sales, and creating an invoice for the customer. Also, this system keeps collecting product quantity product price, and product cost.

14.2 What I have learned

I have learned a lot about the real-life software development process. How a project task is assigned to a developer, How to apply business logic into a system, and many more. I have learned many of these to create this system. I can improve many diagrams such as the use case diagram, ERD diagram activity diagram. Also, I improve my knowledge of testing systems such as black-box testing, white box testing. It also improves my programming skills. In this project, I use PHP(Codeigniter3).

14.3 The problems I have faced

I have faced many problems during the project. I have used the Incremental process model to develop the software also face many problems to overcome this problem. There are many problems drawing diagrams whatever I draw all diagrams properly. I faced a problem with maintaining a time-box.

14.4 What Solution Occurred

I tried to solve all problems also got all solutions. I have worked hard to complete my project within the deadline to maintain the time-box.

Chapter 15 – Conclusion

15.1 Summary of the Project

Working at DCL has given me the opportunity to participate in the design and implementation of software. My most extensive experience was in the field of design. I've learned a lot of new things that I didn't know before. I have also learned some technical issues which help me to do better in my future life. The following indicator will indicate some of the technical issues which I have learned and implemented from this project.

- The designing strategy of a PHP-based project.
- The analyzing strategy of a project.
- Developing a project that can be accessed over the internet.
- Developing software for sharing databases in the private network.
- Real-life experience with MYSQL and PHP (CodeIgniter).

15.2 Goal of the project

- Better shop organization by this system
- Pos record all invoices and easily manage the organization
- This Pos system help to improve better service of the customers.
- In this system improve or change shop image a positive attitude.
- Better promotion management including the type of discount and the time duration.
- Pos system properly manages purchases from suppliers also keeps track of all orders placed.
- Employers select all the product the customer want to buy and automatically calculate the price.

15.3 Success of the project

In this point of sales system, sales chart can be seen through this project authority can observe all situation. The price of the item is given in this system. A notification has come the number of the product is less than 25. Also, this system show all-day sales record, sales product cost, and price. It makes invoices for the customer.

15.4 What I have done in the documentation

I have completed all tasks to complete this documentation. I completed all diagrams, tables, and time-box. This documentation also added goals and added all necessary information for all the chapters.

15.5 Value of the Project

Technology always saves time. This project owner will get many facilities such as shop management, daily sales report product cost, and product price. Staff can sell all products proper price and make invoices for the customer. I gathered many experiences from this project development. This experience will give me an advantage in a professional job.

15.6 My experience

I have gathered a lot of experience from this project. I gather programming skills. As I have done my project PHP(Codeigniter3), thus my knowledge about this framework has improved than before. Before doing this project I had lacked on the database. But after completing this project I have successfully coped with those issues. Besides I can learn how the organization can manage employee, customer and product sell through this system.

Appendix

AFP: Adjusted Function Point

CC- Customer Communicator

DET- Data Element Type

DI- Degree of Influence

EI- External Input

EIF- External Interface File

EO- External Output

EQ- External Query

ERD - Entity Relationship Diagram

FK- Foreign Key, A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables

FP- Function Pint, a measure of the utility delivered by an application

FTR- File Type Referenced

GSC- General System Characteristics

ILF- Internal Logical File

MS- Microsoft

Normalization- Normalization is a step by step decomposition of complex database into a simple one

PK- Primary Key, A primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records

Project Scheduling- project scheduling is an activity of distributing the estimated efforts within the planned project duration

POS: Point of Sale

QA: Quality Assurance

QMS: Quality Management System

RET- Record Element Type

SQL- Structured Query Language/ Standardized Query Language

SQM: System Quality Management TDI- Total Degree of Influence

TQM: Total Quality Management

UFP- Unadjusted Function Point

VAF- Value Adjustment Factor

References

Books

Database Management System. (2018). Retrieved from Database Management System:
https://www.tutorialspoint.com/dbms/dbms_tutorial.pdf

Scott Tilley, & Harry J. Rosenblatt. (February 5, 2016). *Systems Analysis and Design* (11th Edition ed.). Course Technology.

Websites:

ERD. (n.d.). *ERD*. Retrieved from <https://www.visual-paradigm.com/>.

Overflow, S. (2018). *How to calculate function points*. Retrieved from Stack Overflow:
<https://stackoverflow.com/questions/34473698/how-to-calculate-function-points>

2/16/22, 10:44 PM

Turnitin

Turnitin Originality Report

Processed on: 16-Feb-2022 22:05 +06
 ID: 1763849667
 Word Count: 10238
 Submitted: 1

181-16-240_Israt Jahan_Fall 2021.docx By
 Anonymous

Similarity Index

26%

Similarity by Source

Internet Sources: 20%
 Publications: 8%
 Student Papers: 24%

5% match (student papers from 29-Dec-2019)
 Class: Final_project_Fall2019
 Assignment: Project_Fall_2019
 Paper ID: [1238624034](#)

2% match (Internet from 14-Nov-2019)
<https://www.slideshare.net/Kinzarazzag09/risk-management-91110908>

2% match (student papers from 03-Dec-2014)
[Submitted to Laureate Higher Education Group on 2014-12-03](#)

1% match (Internet from 19-Dec-2021)
<https://www.slideshare.net/siddharthchaudhary39/automated-home-security-project-86821068>

1% match (Internet from 29-Jul-2020)
https://mafiadoc.com/software-engineering-9th-edition_598ac8c51723ddd169541bdb.html

1% match (student papers from 06-Jan-2022)
[Submitted to Management Development Institute Of Singapore on 2022-01-06](#)

1% match (Internet from 30-Sep-2019)
<http://www.ijetcse.com/wp-content/plugins/ijetcse/file/upload/docx/540IMPROVING-PRODUCT-RECOMMENDATION-BY-ENUMERATION-pdf.pdf>

1% match (Internet from 27-Sep-2014)
<http://dspace.thapar.edu:8080/dspace/bitstream/10266/603/3/T603.pdf>

https://www.turnitin.com/newreport_printview.asp?eq=0&eb=1&esm=0&oid=1763849667&sid=0&n=0&m=2&svr=26&r=42.96389149418478&lang=en_us

1/21