

# **E-Cosmetics by Fair cosmetics**



**Submitted By:**

**Usman Haider**

**Supervised By:**

**Memoona Afsheen Malik**

**Department of Computer Sciences**

**Quaid-i-Azam University**

**Islamabad**

**2019**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# **Abstract**

E-cosmetics is web based application for Fair Cosmetics Company for its customers and salesmen. It provides Fair Cosmetics Company to advertise its products without paying any penny. It also provides company to manage its stock of products as well as facilitates its customers (shopkeepers) to place online order and view their complete list of orders and paid amount history. It allows shopkeeper to remove order within current date of placing. Administrator has right to block any shopkeeper due to his laziness in paying remaining amount. After that, that shopkeeper can never place order until he might be unblocked. Salesman updates bill of shopkeepers after receiving amount from them and can add order for shopkeepers (except for blocked).

# Table of Contents

List of Figures .....	viii
-----------------------	------

List of Table .....	x
---------------------	---

<b>Chapter 1 Introduction.....</b>	<b>1</b>
------------------------------------	----------

1.1. Problem Definition.....	1
1.2. Proposed Solution .....	1
1.3. Objectives .....	2
1.4. Project Organization.....	2
1.4.1. Software Process Model .....	2
1.4.2. Roles and responsibilities .....	2
1.4.3. Tools and techniques .....	2
1.5. Project Management Plan.....	3
1.5.1. Time table.....	3
1.5.2. Gantt chart .....	3
1.5.1. Project Deliverables .....	9
1.6. Report Structure.....	9

<b>Chapter 2 Requirement Gathering and Analysis.....</b>	<b>10</b>
--	-----------

2.1. Purpose .....	10
2.2. Product Scope .....	10
2.3. Definitions, Abbreviations and Acronyms .....	11
2.4. Overall description .....	11
2.4.1. Product Perspective.....	11
2.4.2. User Characteristics .....	12
2.4.3. Constraints .....	12
2.4.4. Assumptions and dependencies .....	12
2.5. External Interface Requirements.....	12
2.5.1. User Interfaces .....	12
2.5.2. Hardware Interfaces .....	13
2.5.3. Software Interfaces .....	13
2.5.4. Communication Interfaces.....	13
2.6. Software Quality Attributes .....	13

2.6.1. Availability .....	13
2.6.2. Maintainability.....	13
2.6.3. Portability .....	13
2.6.4. Reliability .....	13
2.6.5. Security.....	13
2.6.6. Performance.....	14
2.7. Functional Requirements.....	14
2.8.1. Use case Diagram.....	14
2.8.2. Use case Description .....	16
Use case 1: Login .....	16
Use case 2: Register salesman/shopkeeper .....	16
Use case 3: Add product category .....	17
Use case 4: Update product category .....	17
Use case 5: Remove product category .....	17
Use case 6: Add product .....	18
Use case 7: Update product.....	18
Use case 8: Remove product .....	18
Use case 9: Add stock.....	19
Use case 10: Update stock.....	19
Use case 11: Remove stock.....	19
Use case 12: View list of products .....	20
Use case 13: Place order .....	20
Use case 14: View bill history.....	20
Use case 15: Add order .....	21
Use case 16: Remove order.....	21
Use case 17: Update bill.....	22
Use case 18: Remove shopkeeper.....	22
Use case 19: Remove salesman .....	22
Use case 20: Ask for help.....	23
Use case 21: View list of salesman.....	23
Use case 22: View List of shopkeeper .....	23
Use case 23: Post news .....	24

Use case 24: Logout.....	24
Use case 25: Reply help .....	24
<b>Chapter 3 Software Design Description.....</b>	<b>25</b>
3.1. Introduction .....	25
3.1.1. Design Overview .....	25
3.1.2. Requirement Traceability Matrix.....	25
3.2. System Architecture Design .....	26
3.2.1. Chosen System Architecture.....	26
Presentation layer .....	26
Application layer .....	26
Data layer .....	27
3.3. System Sequence Diagrams (SSD).....	27
SSD-1 (Login).....	28
SSD-2 (Register salesman/shopkeeper).....	28
SSD-3 (Add product).....	29
SSD-4 (Update product).....	29
SSD-5 (Remove product).....	30
SSD-6 (Add stock).....	30
SSD-7 (Place order).....	31
SSD-9 (Update bill).....	32
SSD-10 (Remove shopkeeper) .....	32
SSD-11 (Remove salesman).....	33
SSD-12 (Ask for help) .....	33
SSD-13 (Post news).....	34
3.4. Entity relationship diagram.....	34
3.5. Data flow diagrams (DFD) .....	35
3.5.1. Zero level DFD (Context level) .....	36
3.5.2. First level DFD (Admin) .....	36
3.5.3. First level DFD (Shopkeeper).....	37
3.5.4. First level DFD (Salesman) .....	37
3.6. Activity diagrams.....	38
3.6.1. Manage product (Activity diagram).....	38

3.6.2. Place order, bill (Activity diagram).....	38
3.6.3. Add Order, remove order & Update bill (Activity diagram) .....	39
3.7. Database design .....	40
<b>Chapter 4 Software Implementation Document .....</b>	<b>41</b>
4.1. Language Selection .....	41
4.2. Tools Selection .....	41
4.3. Application Screen Shots .....	41
4.3.1. Screenshot for product categories .....	42
4.3.2. Screenshot for Product Stock.....	42
4.3.3. Screenshot of product display for shopkeeper .....	43
4.3.4. Screenshot for list of shopkeepers.....	43
4.3.5. Screenshot for login .....	44
4.3.6. Screenshot for payment table.....	44
<b>Chapter 5 Software Test Document .....</b>	<b>45</b>
5.1. Introduction .....	45
5.1.1. Test approach.....	45
5.2. Test plan .....	45
5.2.1. Testing tools and environment.....	45
5.3. Test cases.....	46
5.3.1. TC-001 Login .....	46
5.3.2. TC-002 Register shopkeeper/salesman .....	46
5.3.3. TC-003 Add product category .....	46
5.3.4. TC-004 Update product category.....	47
5.3.5. TC-005 Remove product category .....	47
5.3.6. TC-006 Add stock.....	47
5.3.7. TC-007 Place order .....	47
5.3.8. TC-008 Remove order.....	48
5.3.9. TC-009 Update bill .....	48
5.3.10. TC-010 Block shopkeeper .....	48
5.3.11. TC-011 Remove salesman.....	49
5.3.12. TC-012 Ask for help .....	49
5.3.13. TC-013 Post news .....	49

5.3.14. TC-014 Logout .....	49
<b>Chapter 6 Conclusions and Future Enhancements.....</b>	<b>51</b>
6.1. Conclusions .....	51
6.2. Future enhancement .....	51
<b>References .....</b>	<b>52</b>



# List of Figures

Figure 1. 1 Time Schedule [1/3] .....	4
Figure 1. 2 Time Schedule [2/3] .....	5
Figure 1. 3 Time Schedule [3/3] .....	6
Figure 1. 4 Gantt chart [1/4] .....	6
Figure 1. 5 Gantt chart [2/4] .....	7
Figure 1. 6 Gantt chart [3/4] .....	8
Figure 1. 7 Gantt chart [4/4] .....	8
Figure 2. 1 Product Perspective .....	12
Figure 2. 2 Use case diagram.....	15
Figure 3. 1 System Architecture Diagram.....	27
Figure 3. 2 SSD (Login) .....	28
Figure 3. 3 SSD (Register shopkeeper/salesman).....	28
Figure 3. 4 SSD (Add product).....	29
Figure 3. 5 SSD (Update product) .....	29
Figure 3. 6 SSD (Remove product).....	30
Figure 3. 7 SSD (Add stock) .....	30
Figure 3. 8 SSD (Place order).....	31
Figure 3. 9 SSD (Remove order) .....	31
Figure 3. 10 SSD (Update bill) .....	32
Figure 3. 11 SSD (Remove shopkeeper).....	32
Figure 3. 12 SSD (Remove salesman) .....	33
Figure 3. 13 SSD (Ask for help) .....	33
Figure 3. 14 SSD (Post news).....	34
Figure 3. 15 SSD (Logout) .....	34
Figure 3. 16 Entity relationship diagram.....	35
Figure 3. 17 DFD (0 level) .....	36
Figure 3. 18 1st level DFD (Admin) .....	36
Figure 3. 19 1st level DFD (Shopkeeper).....	37
Figure 3. 20 1st level DFD (Salesman) .....	37
Figure 3. 21 Activity diagram (Manage product).....	38
Figure 3. 22 Activity diagram (Place order, order remove, view bill and payment detail) .....	39
Figure 3. 23 Activity diagram (Add order, remove order & update bill).....	39
Figure 3. 24 Database design.....	40
Figure 4. 1 Screenshot for product categories .....	42
Figure 4. 2 Screenshot for product stock.....	42
Figure 4. 3 Screenshot of product display for shopkeeper .....	43
Figure 4. 4 Screenshot for post news .....	43

Figure 4. 5 Screenshot for login.....	44
Figure 4. 6 Screenshot for payments table .....	44

# List of Table

Table 1. 1 Project Deliverables.....	9
Table 2. 1 Definition, Abbreviations and Acronyms .....	11
Table 2. 2 Login Use case Description .....	16
Table 2. 3 Register salesman/shopkeeper use case description.....	16
Table 2. 4 Add product category use case description.....	17
Table 2. 5 Update product category use case description .....	17
Table 2. 6 Remove product category use case description.....	18
Table 2. 7 Add product use case description .....	18
Table 2. 8 Update product use case description .....	18
Table 2. 9 Remove product use case description.....	19
Table 2. 10 Add stock use case description.....	19
Table 2. 11 Update stock use case description .....	19
Table 2. 12 Remove stock use case description .....	20
Table 2. 13 View list of products use case description.....	20
Table 2. 14 Place order use case description .....	20
Table 2. 15 View bill history use case description .....	21
Table 2. 16 Add order into cart use case description .....	21
Table 2. 17 Remove order from cart use case description .....	22
Table 2. 18 Update bill use case description .....	22
Table 2. 19 Block shopkeeper use case description.....	22
Table 2. 20 Remove salesman use case description.....	23
Table 2. 21 Ask for help use case description .....	23
Table 2. 22 View list of salesman use case description .....	23
Table 2. 23 View list of shopkeeper use case description.....	24
Table 2. 24 Post news use case description .....	24
Table 2. 25 Logout use case description .....	24
Table 2. 26 Reply help use case description.....	24
Table 3. 1 Requirement traceability matrix.....	26
Table 5. 1 TC-Login.....	46
Table 5. 2 TC-Register shopkeeper/salesman .....	46
Table 5. 3 TC-Add product category .....	47
Table 5. 4 TC-Update product category.....	47
Table 5. 5 TC-Remove product category .....	47
Table 5. 6 TC-Add stock .....	47
Table 5. 7 TC-Place order .....	48
Table 5. 8 TC-Remove order.....	48
Table 5. 9 TC-Update bill.....	48

Table 5. 10 TC-Block shopkeeper .....	49
Table 5. 11 TC-Remove salesman .....	49
Table 5. 12 TC-Ask for help.....	49
Table 5. 13 TC-Post news .....	49
Table 5. 14 TC-Logout.....	50

# Chapter 1

## Introduction

This chapter briefly describes problem definition and proposed solution of the project. It also elaborates objectives, project organization and project planning. Finally, this chapter explains report structure.

### 1.1. Problem Definition

Fair cosmetics is the beauty products manufacturing company. It manufactures beauty products like beauty cream, beauty soap, hair remover etc. and sells in the market through its salesmen. Company sends its salesmen in different markets to introduce its products in those markets. Salesman introduces shopkeepers with the products of the company. As shopkeepers get to know about the products, they can make an order for the products. Salesman takes their order and provides them their required products. Shopkeepers pay amount on their arrival.

In this modern age of internet, different companies use internet to enhance their revenue. But, Fair cosmetics doesn't have any web-based platform through which it can manage stock of its products and provides a facility to its customers (shopkeepers) to view its products, place order for the products, view their bill, order and payment history at any time from any place. Traditional way of billing is not useful these days as there can be many issues in bills (like extra charges, paid bill lost which might be added in next bill etc.).

Currently, shopkeepers have to call a salesman to write down order or wait for arrival of company's salesman or visit office of the company. This is not fruitful for business of the company. Shopkeepers or indeed the company can't always keep track of the manual bills and may lose record of payments. So, in case of a confusion there is no record to call upon. In both scenarios, one or both of the side may have to pay for it. This may result in a trust deficit between shopkeeper and the company.

### 1.2. Proposed Solution

This is a web-based application which is designed to replace traditional way of stock management as well as selling of products. It allows company to manage stock of its products and remind them if any product has low stock. This application allows company to advertise its new product without paying any amount. Registered and non-registered users can view products. Administrator will register its customers (Shopkeeper) by giving them an account. After that, they will be able to place online order, view products of the company (either available, not available or removed) and view his remaining due amount. If a product is out of stock then this system doesn't

allow shopkeeper to place order for that particular product. Salesman can also add an order which he takes during his visit to shopkeeper and update remaining amount. By using this system, administrator can block any customer and remove any salesman on his misbehavior. Administrator can view list of removed products, removed salesman and removed shopkeepers along with their payments.

### **1.3. Objectives**

This is a web based application for Fair Cosmetics Company which is manufacturing and selling its products into market. This is basically designed to replace traditional way of managing stock or products, making order for products, keeping record of customer (shopkeeper) order, bill and payment history and advertising for products. This application allows shopkeeper to place order, view order, bill and payment in just few clicks. This wouldn't allow a shopkeeper to place order for that product which is not available. In just few click, a salesman can manage order for a shopkeeper, update shopkeeper's bill on receiving amount from shopkeeper during his visit and reply on shopkeeper's question. Administrator can make advertisement for products. The main objective is to run business activities. This leads to increase in the revenue of the company.

### **1.4. Project Organization**

This portion of document describes which software process model is followed, what are the roles and responsibilities assigned and which types of tools and techniques are used for the completion of project.

#### **1.4.1. Software Process Model**

I shall use "Waterfall model" as Software Process Model for the completion of this project because project requirements are clear, easy to understand and simply to understand, each phase is completed first and then start next phase and allows for easy analysis and testing.

It is reasonable to use waterfall model for the project. Indeed, implementation will be gone in systematic sequential way to development. This means before gathering all requirements, we wouldn't go towards design phase and so on.

#### **1.4.2. Roles and responsibilities**

This is individual project.

#### **1.4.3. Tools and techniques**

A tool can be physical or software program which provides easiness to complete a job. Some software program tool will be used for the completion of the project. Microsoft Word will be used for documentation throughout the project, Project Liber will be used for Project Management plan. Microsoft Visio will be used for Use case, System Sequence Diagram (SSD), Entity Relationship Diagram (ERD), domain model. For implementation, Notepad++ will be used to write down code (Frontend and backend), Xampp Server will be used and MySQL database will be used for storing

records. HTML, CSS, Bootstrap and JQuery will be used for frontend and for backend PHP will be used.

## **1.5. Project Management Plan**

The project management plan is a formal, approved document used to define how the project is executed, monitored as well as controlled. It may be a summary or a detailed document that includes baselines, subsidiary management plan and other planning documents. The project management plan is not created all at once. It is gradually expanded, which means it is developed, refined, revisited and updated. This plan contains start to end phase to complete the whole project. There is nothing assign to any other person. Everything will be done by me. Project Management plan contains two thing i.e. timetable and Gantt chart.

### **1.5.1. Time table**

Time table helps to complete documentation as well as implementation on time because it schedules all phases of software development with available resources and time duration. . It will help us to complete our work in an efficient way by managing time duration with available resources. Time table basically contains start and end time for each specific activity or task. It clearly describes that when will a specific activity will start and end and so on. Figure 1.1, 1.2 and 1.3 shows time table for the completion of the project.

### **1.5.2. Gantt chart**

Plan chart describes the schedule which we will follow in documentation and implementation. It will help us to complete our work in an efficient way. In this plan chart project is divided into tasks and subtasks with the deadline. It also describes either two task will run parallel or not. Figure 1.4, 1.5, 1.6 and 1.7 shows Gantt chart for the successful accomplishment of the project.

		Name	Duration	Start	Finish	Predecessors
1		[-Software Project Mana...	4 days?	11/19/18 8:00 AM	11/22/18 5:00 PM	
2		[-Project Introduction	1 day?	11/19/18 8:00 AM	11/19/18 5:00 PM	
3		Project Overview	0.5 days?	11/19/18 8:00 AM	11/19/18 1:00 PM	
4		Project Deliverables	0.5 days?	11/19/18 1:00 PM	11/19/18 5:00 PM	3
5		[-Project Organization	1.5 days?	11/20/18 8:00 AM	11/21/18 1:00 PM	
6		Software Process Model	0.5 days?	11/20/18 8:00 AM	11/20/18 1:00 PM	4
7		Roles and Responsibility	0.5 days?	11/20/18 1:00 PM	11/20/18 5:00 PM	6
8		Tools and Techniques	0.5 days?	11/21/18 8:00 AM	11/21/18 1:00 PM	7
9		[-Project Management ...	1.5 days?	11/21/18 1:00 PM	11/22/18 5:00 PM	
10		Tasks, Deliverables an...	0.5 days?	11/21/18 1:00 PM	11/21/18 5:00 PM	8
11		Deliverables and Milest...	0.5 days?	11/22/18 8:00 AM	11/22/18 1:00 PM	10
12		Meeting	0.5 days?	11/22/18 1:00 PM	11/22/18 5:00 PM	11
14		[-Software Requirement ...	10 days?	11/23/18 8:00 AM	12/6/18 5:00 PM	
15		[-Requirement Collecti...	1.8 days?	11/23/18 8:00 AM	11/26/18 3:24 PM	
16		Finding functional requ...	0.7 days?	11/23/18 8:00 AM	11/23/18 2:36 PM	12
17		Finding non-functional ..	0.7 days?	11/23/18 2:36 PM	11/26/18 11:12 AM	16
18		Meeting & refine requir...	0.4 days?	11/26/18 11:12 AM	11/26/18 3:24 PM	17
19		[-Define Usecases	2.2 days?	11/26/18 3:24 PM	11/28/18 5:00 PM	
20		Write Usecase text for...	0.5 days?	11/26/18 3:24 PM	11/27/18 10:24 AM	18
21		Draw Usecase diagram	1 day?	11/27/18 10:24 AM	11/28/18 10:24 AM	20
22		Review of Usecases	0.6 days?	11/28/18 10:24 AM	11/28/18 4:12 PM	21
23		Meeting	0.1 days?	11/28/18 4:12 PM	11/28/18 5:00 PM	22
24		[-Define Domain Model	2.1 days?	11/29/18 8:00 AM	12/3/18 8:48 AM	
25		Identify real world obj...	0.25 days?	11/29/18 8:00 AM	11/29/18 10:00 AM	23
26		Identify realtionships	0.25 days?	11/29/18 10:00 AM	11/29/18 1:00 PM	25
27		Draw Domain model	1 day?	11/29/18 1:00 PM	11/30/18 1:00 PM	26
28		Refine Domain model	0.25 days?	11/30/18 1:00 PM	11/30/18 3:00 PM	27

Figure 1. 1 Time Schedule [1/3]



ID	Name	Duration	Start	Finish	Predecessors
29	Refinement of require...	0.25 days?	11/30/18 3:00 PM	11/30/18 5:00 PM	28
30	Meeting	0.1 days?	12/3/18 8:00 AM	12/3/18 8:48 AM	29
31	Draw ERD	2 days?	12/3/18 8:48 AM	12/5/18 8:48 AM	
32	Identify entities	0.25 days?	12/3/18 8:48 AM	12/3/18 10:48 AM	30
33	Identify relationships	0.25 days?	12/3/18 10:48 AM	12/3/18 1:48 PM	32
34	Design ERD	1 day?	12/3/18 1:48 PM	12/4/18 1:48 PM	33
35	Review ERD & Meeting	0.5 days?	12/4/18 1:48 PM	12/5/18 8:48 AM	34
36	Develop SRS	1.9 days?	12/5/18 8:48 AM	12/6/18 5:00 PM	
37	Write SRS Document u...	1.1 days?	12/5/18 8:48 AM	12/6/18 9:36 AM	35
38	Review SRS and make ...	0.5 days?	12/6/18 9:36 AM	12/6/18 2:36 PM	37
39	Document Deliverables...	0.2 days?	12/6/18 2:36 PM	12/6/18 4:12 PM	38
40	Meeting	0.1 days?	12/6/18 4:12 PM	12/6/18 5:00 PM	39
42	Software Design Descri...	9 days?	12/7/18 8:00 AM	12/19/18 5:00 PM	
43	System Architecture ...	2 days?	12/7/18 8:00 AM	12/10/18 5:00 PM	
44	Choose appropriate pa...	0.5 days?	12/7/18 8:00 AM	12/7/18 1:00 PM	40
45	Specify structure of th...	0.5 days?	12/7/18 1:00 PM	12/7/18 5:00 PM	44
46	Define communication...	0.5 days?	12/10/18 8:00 AM	12/10/18 1:00 PM	45
47	Review & finalize , Me...	0.5 days?	12/10/18 1:00 PM	12/10/18 5:00 PM	46
48	Database Design	2 days?	12/11/18 8:00 AM	12/12/18 5:00 PM	
49	Define database archit...	0.5 days?	12/11/18 8:00 AM	12/11/18 1:00 PM	47
50	Normalize ERD	1 day?	12/11/18 1:00 PM	12/12/18 1:00 PM	49
51	Meeting	0.5 days?	12/12/18 1:00 PM	12/12/18 5:00 PM	50
52	User Interface Design	2 days?	12/13/18 8:00 AM	12/14/18 5:00 PM	
53	Design front-end	1 day?	12/13/18 8:00 AM	12/13/18 5:00 PM	51
54	Review of front-end	0.4 days?	12/14/18 8:00 AM	12/14/18 11:12 AM	53
55	Front-end finalized	0.4 days?	12/14/18 11:12 AM	12/14/18 3:24 PM	54
56	Meeting	0.2 days?	12/14/18 3:24 PM	12/14/18 5:00 PM	55

Figure 1. 2 Time Schedule [2/3]

	🕒	Name	Duration	Start	Finish	Predecessors
57		<b>Detailed Design</b>	<b>3 days?</b>	<b>12/17/18 8:00 AM</b>	<b>12/19/18 5:00 PM</b>	
58		Draw System Sequenc...	0.7 days?	12/17/18 8:00 AM	12/17/18 2:36 PM	56
59		Review SSD	0.3 days?	12/17/18 2:36 PM	12/17/18 5:00 PM	58
60		Draw Class Diagram	0.7 days?	12/18/18 8:00 AM	12/18/18 2:36 PM	59
61		Review Class Diagram	0.3 days?	12/18/18 2:36 PM	12/18/18 5:00 PM	60
62		Review overall design ...	0.8 days?	12/19/18 8:00 AM	12/19/18 3:24 PM	61
63		Meeting	0.2 days?	12/19/18 3:24 PM	12/19/18 5:00 PM	62
65	👤	<b>Implementation</b>	<b>17 days?</b>	<b>12/20/18 8:00 AM</b>	<b>1/11/19 5:00 PM</b>	
66		<b>Coding</b>	<b>9 days?</b>	<b>12/20/18 8:00 AM</b>	<b>1/1/19 5:00 PM</b>	
67		Coding fornt-end design	2 days?	12/20/18 8:00 AM	12/21/18 5:00 PM	63
68		Coding whole detail de...	4 days?	12/24/18 8:00 AM	12/27/18 5:00 PM	67
69		Review coding	2 days?	12/28/18 8:00 AM	12/31/18 5:00 PM	68
70		Meeting	1 day?	1/1/19 8:00 AM	1/1/19 5:00 PM	69
71		<b>Database Connectivity</b>	<b>8 days?</b>	<b>1/2/19 8:00 AM</b>	<b>1/11/19 5:00 PM</b>	
72		Create Database and li...	2 days?	1/2/19 8:00 AM	1/3/19 5:00 PM	70
73		Fix error and refine wh...	1 day?	1/4/19 8:00 AM	1/4/19 5:00 PM	72
74		Final Product (Working...	4 days?	1/7/19 8:00 AM	1/10/19 5:00 PM	73
75		Meeting	1 day?	1/11/19 8:00 AM	1/11/19 5:00 PM	74
77	👤	<b>Testing</b>	<b>3 days?</b>	<b>1/14/19 8:00 AM</b>	<b>1/16/19 5:00 PM</b>	
78		Verification	1 day?	1/14/19 8:00 AM	1/14/19 5:00 PM	75
79		Validation	1 day?	1/15/19 8:00 AM	1/15/19 5:00 PM	78
80		Tested Software	1 day?	1/16/19 8:00 AM	1/16/19 5:00 PM	79

Figure 1. 3 Time Schedule [3/3]

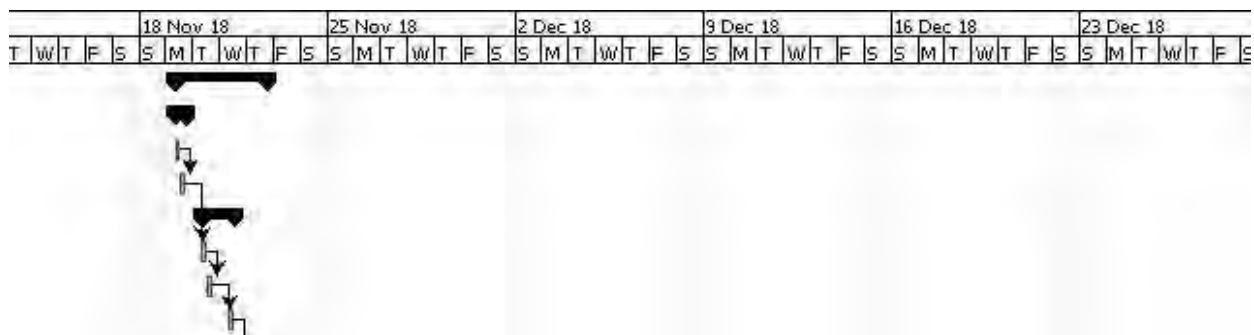


Figure 1. 4 Gantt chart [1/4]

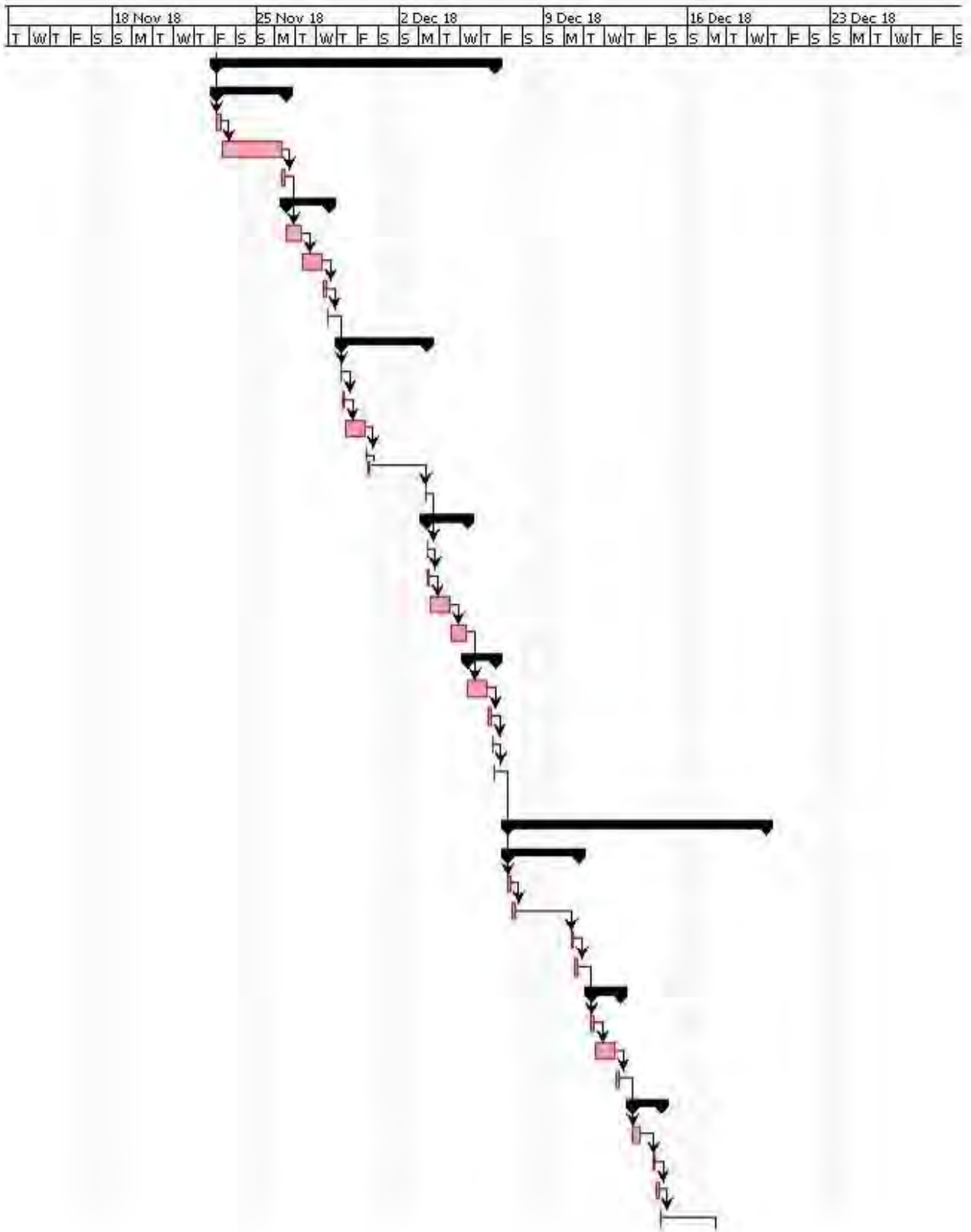


Figure 1. 5 Gantt chart [2/4]

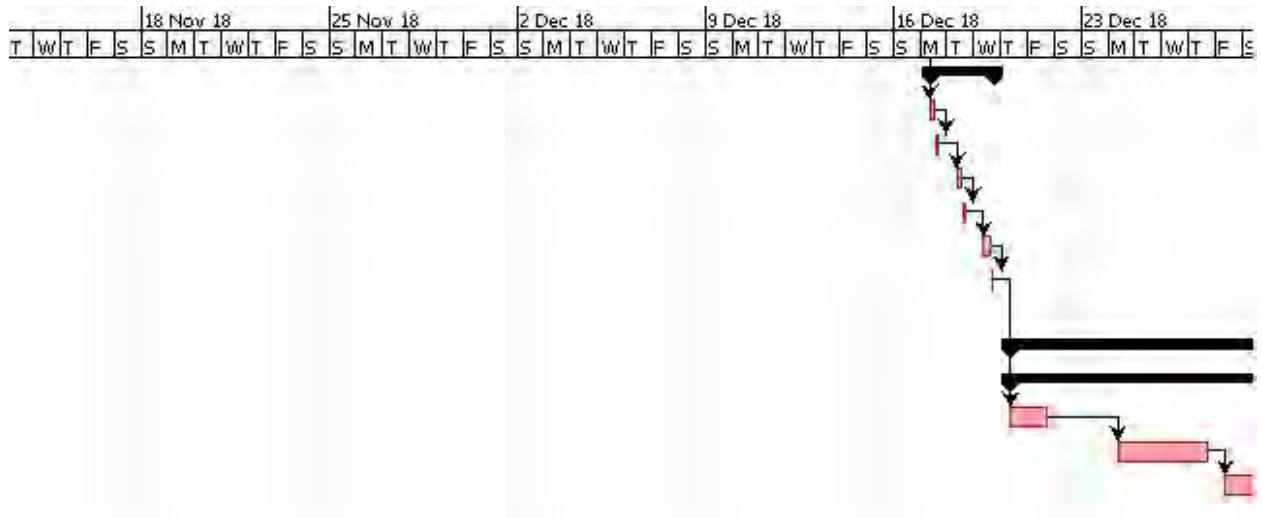


Figure 1. 6 Gantt chart [3/4]

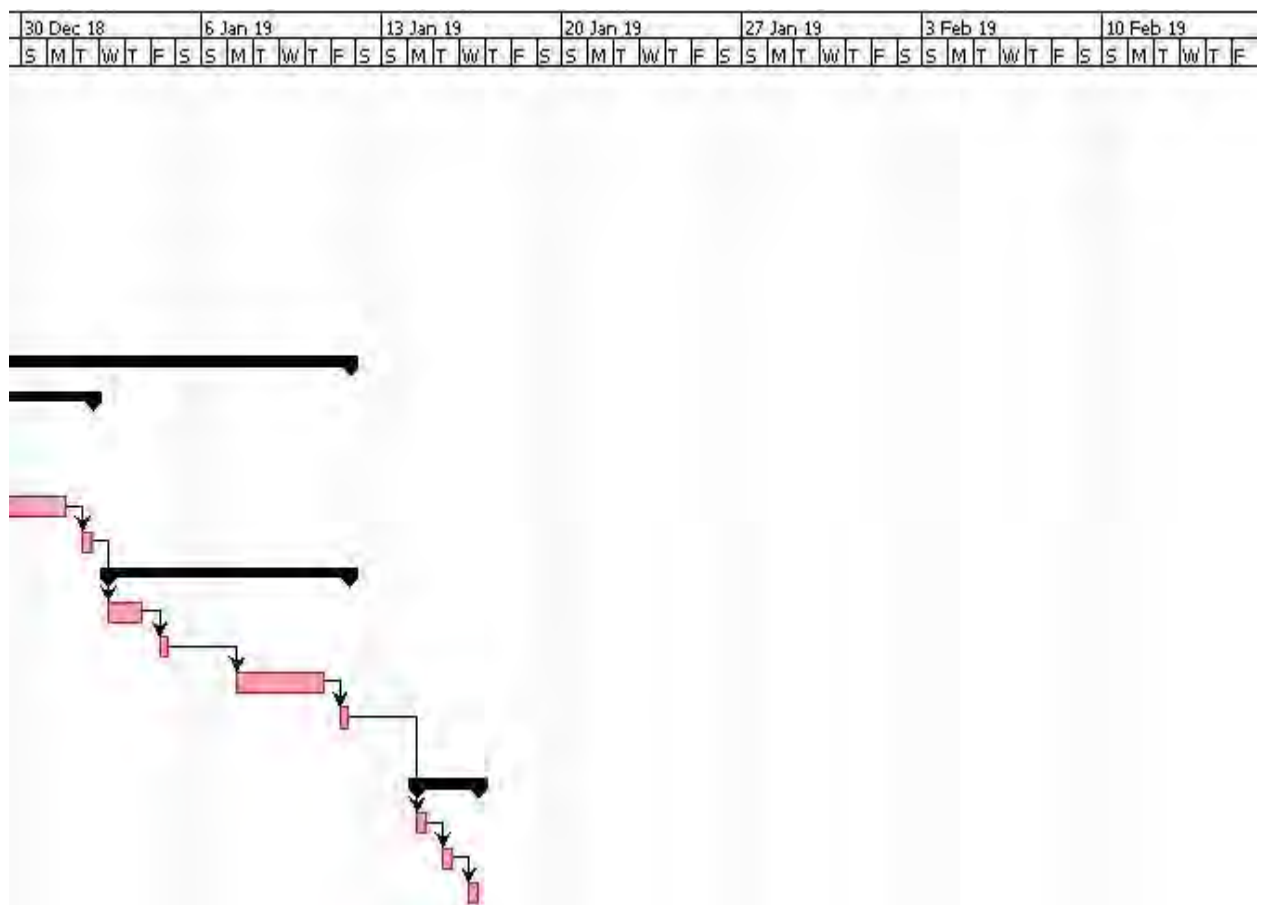


Figure 1. 7 Gantt chart [4/4]

### 1.5.1. Project Deliverables

<b>Deliverable Name</b>	<b>Description</b>
Software Project Management Plan (SPMP)	Description of the software approach and associated milestones
Software requirement Specification (SRS)	Description of expected software features, constraints, interfaces and other attributes.
Software Design Description (SDD)	Description of how the software will meet the requirements. Also describes the rationale of design decisions taken.
Final Product With Source Code	Working product with source code is given in this deliverable.
Software Test Documentation (STD)	Description of the plan and specifications to verify and validate the software and the results.

**Table 1. 1 Project Deliverables**

### 1.6. Report Structure

Chapter 1 has briefly described the introduction of the system, what are actual problem and proposed solution, its objective, the organization of project, and project management plan. Chapter 2 describes the requirements of the system and to decide what the system will do and what the system will not do (constraints). It also briefly describes functional and non-functional as well as major input and output of the system. This chapter contains use case diagram and use case description and domain model for this system.

# Chapter 2

## Requirement Gathering and Analysis

The purpose of this chapter is to clarify the requirement and to decide what the system should do and what the system shouldn't do. It also contains information about major inputs and outputs, definitions, acronyms, abbreviations, constraints, functional and non-functional requirements. It is a communication tool between users and software designers. It provides easiness for reviewing at later stages.

### 2.1. Purpose

The primary purpose of this web based application is to provide an online platform to customers (shopkeepers) registered in the company to place order and view his order, bill and payment. It also manages stock of products for the company. This application updates stock when a shopkeeper buys products. When a product will be out of stock, then this application generates an alert message to administrator for adding new stock. Shopkeeper can place online order, view his order history as well as payment paid history by using this application. A salesman can add order for a shopkeeper and update his bill. This application helps company to advertise its products and facilitates its shopkeepers and salesmen. Administrator blocks any shopkeeper for not paying remaining dues and also remove salesmen. This application will have complete detail of available and removed products, shopkeepers and salesmen etc.

### 2.2. Product Scope

Product scope include major functions, major inputs & outputs. Major functions for this system are given in following:

- Administrator will add /update/view a shopkeeper/salesman.
- Administrator will add/update/view/remove stock of products.
- Administrator can block/remove a shopkeeper.
- Administrator can remove a salesman.
- Administrator will add/remove/update a product.
- Administrator will add/remove/update product category.
- Administrator will view shopkeeper's payment history.
- Administrator will post news (e.g. Discount offer, new product arrival etc.)
- Administrator/Salesman/Shopkeeper/Visitor will view list of products.
- Salesman will add/remove order (on requesting by shopkeeper) for customer.
- Salesman will update shopkeeper's bill or payment.
- Salesman will response of shopkeeper's questions.
- Shopkeeper will place order.
- Shopkeeper will views his order, bill and payment.

- Shopkeeper asks any question (i.e. Product delivery, extra charges etc.)

Major Inputs to this application are given in following:

- Administrator/Salesman/Shopkeeper will login to perform its activities using username and password.
- Administrator can add stock of products.
- Administrator can add new product category and status.
- Administrator can add new product along with product category, price and description, status.
- Salesman can update shopkeeper’s bill.
- Salesman can add/update/remove order for shopkeeper.
- Shopkeeper can place order.

Major Output by this application are given in following:

- Administrator /Salesman/Customer can view customer’s bill.

### 2.3. Definitions, Abbreviations and Acronyms

Abbreviations/Acronyms/Def	Description
IEEE	Institute of Electrical and Electronic Engineers
Owner/Administrator	A person who owns the company.
Salesman	A person who can sale products after assigning role by Owner.
Customer	A person who purchases products from the company
SPMP	Software Project Management Plan
SRS	Software Requirement Specification
PMP	Project Management Plan
SDD	Software Design Description
STD	Software Test Document
User	Owner/Salesman/Shopkeeper/Non-registered visitor
Registered User	Shopkeeper, Salesman and Administrator
Visitor	Non-Registered User

Table 2. 1 Definition, Abbreviations and Acronyms

### 2.4. Overall description

This section of the document describes the factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements.

#### 2.4.1. Product Perspective

This is standalone system. This permits non-registered users (visitor) just to view products. It will allow shopkeeper to place order, view his bill/payment/order history. It will allow salesman to add products into shopkeeper’s cart/make bill/update bill. This system will let company’s administrator (Owner) to manage stock of products.



**Figure 2. 1 Product Perspective**

### **2.4.2. User Characteristics**

User of this system can be person who owns the company or shop and purchases products from the company or salesman who works in the company. This is assumed that user (Owner, Salesman, and Shopkeeper) has basic knowledge of how to use web-based applications and able to perform tasks.

### **2.4.3. Constraints**

This is web-based application. So, user must have laptop/computer/mobile devices to access internet. Only registered shopkeepers/Salesman can perform their tasks. Non-registered users just view list of products. MySQL Database will be used. PHP will be used for backend. HTML, CSS, Bootstrap, JQuery, JavaScript, and Ajax will be used for frontend. GUI will be provided in English language.

### **2.4.4. Assumptions and dependencies**

Following are the assumptions:

- Shopkeepers have basic knowledge how to place order/view his bills/ask question etc.
- Salesman have knowledge how to use this system to complete his job.
- Administrator have complete knowledge about system.
- This is web based application therefore it can be used from all operating system.

Following are the Dependencies:

- Internet should be available 24 hours.
- Database should be connected to the system 24 hours

## **2.5. External Interface Requirements**

### **2.5.1. User Interfaces**

User interface or screens will be designed in such a way that it will look user friendly. It will be single page application that provides major functionality on some clicks. These interface will display proper message on any action by user i.e. guide user if he lost at any place. User guide book (project documentation) will be sufficient to guide the users how to use this product without facing any problems or difficulties. Tunneling technique will be used because before selecting products shopkeeper wouldn't be able to place order. Tailoring technique will also provide to the user such that he/she will be able to view customized information related to his/her interest. The color scheme is friendly and one can feel pleased to use this website.



### **2.5.2. Hardware Interfaces**

- Keyboard
- Mouse
- Screen

### **2.5.3. Software Interfaces**

- Web browser (any)
- MySQL (Database)
- Notepad++
- Operating System (any)
- Xampp Server

### **2.5.4. Communication Interfaces**

Communication protocols required for the communication is Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) for communication over the internet. By using these protocol, anyone can communicate with server.

## **2.6. Software Quality Attributes**

### **2.6.1. Availability**

High Availability is the measure of the quality of a software to keep functioning in spite of problems. The system shall be available 24 hours. All the users are able to access the system at any time.

### **2.6.2. Maintainability**

Source code should be written in well-structured manner with commenting the functions, formulas, difficult portion of source code. If coding is done in such a manner then it would be easy to maintain the system. The database may crash at any certain time due to virus or an operating system failure. Therefore, it is required to keep back up of all data on the database to avoid any data loss.

### **2.6.3. Portability**

As this is a web based system, so it will available on Windows (PC), Linux as well as on Android, Windows Phone, iOS based Smartphone.

### **2.6.4. Reliability**

This web based application is 90% reliable. There is no occurrence of failure in case of submitting empty fields or any invalid data that is entered by user then application handles this user action by giving appropriate message. The database may get crashed at any certain time due to virus, operating system failure or server problem. Therefore, there will be backup of database that used in case of database crash. Fault tolerance will also use in case of server failure or crash.

### **2.6.5. Security**

Main tasks of all types of users can be perform only having username and password. Shopkeeper and salesman would be registered by the Company's administrator. Those users would be registered by administrator.

### **2.6.6. Performance**

Performance of a system defines how much time it will take to perform its functionality. Following are some performance measures for this system:

- Load time for user interface screens shall take no longer than two seconds.
- Queries shall be returned with results within five to ten seconds.
- Login information will be verified within ten seconds.

## **2.7. Functional Requirements**

This section of chapter includes functional requirements and functionalities which shall be provided through this application. These requirements are presented as use cases. Each use case is a requirement.

### **2.8.1. Use case Diagram**

A use case diagram is a graphical representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases. This type of drawing is a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. Use case diagram for this system is on next page.

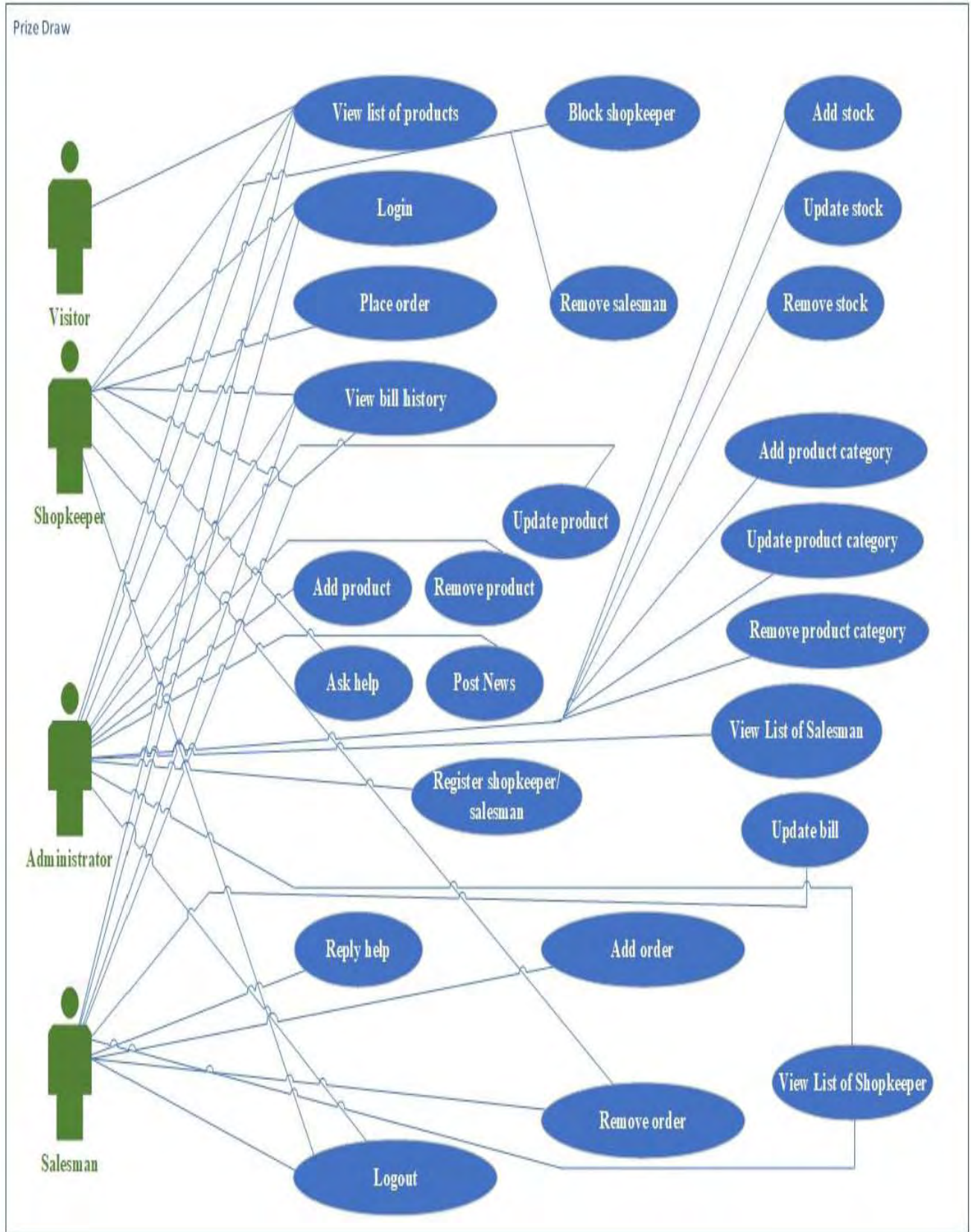


Figure 2. 2 Use case diagram

### 2.8.2. Use case Description

Writing use cases description is an excellent technique to understand and describe requirements. Use cases are a mechanism to help to understand which the system is going to be developed. Informally, they are stories of using a system to meet goals.

#### Use case 1: Login

<b>ID</b>	<b>UC-1</b>
<b>Name</b>	<b>Login</b>
<b>Primary Actor(s)</b>	Administrator/Salesman/Shopkeeper
<b>Pre-Conditions</b>	Administrator Salesman/Shopkeeper is registered in the system.
<b>Post-Conditions</b>	Administrator /Salesman/Shopkeeper logged in and perform activities according to their role.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1) The system requests that the Administrator /Salesman/Shopkeeper to enter username and password.</li> <li>2) Administrator /Salesman/Shopkeeper enters his username and password.</li> <li>3) The system validates the entered username and password and logs Administrator /Salesman/Shopkeeper into the system.</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a) System fails at any time.</li> <li>2a) Wrong or incomplete entry of either username or password.</li> <li>3a) Invalid Username/Password which means system determines that the username/password does not match a username/password for any account.</li> <li>3b) Deactivated username and password.</li> </ol>

**Table 2. 2 Login Use case Description**

#### Use case 2: Register salesman/shopkeeper

<b>ID</b>	<b>UC-2</b>
<b>Name</b>	<b>Register salesman/shopkeeper</b>
<b>Primary Actor(s)</b>	Administrator /Salesman/Shopkeeper
<b>Pre-Conditions</b>	Administrator is authenticated and logged in the system.
<b>Post-Conditions</b>	Administrator will successfully add Salesman/Shopkeeper.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1) Administrator selects Add New Salesman/Shopkeeper.</li> <li>2) Administrator enters their relevant information (i.e. name, username, password, Shop name, address, email &amp; contact number from shopkeeper &amp; name, username, password, email, contact number &amp; address etc.)</li> <li>3) System saves it and shows related success message.</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a) System fails at any time.</li> <li>2a) Administrator fills incorrect or incomplete fields.</li> <li>3a) System doesn't save information due to failure.</li> <li>3b) System saves information but doesn't show success message.</li> </ol>

**Table 2. 3 Register salesman/shopkeeper use case description**

**Use case 3: Add product category**

<b>ID</b>	<b>UC-3</b>
<b>Name</b>	<b>Add product category</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Administrator has authenticated to system and logged in the system. Product category is not already added.
<b>Post-Conditions</b>	Product category added successfully and available to add for any product.
<b>Main Success Scenario</b>	1) Administrator select add product. 2) Administrator adds product category name, status and confirm it. 3) System makes relevant changes.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Administrator may not submit complete information like product category name, status. 2b) Administrator may not confirm. 3a) System may not make relevant changes.

**Table 2. 4 Add product category use case description****Use case 4: Update product category**

<b>ID</b>	<b>UC-4</b>
<b>Name</b>	<b>Update product category</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Product category exist in list of category in the system.
<b>Post-Conditions</b>	Product category information updated successfully
<b>Main Success Scenario</b>	1) Administrator selects update product category. 2) Administrator modifies product category name, status and confirm it. 3) System saves relevant changes and update it.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Administrator may not update any field. 2a) Administrator may not confirm to save changes. 3a) System may not save change and update.

**Table 2. 5 Update product category use case description****Use case 5: Remove product category**

<b>ID</b>	<b>UC-5</b>
<b>Name</b>	<b>Remove product category</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Product category is added in the system and available to add for any product.
<b>Post-Conditions</b>	Product category removed from the system and no longer available to add for against any product
<b>Main Success Scenario</b>	1) Administrator select remove product. 2) Administrator select desired product and remove it. 3) System saves changes and update list of products.

	4) Shopkeeper not be able to request for that product or salesman also not be able to add that product into cart for any shopkeeper.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Product is already removed. 4a) System may not save changes and product is still available to buy.

**Table 2. 6 Remove product category use case description**

**Use case 6: Add product**

<b>ID</b>	<b>UC-6</b>
<b>Name</b>	<b>Add product</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Administrator has authenticated to system and logged in the system. Product is not already added.
<b>Post-Conditions</b>	Product added successfully & available to buy.
<b>Main Success Scenario</b>	1) Administrator select add product. 2) Administrator adds product name, price and detail and confirm it. 3) System makes relevant changes.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Administrator may not submit complete information like product name, price and detail etc. 2b) Administrator may not confirm. 3a) System may not make relevant changes.

**Table 2. 7 Add product use case description**

**Use case 7: Update product**

<b>ID</b>	<b>UC-7</b>
<b>Name</b>	<b>Update product</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Product exist in list of products in the system.
<b>Post-Conditions</b>	Product information updated successfully
<b>Main Success Scenario</b>	1) Administrator selects update product. 2) Administrator modifies product name, price and detail and confirm it. 3) System saves relevant changes and update it.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Administrator may not update any field. 2a) Administrator may not confirm to save changes. 3a) System may not save change and update.

**Table 2. 8 Update product use case description**

**Use case 8: Remove product**

<b>ID</b>	<b>UC-8</b>
<b>Name</b>	<b>Remove product</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Product is added in the system and available to buy.

<b>Post-Conditions</b>	Product removed from the system and shopkeeper could not buy that product anymore.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1) Administrator select remove product.</li> <li>2) Administrator select desired product and remove it.</li> <li>3) System saves changes and update list of products.</li> <li>4) Shopkeeper not be able to request for that product or salesman also not be able to add that product into cart for any shopkeeper.</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a) System may fail at any time.</li> <li>2a) Product is already removed.</li> <li>4a) System may not save changes and product is still available to buy.</li> </ol>

**Table 2. 9 Remove product use case description**

**Use case 9: Add stock**

<b>ID</b>	<b>UC-9</b>
<b>Name</b>	<b>Add stock</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Administrator has authenticated to system and logged in the system. Product is already added.
<b>Post-Conditions</b>	Stock of product added successfully & product available to buy.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1) Administrator selects add stock.</li> <li>2) Administrator selects product and add stock for it.</li> <li>3) System makes relevant changes.</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a) System may fail at any time.</li> <li>2a) Product is not exist in the system.</li> </ol>

**Table 2. 10 Add stock use case description**

**Use case 10: Update stock**

<b>ID</b>	<b>UC-10</b>
<b>Name</b>	<b>Update stock</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Stock of product exist in the system.
<b>Post-Conditions</b>	Stock of product updated successfully.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>4) Administrator selects update stock.</li> <li>5) Administrator modifies stock quantity and confirm it.</li> <li>6) System saves relevant changes and update it.</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a) System may fail at any time.</li> <li>2a) Product may not exist in the system.</li> <li>2a) Administrator may not confirm to save changes.</li> <li>3a) System may not save change and update.</li> </ol>

**Table 2. 11 Update stock use case description**

**Use case 11: Remove stock**

<b>ID</b>	<b>UC-11</b>
<b>Name</b>	<b>Remove stock</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Stock of product is added in the system and available to buy.

<b>Post-Conditions</b>	Stock of product is removed from the system and shopkeeper could not buy that product anymore.
<b>Main Success Scenario</b>	1) Administrator select remove stock. 2) Administrator select stock of desired product and remove it. 3) System saves changes and remove stock of product.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Product is already removed. 3a) System may not save changes and product is still available to buy due to non-removal of stock.

**Table 2. 12 Remove stock use case description**

**Use case 12: View list of products**

<b>ID</b>	<b>UC-12</b>
<b>Name</b>	<b>View list of products</b>
<b>Primary Actor(s)</b>	Administrator/Salesman/Shopkeeper/Visitor
<b>Pre-Conditions</b>	Products added in the system.
<b>Post-Conditions</b>	List of products with their price and description have seen.
<b>Main Success Scenario</b>	1) User select view list of products. 2) List of products shown to him.
<b>Alternatives</b>	1a) System may fail at any time. 2a) List of products may be empty.

**Table 2. 13 View list of products use case description**

**Use case 13: Place order**

<b>ID</b>	<b>UC-13</b>
<b>Name</b>	<b>Place Order</b>
<b>Primary Actor(s)</b>	Shopkeeper
<b>Pre-Conditions</b>	Shopkeeper has an account to place order.
<b>Post-Conditions</b>	Order placed successfully.
<b>Main Success Scenario</b>	1) Shopkeeper select place order. 2) Shopkeeper selects products from list of products and also their quantity and confirm them. 3) Bill of products is shown to him.
<b>Alternatives</b>	1a) System may fail at any time. 2a) List of products may be empty. 2b) The product which shopkeeper wants to buy may not be available. 2c) Shopkeeper may not enter quantity of products. 2d) Shopkeeper may place wrong order. 3a) Bill of products may not be shown to him

**Table 2. 14 Place order use case description**

**Use case 14: View bill history**

<b>ID</b>	<b>UC-14</b>
<b>Name</b>	<b>View bill history</b>
<b>Primary Actor(s)</b>	Shopkeeper, Administrator, Salesman



<b>Pre-Conditions</b>	Shopkeeper logged in the system. He buys some products earlier or pay for them or have some remaining balance. Company has sale products.
<b>Post-Conditions</b>	Shopkeeper, Administrator, Salesman viewed bill or bill history.
<b>Main Success Scenario</b>	<b>Shopkeeper:</b> 1) Selects his bill/bill history 2) He views his paid/unpaid bills. <b>Administrator /Salesman:</b> 3) Selects bill history. 4) They view bill/bill history of a specific shopkeeper or all shopkeepers.
<b>Alternatives</b>	1/3a) System may fail at any time. 2a) Shopkeeper may not view his paid/unpaid bills. 2b) Some bill may be missed. 4a) Administrator /Salesman may not view any specific shopkeeper's bill history. 4b) Some bills or some shopkeeper's bill history may not be available.

**Table 2. 15 View bill history use case description**

**Use case 15: Add order**

<b>ID</b>	<b>UC-15</b>
<b>Name</b>	<b>Add order</b>
<b>Primary Actor(s)</b>	Salesman
<b>Pre-Conditions</b>	Shopkeeper is registered with the system and he is not block.
<b>Post-Conditions</b>	Ordered products added into the cart.
<b>Main Success Scenario</b>	1) Salesman selects add order into cart. 2) Salesman enters products and quantity for shopkeeper and saves it which modifies shopkeeper's remaining amount. 3) System saves all changes and modifies it.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Block shopkeeper is showing and salesman select him and book a order for him. 2b) Salesman enter incorrect product and quantity. 2c) Remaining balance may not be modified due any issue by system.

**Table 2. 16 Add order into cart use case description**

**Use case 16: Remove order**

<b>ID</b>	<b>UC-16</b>
<b>Name</b>	<b>Remove order</b>
<b>Primary Actor(s)</b>	Salesman
<b>Pre-Conditions</b>	Order is added into the cart.
<b>Post-Conditions</b>	Order is removed from the cart.
<b>Main Success Scenario</b>	1) He selects order and confirm to remove it. 2) System saves changes and order.
<b>Alternatives</b>	*a) System may fail at any time. 1a) Order has been already removed.

	1b) Order has been delivered.
--	-------------------------------

Table 2. 17 Remove order from cart use case description

**Use case 17: Update bill**

<b>ID</b>	<b>UC-17</b>
<b>Name</b>	<b>Update bill</b>
<b>Primary Actor(s)</b>	Salesman
<b>Pre-Conditions</b>	Shopkeeper has purchased no product yet or purchased either by placing online order or order on salesman's visit
<b>Post-Conditions</b>	Total amount of bill and number of products modified/updated successfully.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1) Salesman select update bill.</li> <li>2) Salesman adds payment received or when he adds products into cart for shopkeeper during his visit to shopkeeper's shop.</li> <li>3) System make changes and update shopkeeper's bill.</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a) System may fail at any time.</li> <li>2a) Shopkeeper has nothing to pay.</li> <li>2b) Salesman doesn't order anything.</li> <li>2c) When order is added into cart by salesman, shopkeeper's bill may not be updated.</li> </ol>

Table 2. 18 Update bill use case description

**Use case 18: Remove shopkeeper**

<b>ID</b>	<b>UC-18</b>
<b>Name</b>	<b>Remove shopkeeper</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Shopkeeper is registered in the system.
<b>Post-Conditions</b>	Shopkeeper had not login.
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1) Administrator selects remove shopkeeper.</li> <li>2) List of shopkeeper shown to him.</li> <li>3) Administrator selects one of the shopkeeper from the list and remove him.</li> <li>4) Shopkeeper moves to trash.</li> <li>5) System saves changes.</li> </ol>
<b>Alternatives</b>	<ol style="list-style-type: none"> <li>1a) System may fail at any time.</li> <li>2a) List of shopkeeper may not show to administrator.</li> <li>3a) Shopkeeper may already remove.</li> <li>4a) Shopkeeper may delete permanently.</li> </ol>

Table 2. 19 Block shopkeeper use case description

**Use case 19: Remove salesman**

<b>ID</b>	<b>UC-19</b>
<b>Name</b>	<b>Remove salesman</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Salesman is registered with the system and able to perform its job.
<b>Post-Conditions</b>	Salesman removed and not able to do his tasks anymore.

<b>Main Success Scenario</b>	1) Administrator selects block salesman. 2) Administrator blocks undesired salesman and confirm it. System saves changes and doesn't allow to perform any activity in future.
<b>Alternatives</b>	1a) System fails at any time. 2a) Administrator blocks undesired person. 2b) Administrator blocks but not confirm it. 2c) Administrator neither block nor confirm it. 3a) System doesn't update changes. 3b) System may allow to perform any activity even after blocking.

**Table 2. 20 Remove salesman use case description**

**Use case 20: Ask for help**

<b>ID</b>	<b>UC-20</b>
<b>Name</b>	<b>Ask for help</b>
<b>Primary Actor(s)</b>	Shopkeeper
<b>Pre-Conditions</b>	Shopkeeper is registered with the system.
<b>Post-Conditions</b>	Question is submitted to salesman/administrator
<b>Main Success Scenario</b>	1) Shopkeeper selects ask for help. 2) Shopkeeper write his issue and submit it. 3) Question is received to salesman.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Shopkeeper's query may not be submitted to salesman or administrator.

**Table 2. 21 Ask for help use case description**

**Use case 21: View list of salesman**

<b>ID</b>	<b>UC-21</b>
<b>Name</b>	<b>View list of salesman</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Salesman is registered with the system.
<b>Post-Conditions</b>	Administrator has viewed list of salesman.
<b>Main Success Scenario</b>	1) Administrator selects view list of salesman. 2) List of salesman shown to him.
<b>Alternatives</b>	1a) System may fail at any time. 2a) List of salesman is empty.

**Table 2. 22 View list of salesman use case description**

**Use case 22: View List of shopkeeper**

<b>ID</b>	<b>UC-22</b>
<b>Name</b>	<b>View list of shopkeeper</b>
<b>Primary Actor(s)</b>	Administrator/Salesman
<b>Pre-Conditions</b>	Shopkeeper is registered with this system.
<b>Post-Conditions</b>	Administrator/Salesman viewed list of shopkeeper and perform desired operation for them.
<b>Main Success</b>	1) Administrator/Salesman selects view list of shopkeeper.

<b>Scenario</b>	2) Administrator/Salesman viewed list of shopkeeper.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Shopkeeper may not be added.

Table 2. 23 View list of shopkeeper use case description

**Use case 23: Post news**

<b>ID</b>	<b>UC-23</b>
<b>Name</b>	<b>Post news</b>
<b>Primary Actor(s)</b>	Administrator
<b>Pre-Conditions</b>	Administrator is logged in the system.
<b>Post-Conditions</b>	Administrator posted news.
<b>Main Success Scenario</b>	1) Administrator selects post news. 2) Administrator adds relevant news and post it. 3) System saves changes and make available for type of users.
<b>Alternatives</b>	1a) System may fail at any time. 2a) News may not be posted. 3a) News may not available for each type of users.

Table 2. 24 Post news use case description

**Use case 24: Logout**

<b>ID</b>	<b>UC-24</b>
<b>Name</b>	<b>Logout</b>
<b>Primary Actor(s)</b>	Registered users
<b>Pre-Conditions</b>	Registered users logged in.
<b>Post-Conditions</b>	Registered users successfully logout and not be able to perform any activity which is allow to registered user according to their roles.
<b>Main Success Scenario</b>	1) Registered users select logout. 2) They get logout. 3) They are unable to perform any activity after logout action.
<b>Alternatives</b>	1a) System may fail at any time. 2a) Registered users may not logout. 3a) Registered users may still be able to perform activity which is only for registered users according to their role.

Table 2. 25 Logout use case description

**Use case 25: Reply help**

<b>ID</b>	<b>UC-25</b>
<b>Name</b>	<b>Reply help</b>
<b>Primary Actor(s)</b>	Salesmen
<b>Pre-Conditions</b>	Salesmen logged in.
<b>Post-Conditions</b>	Salesmen replied successfully.
<b>Main Success Scenario</b>	1. Salesmen select shopkeepers help request and replied it. 2. Help response received to shopkeeper.
<b>Alternatives</b>	1a) System may fail at any time or help request already respond.

Table 2. 26 Reply help use case description

# Chapter 3

## Software Design Description

The purpose of this chapter is to elaborate required diagrams which covers working of this system. This chapter specifies the general factors that can affect the product and its requirements, providing a background for the requirements of the software.

### 3.1. Introduction

To describe the key concepts of the problem domain and data items used in our system, the following models have been used system sequence diagram, architectural diagram, data flow diagram, entity relationship diagram and user interaction diagrams. Software Design Description (SDD) is a representation of a system that how the users can interact with the system. This can be shown with the help of diagrams.

#### 3.1.1. Design Overview

In the design overview, we can describe the system at architecture level using architectural diagram. In this we can explain how the user interacts with the system. In System sequence diagrams, these diagrams show how system is completely fulfill the requirement. Data flow is represented using Data flow diagrams. User interaction diagrams depicts how interfaces look like.

#### 3.1.2. Requirement Traceability Matrix

Requirement Traceability Matrix captures all requirements proposed by the client or development team.

Use Case ID	Use case name	Test case	SSD
UC-1	Login	TC-001	SSD-1
UC-2	Register salesman/shopkeeper	TC-002	SSD-2
UC-3	Add product category	TC-003	No
UC-4	Update product category	TC-004	No
UC-5	Remove product category	TC-005	No
UC-6	Add product	No	SSD-3
UC-7	Update product	No	SSD-4
UC-8	Remove product	No	SSD-5
UC-9	Add stock	TC-006	SSD-6
UC-10	Update stock	No	No
UC-11	Remove stock	No	No

UC-12	View list of products	No	No
UC-13	Place order	TC-007	SSD-7
UC-14	View bill history	No	No
UC-15	Add order	No	No
UC-16	Remove order	TC-008	SSD-8
UC-17	Update bill	TC-009	SSD-9
UC-18	Remove shopkeeper	TC-010	SSD-10
UC-19	Remove salesman	TC-011	SSD-11
UC-20	Ask for help	TC-012	SSD-12
UC-21	View list of salesman	No	No
UC-22	View list of shopkeeper	No	No
UC-23	Post news	TC-013	SSD-13
UC-24	Logout	TC-014	SSD-14
UC-25	Reply help	No	No

**Table 3. 1 Requirement traceability matrix**

## 3.2. System Architecture Design

A system architecture is the conceptual model that defines the structure, behavior, and more aspects of a system. An architecture depiction is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

### 3.2.1. Chosen System Architecture

3-Tier architecture is used to make e-Cosmetics by Fair cosmetics. Further, in 3-Tier architecture I used structural approach to build e-Cosmetics. Basically, Three-tier (layer) is a client-server architecture in which the user interface, business process (business rules) and data storage are developed and maintained as independent modules or most often on separate platforms. Basically, there are 3 layers, tier 1 (presentation layer, GUI layer), tier 2 (business logic layer) and tier 3 (data layer). These tiers can be developed and tested separately. These three-tier architecture contains:

#### **Presentation layer**

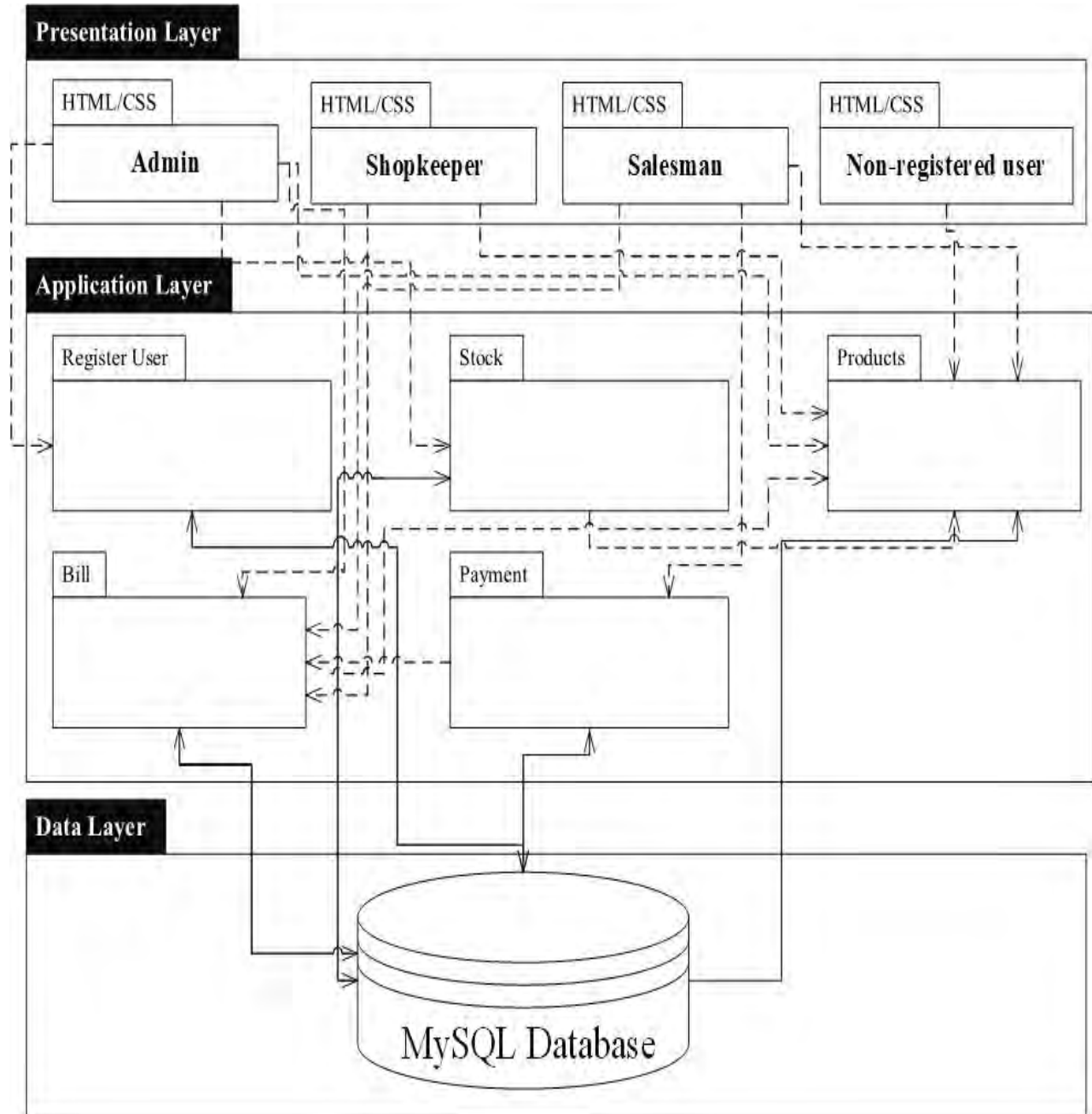
This is the topmost level of the application. The presentation tier displays information related to buy products, view bill, stock management etc. It communicates with other tiers by which it puts out the results to the browser/client tier and all other tiers in the network. In simple terms, it is a layer which users can access directly (such as a web page, or an operating system's GUI).

#### **Application layer**

The application layer is pulled out from the presentation layer and, as its own layer, it controls an application's functionality by performing detailed processing.

**Data layer**

This layer contains technical services which are necessary for proper running of the application.



**Figure 3. 1 System Architecture Diagram**

**3.3. System Sequence Diagrams (SSD)**

A SSD is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events. SSD is visual summaries of the individual use cases. There are thirty use cases, but I am making SSD of basics use cases.

SSD-1 (Login)

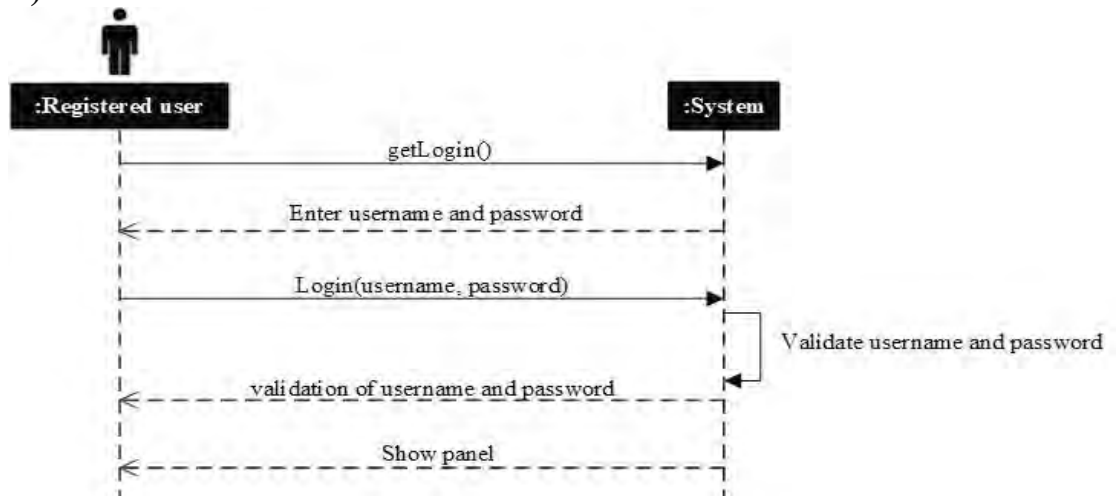


Figure 3. 2 SSD (Login)

SSD-2 (Register salesman/shopkeeper)

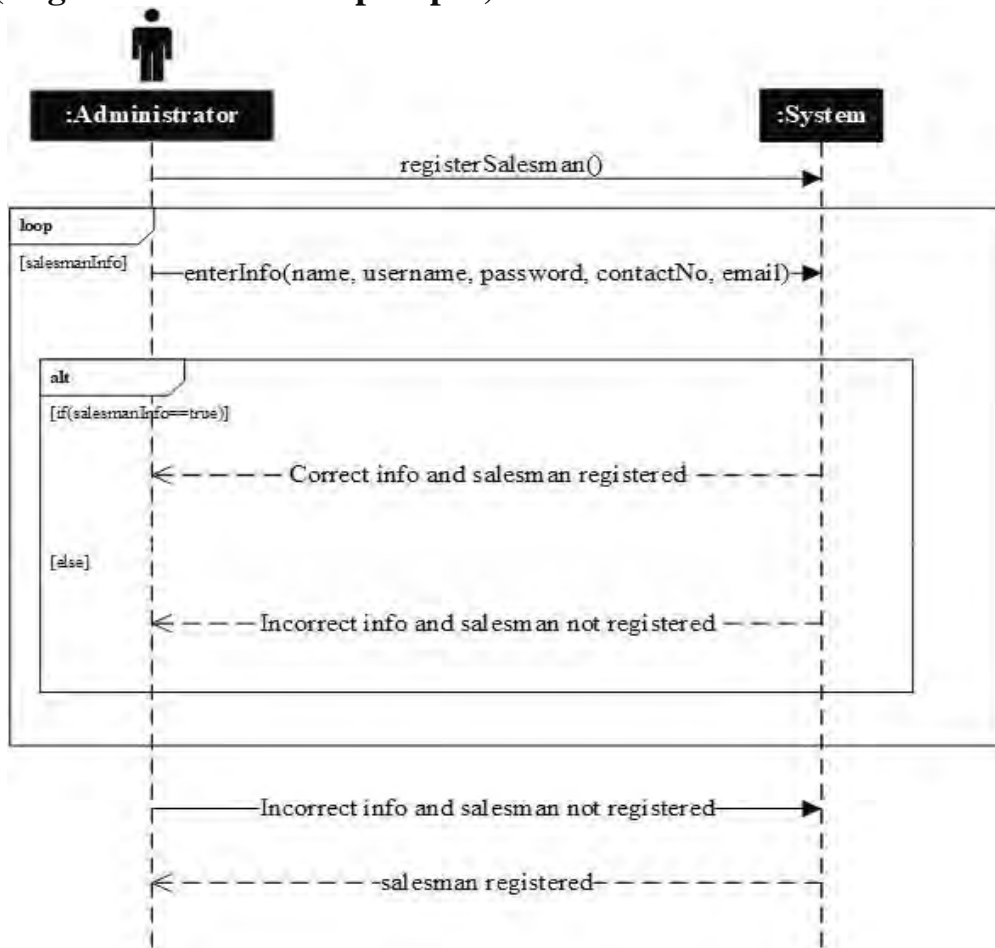


Figure 3. 3 SSD (Register shopkeeper/salesman)



SSD-3 (Add product)

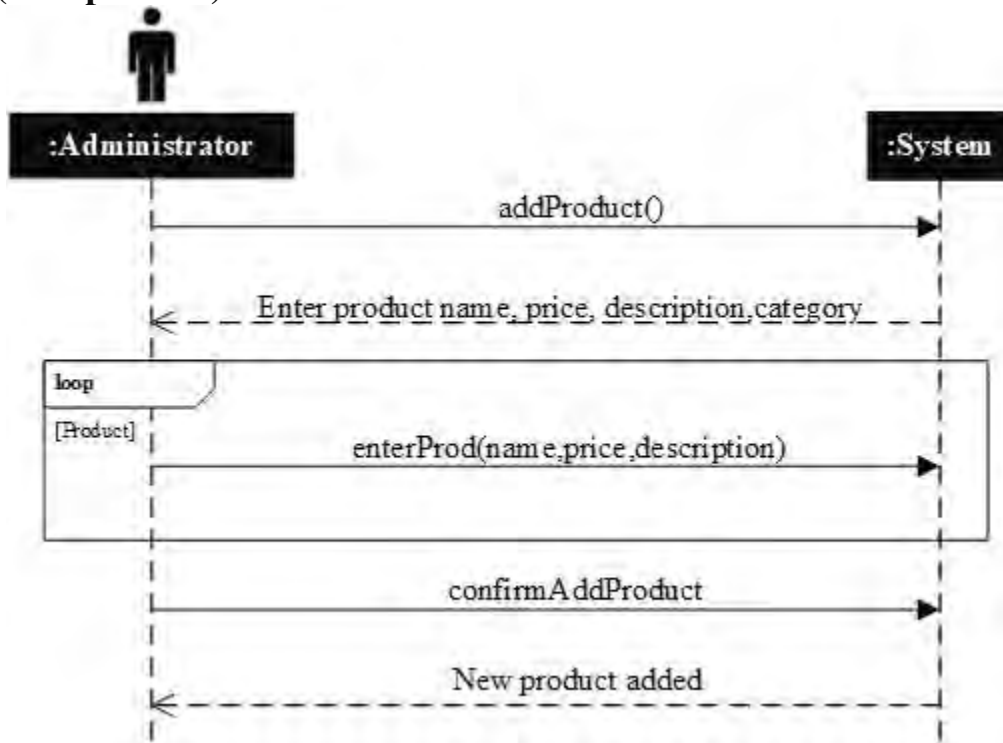


Figure 3. 4 SSD (Add product)

SSD-4 (Update product)

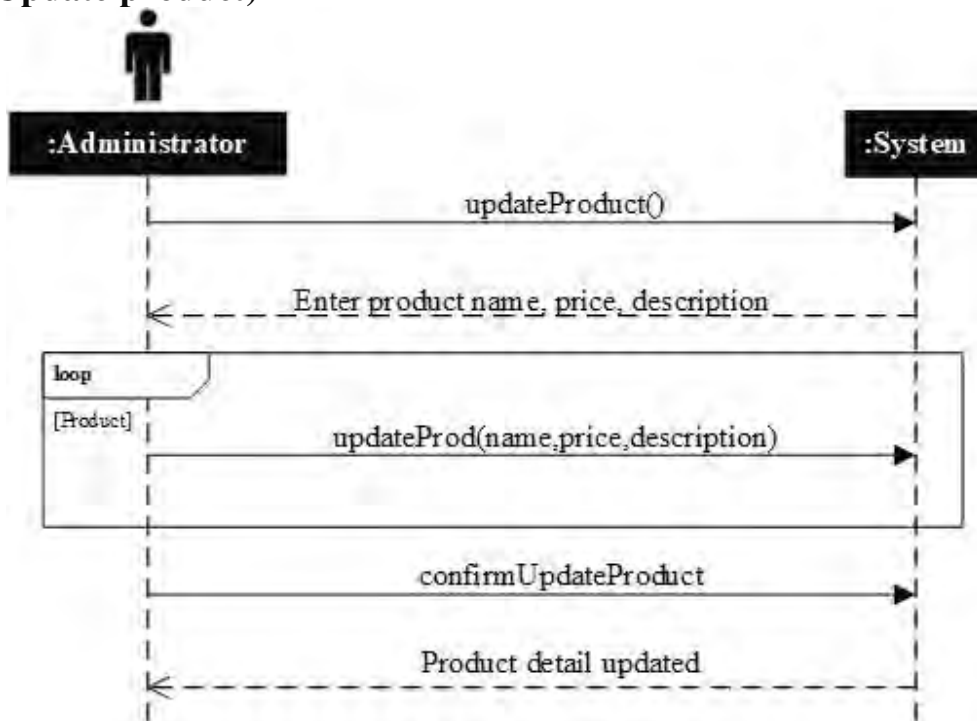


Figure 3. 5 SSD (Update product)

SSD-5 (Remove product)

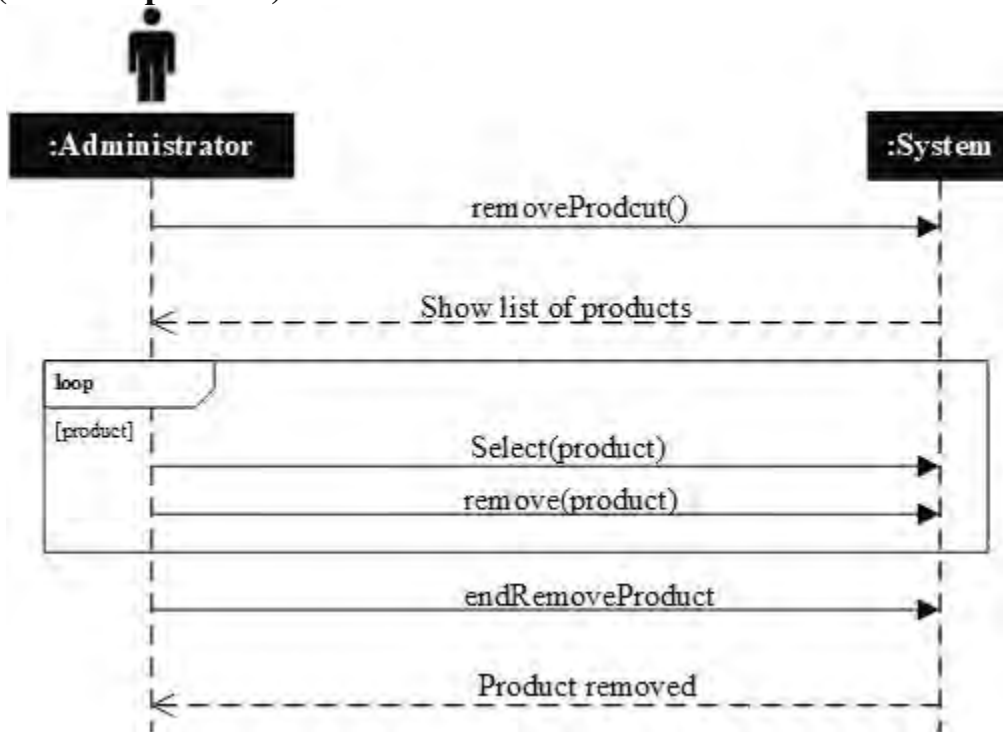


Figure 3. 6 SSD (Remove product)

SSD-6 (Add stock)

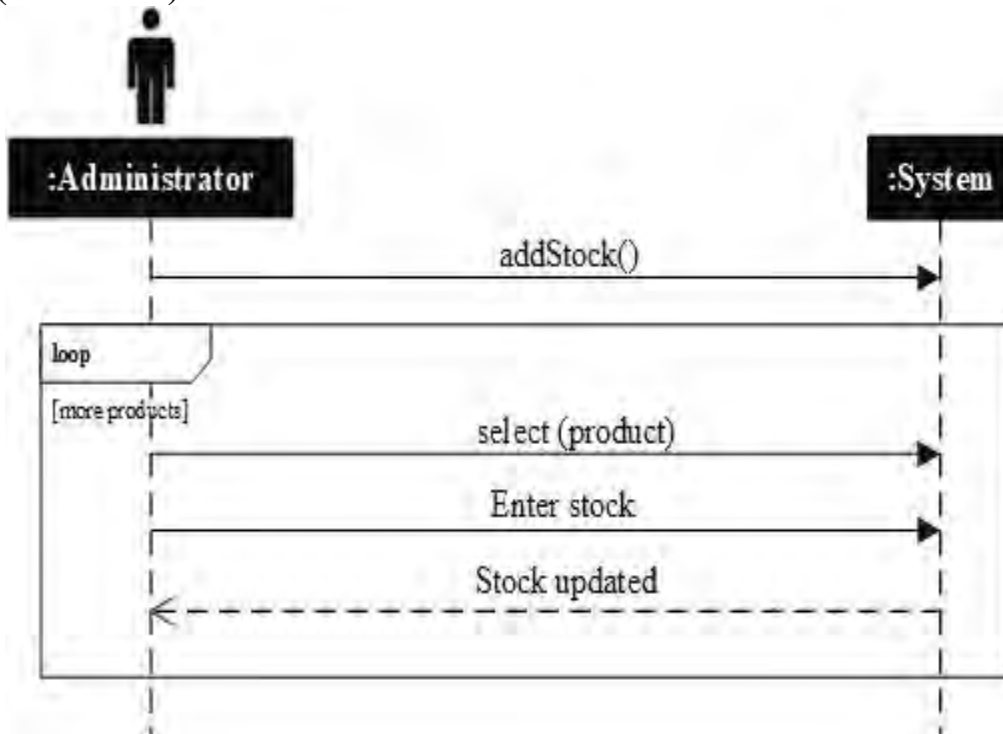


Figure 3. 7 SSD (Add stock)

SSD-7 (Place order)

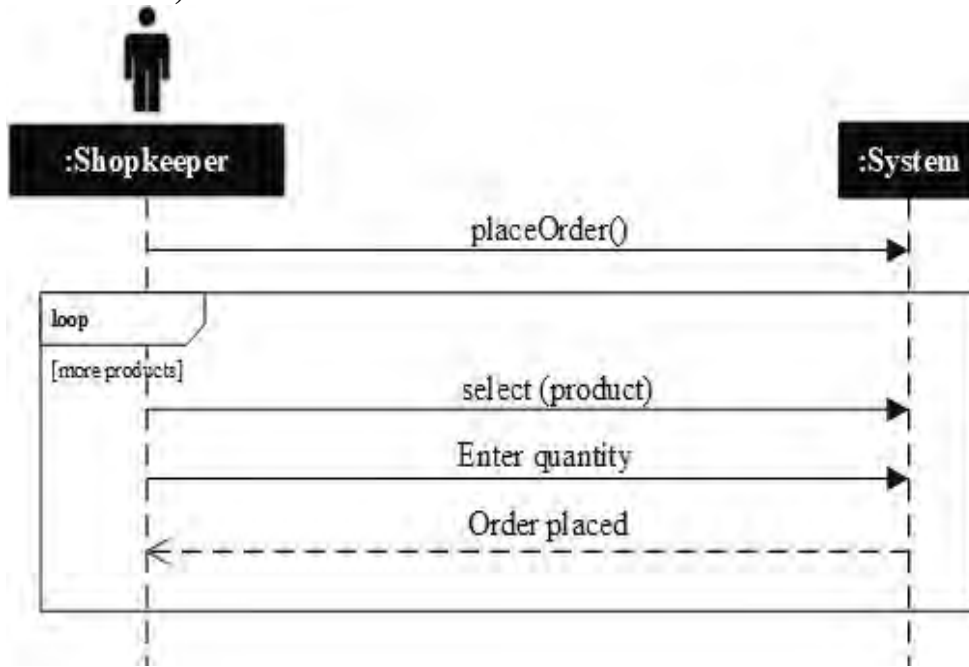


Figure 3. 8 SSD (Place order)

SSD-8 (Remove order)

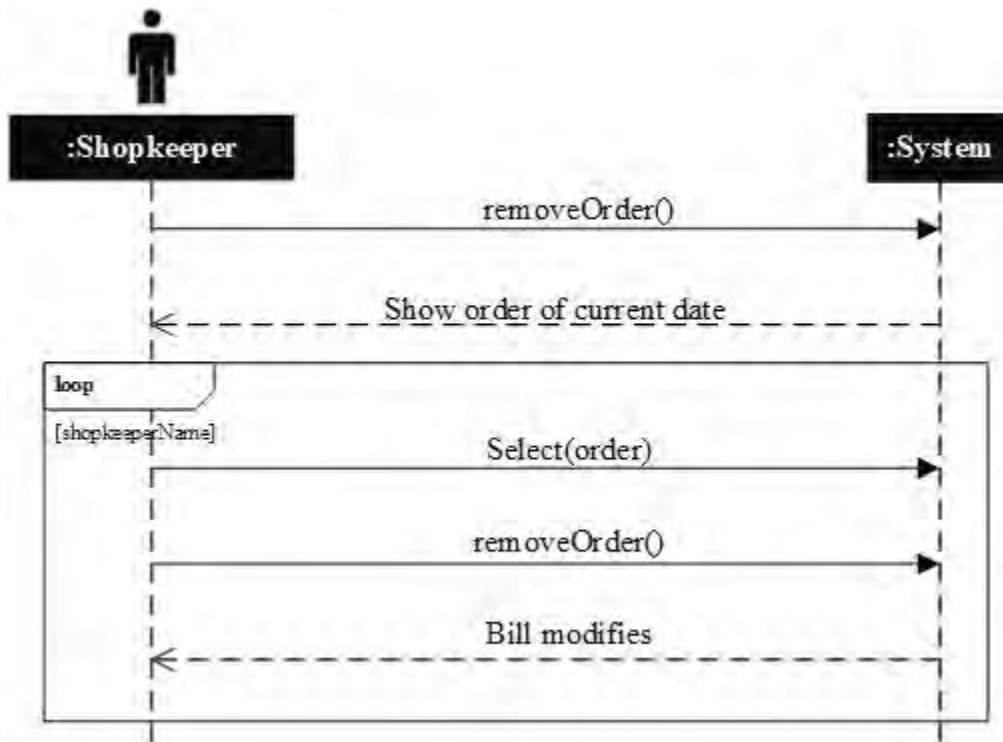


Figure 3. 9 SSD (Remove order)

SSD-9 (Update bill)

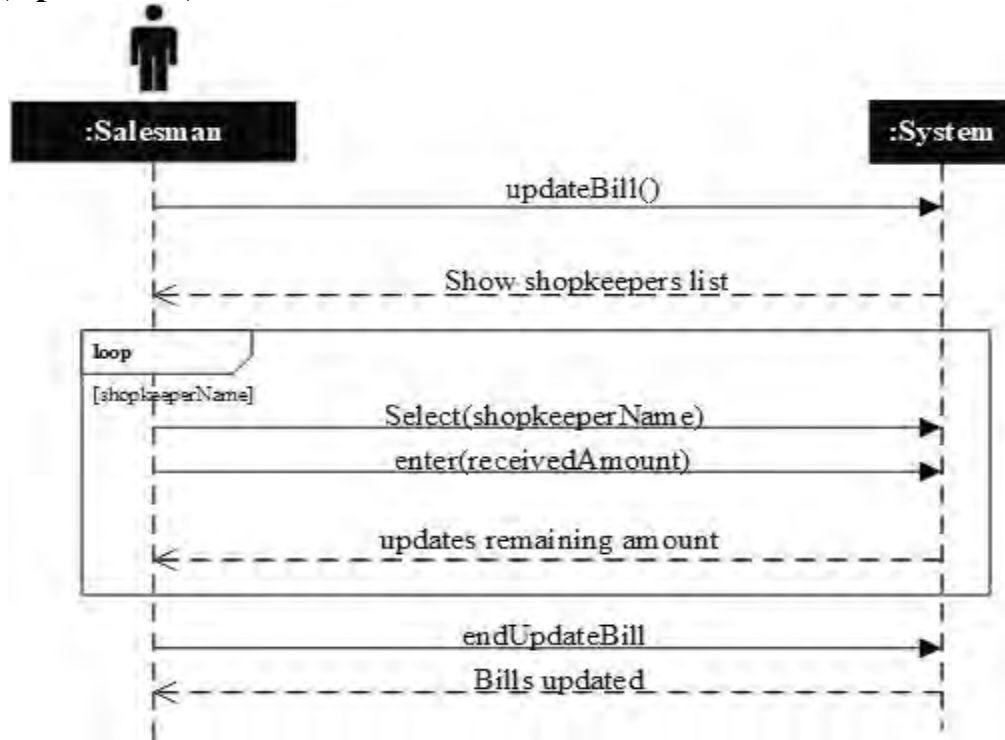


Figure 3. 10 SSD (Update bill)

SSD-10 (Remove shopkeeper)

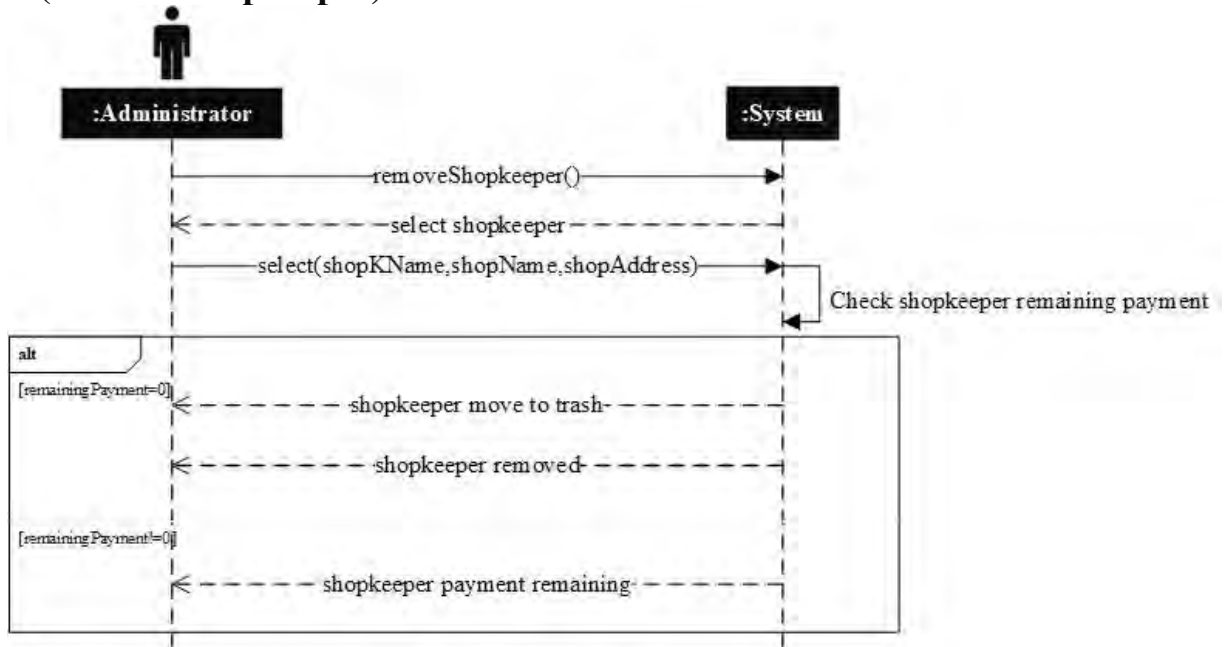


Figure 3. 11 SSD (Remove shopkeeper)

SSD-11 (Remove salesman)

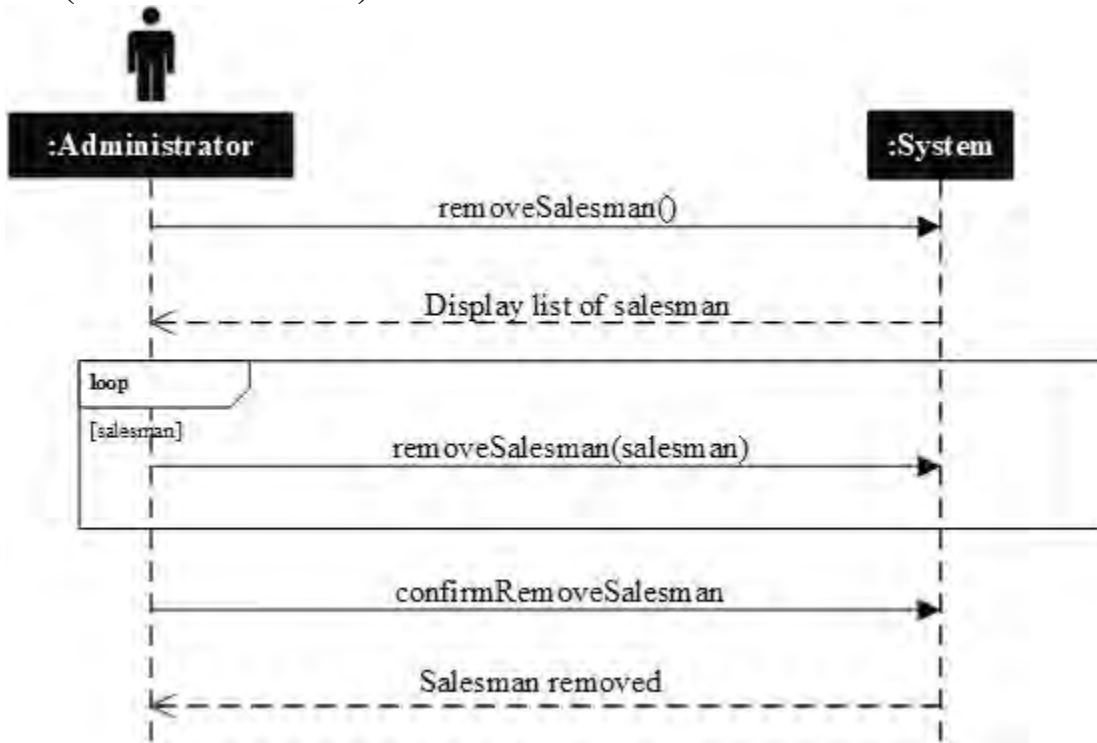


Figure 3. 12 SSD (Remove salesman)

SSD-12 (Ask for help)

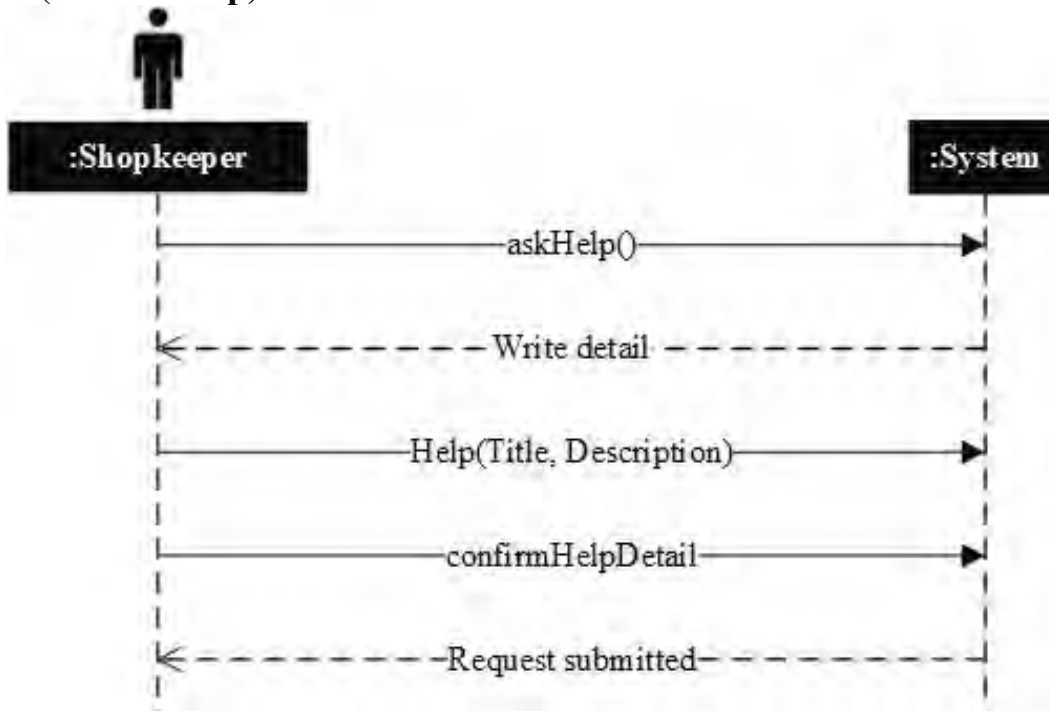


Figure 3. 13 SSD (Ask for help)

SSD-13 (Post news)

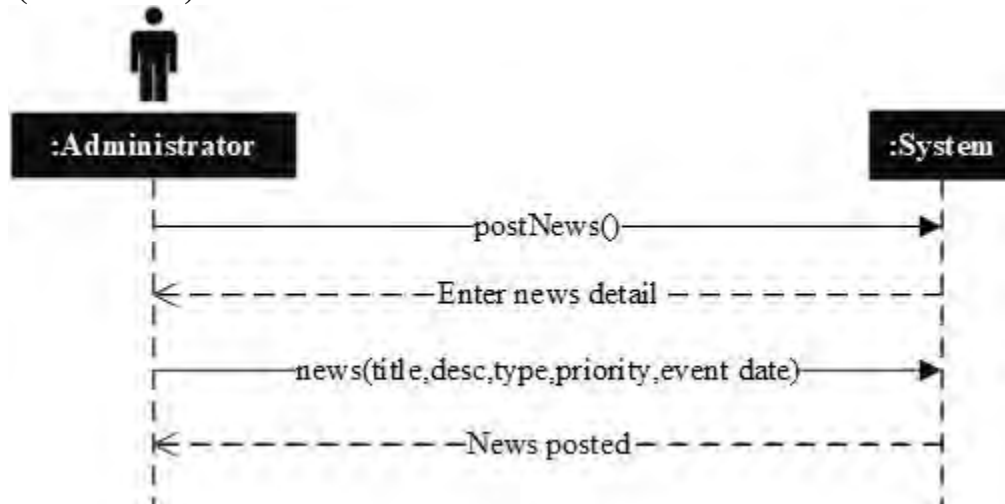


Figure 3. 14 SSD (Post news)

SSD-14 (Logout)

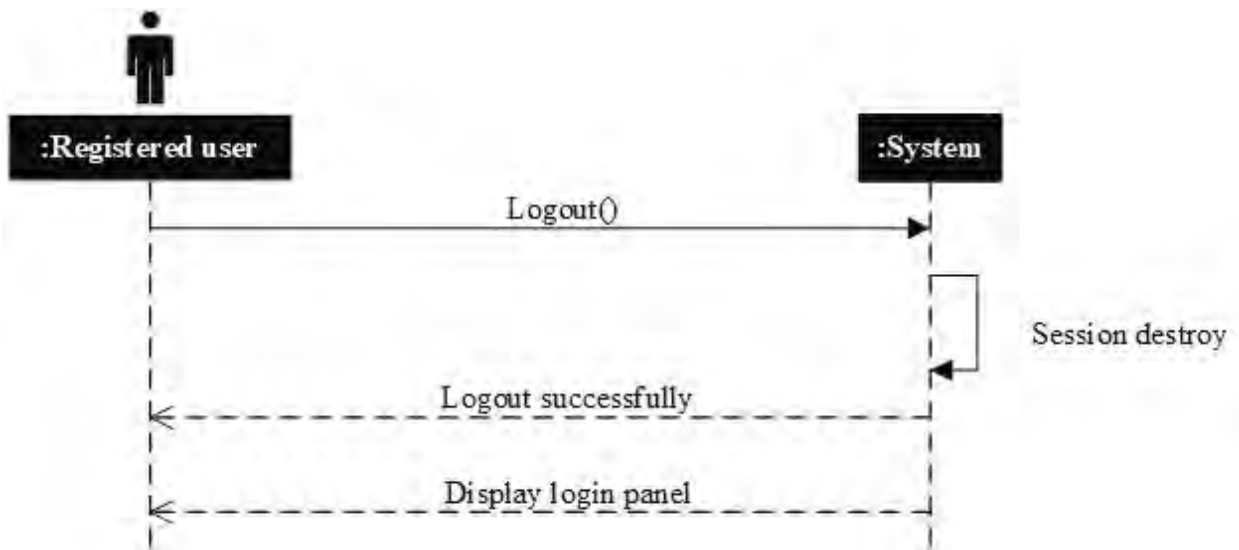


Figure 3. 15 SSD (Logout)

**3.4. Entity relationship diagram**

An entity relationship diagram (ERD) demonstrates the interactions of entity sets stored in a database. An entity set is a collection of similar entities. These entities can have attributes that define its properties. There are different notions uses for this purpose. But, I have used “Chen” style for designing entity relationship diagram. ERD is given below:

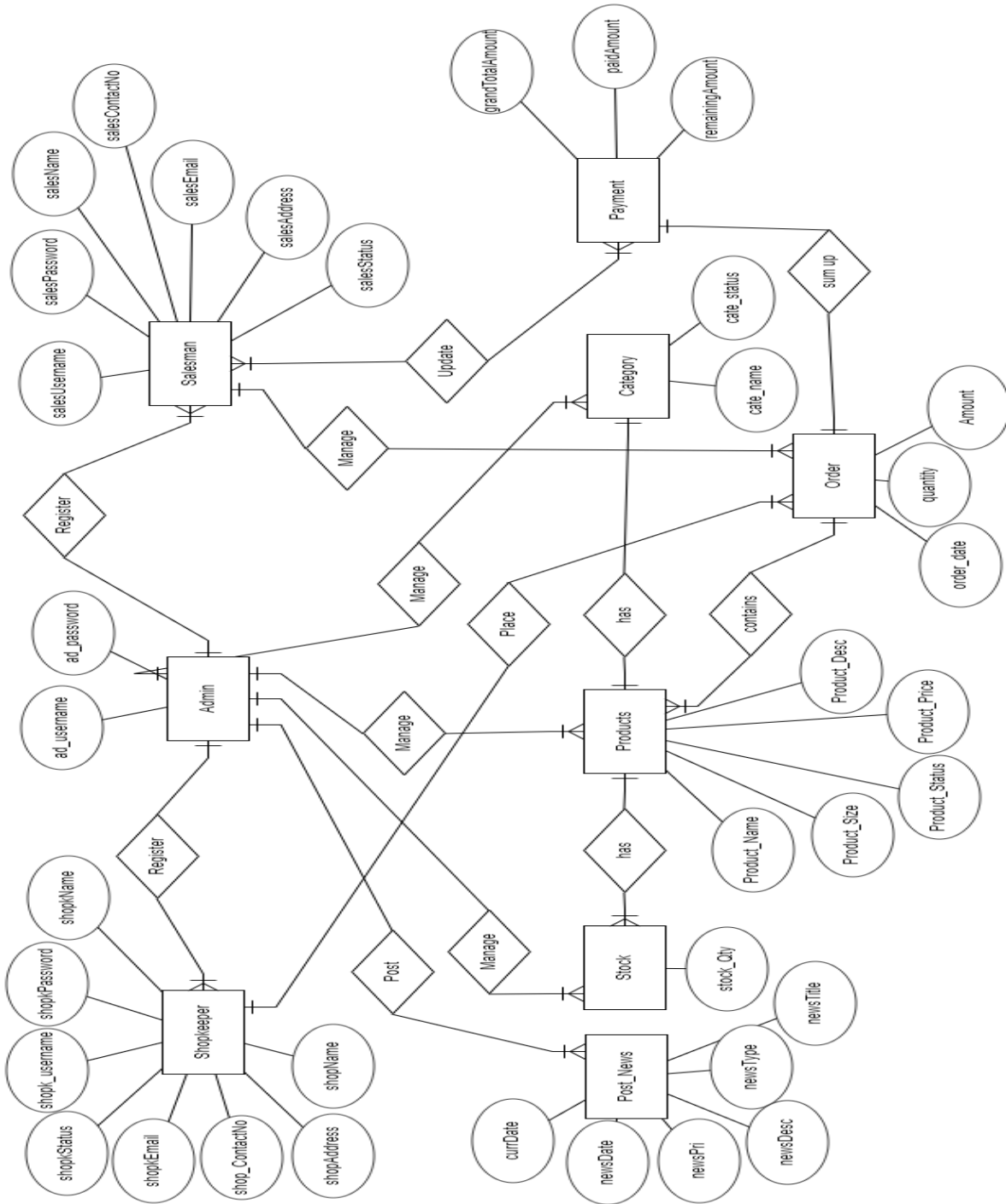


Figure 3. 16 Entity relationship diagram

### 3.5. Data flow diagrams (DFD)

A data flow diagram represents the way information flows through a process or system. It contains data inputs and outputs, data stores, and the various sub-processes the data moves through. I am following “Yourdon and Coad” notation for making DFD.

3.5.1. Zero level DFD (Context level)

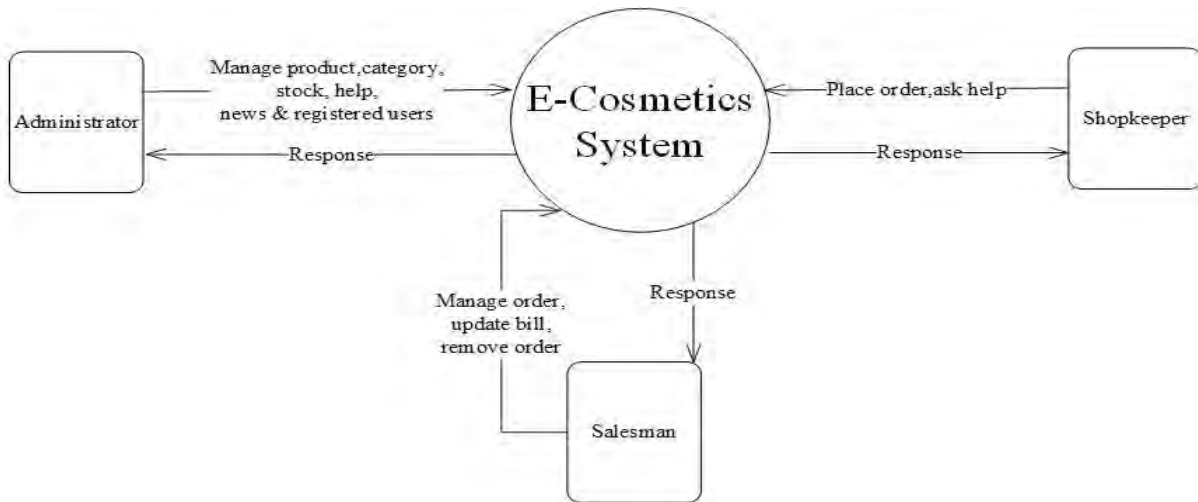


Figure 3. 17 DFD (0 level)

3.5.2. First level DFD (Admin)

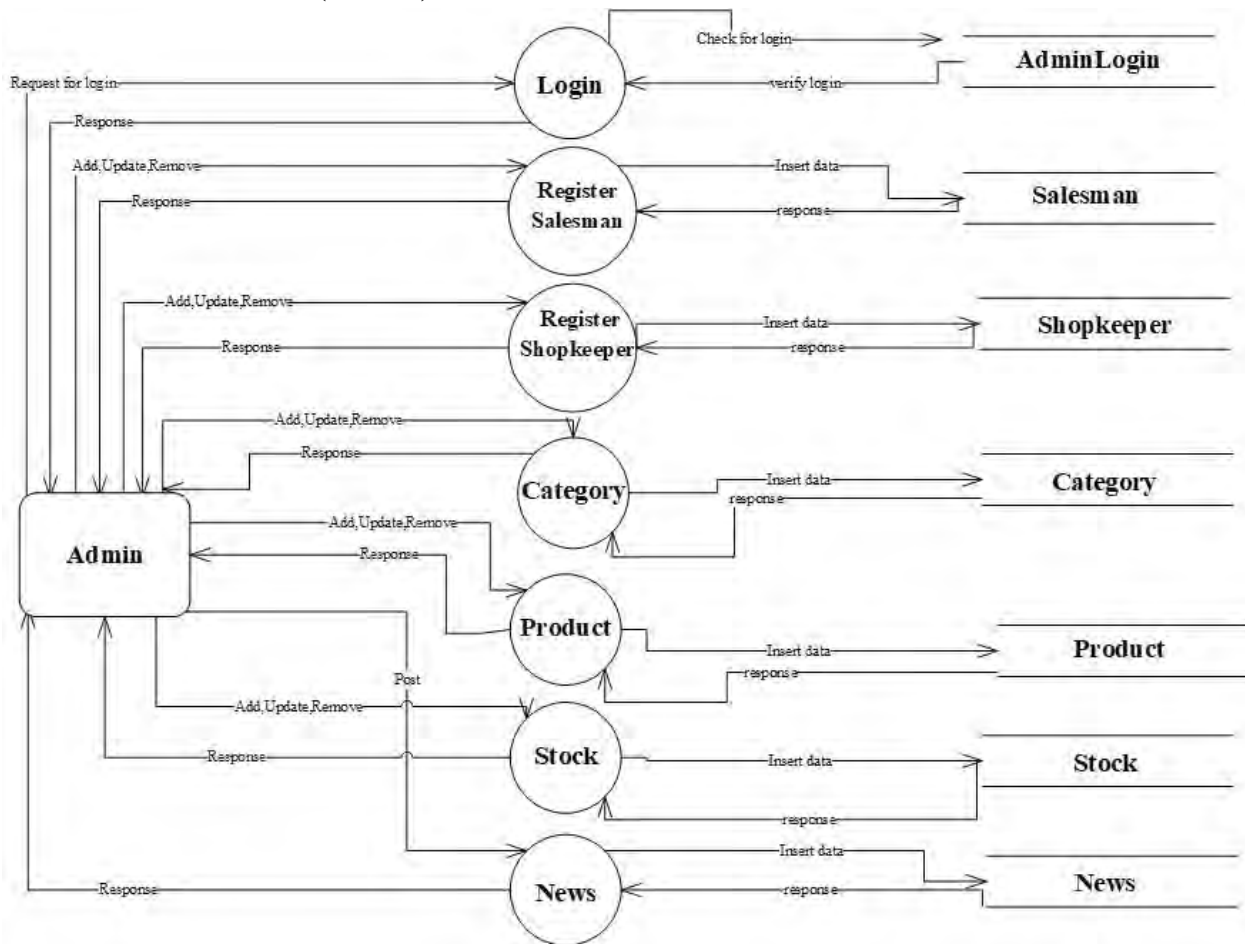


Figure 3. 18 1<sup>st</sup> level DFD (Admin)



3.5.3. First level DFD (Shopkeeper)

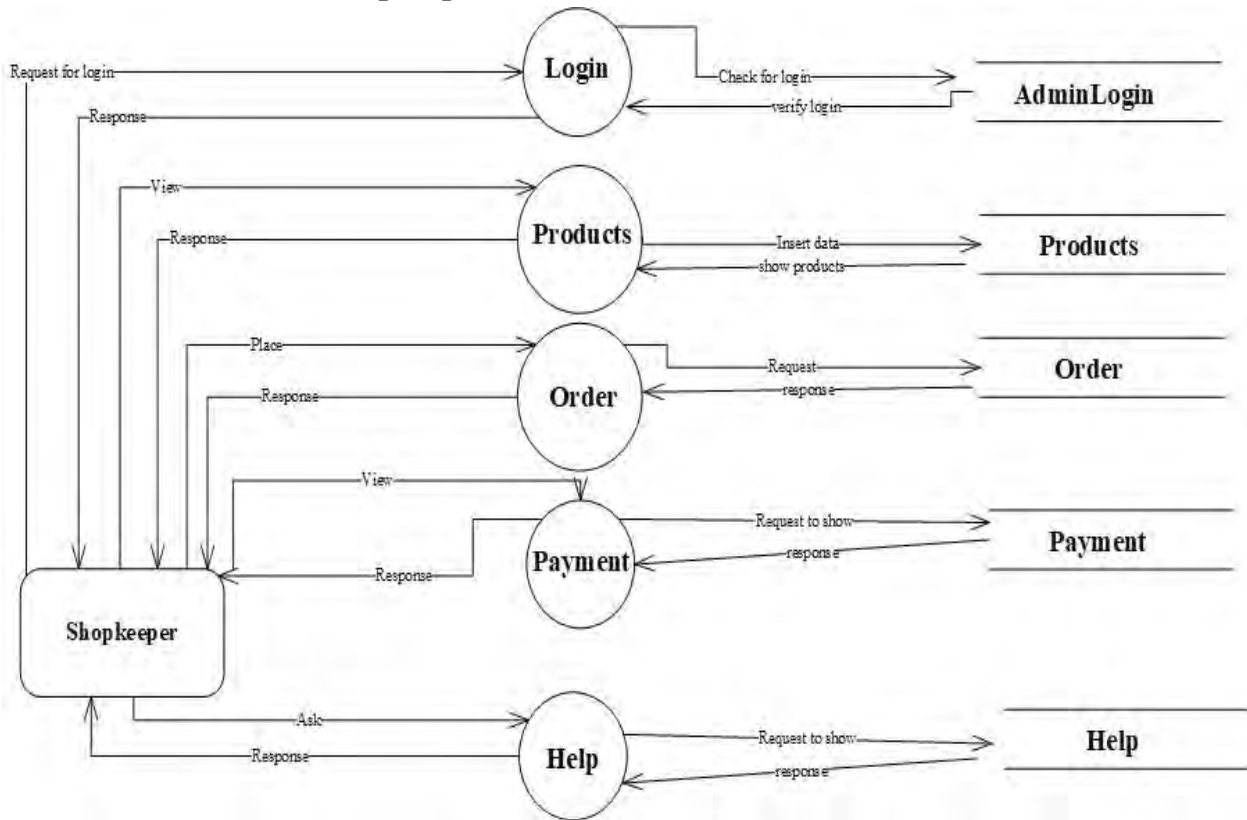


Figure 3. 19 1st level DFD (Shopkeeper)

3.5.4. First level DFD (Salesman)

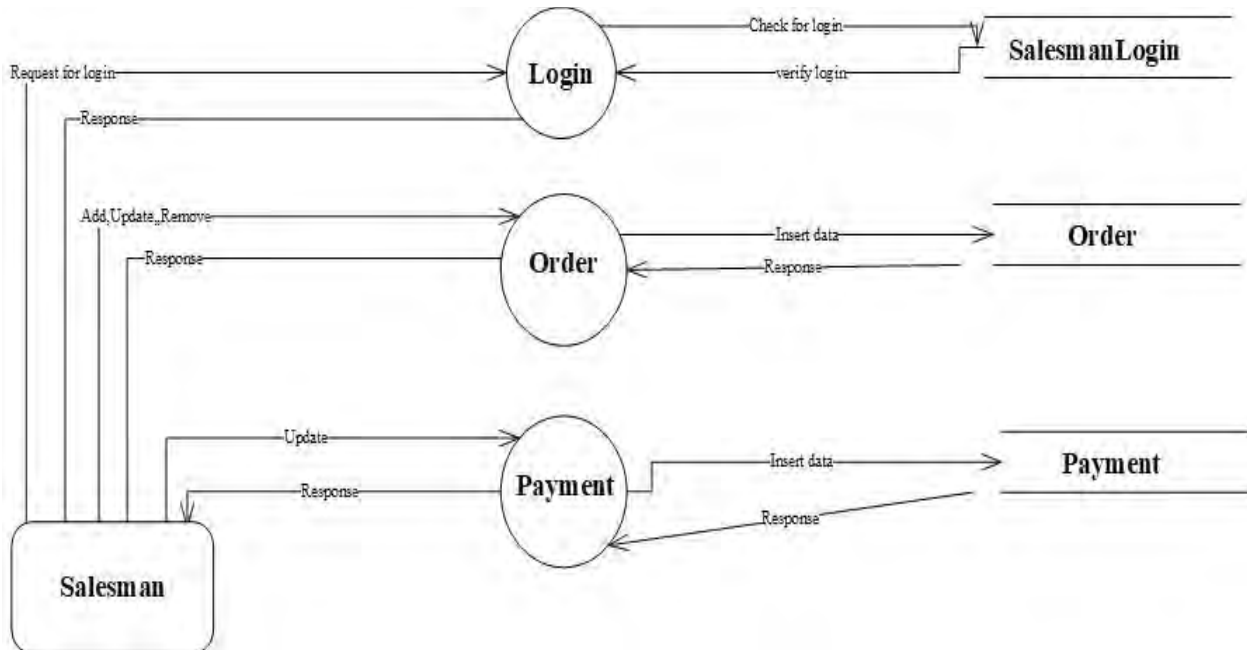


Figure 3. 20 1st level DFD (Salesman)

### 3.6. Activity diagrams

An activity is a state of doing something. The activity diagram describes the sequencing of activities. Activity diagram depicts the dynamic behavior of a system or part of a system through the flow of control between actions that the system performs. It is similar to a flowchart except that an activity diagram can show concurrent flows. Activity diagrams for this system is given in following:

#### 3.6.1. Manage product (Activity diagram)

Add, View, Update and Remove product by administrator is under the umbrella of manage product.

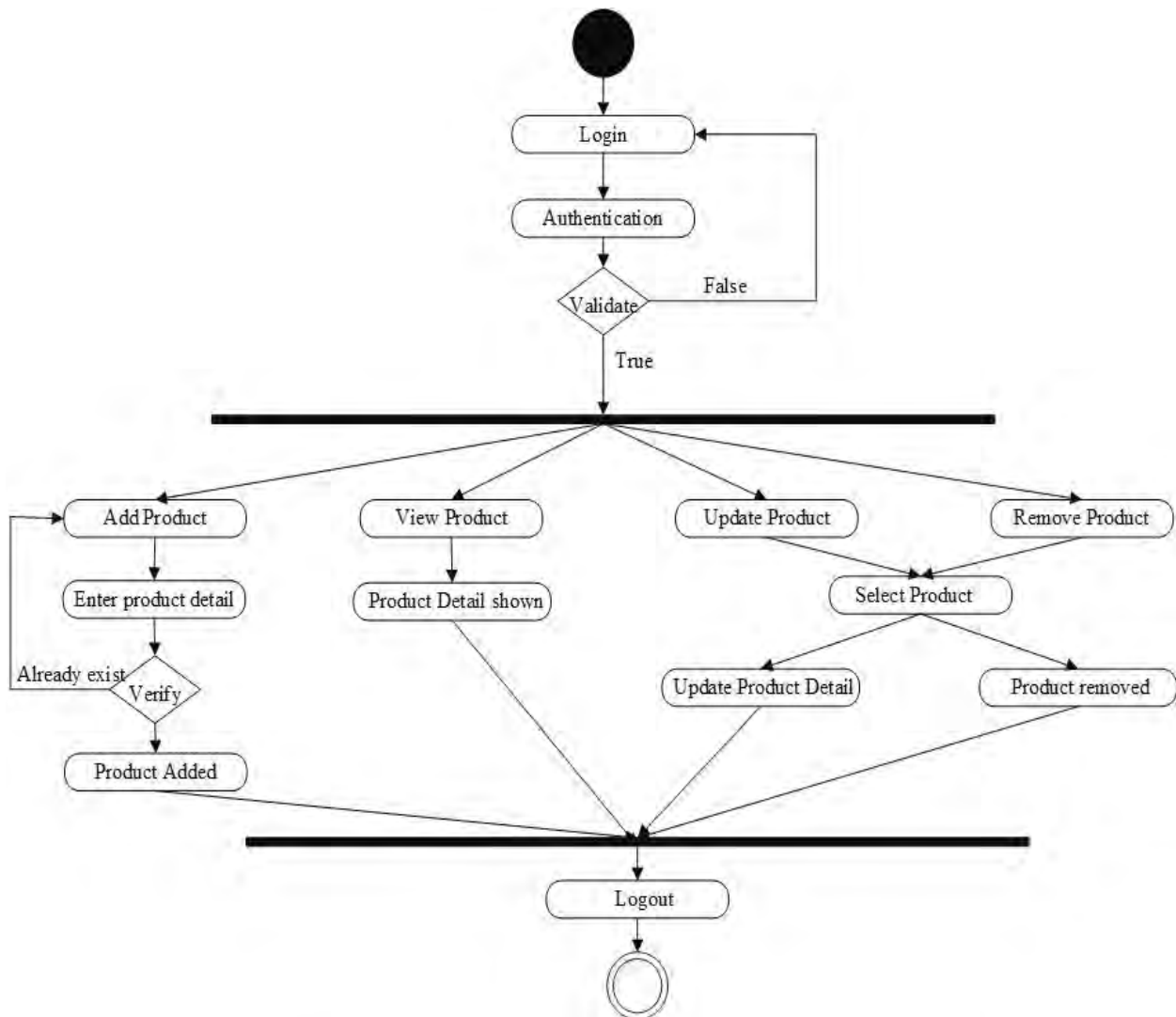


Figure 3. 21 Activity diagram (Manage product)

#### 3.6.2. Place order, bill (Activity diagram)

These activities will be performed by shopkeeper.

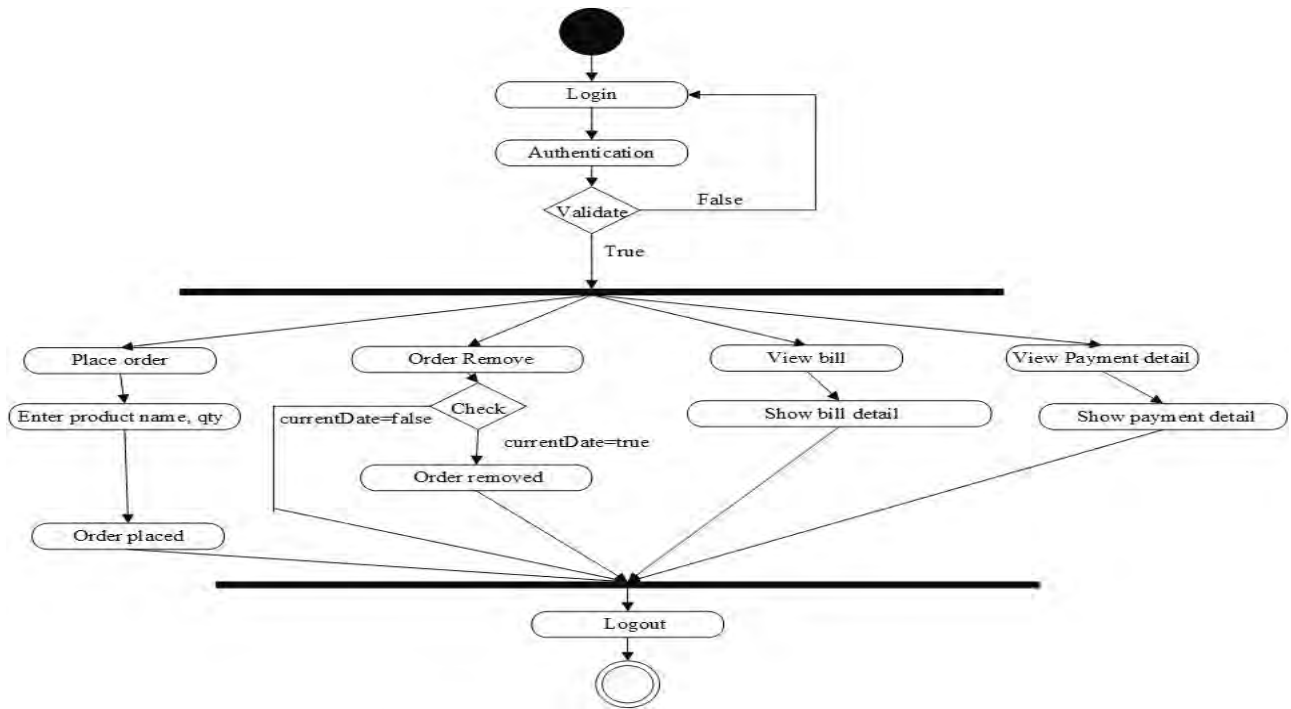


Figure 3. 22 Activity diagram (Place order, order remove, view bill and payment detail)

### 3.6.3. Add Order, remove order & Update bill (Activity diagram)

These activities will be performed by salesman.

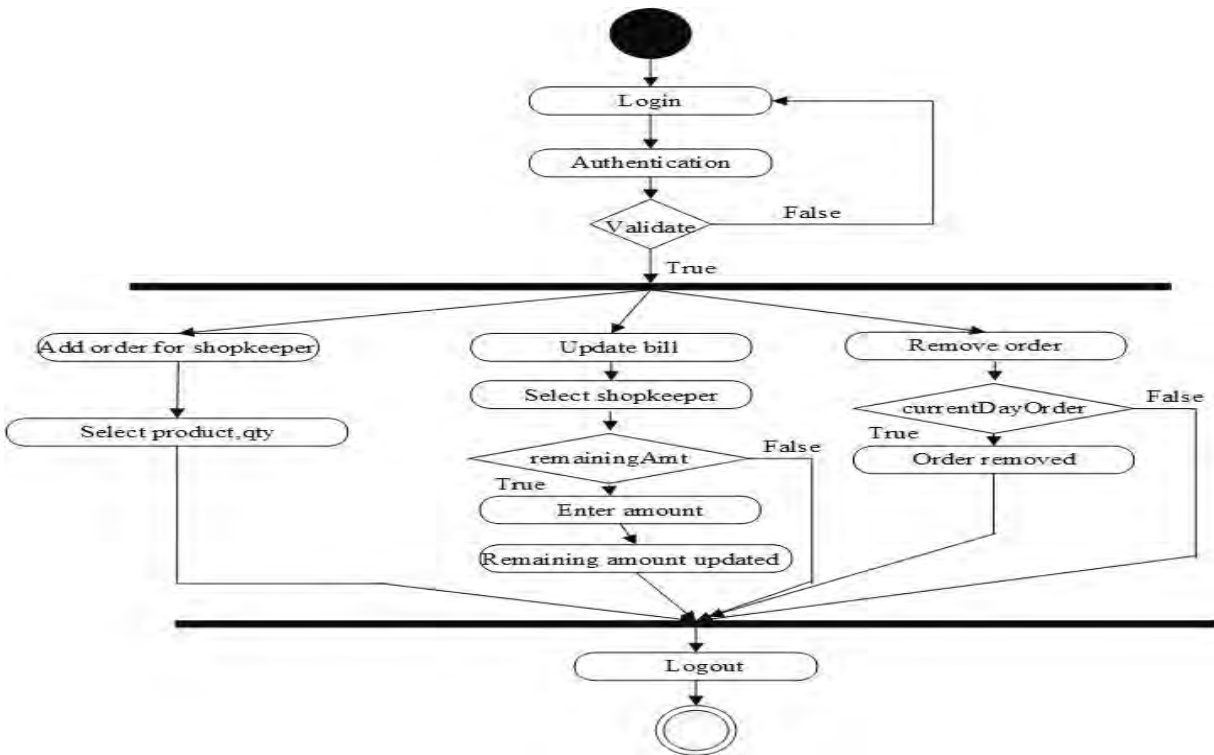


Figure 3. 23 Activity diagram (Add order, remove order & update bill)

### 3.7. Database design

MySQL database is using for storing all information for the running of web-bases e-cosmetic by Fair cosmetics. There are twelve tables in which information will be stored and retrieved for running of system.

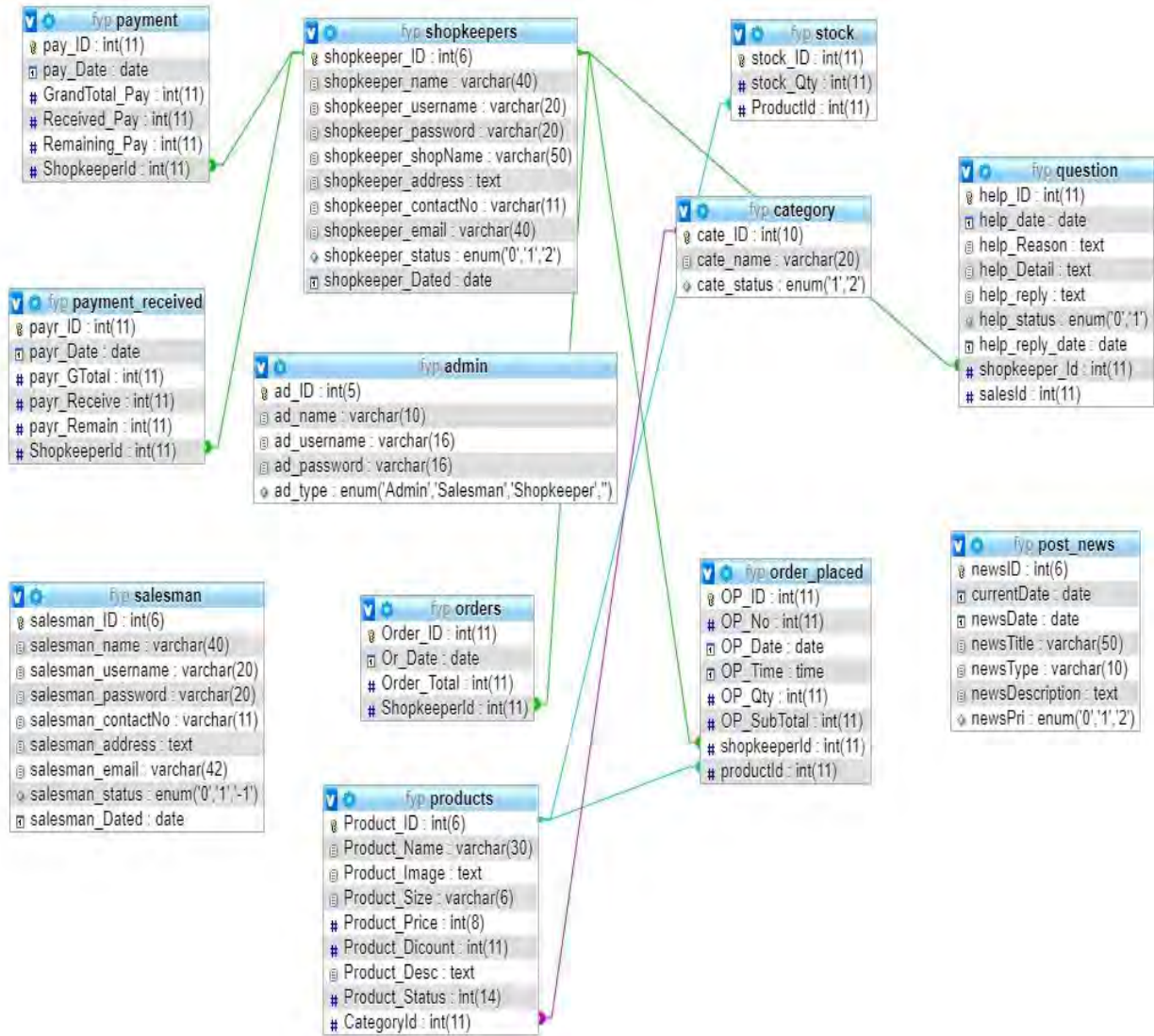


Figure 3. 24 Database design

# Chapter 4

## Software Implementation Document

This chapter describes the project implementation for “E-cosmetics for Fair cosmetics” for beauty products selling company. It has list of tools, language and used in the development of project.

### 4.1. Language Selection

This project implements in following languages:

- **HTML/CSS/Bootstrap**
  - Use for designing front-end web pages.
- **Ajax**
  - Use for the client-side browser to communicate with the server without having to perform a page refresh.
- **JQuery/JavaScript**
  - Use for scripting and validation.
- **PHP**
  - Use for server side scripting
- **MySQL**
  - Use for database.

### 4.2. Tools Selection

List of tools used for implementation of the project is given below:

- **Notepad++**
  - Use for writing code for front-end and backend languages.
- **Web browser**
  - Use for running of OPD web based application.
- **XAMPP Server**
  - Use for running of database.

### 4.3. Application Screen Shots

Application screen shots helps to understand the system, its design, input fields and output fields. These screenshots helps to understand the application views. These are the screenshots of the application e-Cosmetics by Fair cosmetics.

### 4.3.1. Screenshot for product categories

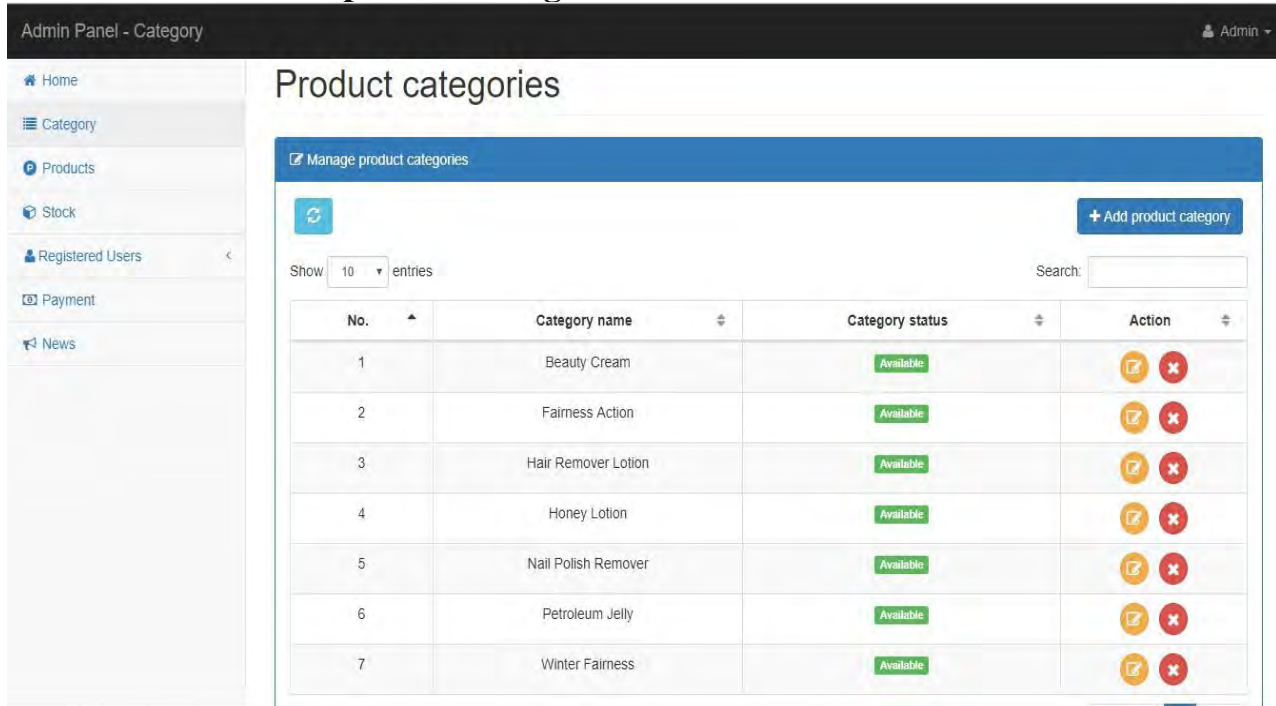


Figure 4. 1 Screenshot for product categories

### 4.3.2. Screenshot for Product Stock

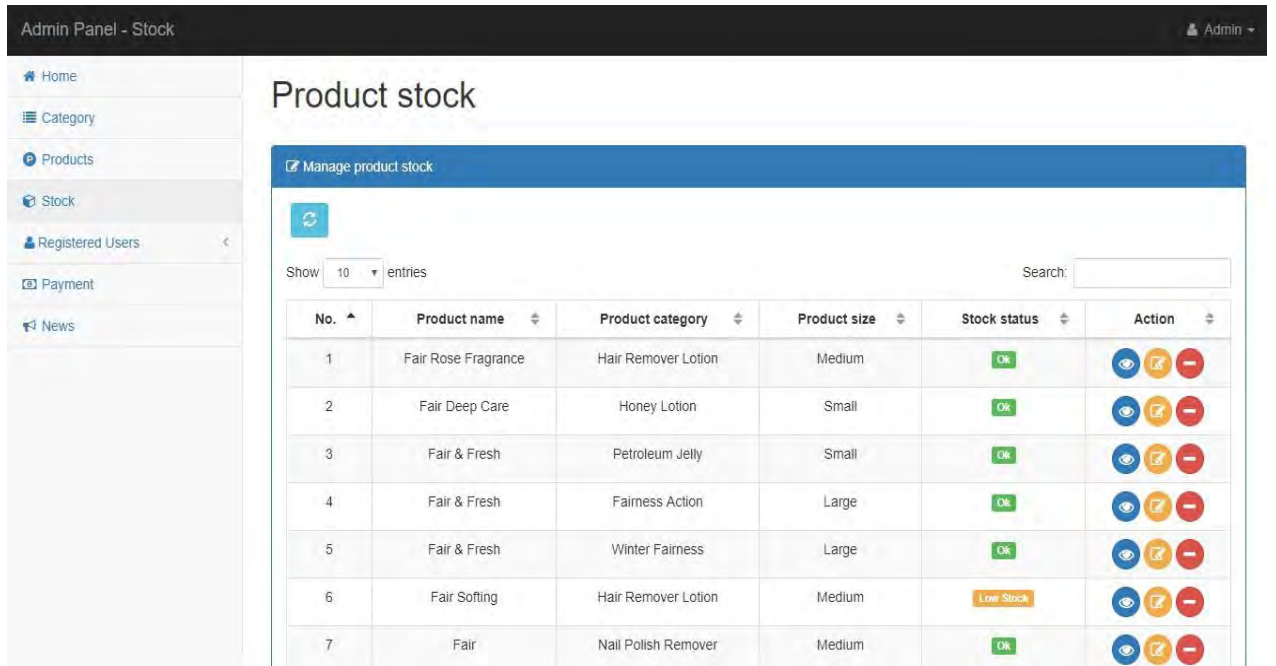


Figure 4. 2 Screenshot for product stock

### 4.3.3. Screenshot of product display for shopkeeper

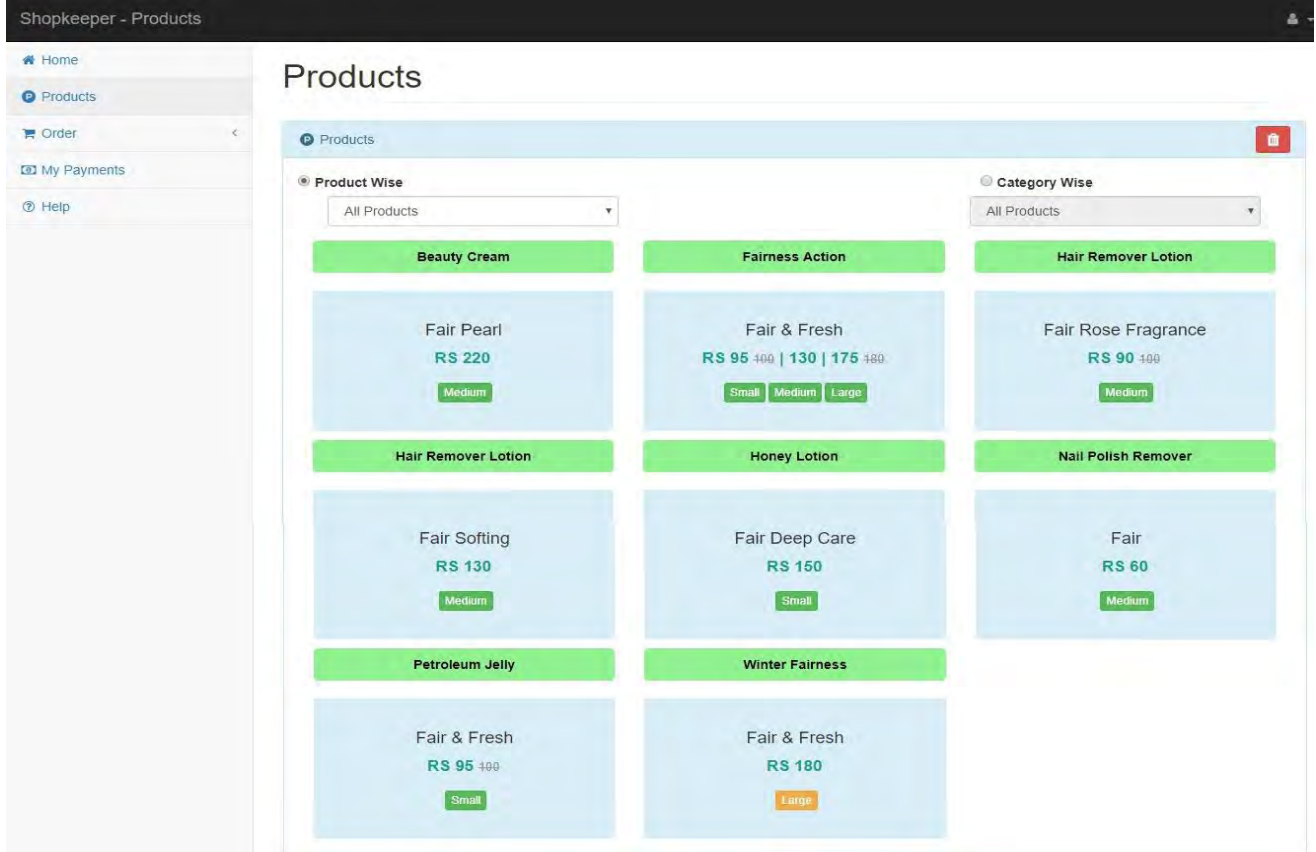


Figure 4. 3 Screenshot of product display for shopkeeper

### 4.3.4. Screenshot for list of shopkeepers

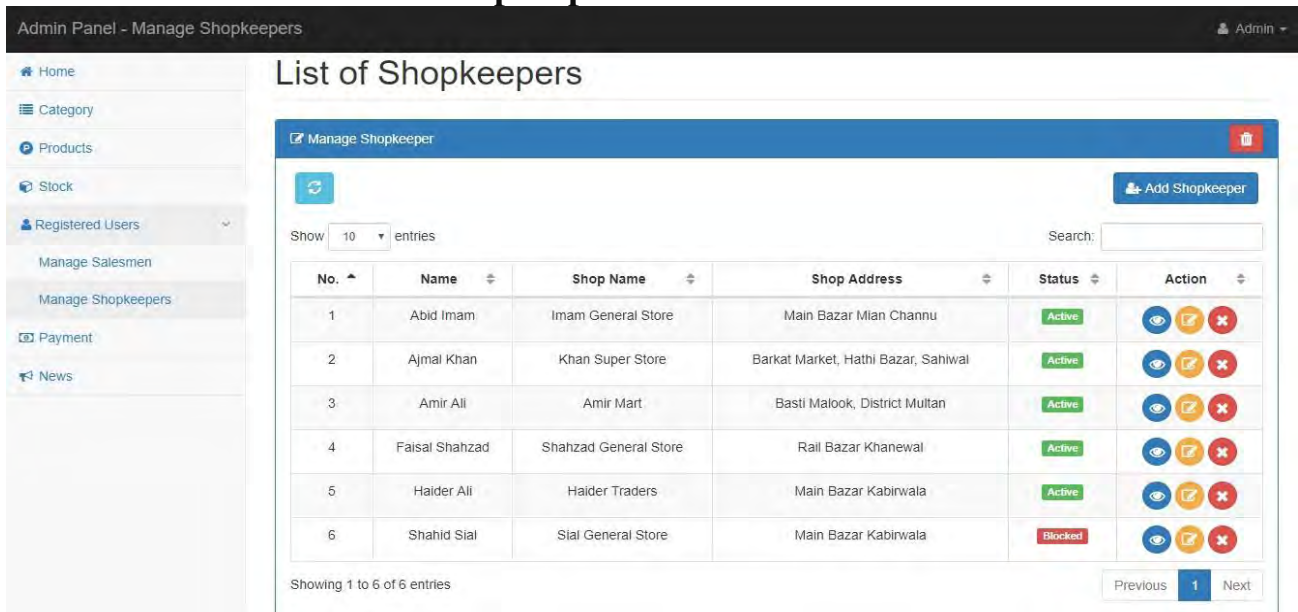


Figure 4. 4 Screenshot for post news

### 4.3.5. Screenshot for login

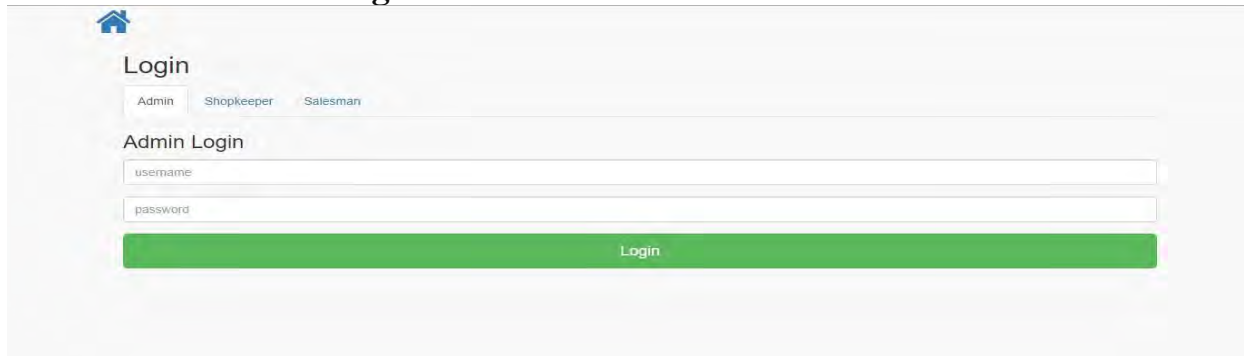


Figure 4. 5 Screenshot for login

### 4.3.6. Screenshot for payment table

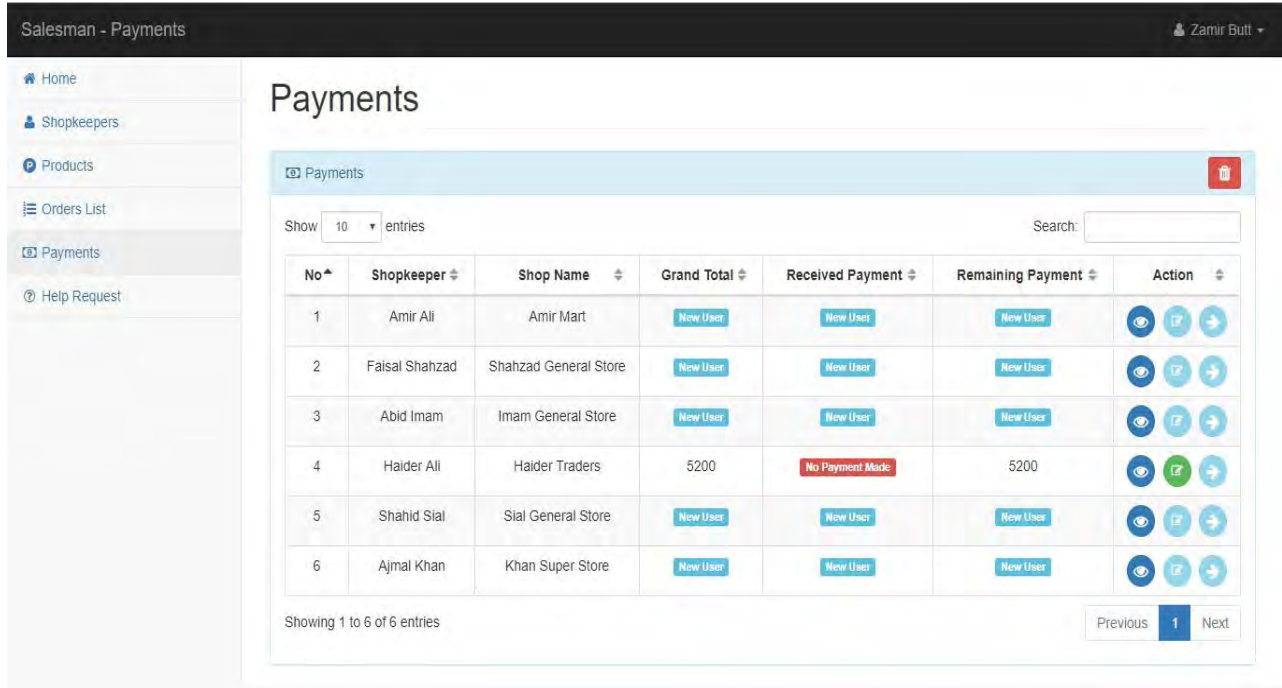


Figure 4. 6 Screenshot for payments table



# Chapter 5

## Software Test Document

This chapter describes software testing and software testing processes. This chapter further elaborates the acceptance test cases which are used to test the functional and non-functional requirements after coding of software.

### 5.1. Introduction

Software test document involves the documentation of artifacts that should be developed before or during the testing of software. Software testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

#### 5.1.1. Test approach

Manual testing includes testing a software manually without using any automated tool or any script. The tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, system testing, and user acceptance testing. Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it. Unit testing is the process of testing program components, such as methods or object and classes. Individual method or function are simplest type of the component. Test-first development is an approach to development where tests are written before the code to be tested. Small code changes are made and the code is refactored until all tests execute successfully.

### 5.2. Test plan

Test planning is an activity that ensures that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. Test plan determines the scope and the risk that need to be tested and are not to be tested. Deciding fail and pass criteria.

#### 5.2.1. Testing tools and environment

This setup consists of the physical setup which includes hardware, and logical setup that includes operating system, client operating system, front end running environment, browser (if web application), or any other software components required to run this software product.

### 5.3. Test cases

#### 5.3.1. TC-001 Login

<b>ID</b>	TC-001
<b>Description</b>	Checking either registered user can get login to system or not.
<b>Actor</b>	Admin/Registered users (shopkeeper, salesman)
<b>Setup</b>	Admin browsed admin login page.
<b>Input(s)</b>	<ol style="list-style-type: none"> <li>1. Enter correct login credentials.</li> <li>2. Enter incorrect login credentials.</li> </ol>
<b>Expected result</b>	<ol style="list-style-type: none"> <li>1. Registered user with correct credentials get logged in the system.</li> <li>2. Invalid username or password.</li> </ol>
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 1 TC-Login

#### 5.3.2. TC-002 Register shopkeeper/salesman

<b>ID</b>	TC-002
<b>Description</b>	Checking shopkeeper/salesman is registered into the system.
<b>Actor</b>	Admin
<b>Setup</b>	Admin logged in the system.
<b>Input(s)</b>	Enters required detail for: <ol style="list-style-type: none"> <li>1. Shopkeeper (e.g. Name, shop name, shop address, cell no, username, email, status) and confirm it.</li> <li>2. Salesman (e.g. Name, address, cell no, username, email, status) and confirm it.</li> </ol>
<b>Expected result</b>	In both above cases, <ul style="list-style-type: none"> <li>○ It will save registered user information if it doesn't already exist.</li> <li>○ In other case, it will not save information.</li> </ul>
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 2 TC-Register shopkeeper/salesman

#### 5.3.3. TC-003 Add product category

<b>ID</b>	TC-003
<b>Description</b>	Checking product category is added or not.
<b>Actor</b>	Admin
<b>Setup</b>	Admin is logged in the system.
<b>Input(s)</b>	<ol style="list-style-type: none"> <li>1. Admin enters product category name, select status and confirm.</li> <li>2. Admin enters product category name, doesn't select status and confirm.</li> <li>3. Admin doesn't enter product category name, select status and confirm.</li> </ol>
<b>Expected result</b>	<ol style="list-style-type: none"> <li>1. Product category will be added.</li> <li>2. Product category will not be added.</li> <li>3. Product category will not be added.</li> </ol>
<b>Actual result</b>	As expected

<b>Conclusion</b>	Pass
-------------------	------

Table 5. 3 TC-Add product category

#### 5.3.4. TC-004 Update product category

<b>ID</b>	TC-004
<b>Description</b>	Checking product category is updated or not.
<b>Actor</b>	Admin
<b>Setup</b>	Admin is logged in the system. Product category is already added in the system.
<b>Input(s)</b>	<ol style="list-style-type: none"> <li>Admin rename product category or select change status &amp; confirm it.</li> <li>Admin don't rename product category or select other status and confirm it.</li> <li>Admin leaves one of the field blanks.</li> </ol>
<b>Expected result</b>	<ol style="list-style-type: none"> <li>Product category will be updated.</li> <li>Product category will not be updated.</li> <li>Product category will not be updated.</li> </ol>
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 4 TC-Update product category

#### 5.3.5. TC-005 Remove product category

<b>ID</b>	TC-005
<b>Description</b>	Checking product category is removed or not.
<b>Actor</b>	Admin
<b>Setup</b>	Admin is logged in the system. Product category is already added in the system.
<b>Input(s)</b>	Admin select product category and confirm it to remove.
<b>Expected result</b>	Product category is removed
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 5 TC-Remove product category

#### 5.3.6. TC-006 Add stock

<b>ID</b>	TC-006
<b>Description</b>	Checking stock is added or not.
<b>Actor</b>	Admin
<b>Setup</b>	Admin is logged in the system. Product is already added in the system.
<b>Input(s)</b>	Admin select product stock, add quantity for it and confirm it.
<b>Expected result</b>	Product stock is added.
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 6 TC-Add stock

#### 5.3.7. TC-007 Place order

<b>ID</b>	TC-007
<b>Description</b>	Checking order is placed

<b>Actor</b>	Shopkeeper
<b>Setup</b>	Shopkeeper is registered in the system and not blocked.
<b>Input(s)</b>	<ol style="list-style-type: none"> <li>1. He selects products from list of products and enter correct quantity.</li> <li>2. He selects products from list of products and enter incorrect quantity.</li> <li>3. He selects no product and enter quantity.</li> </ol>
<b>Expected result</b>	<ol style="list-style-type: none"> <li>1. Order no and bill is generated and shown to him.</li> <li>2. Order will not be placed.</li> <li>3. Order will not be placed.</li> </ol>
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 7 TC-Place order

### 5.3.8. TC-008 Remove order

<b>ID</b>	TC-008
<b>Description</b>	Checking shopkeeper's order is removed or not.
<b>Actor</b>	Shopkeeper
<b>Setup</b>	Shopkeeper is logged in the system.
<b>Input(s)</b>	<ol style="list-style-type: none"> <li>1. Shopkeeper select order of current date and confirm to remove it.</li> <li>2. Shopkeeper select order of previous date and try to remove it.</li> </ol>
<b>Expected result</b>	1. Order has been remove and payment updated.
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 8 TC-Remove order

### 5.3.9. TC-009 Update bill

<b>ID</b>	TC-009
<b>Description</b>	Checking shopkeeper's bill is updated or not.
<b>Actor</b>	Salesman
<b>Setup</b>	Salesman is logged in the system.
<b>Input(s)</b>	<ol style="list-style-type: none"> <li>3. Salesman selects shopkeeper, add amount which shopkeeper paid to him and confirm it.</li> <li>4. Salesman selects shopkeeper, doesn't add amount and confirm it</li> <li>5. Shopkeeper has no remaining amount to pay.</li> </ol>
<b>Expected result</b>	<ol style="list-style-type: none"> <li>2. Bill will be updated</li> <li>3. Bill will not be updated.</li> <li>4. Over amount be consider for next bill.</li> </ol>
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 9 TC-Update bill

### 5.3.10. TC-010 Block shopkeeper

<b>ID</b>	TC-010
<b>Description</b>	Checking shopkeeper is blocked.
<b>Actor</b>	Admin
<b>Setup</b>	Shopkeeper is registered in the system and not blocked.
<b>Input(s)</b>	Select a shopkeeper and confirm to block him.

<b>Expected result</b>	Shopkeeper is to be blocked and cannot place any order.
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 10 TC-Block shopkeeper

**5.3.11. TC-011 Remove salesman**

<b>ID</b>	TC-011
<b>Description</b>	Checking salesman is removed or not.
<b>Actor</b>	Admin
<b>Setup</b>	Salesman is registered in the system.
<b>Input(s)</b>	Admin selects a salesman which he want to remove and confirm it.
<b>Expected result</b>	Salesman is removed and he no more able to perform any activity.
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 11 TC-Remove salesman

**5.3.12. TC-012 Ask for help**

<b>ID</b>	TC-012
<b>Description</b>	Checking request for any issue is submitted to admin
<b>Actor</b>	Shopkeeper
<b>Setup</b>	Shopkeeper is logged in the system.
<b>Input(s)</b>	1. Shopkeeper select issue type, enter detail and confirm it. 2. Shopkeeper leaves any field blank.
<b>Expected result</b>	1. Shopkeeper's request will be submitted. 2. Shopkeeper's request will not be submitted.
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 12 TC-Ask for help

**5.3.13. TC-013 Post news**

<b>ID</b>	TC-013
<b>Description</b>	Checking request for any issue is submitted to admin
<b>Actor</b>	Admin
<b>Setup</b>	Admin is logged in the system.
<b>Input(s)</b>	1. Admin select date, enter news detail (title, description etc.) and confirm it. 2. Admin tries to select previous dates.
<b>Expected result</b>	1. News is posted and available to users. 2. Previous date will not be selected.
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 13 TC-Post news

**5.3.14. TC-014 Logout**

<b>ID</b>	TC-014
<b>Description</b>	Checking registered users logout or not.

<b>Actor</b>	Admin/Shopkeeper/Salesman
<b>Setup</b>	Registered users is logged in the system.
<b>Input(s)</b>	1. Registered users select logout.
<b>Expected result</b>	1. Registered users get logout and no longer to do registered user activities.
<b>Actual result</b>	As expected
<b>Conclusion</b>	Pass

Table 5. 14 TC-Logout

# Chapter 6

## Conclusions and Future Enhancements

This document describes the project conclusions and future enhancements i.e. what type of new features can be added with time.

### 6.1. Conclusions

- Shopkeeper no longer to wait for the arrival of salesman to order for products.
- Shopkeepers can view their complete bill history at any place.
- Salesman feels relax for carrying lot of previous record of shopkeeper.
- Salesman can update bill of any shopkeeper.
- Administrator/Salesman can view complete bill history of any shopkeeper.
- This application maintain record of shopkeeper's sales and payment paid.

### 6.2. Future enhancement

This application is not contain any type of online payment method. In future, online payment method can be added to relax burden of visit by salesman. This application also extended to maintain stock of the company. Initially, this is not required by the company. Currently, this application doesn't has functionality of mobile phone alerts. In future, this type of can also be added. This application could be android application in future.

# References

- [1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- [2] “Datatables” [Online].  
<https://datatables.net/examples/styling/bootstrap.html>
- [3] R. S. Pressman, Software engineering: a practitioner’s approach, McGraw-Hill, 2007.
- [4] “Data flow diagram” [Online].  
<https://www.lucidchart.com/pages/data-flow-diagram?a=0>
- [5] “Entity Relationship Diagrams” [Online].  
<https://erdplus.com/#/standalone>