

# SUPERPROPOS: PRODUCT ORDERING SYSTEM

RINA BINTI ISHAK

Bachelor of Computer Science (Software  
Engineering) with Honours

UNIVERSITI MALAYSIA PAHANG

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : RINA BINTI ISHAK

Date of Birth

Title : SUPERPROPOS: PRODUCT ORDERING SYSTEM

Academic Session : 2020/2021

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Student's Signature)

RINA BINTI ISHAK

Date:02/07/2023

\_\_\_\_\_  
(Supervisor's Signature)

Azma Abdullah

\_\_\_\_\_  
Name of Supervisor

Date: 28 July 2023

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



## SUPERVISOR'S DECLARATION

I/We\* hereby declare that I/We\* have checked this thesis/project\* and in my/our\* opinion, this thesis/project\* is adequate in terms of scope and quality for the award of the degree of \*Doctor of Philosophy/ Master of Engineering/ Master of Science in .....

---

(Supervisor's Signature)

Full Name : Azma Abdullah

Position : Lecturer

Date : 28 July 2023



## STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in black ink, appearing to be 'Rina Binti Ishak', is written above a horizontal line.

(Student's Signature)

Full Name : RINA BINTI ISHAK

ID Number : CB20009

Date : 02/07/2023

SUPERPROPOS: PRODUCT ORDERING SYSTEM

RINA BINTI ISHAK

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computing  
UNIVERSITI MALAYSIA PAHANG

JULY 2023

## **ACKNOWLEDGEMENTS**

I would like to extend my deepest appreciation to the Almighty Allah for His unwavering guidance and blessings throughout the journey of developing SuperProPOS: Product Ordering System. I am truly grateful for His divine presence, which has been a constant source of strength and inspiration.

My deepest gratitude is extended to my supervisor, whose knowledge and excellent counsel were crucial in forming this project. Her advice and support were crucial in helping me to bring out my best for the completion of the project.

I also want to express my sincere gratitude to my family for their constant support and tolerance. Throughout the periods of this project, their confidence in me kept me motivated.

I'm also appreciative that Promaxx Marketing gave me the chance to work on this project. SuperProPOS: Product Ordering System was enhanced and made more pertinent to the client's demands thanks to their input and collaboration.

Lastly, I am thankful to all the individuals, friends, and lecturers who provided their support and encouragement throughout this endeavor. Their words of encouragement and belief in my abilities were a constant source of inspiration.

## ABSTRAK

Minat yang meningkat dalam e-dagang telah menjadikan internet sebagai platform utama untuk menjalankan perniagaan. Walau bagaimanapun, dalam beberapa kes, penjual menghadapi cabaran ketika mengendalikan pesanan besar daripada pelanggan. Penjual sering perlu menghubungi admin untuk mengesahkan pembayaran sebelum meneruskan pesanan kepada staf gudang, yang menyebabkan kelewatan dan menjejaskan kepercayaan pelanggan. Kesilapan dalam pembungkusan dan penghantaran produk juga berlaku disebabkan persamaan butiran pesanan. Selain itu, penjual menghadapi kesulitan untuk membuat aduan dan memastikan pembungkusan pesanan yang tepat.

Untuk menangani masalah ini, satu kajian mengenai kaedah semasa memesan produk dalam sistem yang dimodenkan telah dijalankan. Kajian ini membawa kepada pembangunan *SuperProPOS: Product Ordering System*, yang bertujuan untuk menyelaraskan dan mempercepatkan proses pesanan. Selepas menyelesaikan pembangunan sistem, ujian telah dijalankan untuk mengukur fungsionaliti sistem.

Keputusan ujian menunjukkan bahawa sistem ini membolehkan penjual meletakkan pesanan dengan lebih cekap, menghantar pesanan kepada admin untuk kelulusan, dan seterusnya memberi makluman kepada staf gudang untuk pembungkusan dan mengemaskini status pesanan. Pemasar juga dapat memantau aktiviti penjual melalui sistem, dan penjual dapat dengan mudah membuat aduan mengenai sebarang masalah pesanan. Keseluruhan, sistem ini meningkatkan kecekapan proses perniagaan dan kepuasan pelanggan.

## **ABSTRACT**

The growing interest in e-commerce has made the internet the primary platform for conducting business. However, in some cases, sellers face challenges when handling large orders from customers. Sellers often need to contact the admin to confirm payments before forwarding orders to the warehouse staff, leading to delays and impacting customer trust. Mistakes in product packaging and delivery also occur due to the similarity of order details. Additionally, sellers struggle to make complaints and ensure accurate order packing.

To address these issues, a study of the current method of ordering products in an improvised system was conducted. This study led to the development of the SuperProPOS: Product Ordering System, aimed at streamlining and expediting the order process. After completing the system development, testing was carried out to measure its functionality.

The results of the testing showed that the system allows sellers to place orders more efficiently, sending the orders to the admin for approval and subsequently notifying the warehouse staff for packing and updating the order status. Marketers can also monitor seller activities through the system, and sellers can easily lodge complaints regarding any order issues. Overall, this system enhances business process efficiency and customer satisfaction.



## TABLE OF CONTENT

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF APPENDICES</b>	<b>xiv</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>15</b>
1.1 Introduction	15
1.2 Problem Statement	16
1.3 Objectives	16
1.4 Scope	17
1.5 Thesis Organization	17
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>19</b>
2.1 Introduction	19
2.2 Current System Operation	19
2.3 Related Systems	22
2.3.1 Obsess Cosmetics	22
2.3.2 Bio-Asli	23
2.3.3 D’Herbs	25
2.4 Analysis/Comparison of Existing System	26
2.4.1 Analysis of comparison on existing system	26

2.4.2	Relevance of comparison with project title	31
2.5	Summary	31
<b>CHAPTER 3 METHODOLOGY</b>		<b>32</b>
3.1	Introduction	32
3.2	Project Methodology	32
3.2.1	Requirement Phase	33
3.2.2	Design Phase	34
3.2.3	Development Phase	34
3.2.4	Testing Phase	34
3.2.5	Deployment Phase	34
3.2.6	Maintenance Phase	35
3.3	Project Requirement	35
3.3.1	Functional requirement	35
3.3.2	Non-Functional Requirement	36
3.3.3	Constraint Requirement	36
3.3.4	Limitation Requirement	37
3.4	Propose Design	37
3.4.1	Flowchart	38
3.4.2	Context Diagram	43
3.4.3	Use Case Diagram	44
3.4.4	Use Case Description	45
3.4.5	Activity Diagram	55
3.5	Data Design	59
3.5.1	Entity Relationship Diagram	59
3.5.2	Data Dictionary	60

3.6	Proof of Initial Concept	65
3.6.1	Admin Page	65
3.6.2	Seller Page	72
3.6.3	Marketer Page	78
3.6.4	Warehouse Staff Page	84
3.7	Testing Plan	88
3.8	Potential Use of Proposed Solution	89
3.9	Gantt Chart	90
<b>CHAPTER 4 IMPLEMENTATION, RESULT AND DISCUSSION</b>		<b>91</b>
4.1	Introduction	91
4.2	Implementation Process	91
4.2.1	Developer Environment	91
4.2.2	System Functionality	98
4.3	System Testing	124
4.4	Result Discussion	125
<b>CHAPTER 5</b>		<b>130</b>
5.1	INTRODUCTION	130
5.2	System Constraint	131
5.3	Future Work	131
<b>REFERENCES</b>		<b>132</b>
<b>APPENDICES</b>		<b>0</b>

## LIST OF TABLES

Table 2.1	Comparison of Three Existing Systems	28
Table 2.2	Advantages and Disadvantages of Three Existing System	29
Table 3.1	Use Case Description for Manage Order	45
Table 3.2	Use Case Description for Manage Complaint	50
Table 3.3	Short Description of Use Case for SuperProPOS: Product Ordering System	53
Table 3.4	Table for admin	60
Table 3.5	Table for marketer	60
Table 3.6	Table for seller	60
Table 3.7	Table for warehouse	61
Table 3.8	Table for customer	61
Table 3.9	Table for product	62
Table 3.10	Table for cart	62
Table 3.11	Table for orders	63
Table 3.12	Table for complaint	64
Table 3.13	UAT of SuperProPOS: Product Ordering System	88

## LIST OF FIGURES

Figure 2.1	Bizzapp Website Main Page	20
Figure 2.2	WhatsApp Chat of Order and Payment Approval	20
Figure 2.3	Example of Printed Successful Order Information	21
Figure 2.4	Whatsapp Chat Group for Order Complaint	21
Figure 2.5	Obsess Cosmetics Registration Form	22
Figure 2.6	Obsess Cosmetics Shipping Form of Ordered Product	23
Figure 2.7	Bio-Asli Website Main Page	23
Figure 2.8	Bio-Asli Shipping Form and Total Price of Ordered Product	24
Figure 2.9	Bio-Asli Payment Form	25
Figure 2.10	D’Herbs Website Main Page	25
Figure 2.11	D’Herbs Total Payment and Shipping Form	26
Figure 3.1	Waterfall Model Methodology	33
Figure 3.2	Flowchart of Register Employee Module	38
Figure 3.3	Flowchart of Module Login	39
Figure 3.4	Flowchart of Manage Product Detail Module.	40
Figure 3.5	Flowchart of Manage Order Module	41
Figure 3.6	Flowchart of Manage Complaint Module.	42
Figure 3.7	Context Diagram of SuperProPOS: Product Ordering System	43
Figure 3.8	Use Case Diagram of SuperProPOS: Product Ordering System	44
Figure 3.9	Activity Diagram for Login	55
Figure 3.10	Activity Diagram for Register Employee.	55
Figure 3.11	Activity Diagram for Manage User Profile	56
Figure 3.12	Activity Diagram for Manage Product	56
Figure 3.13	Activity Diagram for Manage Complaint	57
Figure 3.14	Activity Diagram for Manage Order	58
Figure 3.15	ERD of SuperProPOS: Product Ordering System	59
Figure 3.16	SuperProPOS: Product Ordering System Main Page	65
Figure 3.17	UI of Admin Login Page	65
Figure 3.18	UI Homepage for Admin	66
Figure 3.19	UI Register Seller by Admin	66
Figure 3.20	UI Register Marketer by Admin	66
Figure 3.21	UI Register Warehouse Staff by Admin	67
Figure 3.22	UI Seller List for Admin	67

Figure 3.23	UI Marketer List for Admin	67
Figure 3.24	UI Warehouse staff List for Admin	68
Figure 3.25	UI Update Admin Detail for Admin	68
Figure 3.26	UI Product List for Admin	68
Figure 3.27	UI Add New Product for Admin	69
Figure 3.28	UI Update Product for Admin	69
Figure 3.29	UI Order List for Admin	69
Figure 3.30	UI Update Order Status for Admin	70
Figure 3.31	UI Complaint List for Admin	70
Figure 3.32	UI Update Complaint Status for Admin	70
Figure 3.33	UI List Sales Report for Admin	71
Figure 3.34	UI Graph Sales Report for Admin	71
Figure 3.35	UI Login for Seller	72
Figure 3.36	UI Homepage for Seller	72
Figure 3.37	UI Product List for Seller	73
Figure 3.38	UI Product Detail for Seller	73
Figure 3.39	UI Shop for Seller	73
Figure 3.40	UI Cart for Seller	74
Figure 3.41	Check Out Form Order for Seller	74
Figure 3.42	UI Successful Order Message for Seller	74
Figure 3.43	UI Order List for Seller	74
Figure 3.44	UI Add New Customer for Seller	75
Figure 3.45	UI Customer List for Seller	75
Figure 3.46	UI Edit Customer Detail for Seller	75
Figure 3.47	UI Complaint List for Seller	76
Figure 3.48	UI Add New Complaint for Seller	76
Figure 3.49	Edit Complaint for Seller	76
Figure 3.50	View Complaint for Seller	76
Figure 3.51	UI Update Seller Profile for Seller	77
Figure 3.52	UI List Sales Report for Seller	77
Figure 3.53	UI Graph Sales Report for Seller	77
Figure 3.54	UI Login for Marketer	78
Figure 3.55	UI Homepage for Marketer	78
Figure 3.56	UI Product List for Marketer	79
Figure 3.57	UI Product Detail for Marketer	79

Figure 3.58	UI Order List for Marketer	79
Figure 3.59	View Order Detail for Marketer	80
Figure 3.60	UI Customer List for Marketer	80
Figure 3.61	UI Customer Detail for Marketer	80
Figure 3.62	UI Complaint List for Marketer	81
Figure 3.63	View Complaint Detail for Marketer	81
Figure 3.64	UI Seller List for Marketer	82
Figure 3.65	List Order of Seller for Marketer	82
Figure 3.66	UI Update Marketer Profile for Marketer	82
Figure 3.67	UI List of Seller Sales Report for Marketer	83
Figure 3.68	UI List Sales Report for Marketer	83
Figure 3.69	UI Graph Sales Report for Marketer	83
Figure 3.70	UI Login for Warehouse Staff	84
Figure 3.71	UI Homepage for Warehouse staff	84
Figure 3.72	UI Product List for Warehouse staff	85
Figure 3.73	UI Update Product for Warehouse staff	85
Figure 3.74	UI Order List for Warehouse staff	86
Figure 3.75	UI Order Detail for Warehouse staff	86
Figure 3.76	UI Complaint List for Warehouse staff	86
Figure 3.77	UI Complaint Detail for Warehouse staff	87
Figure 3.78	UI Update Warehouse staff Profile for Warehouse staff	87
Figure 4.1	Folder of superpropos Inside xampp/htdocs (Localhost)	92
Figure 4.2	Details Folder Inside of superpropos Folder	92
Figure 4.3	Details of ApplicationLayer Folder	93
Figure 4.4	Details of ManageAdmin Folder	93
Figure 4.5	Details of ManageMarketer Folder	94
Figure 4.6	Details of ManageSeller Folder	94
Figure 4.7	Details of ManageWarehouse Folder	95
Figure 4.8	Details of BusinessServiceLayer Folder	95
Figure 4.9	Details of Controller Folder	96
Figure 4.10	Details of Model Folder	96
Figure 4.11	Details of Library Folder	96
Figure 4.12	Database of SuperProPOS: Product Ordering System	97
Figure 4.13	Login Page for Admin	98
Figure 4.14	Login Page for Admin	98

Figure 4.15	Login Page for Marketer	99
Figure 4.16	Login Page for Warehouse staff	99
Figure 4.17	List of Seller for Admin	100
Figure 4.18	Register New Seller Page for Admin	100
Figure 4.19	Register New Marketer Page for Admin	100
Figure 4.20	Register New Warehouse staff Page for Admin	101
Figure 4.21	Update Profile Page for Admin	109
Figure 4.22	Update Profile Page for Seller	109
Figure 4.23	Update Profile Page for Marketer	110
Figure 4.24	Update Profile Page for Warehouse staff	110
Figure 4.25	Add New Product Page for Admin.	111
Figure 4.26	Update Product Page for Warehouse staff	111
Figure 4.27	List Product Page for Seller	112
Figure 4.28	List of Product Page for Marketer	112
Figure 4.29	Product Detail Page for Seller	112
Figure 4.30	List Customer for Seller	113
Figure 4.31	Add New Customer Page for Seller	113
Figure 4.32	Update Customer Page for Seller	113
Figure 4.33	Add Order Page for Seller	117
Figure 4.34	Add to Cart Pages for Seller.	117
Figure 4.35	Check Out Order Form Page for Seller.	118
Figure 4.36	Update Order Payment Status Page for Admin.	118
Figure 4.37	Update Delivery Status Page for Warehouse staff.	119
Figure 4.38	Add Complaint Form Page for Seller.	119
Figure 4.39	Update Complaint Status Page for Admin.	120
Figure 4.40	Update Complaint Status Page for Warehouse staff.	120
Figure 4.41	List of Seller Complaint Page for Marketer	120
Figure 4.42	Daily Sales Report Page for Admin.	121
Figure 4.43	Monthly Sales Report for Seller.	122
Figure 4.44	List of Marketer's Seller Sales Page for Marketer.	122
Figure 4.45	Marketer's Seller Daily Sales Report Page for Marketer.	123
Figure 4.46	Register New Seller Page for Admin	125
Figure 4.47	List of Seller Page for Admin	125
Figure 4.48	Add New Product Page for Admin	126
Figure 4.49	Cart Pages for Seller	126



Figure 4.50	Check Out Order Form Page for Seller	126
Figure 4.51	List Order Page for Admin	126
Figure 4.52	List Order Page for Warehouse staff	127
Figure 4.53	Add Complaint Form Page for Seller	127
Figure 4.54	List Complaint Page for Warehouse staff	127
Figure 4.55	List Seller's Sales Report Page for Marketer	128
Figure 4.56	Daily Report Sales Page for Admin	128
Figure 4.57	Daily Report Sales Graph Page for Admin	128

## LIST OF APPENDICES

Appendix A: Software Requirement Specification (SRS) of SuperProPOS: Product Ordering System	1
Appendix B: Software Design Document	30
Appendix C: User Acceptance Testing (UAT)	44
Appendix D: Gantt Chart	63

## CHAPTER 1

### INTRODUCTION

#### 1.1 Introduction

Nowadays a lot of people are interested in doing business, especially the e-commerce business. This is because the internet has become the main platform of people conducting their business in this era.

In business, there are some cases where the customers ordered hundreds of products from seller. The seller needs to ask for confirmation of the payment from the admin by call or message to proceed their order to the warehouse staff. Then, the warehouse staff process and update the status of the orders to the seller.

From this scenario, it shows that there are some disadvantages facing by the seller to place the order from the customers. It may lead to late reply from the admin that can cause the seller having difficulties to get the trust from the customers again. Next, the warehouse staff may make some mistake in packing the product that should be sent to right customers because of the similarities of customers details. The customers may feel dissatisfied with the wrong product that have been received by them.

By developing the SuperProPOS: Product Ordering System, it helps the seller to make an order for customer in an easier and faster way. The system can send the order to admin for the approval. Once the admin approves the order, the warehouse staff receive the order and start packing the request. After that, the warehouse staff can update the status of the order to the seller and customer in the system. The marketer who is in charge monitoring the seller, can view the activities of the seller from the system. Complaint also can be made for the order by seller to inform the warehouse staff about the problem of the order.

## **1.2 Problem Statement**

The first problem that a business company faces is with their way of submitting and getting an approval for the order payment. This is because the seller must get approval of the order payment from admin via call or message. The admin may take a slow time in approving the payment or overlooked the message of the payment sent by seller. Next, the warehouse staff also faces a problem by mistaking the order details. This is because the warehouse staff normally confuses with the naming and the quantity of the product that need to be packed from the receipt. Lastly, the seller faces a hard time to make a complaint of their order. This is because the seller has to make a complaint via message in a group chat and remind the warehouse staff of their order frequently, in case the warehouse staff might overlooked the complaint message from other message in the group chat.

To solve these problems, the SuperProPOS: Product Ordering System can help the seller to make an order in a faster way and the admin can approve the order payment through the system. Next, SuperProPOS: Product Ordering System can help the warehouse staff by providing an accurate detail of the product including the image of product that should be pack from the system. Through the product images, the warehouse staff is more aware with the product that need to be packed. Lastly, the SuperProPOS: Product Ordering System can help the seller making a complaint easily. This is because the seller can see their complaint status and description of the complaint in the system. Thus, by developing SuperProPOS: Product Ordering System can help to solve these problems.

## **1.3 Objectives**

- i. To study the current method of ordering product in an improvised system.
- ii. To develop a prototype of SuperProPOS: Product Ordering System in web based.

- iii. To evaluate the proposed prototype of the SuperProPOS: Product Ordering System in web based.

#### **1.4 Scope**

The SuperProPOS: Product Ordering System involved with 4 types of users, Seller, Marketer, Admin and Warehouse staff. Marketer is allowed to monitor the sellers' activities. Seller is allowed to make an order in the SuperProPOS: Product Ordering System for the customers. Admin then approve the payment of the products to the warehouse staff. Warehouse staff received the order and update the order status to the seller.

The function that available in the SuperProPOS: Product Ordering System is the seller, marketer, admin, and warehouse staff can access to the system by login with their username and password. Admin is responsible in registering Marketer, Seller and Warehouse staff to the system. Marketer is assigning to each seller. Next, the marketer, seller, admin, and warehouse staff can manage their profile. Then, admin and warehouse staff can manage the product. Later, the marketer, seller, admin and warehouse staff able to manage the order for customer. Lastly, the marketer, seller and warehouse staff are able to manage complaint for the product ordered.

The platforms that are being used for developing the SuperProPOS: Product Ordering System are using PHP, CSS and JavaScript for the framework. MySQL and XAMPP are being used as the database platform. The local host is being used to run the system.

#### **1.5 Thesis Organization**

Chapter 1 discusses the introduction of the SuperProPOS: Product Ordering System project.

Chapter 2 discusses the comparison of existing systems related to the SuperProPOS: Product Ordering System. These existing systems could be improved and implemented in the SuperProPOS: Product Ordering System.

Chapter 3 discusses the methodology suitable for the SuperProPOS: Product Ordering System. It covers the system requirements and the flow of the entire process.

Chapter 4 discusses the implementation, results, and discussion of the SuperProPOS: Product Ordering System. This chapter explains the development environment, functionality, system testing, and the discussion of results.

Chapter 5 discusses the conclusion of the SuperProPOS: Product Ordering System project. It explains the system constraints and future work that may be related to the system in the future.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

This chapter discuss about the literature review that have been studied for the SuperProPOS: Product Ordering System. It discusses about chapter 2.2 called *Current System Operation* and chapter 2.3 *Related Systems*. The chapter 2.3 consist of sub-chapter 2.3.1 *Obsess Cosmetics*, sub-chapter 2.3.2 *Bio-Asli* and sub-chapter 2.3.3 *D'Herbs*. Chapter 2.4 called *Analysis/Comparison of Existing System* is discussing about the sub-chapter 2.3.1 *Analysis of comparison on existing system* and sub-chapter 2.3.3 called *Relevance of comparison with the project title*. Lastly, it discusses about the summary of chapter 2 in chapter 2.4 called *Summary*.

#### 2.2 Current System Operation

The current system operation of the business company is, the company have to make an order from a third-party website called *Bizapp*. The customer needs to chat the seller in *WhatsApp* to buy the products. Then, the seller need make the order for the customer through the *Bizapp* website and send the proof of payment through *WhatsApp* like image of receipt to the admin. The admin verifies and updates the status of the order in the *Bizapp* and the warehouse staff then process the sellers' orders. If any complaint is made by the customer about their order, the seller have to make a complaint in a WhatsApp group of the company to inform about the status of the customers' orders. Figure 2.1 shows the current third-party system called *Bizapp* used by the company.

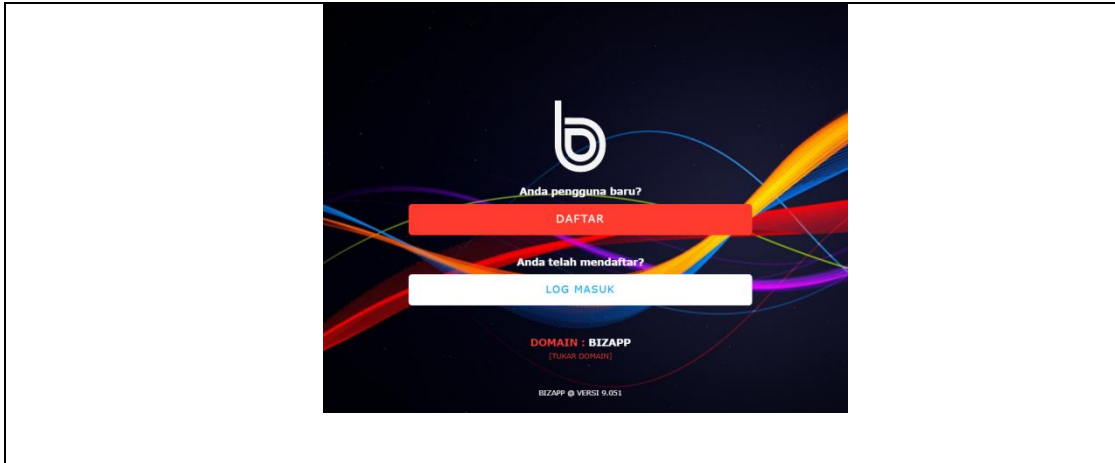


Figure 2.1 Bizzapp Website Main Page

Seller sends the customer's order detail and image of receipt payment to the admin. Admin of the company reply with an emoji thumbs up representing that the order and payment is successful. Figure 2.2 shows the chat between admin and seller to approve the order and payment of customer.

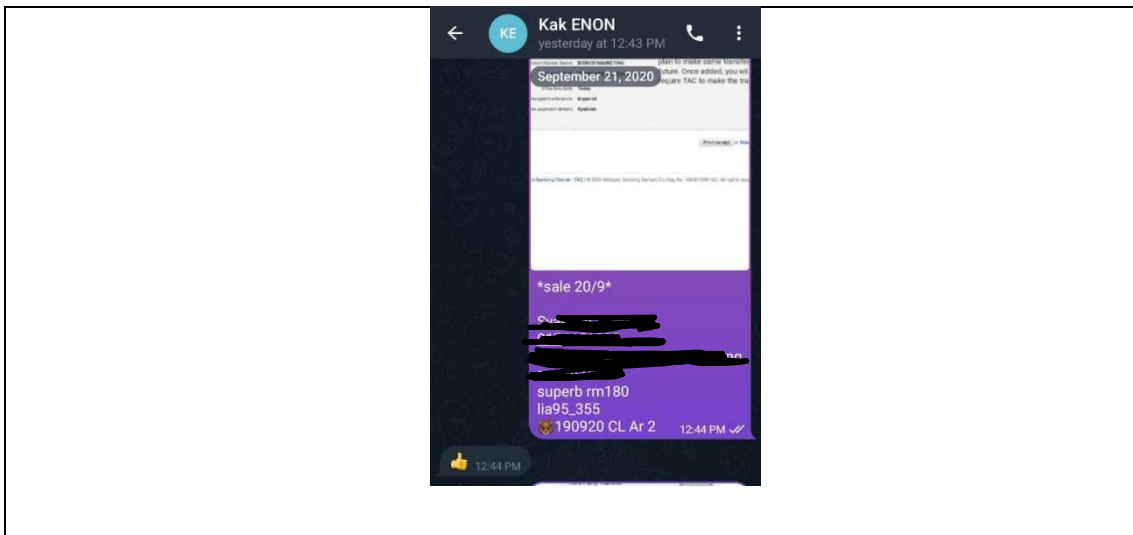


Figure 2.2 WhatsApp Chat of Order and Payment Approval



Then, the warehouse staff prints out the successful order to packing it according to the ordered product information given. Figure 2.3 shows the example of printed successful order information.



Figure 2.3 Example of Printed Successful Order Information

The complaint of the ordered product needs to be made by the seller in a group *WhatsApp* called “Customer Report” to inform the warehouse staff about the problem with the order. Figure 2.4 shows the example of the complaint sent by the seller to the warehouse staff in the group chat.

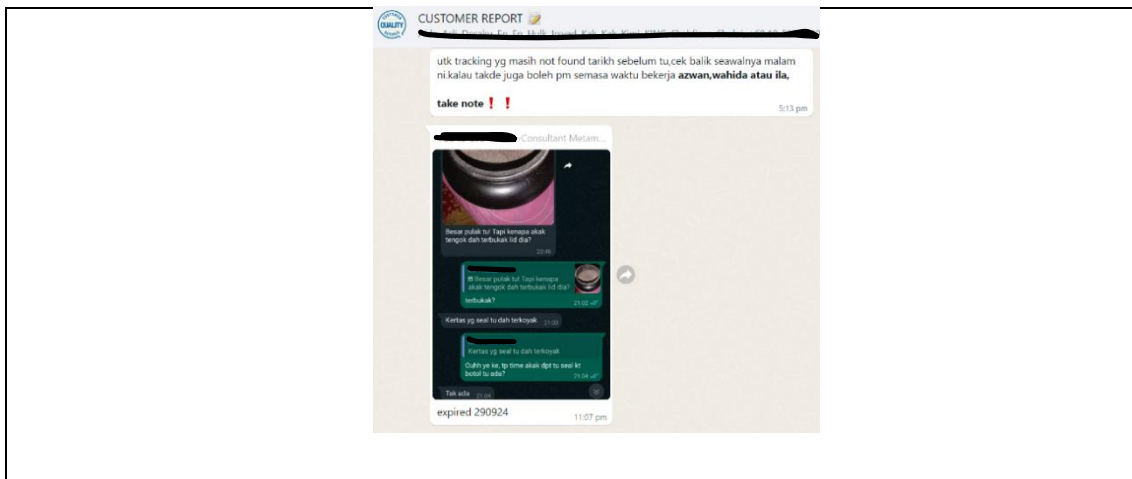


Figure 2.4 Whatsapp Chat Group for Order Complaint

## 2.3 Related Systems

The related systems for this sub-chapter 2.3 are *Obsess Cosmetics*, *Bio-Asli* and *D’Herbs*.

### 2.3.1 Obsess Cosmetics

*Obsess Cosmetics* is an e-commerce company that used website as their platform to sell their product like cosmetics and skin care for woman. This website is a web-based system that allow customer to buy product from the company easily. The customer must sign-in to the website or reach their agent to buy the product from the company. To register to the website, the customer must get the registration link from their leader or upline. Figure 2.5 shows the interface of the registration of the system.

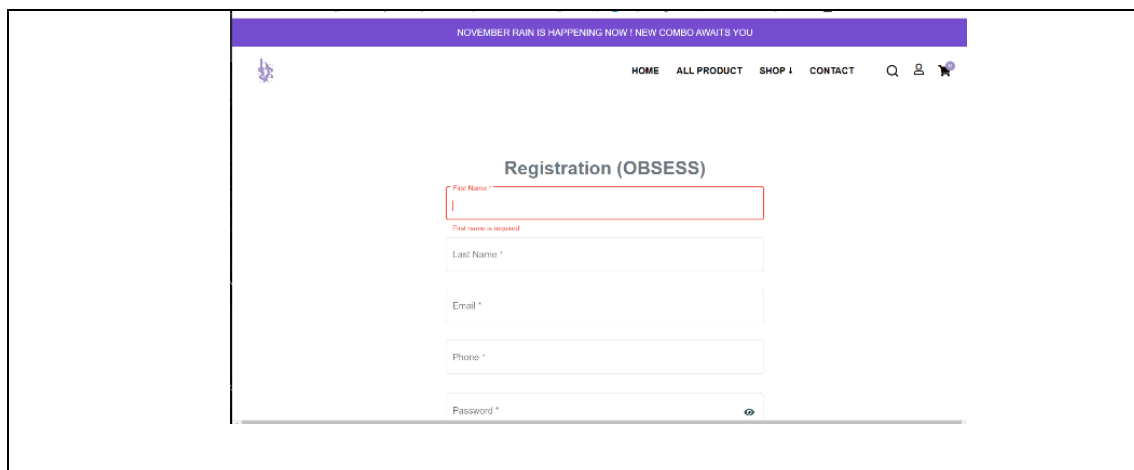


Figure 2.5 Obsess Cosmetics Registration Form

*Obsess Cosmetics* website allows the customers to choose and buy the products easily. The sign-in customers must choose the products and fill in the shipment form to allow them to buy the products online. The customer can choose two option of shipping which is “Shipping by Courier” or “Self-Pickup”. The payment of the products can be made through online banking. Figure 2.6 shows the shipment information of the ordered product.

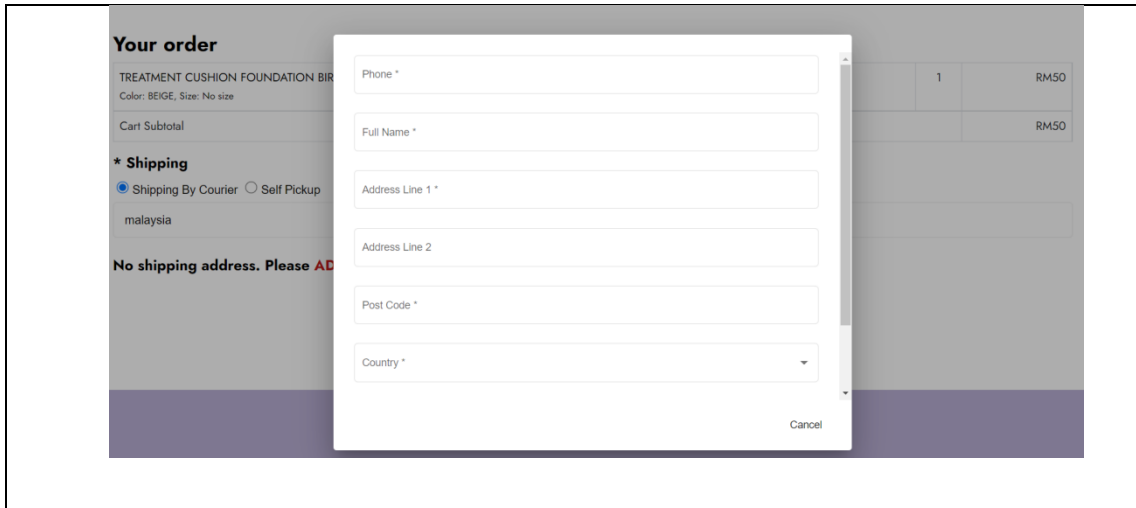


Figure 2.6 Obsess Cosmetics Shipping Form of Ordered Product

### 2.3.2 Bio-Asli

*Bio-Asli* is a trading company that used website as their main platform to do their business. This company sell health products for all types of people. This company used web-based system that allow customer to buy product from the company easily. Figure 2.7 shows the main page of *Bio-Asli* website.

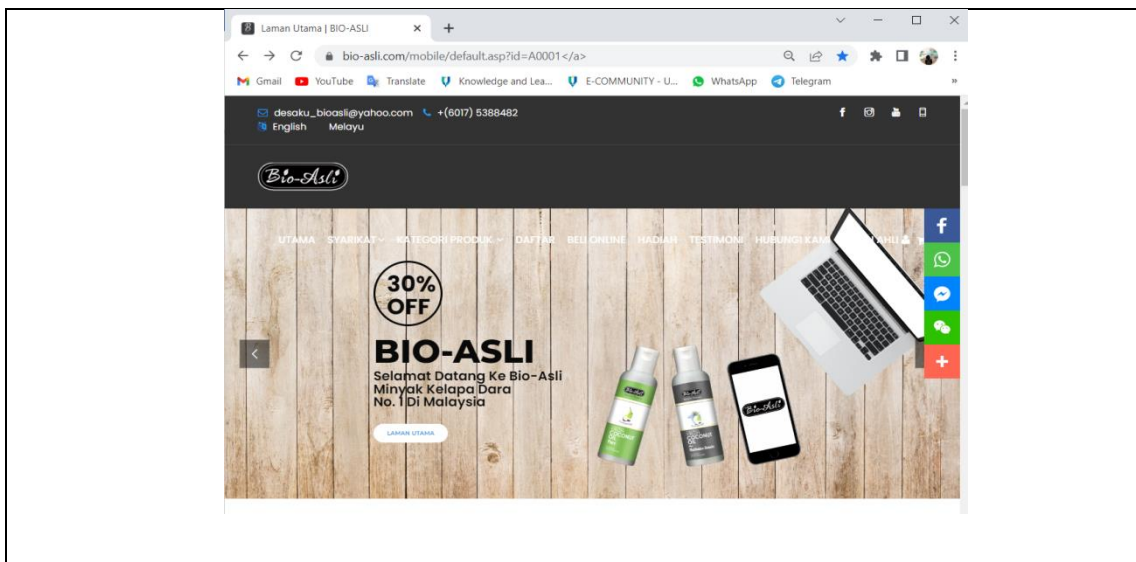


Figure 2.7 Bio-Asli Website Main Page

*Bio-Asli* customer is allowed to sign-in or not sign-in to order their product from the websites. Sign-in customer can get more discount based on their order quantity on the website. The staff also can make an order for the customer by using the customer ID to help the novice users especially senior citizen. The customer or staff need to fill the shipment information to allow the product to be ordered. Customer or staff can choose three methods of payment for the products which is “Perbankan Internet”, “Kaunter Bank” or “Mesin ATM/CDM”. Once the ordered product is successful, the customer needs to fill the payment form as a proof of successful payment by the customer. Figure 2.8 shows the total price and shipping form of customer. While figure 2.9 shows the payment form that need to be fill by the customers.

The screenshot displays a web form for shipping information and a summary of the order. The form is titled "Maklumat Penghantaran Anda" and includes the following fields:

- Nama Anda \* (Rina test)
- No. H/P Anda \* (1231313)
- Alamat E-Mail \* (1212131)
- Kaedah Pembayaran \* (Perbankan Internet)

The summary section shows:

- Sub-Jumlah: RM 40.00
- Lokasi: MALAYSIA
- Kos Pos: RM 0.00
- Berat: 200 gram
- JUMLAH BESAR: RM 40.00**

The page also features a navigation menu at the top with links for UTAMA, KATEGORI PRODUK, TESTIMONI, PEMBAYARAN, PENGHANTARAN, HUBUNGI KAMI, and LOGIN AHLI. Social media icons for Facebook, WhatsApp, Telegram, and Messenger are visible on the right side.

Figure 2.8 Bio-Asli Shipping Form and Total Price of Ordered Product

Figure 2.9 Bio-Asli Payment Form

### 2.3.3 D'Herbs

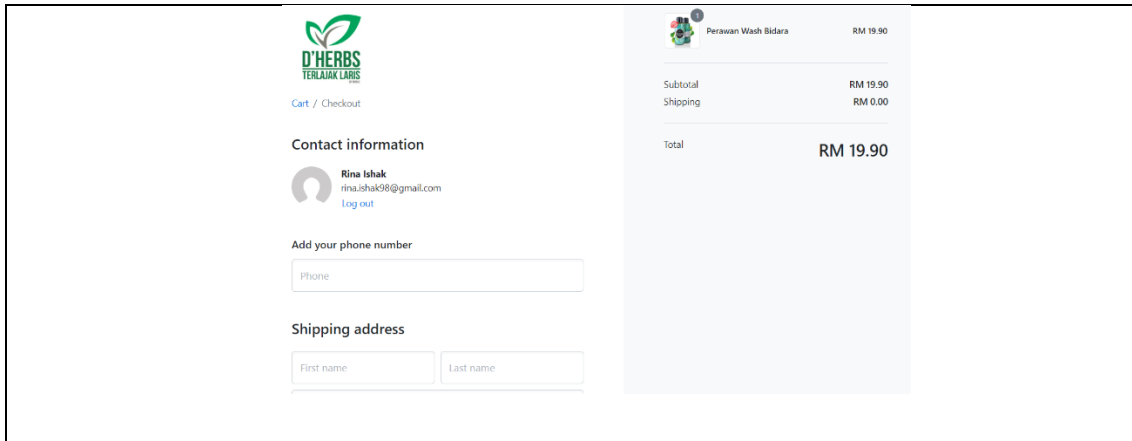
*D'Herbs* is a company that sell health product that can be used by all generation. This company used web-based system that allow customer to buy product from the company easily. Figure 2.2.3.1 shows the main page of *D'Herbs* websites.



Figure 2.10 D'Herbs Website Main Page

*D'Herbs*'s customers need to go to the "Catalog" menu to order the product. All the option of the product is displayed including the picture, name, and price. Once the products were chosen, the customer need to fill their contact information for shipment of

product and proceed with the payment of product. The ordered is successful once the payment is successful. Figure 2.11 shows the total payment and shipping information form of customer.



The screenshot displays the checkout interface for D'Herbs. On the left, the logo 'D'HERBS TERBUKTI LAGI' is visible. Below it, the user is logged in as 'Rina Ishak' with the email 'rina.ishak98@gmail.com'. There is a 'Log out' link. A section for 'Add your phone number' contains a 'Phone' input field. Below that, the 'Shipping address' section has 'First name' and 'Last name' input fields. On the right, a summary table shows the order details:

Perawan Wash Bidara	RM 19.90
Subtotal	RM 19.90
Shipping	RM 0.00
<b>Total</b>	<b>RM 19.90</b>

Figure 2.11 D'Herbs Total Payment and Shipping Form

## 2.4 Analysis/Comparison of Existing System

### 2.4.1 Analysis of comparison on existing system

Based on the Existing Systems 2.2, Table 2.3.1 shows the comparison of the three existing product ordering websites. The features that are being compared includes Graphical User Interface (GUI), functions, advantages, and disadvantages of the websites.

Through the three compared websites, those websites have its own advantage and disadvantages. *Obsess Cosmetics* have a very interesting GUI. The combination of the template colour match with their brand and suitable to attract their target market compared to *Bio-Asli* and *D'herbs*. *Obsess Cosmetics* also provide two options of shipping for the customer which is "Shipping by Courier" or "Self -Pickup" compared to *Bio-Asli* and *D'Herbs* that does not have this option in their websites.

*Bio-Asli* advantages are the websites is full of information. It is suitable for all generation to get through the website compared to *Obsess Cosmetics* and *D'herbs*. *Bio-Asli* also

provides a lot of payment option for customers who want to pay by using internet banking, bank counter or ATM/CDM machine. This can ease the customers like senior citizen to make payment through the ATM/CDM option compared *Obsess Cosmetics* and *D'Herbs* that only provide online transfer for their customers. The proof of payment needs to be fill in the payment form for the customer who choose to make payment through bank counter or ATM/CDM machine.

As for *D'Herbs*, it has the option for billing address that allow the customer to choose between using the same with shipping address or use a different billing address compared to *Obsess Cosmetics* and *Bio-Asli* that does not have this option. This eases the customer to avoid repeating the same process of filling the shipping and billing form twice.

Table 2.1 Comparison of Three Existing Systems

Website Name	<i>Obsess Cosmetics</i>	<i>Bio-Asli</i>	<i>D'Herbs</i>
Features of Graphical User Interface (GUI).	Interesting and attractive colour combination to attract target customer.	Too crowded and not suitable for all ages customers.	Simple and minimalist with one theme colour.
Functions	<ol style="list-style-type: none"> <li>1. Display all product information.</li> <li>2. Allow customer to order the product by sign-in to the website.</li> <li>3. Allow customer to update their profile.</li> </ol>	<ol style="list-style-type: none"> <li>1. Display all product information.</li> <li>2. Allow customer to order product without sign-in or sign-in to the website.</li> <li>3. Allow customer to update their profile.</li> </ol>	<ol style="list-style-type: none"> <li>1. Display all product information.</li> <li>2. Allow customer to order product without sign-in or sign-in to the website.</li> <li>3. Allow customer to update their profile.</li> </ol>



Table 2.2

Advantages and Disadvantages of Three Existing System

Website Name	<i>Obsess Cosmetics</i>	<i>Bio-Asli</i>	<i>D'Herbs</i>	POS: SuperProPOS: Product Ordering System
Advantages	<ol style="list-style-type: none"> <li>1. Have nice GUI.</li> <li>2. Allow customer to make order easily.</li> <li>3. Provide two options of shipping for the customer which is Shipping by Courier of Self Pick-up.</li> </ol>	<ol style="list-style-type: none"> <li>1. Allow customer to make order easily.</li> <li>2. Provide a lot of payment option for customer.</li> </ol>	<ol style="list-style-type: none"> <li>1. Allow customer to make order easily.</li> <li>2. Allow customers to choose option for billing address.</li> </ol>	<ol style="list-style-type: none"> <li>1. Have nice GUI</li> <li>2. Allow customer to make order easily.</li> <li>3. Allow customer to make order easily.</li> <li>4. Allow customers to choose option for billing address.</li> <li>5. Provide two options of shipping for the customer which</li> </ol>

				is Shipping by Courier of Self Pick-up.
Disadvantages	The customer needs to ask leader/upline to get the registration link for the website.	The cart pages are confusing for customer to proceed with shopping or proceed to place order.	The customer only has one option to make payment which is online banking only.	Customers need to order the product through seller only.

### **2.4.2 Relevance of comparison with project title**

In this project, SuperProPOS: Product Ordering System is relevance with these three existing websites. This is because of all websites have manage order function and it match with the project title which is SuperProPOS: Product Ordering System. The SuperProPOS: Product Ordering System also have registration for admin, seller, marketer, and warehouse staff to allow them to manage order the product. The system allows the seller to make order for the customer by choosing the product and fill-in the shipment form to proceed with ordering the product from the system. The system also allows the seller to choose a few payments option like online banking, bank counter or ATM/CDM payment. The seller needs to send the proof of payment like receipt image to the system to allow the admin to give approval the order. This system also allowed the seller to choose option of the billing address for customer either it is the same as shipping address or different address. To make it interesting, the SuperProPOS: Product Ordering System also allow the seller to make a complaint of their order if any problems happened to the ordered product. The warehouse staff can view and update the status of the complaint to inform the customer or seller about the order complaint. The marketer is able to view all the activities of their assigned seller in the system.

### **2.5 Summary**

As a conclusion the SuperProPOS: Product Ordering System is relevant and relate to the existing systems. These existing systems have some disadvantages and could be improved to be implemented in the SuperProPOS: Product Ordering System.

## CHAPTER 3

### METHODOLOGY

#### 3.1 Introduction

This chapter discuss about the methodology that is suitable for the SuperProPOS: Product Ordering System. It discusses about chapter 3.2 *Project Methodology*, chapter 3.3 *Project Requirement* which consist of functional and non-functional requirement, constraints and limitation of system. The chapter 3.4 *Propose Design* shows the flowchart, context diagram, use case diagram and description, activity diagram and storyboard of the system. Chapter 3.5 called *Data Design* is discussing about the ERD and Database Dictionary of the system. Chapter 3.6 called *Proof of Initial Concept* provide the design of the proof concept with explanation of the system. Chapter 3.7 called *Testing Plan* and Chapter 3.8 called *Potential Use of Proposed Solution*. Lastly, it shows the project of the SuperProPOS: Product Ordering System in chapter 3.9 called *Gantt Chart*.

#### 3.2 Project Methodology

The methodology of the SuperProPOS: Product Ordering System implemented is Waterfall model of Software Development Life Cycle (SDLC). There are five phases involved in this model which is requirement, design, implementation, verification, and maintenance. Each phase is interacted sequentially into next phase. This methodology helps to prevent a well-defined and predictable project from going over its schedule and budget limits since each phase of the waterfall models are plan thoroughly. The duration of the project also takes around 6 months to be finished until the deadline and it is a sufficient time of the project to be done successfully.

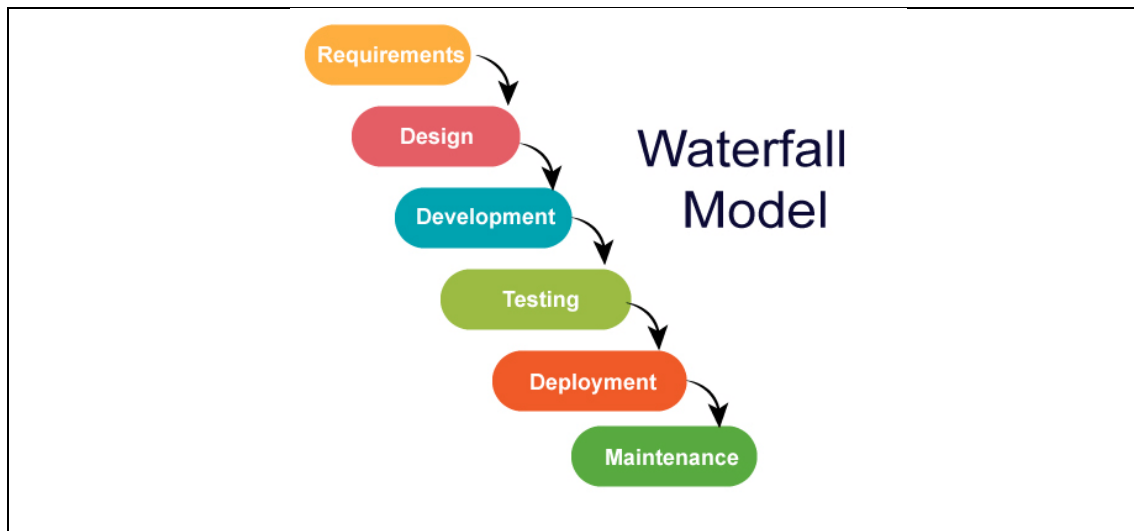


Figure 3.1 Waterfall Model Methodology

Figure 3.1 above shows the process of the Waterfall Model, which follows a sequential approach with distinct phases including Requirement, Design, Development, Testing, Deployment, and Maintenance. Throughout the development of SuperProPOS: Product Ordering System, the project adhered to this rigorous methodology. To ensure efficient project management and timely completion, a Gantt Chart was employed. The SuperProPOS Gantt Chart, presented in Appendix D, facilitated comprehensive planning and monitoring of the project phases. It served as a valuable visual tool to ensure that each stage of the Waterfall Model was executed smoothly and contributed to the successful development of the system.

### 3.2.1 Requirement Phase

The first phase is Requirement Gathering and Analysis. In here, requirements are gathered and understood by the meeting with the stakeholders like admin, seller, warehouse staff, marketer, software manager for the SuperProPOS: Product Ordering System. Opinions of the stakeholders also is asked to develop the system into a high-quality system. The project scope, costs, risks and timeline are also being set in this phase. The system requirements also have been diligently documented and can be referenced in the Software Requirement Specification document provided in Appendix A.

### **3.2.2 Design Phase**

Next, the second phase is System Design phase. In this phase, research of other existing systems that have similarity with SuperProPOS: Product Ordering System is being done. The listed systems are being compared to improvise the system. The technical solution for requirement of SuperProPOS: Product Ordering System is set out. The design of the system model and interface of the SuperProPOS: Product Ordering System of are done in this phase. The detailed design of the system has also been thoroughly documented and can be referred to in the Software Design Document provided in Appendix B.

### **3.2.3 Development Phase**

Later, the third phase is Development phase. The architecture of the system is built to see the flow of the system. The computer language like HTML, PHP, CSS, and JavaScript is being used to code the SuperProPOS: Product Ordering System based on the requirements and specification that have been gathered.

### **3.2.4 Testing Phase**

The fourth phase is Testing phase. Testing of the SuperProPOS: Product Ordering System is done to ensure the system is free error and there is no failure before the system can be released to customers. The testing of the system is performed by the stakeholders and UAT form is given. The User Acceptance Testing of the system has also been thoroughly tested and can be referred to in the User Acceptance Testing document provided in Appendix C.

### **3.2.5 Deployment Phase**

Later, the fifth phase is Deployment phase. The SuperProPOS: Product Ordering System is deployed to the FK INDAH server to ensure that the system is workable and zero-failure.

### **3.2.6 Maintenance Phase**

Lastly, once the deployment phase has passed, the system can be reviewed by the stakeholders. Maintenance is carried out after the deployment, and if any defects are found or change requests come from stakeholders, the SuperProPOS: Product Ordering System needs to update to a new version of the system.

## **3.3 Project Requirement**

The project requirement for the SuperProPOS: Product Ordering System consist of functional requirement, non-functional requirement, constraint requirement, limitation requirement and user requirement.

### **3.3.1 Functional requirement**

- The SuperProPOS: Product Ordering System must allow the marketer, seller, admin, and warehouse staff to login to the system.
- The SuperProPOS: Product Ordering System must allow the admin to make registration for seller, marketer and warehouse staff.
- The SuperProPOS: Product Ordering System must allow the admin, warehouse staff, marketer and seller to manage the product.
- The SuperProPOS: Product Ordering System must allow the marketer, seller, admin and warehouse staff to manage the order.
- The SuperProPOS: Product Ordering System must allow the seller and admin to manage the payment of the order.
- The SuperProPOS: Product Ordering System should allow the seller, admin, marketer and warehouse staff to manage the complaint of the order.

### **3.3.2 Non-Functional Requirement**

- Security: Only admin can see all the seller, marketer and warehouse staff personal information.
- Performance: The SuperProPOS: Product Ordering System should be compatible with any Web browser.
- Performance: The SuperProPOS: Product Ordering System should response the interaction between user and the systems not exceed 5 seconds.

### **3.3.3 Constraint Requirement**

- Access Control Constraint: The seller, admin, marketer and warehouse staff must log in to perform any task in SuperProPOS: Product Ordering System. It restricted access to authorised users only.
- Connectivity Constraint: The: SuperProPOS: Product Ordering System must have internet connection to allowed seller, admin, marketer and warehouse staff to use the system.
- Usability Constraint: The SuperProPOS: Product Ordering System should be user-friendly that allow user like seller, admin, marketer and warehouse staff to use the system without errors.
- Time Constraint: The SuperProPOS: Product Ordering System must be developed within the 6 months period given by the clients.
- Availability Constraint: The SuperProPOS: Product Ordering System should be available 99.99% of the time for any 24 hours.



### **3.3.4 Limitation Requirement**

- **Technical Limitation:** The reliance on an internet connection and the need for SuperProPOS: Product Ordering System availability are technical limitations that impose certain constraints on the system's functionality and usage.
- **Time Limitation:** The fixed timeline of 6 months for SuperProPOS: Product Ordering System development is a limitation related to time management and project completion.
- **Usability Limitation:** The expectation for the system to be user-friendly without errors imposes a limitation on the system's design and interface.
- **Usability Limitation:** The expectation for the system to be user-friendly without errors imposes a limitation on the system's design and interface.

### **3.4 Propose Design**

In this section, the design that related to SuperProPOS: Product Ordering System requirements was described. The Flowchart, Context Diagram, Use Case Diagram and Description and the Activity Diagram of the SuperProPOS: Product Ordering System is shown below.

The detailed of the proposed design also can be found in Appendix A which is the Software Requirement Specification (SRS) document of SuperProPOS: Product Ordering System.

### 3.4.1 Flowchart

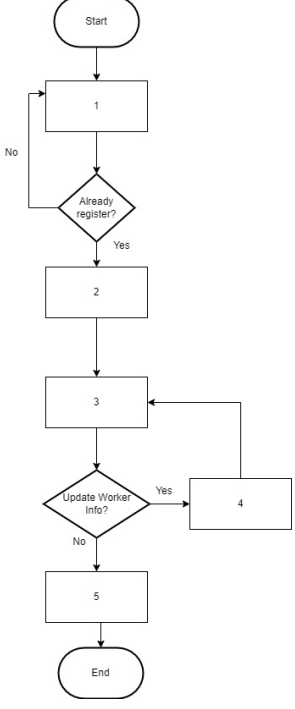
PROCESS	ACTION	RESPONSIBLE PERSON
 <pre> graph TD     Start([Start]) --&gt; 1[1]     1 --&gt; D1{Already register?}     D1 -- No --&gt; 1     D1 -- Yes --&gt; 2[2]     2 --&gt; 3[3]     3 --&gt; D2{Update Worker info?}     D2 -- Yes --&gt; 4[4]     D2 -- No --&gt; 5[5]     4 --&gt; 3     5 --&gt; End([End])         </pre>	<ol style="list-style-type: none"> <li>1. Login to the system</li> <li>2. Register Employee</li> <li>3. System displayed list of employees by position.</li> <li>4. Update employee Detail</li> <li>5. System updated list of employees</li> </ol>	<p>Admin</p> <p>Admin</p> <p>Admin</p>

Figure 3.2 Flowchart of Register Employee Module

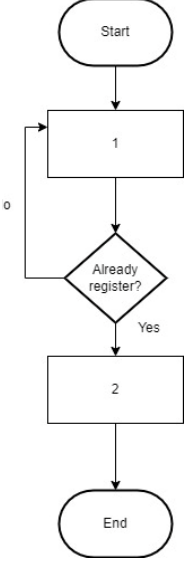
PROCESS	ACTION	RESPONSIBLE PERSON
 <pre> graph TD     Start([Start]) --&gt; 1[1]     1 --&gt; Decision{Already register?}     Decision -- No --&gt; 1     Decision -- Yes --&gt; 2[2]     2 --&gt; End([End]) </pre>	<ol style="list-style-type: none"> <li>1. Login to the system.</li> <li>2. Update Profile Information</li> </ol>	<p>Seller, Marketer, Warehouse Staff and Admin.</p> <p>Seller, Marketer, Warehouse Staff and Admin.</p>

Figure 3.3 Flowchart of Module Login

PROCESS	ACTION	RESPONSIBLE PERSON
<pre> graph TD     Start([Start]) --&gt; 1[1]     1 --&gt; D1{Already register?}     D1 -- No --&gt; 1     D1 -- Yes --&gt; 2[2]     2 --&gt; 3[3]     3 --&gt; D2{Update Product Info?}     D2 -- Yes --&gt; 4[4]     D2 -- No --&gt; 5[5]     4 --&gt; 3     5 --&gt; 6[6]     6 --&gt; End([End]) </pre>	<ol style="list-style-type: none"> <li>1. Login to the system.</li> <li>2. Add Product Information.</li> <li>3. System displayed list of product information.</li> <li>4. Update Product Information.</li> <li>5. System update list of products.</li> <li>6. View products information.</li> </ol>	<p>Admin</p> <p>Admin</p> <p>Admin and Warehouse staff</p> <p>Seller, Marketer, Warehouse staff and Admin</p>

Figure 3.4 Flowchart of Manage Product Detail Module.

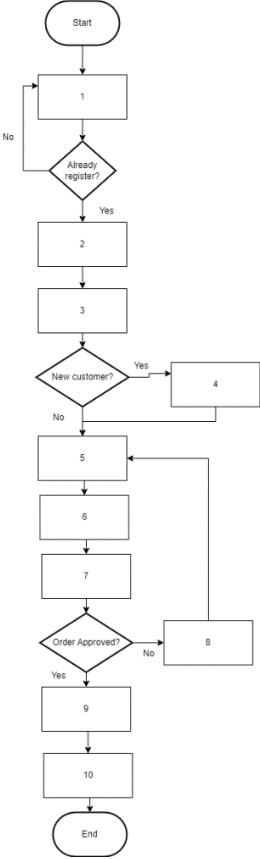
PROCESS	ACTION	RESPONSIBLE PERSON
 <pre> graph TD     Start([Start]) --&gt; 1[1]     1 --&gt; D1{Already register?}     D1 -- No --&gt; 1     D1 -- Yes --&gt; 2[2]     2 --&gt; 3[3]     3 --&gt; D2{New customer?}     D2 -- Yes --&gt; 4[4]     D2 -- No --&gt; 5[5]     4 --&gt; 5     5 --&gt; 6[6]     6 --&gt; 7[7]     7 --&gt; D3{Order Approved?}     D3 -- No --&gt; 8[8]     D3 -- Yes --&gt; 9[9]     8 --&gt; 5     9 --&gt; 10[10]     10 --&gt; End([End]) </pre>	<ol style="list-style-type: none"> <li>1. Login to the system.</li> <li>2. Choose product.</li> <li>3. Choose customer.</li> <li>4. Create new customer.</li> <li>5. Make order to system.</li> <li>6. Successfully created order.</li> <li>7. Check payment.</li> <li>8. Update order information.</li> <li>9. Packing order.</li> <li>10. Deliver order</li> </ol>	<p>Seller</p> <p>Seller</p> <p>Seller</p> <p>Seller</p> <p>Seller</p> <p>Seller</p> <p>Admin</p> <p>Seller</p> <p>Warehouse staff</p> <p>Warehouse staff</p>

Figure 3.5 Flowchart of Manage Order Module

PROCESS	ACTION	RESPONSIBLE PERSON
<pre> graph TD     Start([Start]) --&gt; 1[1]     1 --&gt; D1{Already register?}     D1 -- No --&gt; 1     D1 -- Yes --&gt; 2[2]     2 --&gt; 3[3]     3 --&gt; D2{Complaint accepted?}     D2 -- No --&gt; 4[4]     4 --&gt; 3     D2 -- Yes --&gt; 5[5]     5 --&gt; End([End]) </pre>	<ol style="list-style-type: none"> <li>1. Login to system.</li> <li>2. Make complaint in the system.</li> <li>3. Investigate complaints in system.</li> <li>4. Update complaint information.</li> <li>5. Complaint solved.</li> </ol>	<p>Seller, Marketer Seller</p> <p>Admin, Warehouse staff, Marketer. Seller</p> <p>Admin, Warehouse staff</p>

Figure 3.6 Flowchart of Manage Complaint Module.

### 3.4.2 Context Diagram

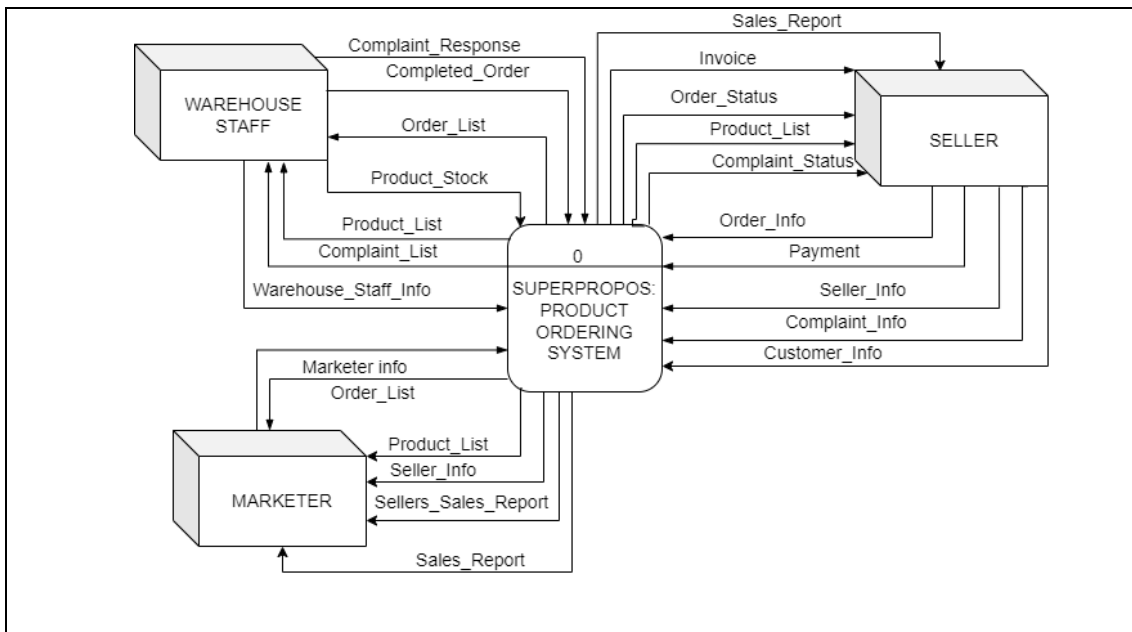


Figure 3.7 Context Diagram of SuperProPOS: Product Ordering System

### 3.4.3 Use Case Diagram

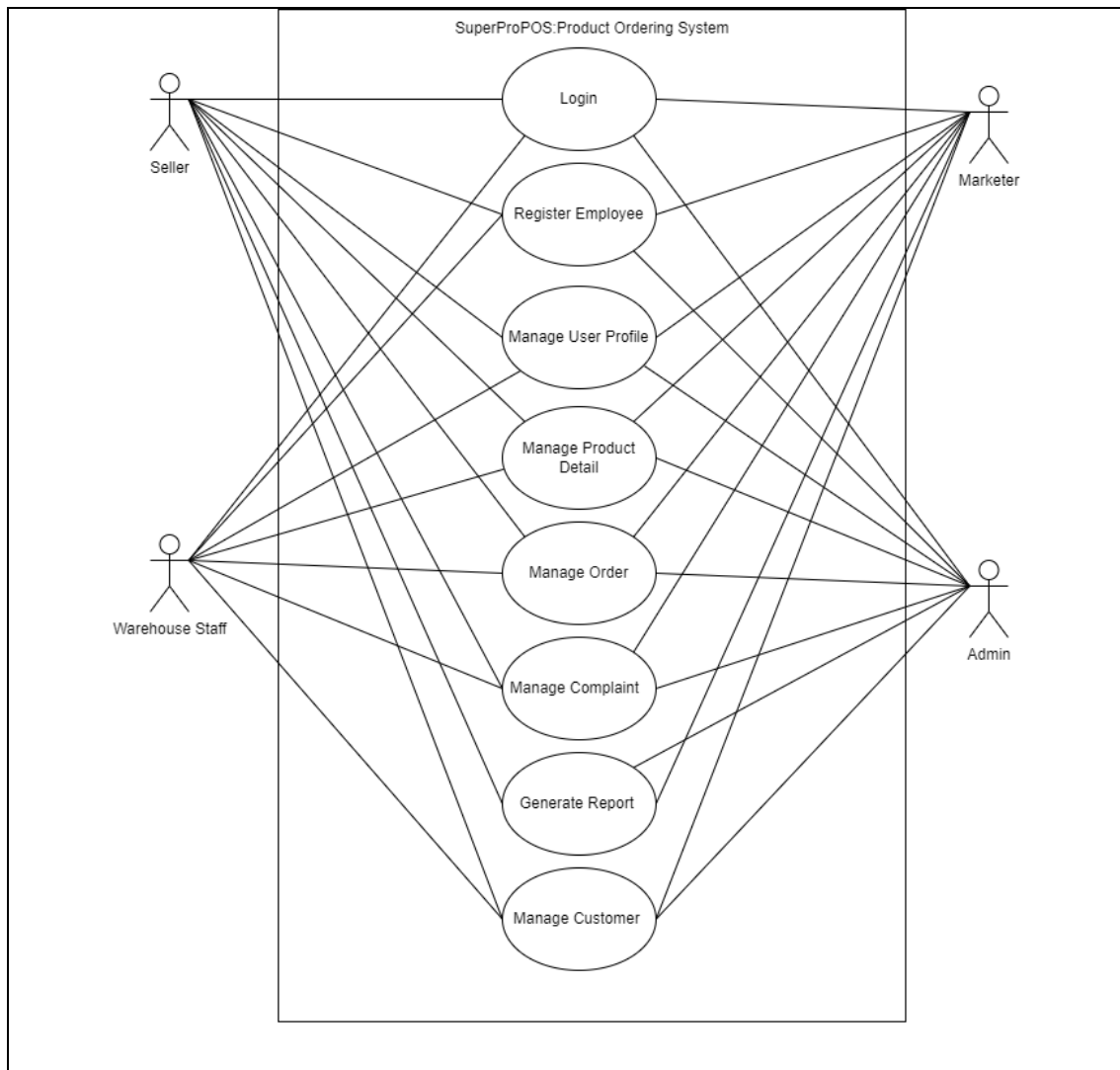


Figure 3.8 Use Case Diagram of SuperProPOS: Product Ordering System



### 3.4.4 Use Case Description

The use case description for Manage Order module and Manage Complaint module for SuperProPOS: Product Ordering System example are being shown below in table 3.1 and table 3.2. The details of the Use Case Description for the whole system can be referred in Software Requirement System in Appendix A.

#### 3.4.4.1 Manage Order

Table 3.1 Use Case Description for Manage Order

Use Case ID	POS-UC005
Use Case Name	Manage Order
Brief Description	<p>This use case is initiated by seller, admin, marketer and warehouse staff. It allowed the seller to make and order for the product. The seller also can edit the order information.</p> <p>The admin can view and update the order payment status from the order list.</p> <p>The warehouse staff can view and update the order status and tracking number after the payment has been approved by admin from the order list.</p> <p>The marketer can view all the order list information from their assigned seller from the database.</p>
Actor	Admin, Warehouse staff, Seller and Marketer.

Pre-Condition	<ul style="list-style-type: none"> <li>• The admin, warehouse staff, seller and marketer already have a registered account in the system.</li> <li>• The admin, warehouse staff, seller and marketer already login to the system.</li> <li>• The admin, warehouse staff, seller and marketer must have internet connection.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>1. The use case starts when the seller goes to Order Now menu page.</li> <li>2. The system displays the products to be ordered by seller.</li> <li>3. The seller clicks &lt;&lt;ADD TO CART&gt;&gt; button on the product.</li> <li>4. The system submits the product to the cart.</li> <li>5. The seller clicks the &lt;&lt;CART&gt;&gt; button to go to the cart page.</li> <li>6. The system displays the chosen products details and its total price and payment that need to be made.</li> <li>7. The seller is able to <ol style="list-style-type: none"> <li>a. Update quantity of products in cart. <b>[A1: Update Cart]</b></li> </ol> </li> </ol>

	<ul style="list-style-type: none"><li>b. Delete the selected product from cart. <b>[A2: Delete Cart]</b></li><li>8. The seller then clicks on the &lt;&lt;Proceed to Check Out&gt;&gt; button.</li><li>9. The system displays the total price, total products and order form.</li><li>10. The seller is able to:<ul style="list-style-type: none"><li>a. Add order details choose existing customer details, payment method and upload the receipt image to the system <b>[A3: Add Order]</b></li><li>b. Update order details in the system. <b>[A4: Edit Order]</b></li></ul></li><li>11. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</li><li>12. The system validates the order information.</li><li>13. The system displays successful message to the seller.</li><li>14. The admin is able to update the status of order payment in the database.</li><li>15. The warehouse staff is able to update the delivery status and tracking number of order after admin approved the payment in the database.</li></ul>
--	---

	<p>16. The seller, marketer, warehouse staff and admin are able to view the updated order details by click on the Order List menu.</p> <p>17. The system list all the order details from the database.</p> <p>18. The use case ends.</p>
Alternative Flow	<p><b>A1: Update Cart</b></p> <ol style="list-style-type: none"> <li>1. The seller inserts the quantity of selected products.</li> <li>2. The seller clicks the &lt;&lt;UPDATE&gt;&gt; button.</li> <li>3. The use case continues to use case 8 in basic flow.</li> </ol> <p><b>A2: Delete Cart</b></p> <ol style="list-style-type: none"> <li>1. The seller chooses the existing products in cart.</li> <li>2. The seller clicks &lt;&lt;DELETE&gt;&gt; button.</li> <li>3. The use case continues to use case 8 in basic flow.</li> </ol> <p><b>A3: Add Order</b></p> <ol style="list-style-type: none"> <li>1. The seller chooses the existing customer from the list. <b>[A3.1 Add New Customer]</b></li> <li>2. The seller inserts the payment method and upload the receipt image to the system.</li> </ol>

	<p>3. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</p> <p>4. The use case continues to step 11 in basic flow.</p> <p><b>A4: Edit Order</b></p> <p>1. The seller chooses the order detail that he/she already done it before this.</p> <p>2. The seller clicks &lt;&lt;EDIT&gt;&gt; button.</p> <p>3. The seller edits the order's payment receipt image that he/she key-in before this.</p> <p>4. The use case continues to step 11 in basic flow.</p> <p><b>A3.1 Add New Customer</b></p> <p>1. The seller clicks on the &lt;&lt;+NEW CUSTOMER&gt;&gt; button.</p> <p>2. The seller inserts customer name, phone number and address to the system.</p> <p>3. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</p> <p>4. The use case continues to step 1 in alternative flow A3.</p>
Exception Flow	None
Post Conditions	The list of order is successfully updated.
Rules	None
Constraints	None

### 3.4.4.2 Manage Complaint

Table 3.2 Use Case Description for Manage Complaint

Use Case ID	POS-UC006
Use Case Name	Manage Complaint
Brief Description	<p>This use case is initiated by seller, admin, warehouse staff and marketer. It allowed the seller to add a complaint for the order in the system. The seller also can edit the complaint information.</p> <p>The admin and warehouse staff can view and update the status of complaint in the complaint information.</p> <p>The marketer can view the complaint information of their assigned seller from the database.</p>
Actor	Admin, Warehouse staff, Seller and Marketer.
Pre-Condition	<ul style="list-style-type: none"><li>• The admin, warehouse staff, seller and marketer already have a registered account in the system.</li><li>• The admin, warehouse staff, seller and marketer already login to the system.</li><li>• The admin, warehouse staff, seller and marketer must have internet connection.</li></ul>
Basic Flow	<ol style="list-style-type: none"><li>1. The use case starts when the seller goes to complaint menu page.</li><li>2. The system displays the complaint list detail.</li></ol>

	<ol style="list-style-type: none"> <li>3. The seller is able to: <ol style="list-style-type: none"> <li>a. Add new complaint by order id, problem, details of problem and image in the system. [<b>A1: Add Complaint</b>]</li> <li>b. Edit the complaint problem, details of problem and image in the system [<b>A2: Edit Complaint</b>]</li> </ol> </li> <li>4. The seller clicks &lt;&lt;SAVE&gt;&gt; button.</li> <li>5. The system validates the complaint information.</li> <li>6. The system displays successful message to the seller.</li> <li>7. The admin and warehouse staff are able to update the status of complaint in the database.</li> <li>8. The marketer is able to view the complaint list made by their assigned sellers.</li> <li>9. The system list all the complaint details from the database.</li> <li>10. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1: Add Complaint</b></p> <ol style="list-style-type: none"> <li>1. The seller goes to Add New Complaint menu.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The seller inserts the order id, problem, details of problem and image in the system.</li> <li>3. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</li> <li>4. The use case continues to step 4 in basic flow.</li> </ol> <p><b>A2: Edit Complaint</b></p> <ol style="list-style-type: none"> <li>1. The seller chooses the complaint detail that he/she already done it before this.</li> <li>2. The seller clicks &lt;&lt;EDIT&gt;&gt; button.</li> <li>3. The seller edits the complaint detail that he/she key-in before this.</li> <li>4. The use case continues to step 4 in basic flow.</li> </ol>
Exception Flow	None
Post Conditions	The list of complaints is successfully updated.
Rules	None
Constraints	None



Table 3.3 Short Description of Use Case for SuperProPOS: Product Ordering System

Actor	Use Case Name	Short Description
Marketer, Seller, Admin and Warehouse Staff	Login	The marketer, seller, admin, and warehouse staff need to use their email and password to login into the system.
Admin, Seller, Marketer and Warehouse Staff	Register Employee	The admin needs to register marketer, seller and warehouse staff account before using the system. The seller, marketer and warehouse staff must be able to view their registered profile.
Seller, Warehouse staff, Admin and Marketer	Manage User Profile	Provide the capability of seller, warehouse staff, marketer and admin to update their user profile information. The marketer also can view the seller's profile. The admin also can view and update seller, marketer and warehouse profile information.
Admin, Warehouse staff, Seller and Marketer	Manage Product Detail	The admins are able to view, add, update and

		delete the product. Warehouse staff can update the products stocks in the system. Seller and Marketer can view the product list.
Seller, Marketer, Admin, and Warehouse staff	Manage Order	The seller can add and edit their order before payment. The admin can update the payment status of the order. The warehouse staff can view and update the delivery status and tracking number of the order. The marketer can view the order made by their assigned seller.
Marketer, Seller, Admin and Warehouse staff	Manage Complaint	The seller can add and update the complaint for the order. Warehouse staff and admin can view and update the status of the complaint. Marketer can view the complaint made by the seller.
Admin, Seller and Marketer	Generate Report	The system allows the admin, seller and marketer to view their sales report by daily, monthly and yearly.

### 3.4.5 Activity Diagram

#### 3.4.5.1 Login

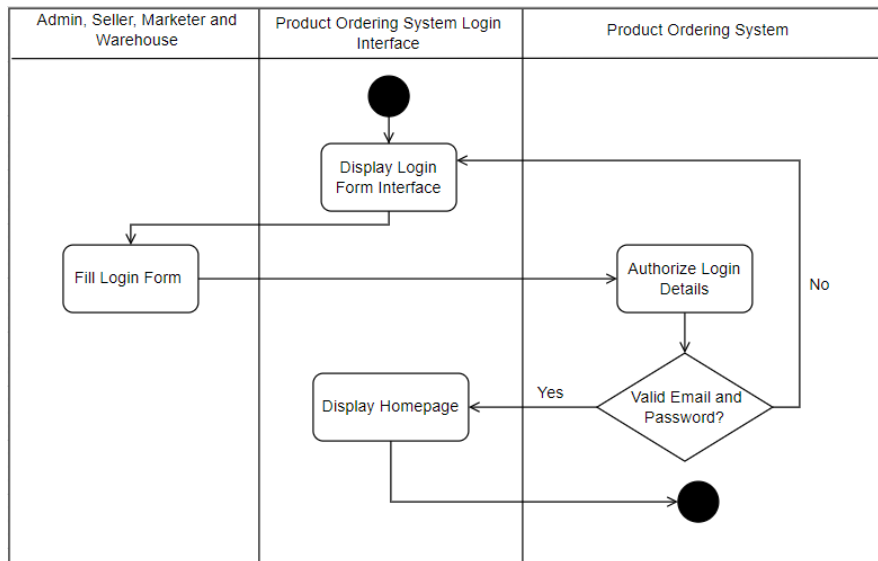


Figure 3.9 Activity Diagram for Login

#### 3.4.5.2 Register Employee

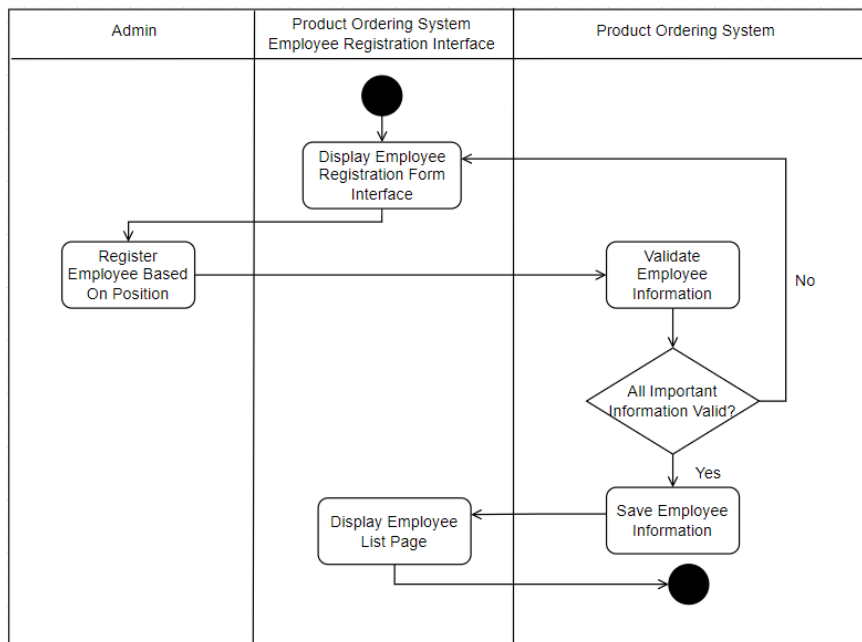


Figure 3.10 Activity Diagram for Register Employee.

### 3.4.5.3 Manage User Profile

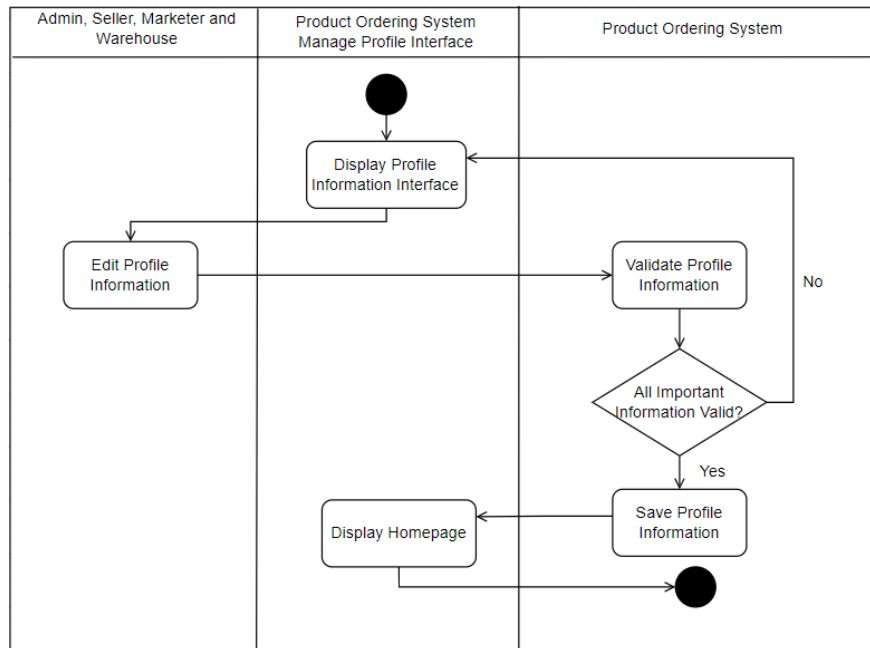


Figure 3.11 Activity Diagram for Manage User Profile

### 3.4.5.4 Manage Product

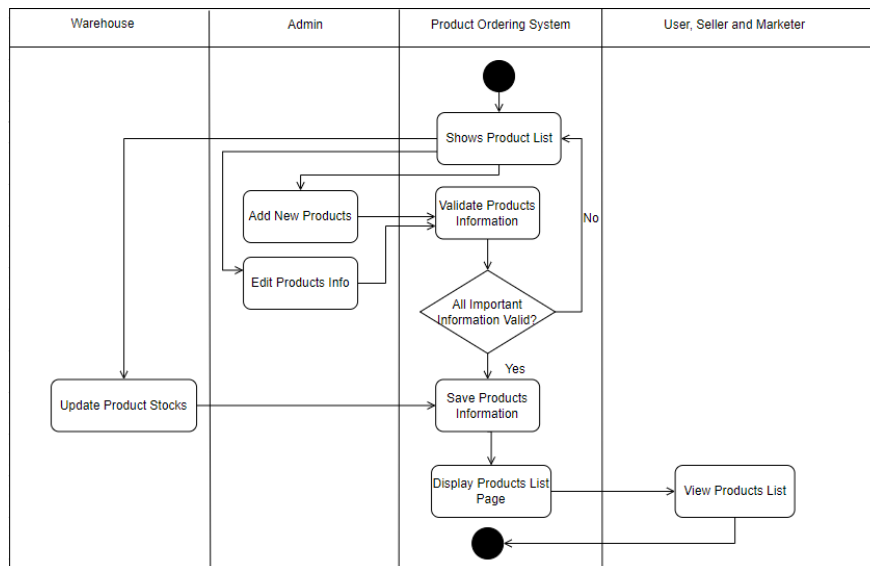


Figure 3.12 Activity Diagram for Manage Product

### 3.4.5.5 Manage Complaint

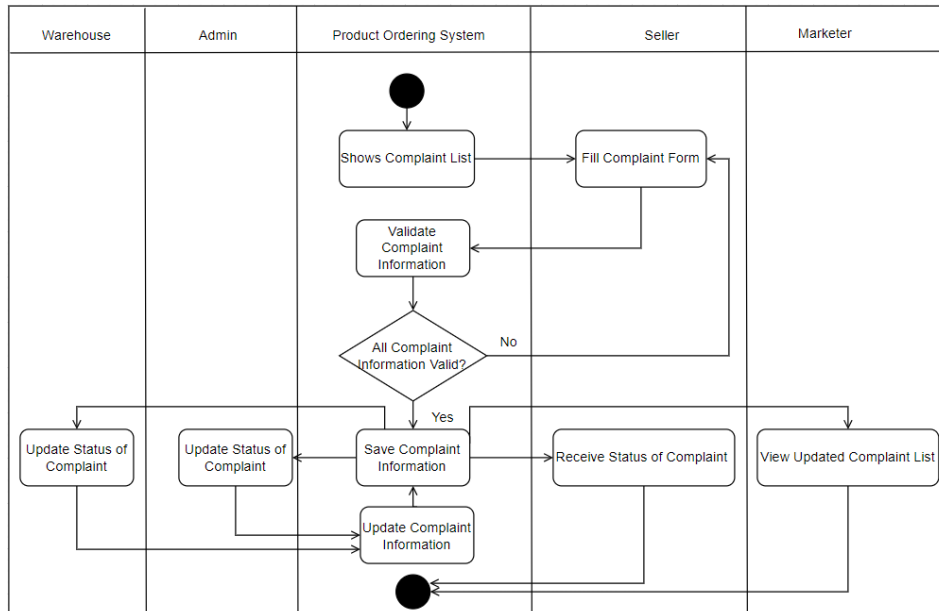


Figure 3.13 Activity Diagram for Manage Complaint

### 3.4.5.6 Manage Order

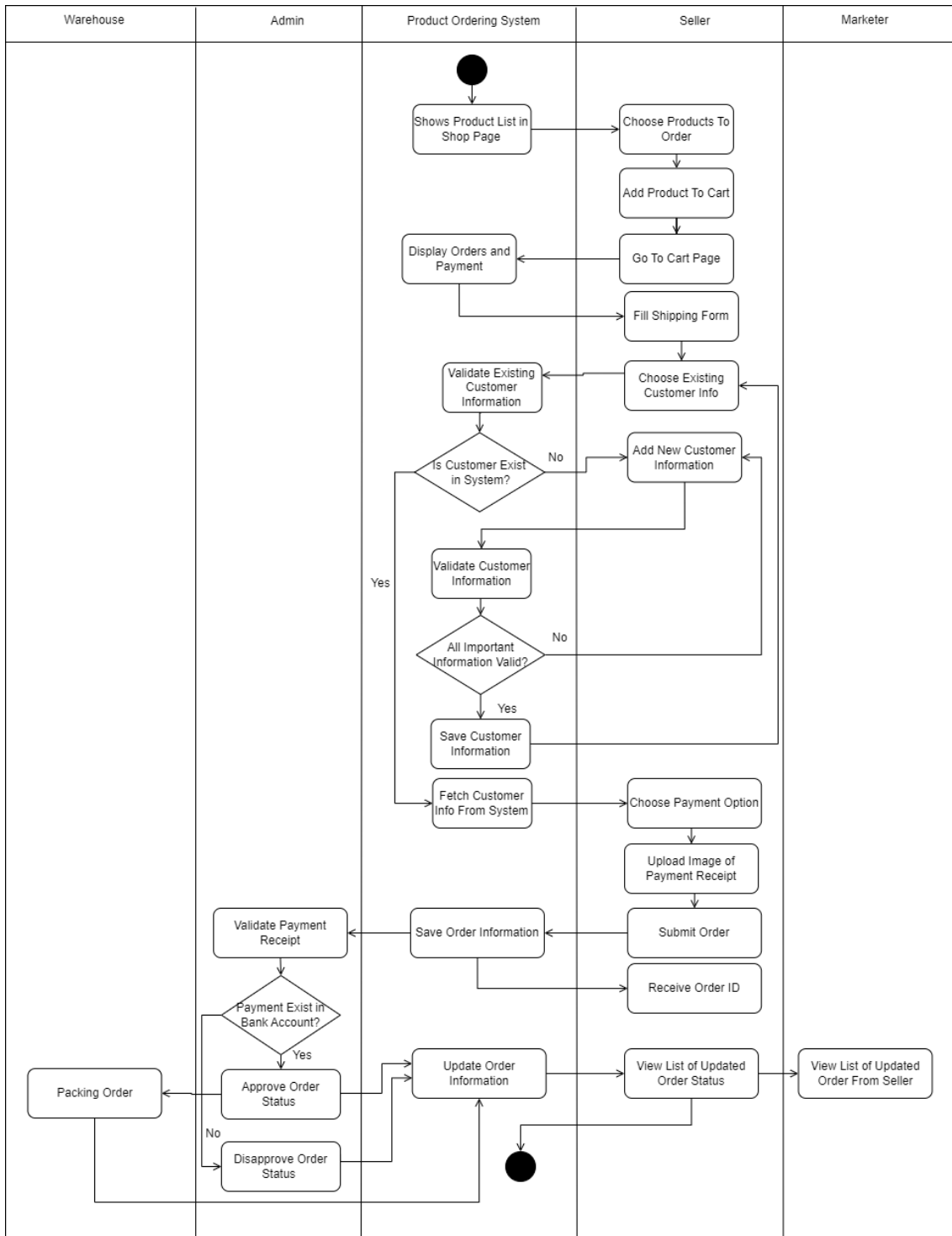


Figure 3.14 Activity Diagram for Manage Order

### 3.5 Data Design

Below are the details of the data design for SuperProPOS: Product Ordering System. The explanation of the data designs also available in Appendix B as Software Design Description (SDD) document.

#### 3.5.1 Entity Relationship Diagram

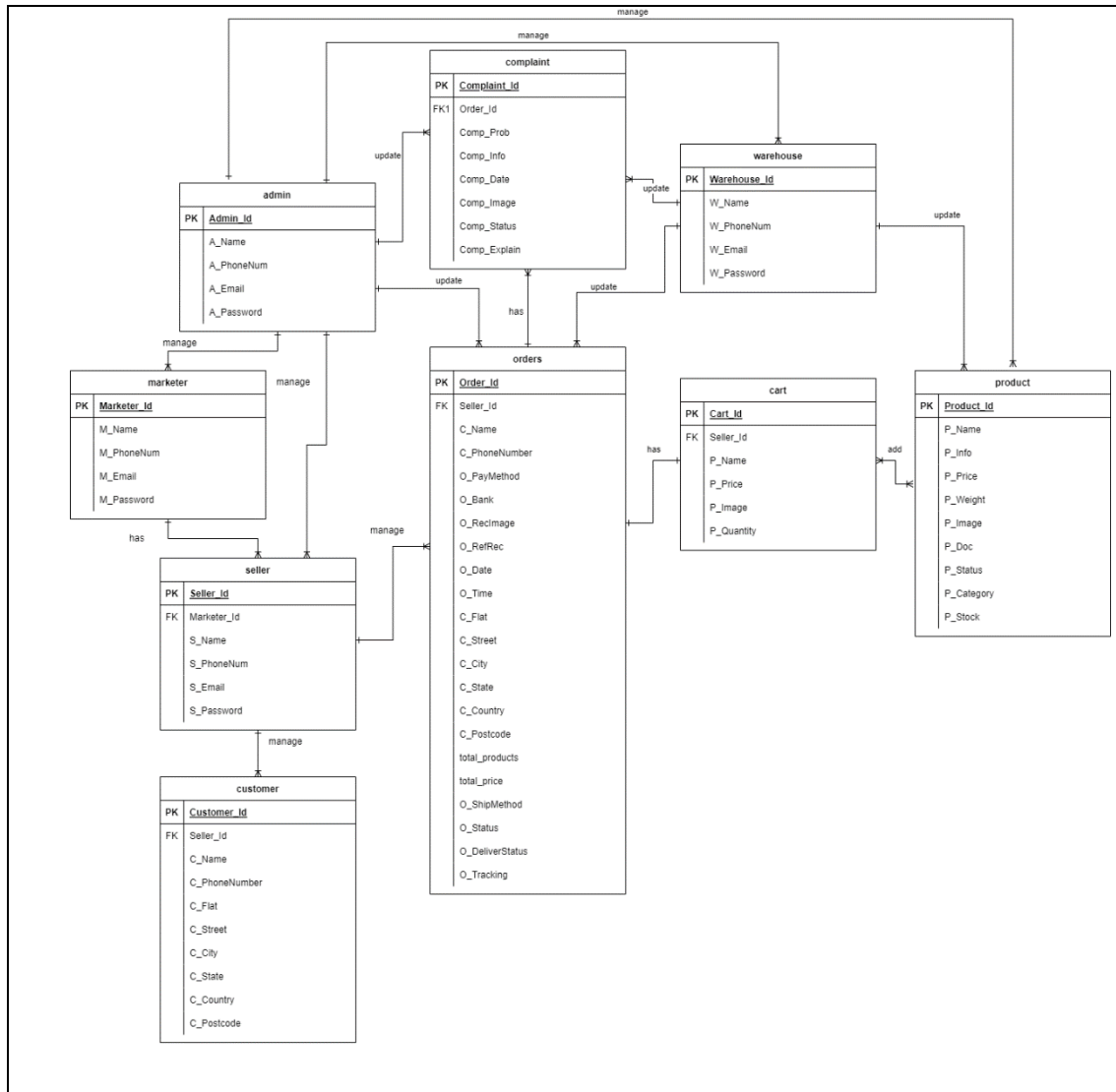


Figure 3.15 ERD of SuperProPOS: Product Ordering System

### 3.5.2 Data Dictionary

#### 3.5.2.1 admin

Table 3.4 Table for admin

Field Name	Description	Data Type	Constraint
Admin_Id	Admin ID	INT (10)	PK
A_Name	Admin Name	VARCHAR (200)	
A_PhoneNum	Admin Phone Number	VARCHAR (11)	
A_Email	Admin Email	VARCHAR (100)	
A_Password	Admin Password	VARCHAR (20)	

#### 3.5.2.2 marketer

Table 3.5 Table for marketer

Field Name	Description	Data Type	Constraint
Marketer_Id	Marketer ID	INT (10)	PK
M_Name	Marketer Name	VARCHAR (200)	
M_PhoneNum	Marketer Phone Number	VARCHAR (11)	
M_Email	Marketer Email	VARCHAR (100)	
M_Password	Marketer Password	VARCHAR (20)	

#### 3.5.2.3 seller

Table 3.6 Table for seller

Field Name	Description	Data Type	Constraint
Seller_Id	Seller ID	INT (10)	PK
Marketer_Id	Marketer ID	INT (10)	FK
S__Name	Seller Name	VARCHAR (200)	
S_PhoneNumber	Seller Phone Number	VARCHAR (11)	



S_Email	Seller Email	VARCHAR (100)	
S_Password	Seller Password	VARCHAR (20)	

### 3.5.2.4 warehouse

Table 3.7 Table for warehouse

Field Name	Description	Data Type	Constraint
Warehouse_Id	Warehouse staff ID	INT (10)	PK
W_Name	Warehouse staff Name	VARCHAR (200)	
W_PhoneNum	Warehouse staff Phone Number	VARCHAR (12)	
W_Email	Warehouse staff Email	VARCHAR (100)	
W_Password	Warehouse staff Password	VARCHAR (50)	

### 3.5.2.5 customer

Table 3.8 Table for customer

Field Name	Description	Data Type	Constraint
Customer_Id	Customer ID	INT (10)	PK
Seller_Id	Seller ID	INT (10)	FK
C__Name	Customer Name	VARCHAR (200)	
C_PhoneNum	Customer Phone Number	VARCHAR (11)	
C_Flat	Customer Flat Address	VARCHAR (255)	
C_Street	Customer Street Address	VARCHAR (255)	
C_City	Customer City Address	VARCHAR (255)	
C_State	Customer State Address	VARCHAR (255)	
C_Country	Customer Country Address	VARCHAR (255)	
C_Postcode	Customer Postcode Address	VARCHAR (255)	
C_Date	Customer Information Date Inserted	VARCHAR (10)	

### 3.5.2.6 product

Table 3.9 Table for product

Field Name	Description	Data Type	Constraint
Product_Id	Product ID	INT (11)	PK
P_Name	Product Name	VARCHAR (200)	
P_Info	Product Info	VARCHAR (500)	
P_Price	Product Price	DECIMAL (7,2)	
P_Weight	Product Weight	FLOAT	
P_Image	Product Image	VARCHAR (500)	
P_Doc	Product Document	VARCHAR (500)	
P_Status	Product Status	VARCHAR (20)	
P_Category	Product Category	VARCHAR (20)	
P_Stock	Product Stock	INT (11)	

### 3.5.2.7 cart

Table 3.10 Table for cart

Field Name	Description	Data Type	Constraint
Cart_Id	Cart ID	INT (11)	PK
Seller_Id	Seller ID	INT (255)	FK
P_Name	Product Name	VARCHAR (255)	
P_Price	Product Price	DECIMAL (7,2)	
P_Image	Product Image	VARCHAR (255)	
P_Quantity	Product Quantity	VARCHAR (255)	

### 3.5.2.8 orders

Table 3.11 Table for orders

Field Name	Description	Data Type	Constraint
Order_Id	Order ID	INT (255)	PK
Seller_Id	Seller ID	INT (12)	FK
C_Name	Customer Name	VARCHAR (255)	
C_PhoneNum	Customer Phone Number	VARCHAR (255)	
O_PayMethod	Payment Method	VARCHAR (255)	
O_Bank	Bank Name	VARCHAR (255)	
O_ReclImage	Receipt Image	VARCHAR (255)	
O_RefRec	Receipt Reference	VARCHAR (255)	
O_Date	Receipt Payment Date	DATE	
O_Time	Receipt Payment Time	TIME	
C_Flat	Customer Flat Address	VARCHAR (255)	
C_Street	Customer Street Address	VARCHAR (255)	
C_City	Customer City Address	VARCHAR (255)	
C_State	Customer State Address	VARCHAR (255)	
C_Country	Customer Country Address	VARCHAR (255)	
C_Postcode	Customer Postcode Address	VARCHAR (255)	
total_products	Total Product Ordered	VARCHAR (255)	
total_price	Total Price Paid	DECIMAL (7,2)	

O_ShipMethod	Order Shipment Method	VARCHAR (255)	
O_Status	Order Payment Status	VARCHAR (255)	
O_DeliverStatus	Order Delivery Status	VARCHAR (255)	
O_Tracking	Order Tracking Number	VARCHAR (255)	

### 3.5.2.9 complaint

Table 3.12 Table for complaint

Field Name	Description	Data Type	Constraint
Complaint_Id	Complaint ID	INT (10)	PK
Order_Id	Order ID	INT (10)	FK
Comp_Prob	Complaint Problem	VARCHAR (20)	
Comp_Info	Complaint Info	VARCHAR (500)	
Comp_Date	Complaint Date	TIMESTAMP	
Comp_Image	Complaint Image	VARCHAR (500)	
Comp_Status	Complaint Status	VARCHAR (20)	
Comp_Explain	Complaint Status Explanation	VARCHAR (500)	

### 3.6 Proof of Initial Concept

In this initial concept part, it shows the proof of concept (POC) of the SuperProPOS: Product Ordering System. It is done to test the general idea and verify that the system can be executed from technical point of view. The POC of the SuperProPOS: Product Ordering System is shown by the prototype shown below by categories of Admin, Seller, Marketer and Warehouse staff. Before the users go to their own site, the mainpage of SuperProPOS: Product Ordering System is displayed to let the users to choose their page based on their position as figure 3.16 below.



Figure 3.16 SuperProPOS: Product Ordering System Main Page

#### 3.6.1 Admin Page

The admin firstly needs to login to the SuperProPOS: Product Ordering System based on figure 3.17.

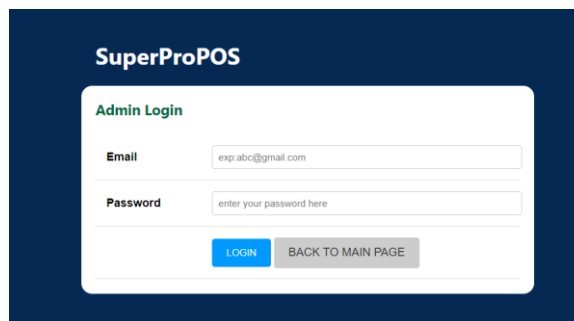


Figure 3.17 UI of Admin Login Page

Once the admin success to login, the system will display the Home Page for admin. Figure 3.18 shows the Home Page below.

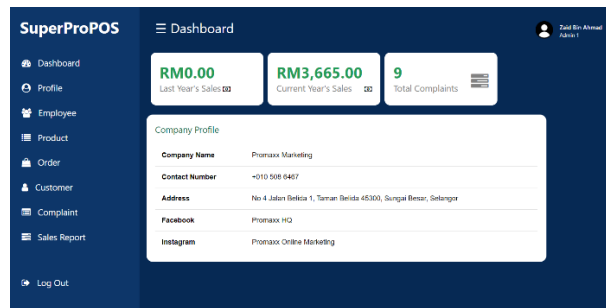


Figure 3.18 UI Homepage for Admin

The admin then can make a registration for employee which is seller, marketer and warehouse staff in the Employee List page. Figure 3.19 and 3.21 below shows the Register Employee page.

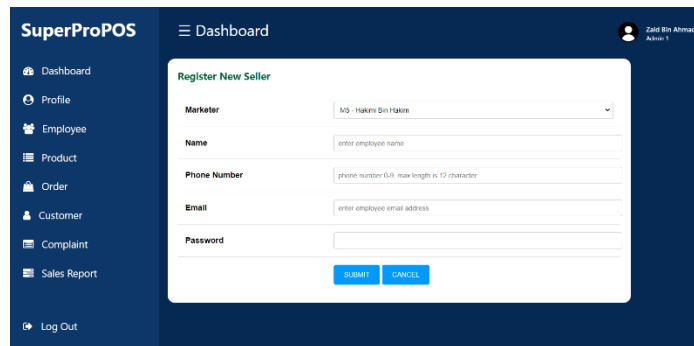


Figure 3.19 UI Register Seller by Admin

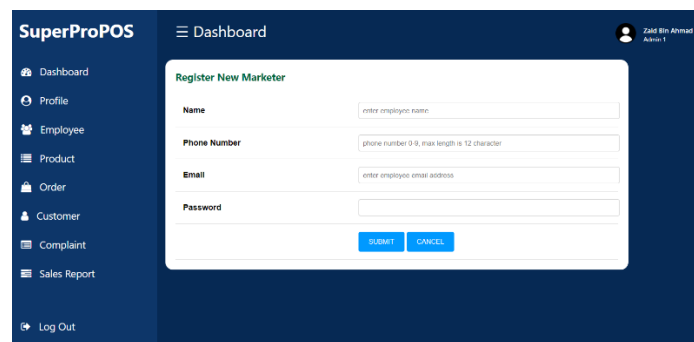


Figure 3.20 UI Register Marketer by Admin

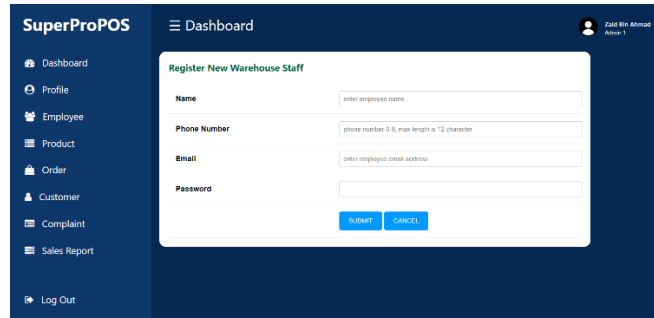


Figure 3.21 UI Register Warehouse Staff by Admin

Once the registration of employee done, the admin can view the list of employees in menu “Seller List”, “Marketer List” and “Warehouse staff List”. Figure 3.22 until figure 3.24 shows the list of the employee by position.

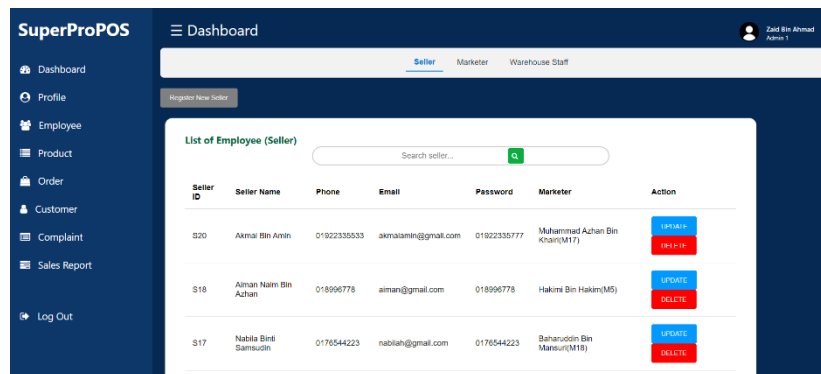


Figure 3.22 UI Seller List for Admin

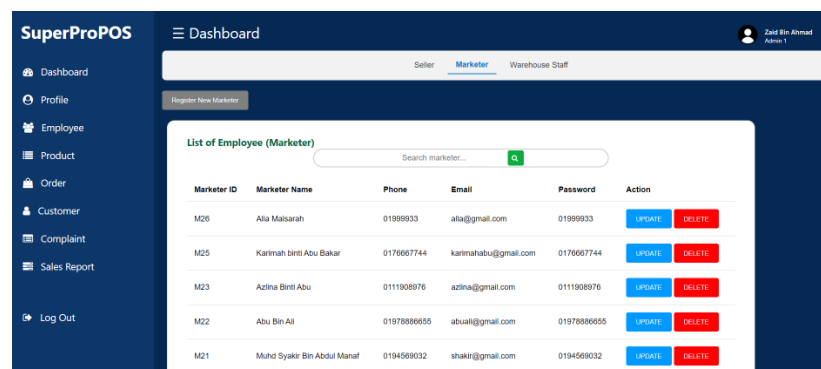


Figure 3.23 UI Marketer List for Admin

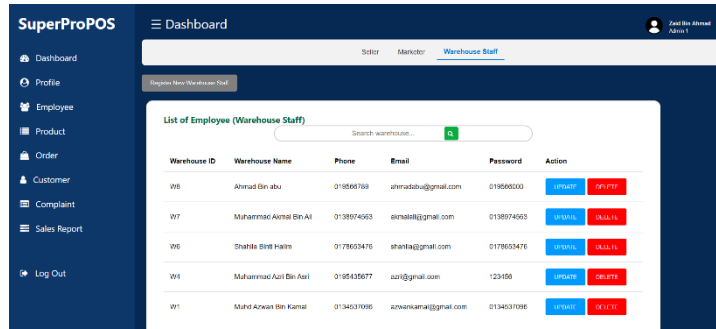


Figure 3.24 UI Warehouse staff List for Admin

The admin profile can be updated by the admin as show in figure 3.25 below.

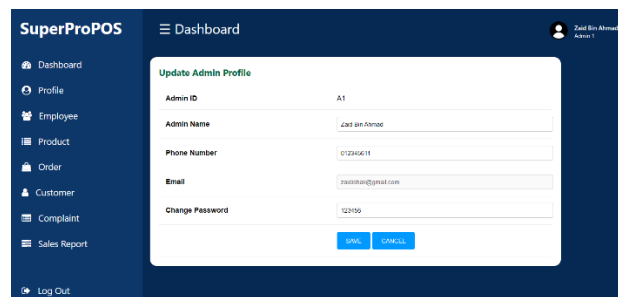


Figure 3.25 UI Update Admin Detail for Admin

The admin also can manage the product by add, update and delete the products information to the system. Below is the page for admin to manage the product based on figure 3.26 until figure 3.28.

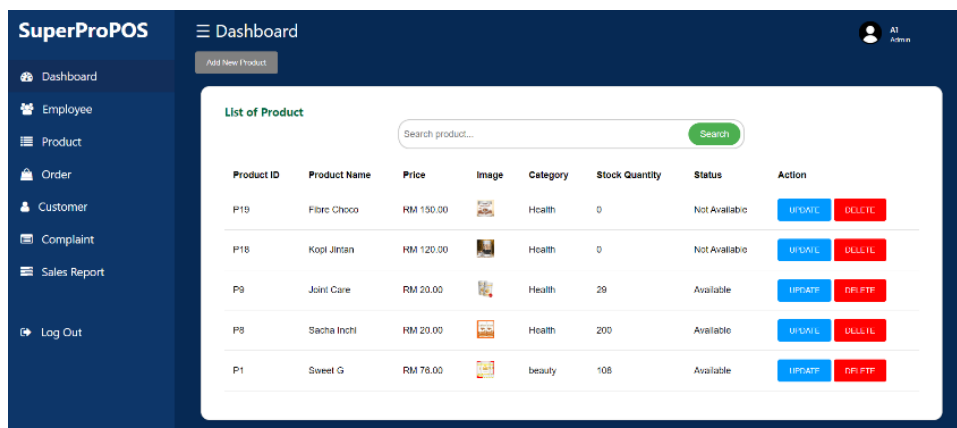


Figure 3.26 UI Product List for Admin



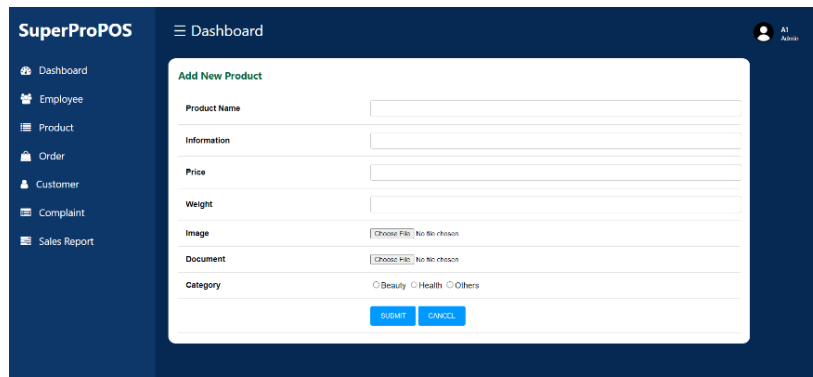


Figure 3.27 UI Add New Product for Admin

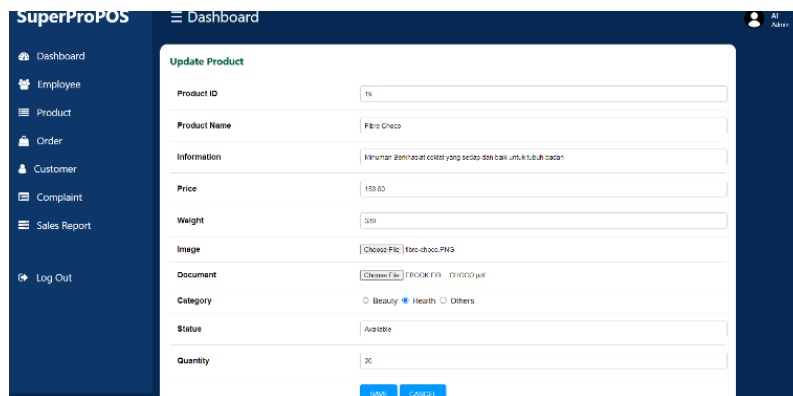


Figure 3.28 UI Update Product for Admin

Admin can manage the order by view order list and update the order status. Below shows the interface for order list and update order based on figure 3.29 and figure 3.30.

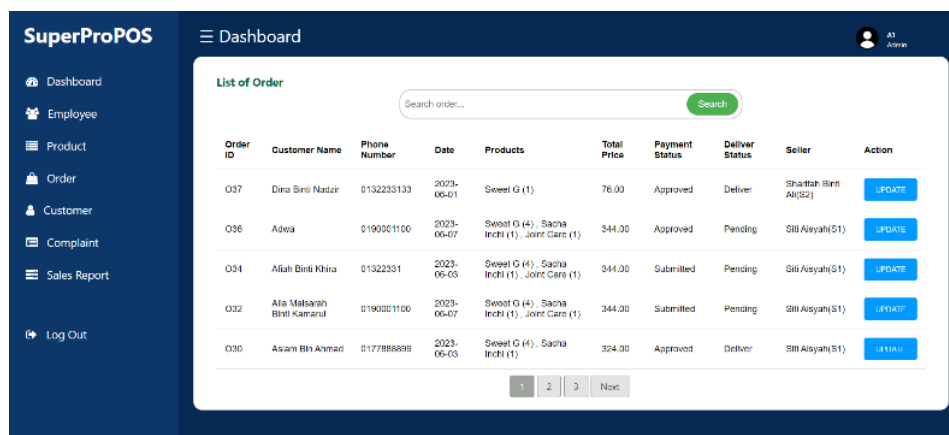


Figure 3.29 UI Order List for Admin

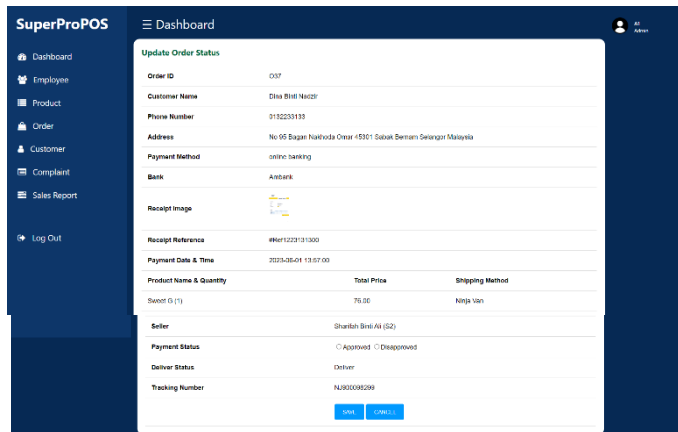


Figure 3.30 UI Update Order Status for Admin

The admin also can manage the complaint by view and update the status of the complaint. Figure 3.31 and figure 3.32 shows the interface of complaint for admin.

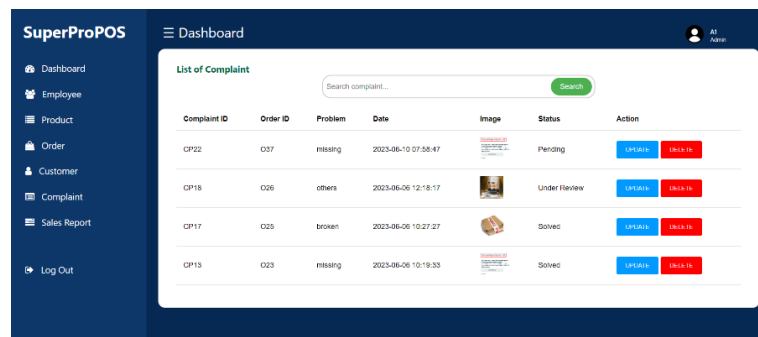


Figure 3.31 UI Complaint List for Admin

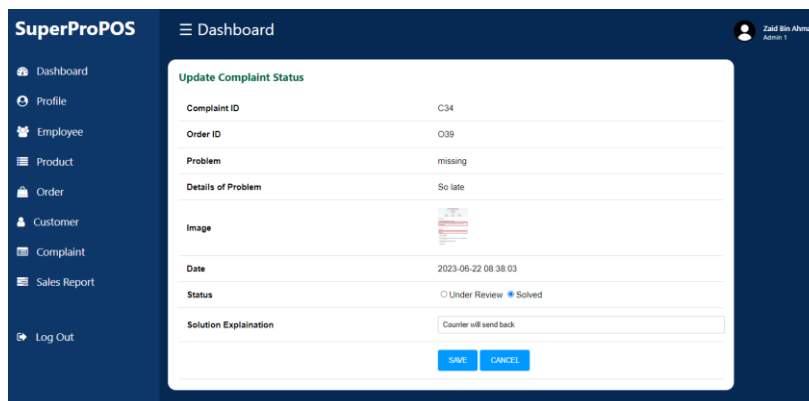


Figure 3.32 UI Update Complaint Status for Admin

Lastly admin also can view the sales report in the system. Figure 3.33 and figure 3.34 shows the interface of the sales report.

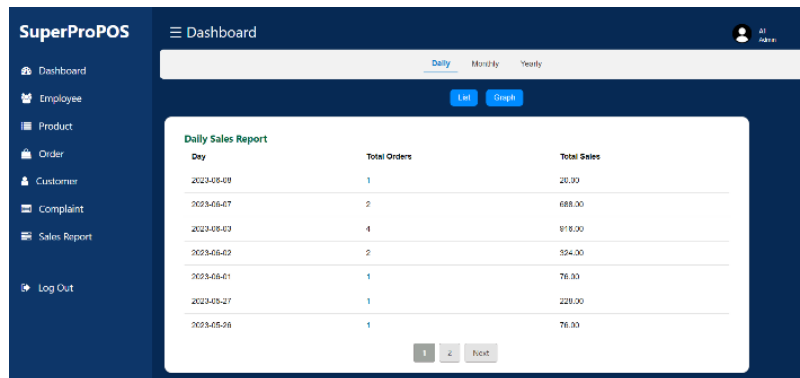


Figure 3.33 UI List Sales Report for Admin

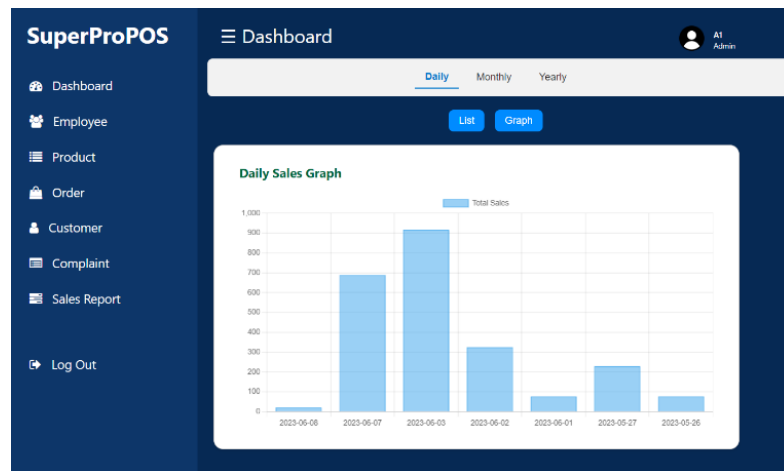


Figure 3.34 UI Graph Sales Report for Admin

### 3.6.2 Seller Page

Firstly, the seller needs to login. Figure 3.35 below shows the login interface.

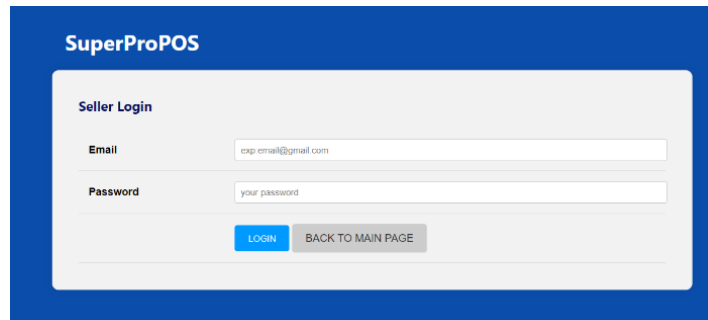


Figure 3.35 UI Login for Seller

Once seller successfully login to system, seller is able to view the homepage. Figure 3.36 below shows the homepage of seller.

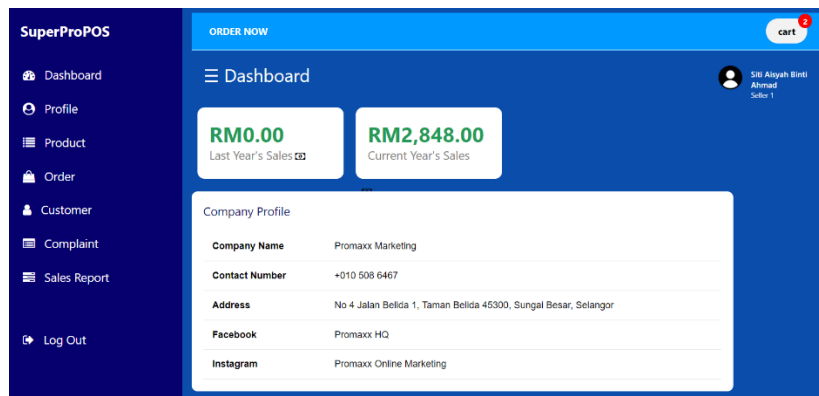


Figure 3.36 UI Homepage for Seller

The seller then can be able to view the products by click on the Product Menu. Figure 3.37 and figure 3.38 shows the product list and details for seller.

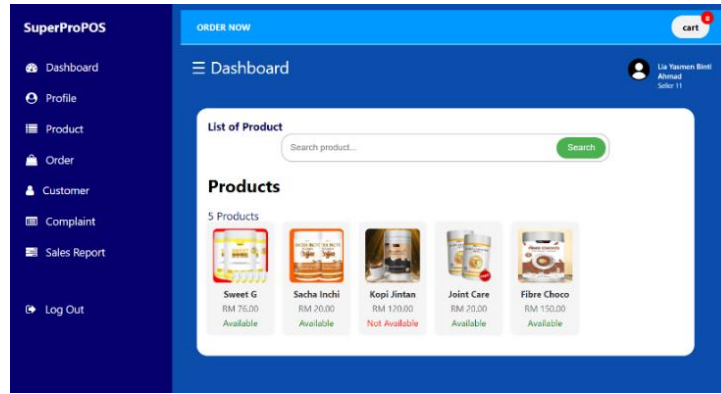


Figure 3.37 UI Product List for Seller

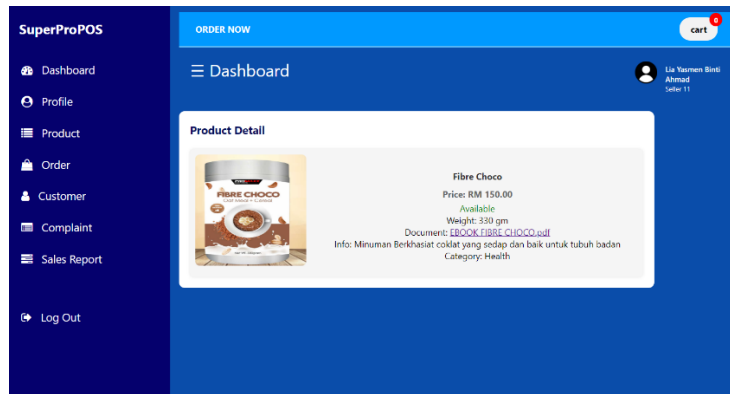


Figure 3.38 UI Product Detail for Seller

The seller also can make an order by going to the Shop menu. Figure 3.39 until figure 3.43 shows the process of ordering product from the shop menu.

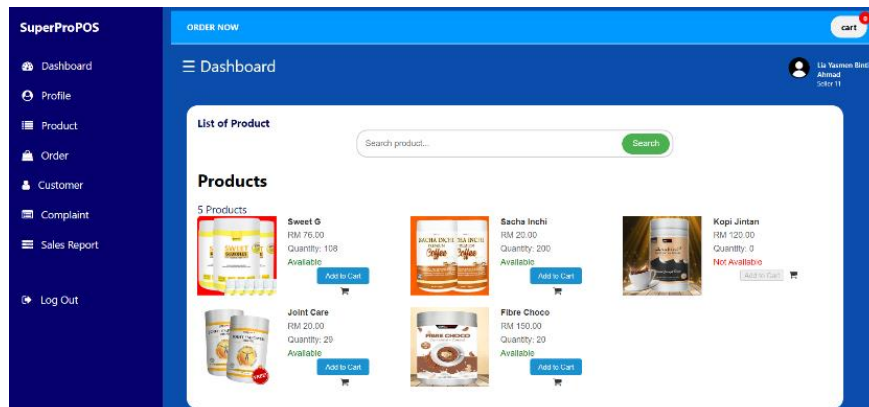


Figure 3.39 UI Shop for Seller

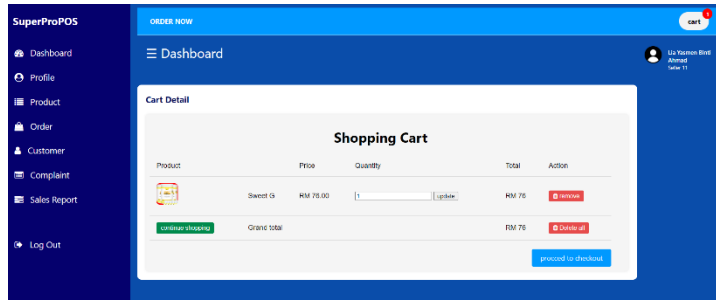


Figure 3.40 UI Cart for Seller

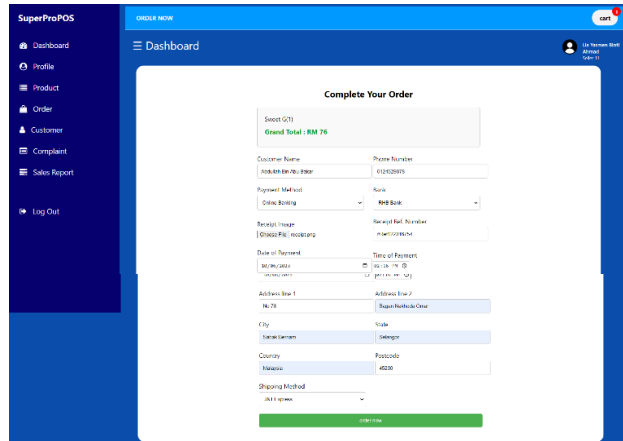


Figure 3.41 Check Out Form Order for Seller

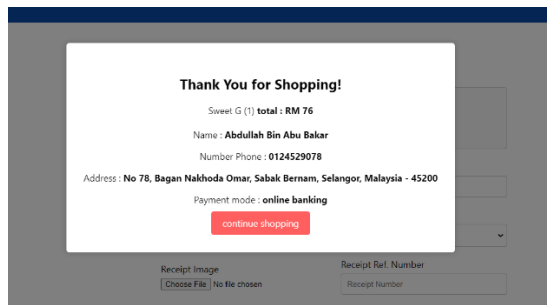


Figure 3.42 UI Successful Order Message for Seller

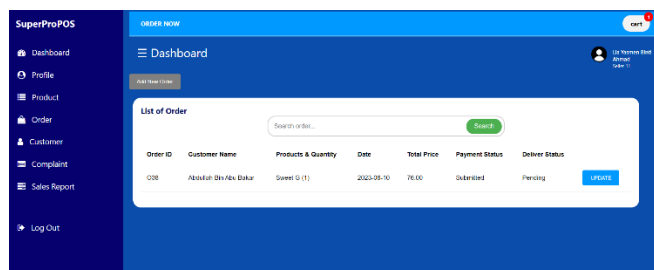


Figure 3.43 UI Order List for Seller

The seller also can add new customer from the order form if there are no existing customer there. The seller also can view and edit the customer list details. Figure 3.44 until figure 3.46 shows the interface for manage customer.

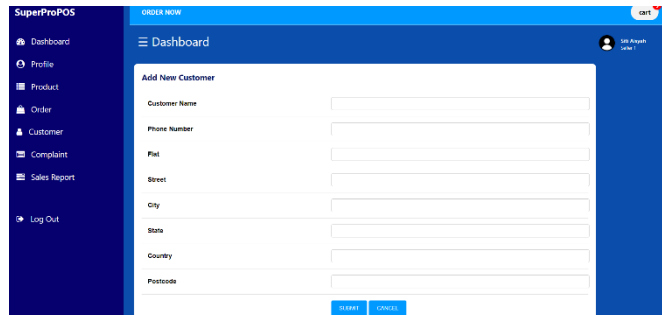


Figure 3.44 UI Add New Customer for Seller

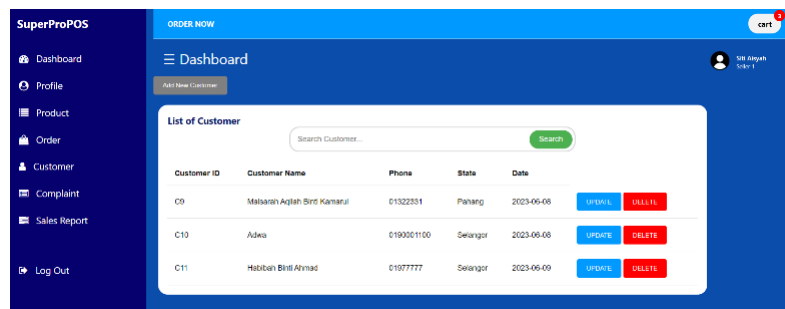


Figure 3.45 UI Customer List for Seller

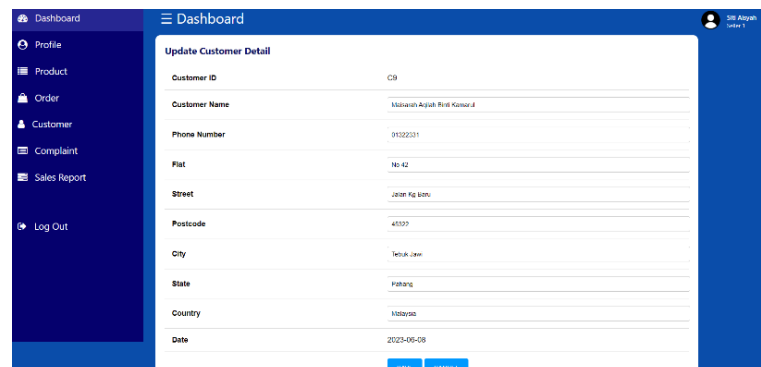


Figure 3.46 UI Edit Customer Detail for Seller

Then, seller also can manage complaint by add, view and edit the complaint. Figure 3.47 until figure 3.50 shows the complaint interface for seller.

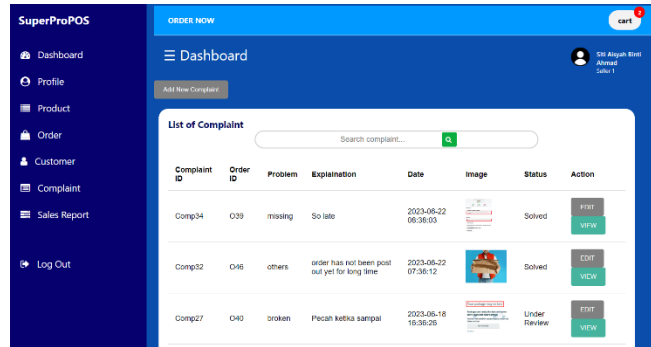


Figure 3.47 UI Complaint List for Seller

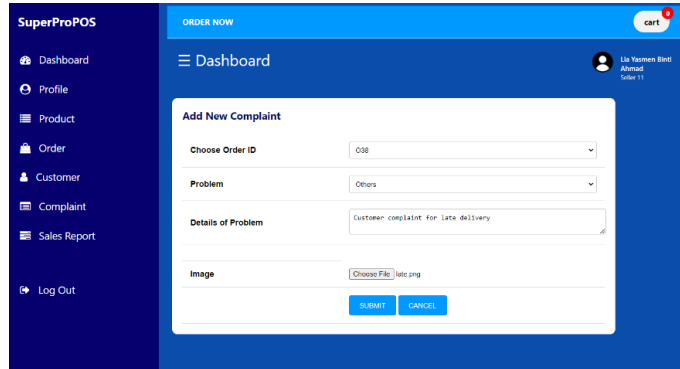


Figure 3.48 UI Add New Complaint for Seller

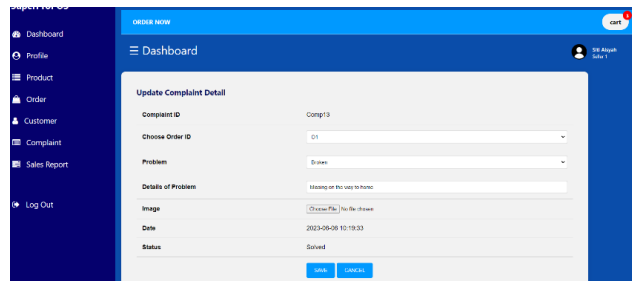


Figure 3.49 Edit Complaint for Seller

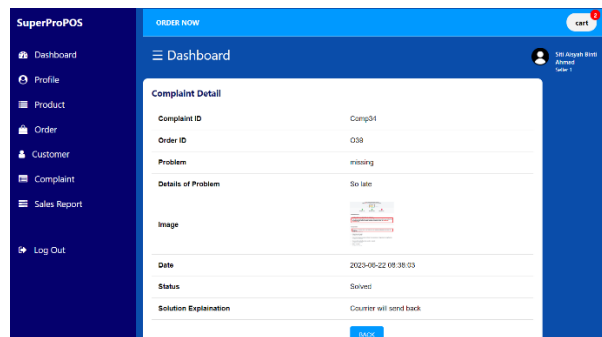


Figure 3.50 View Complaint for Seller



The seller also can update their profile in the menu profile. Figure 3.51 below shows the update profile page for seller.

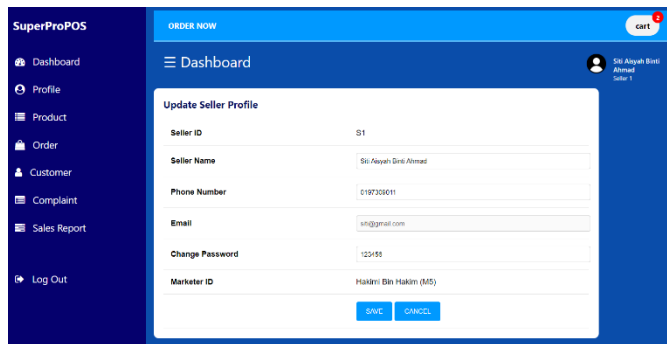


Figure 3.51 UI Update Seller Profile for Seller

Lastly, the seller can view the sales report. Figure 3.52 until 3.53 shows the sales report of seller.

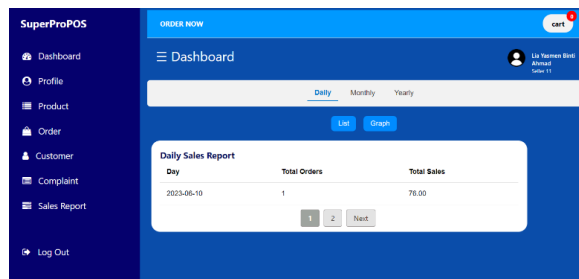


Figure 3.52 UI List Sales Report for Seller

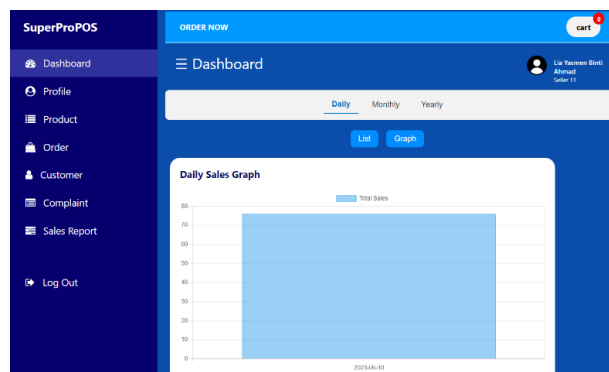


Figure 3.53 UI Graph Sales Report for Seller

### 3.6.3 Marketer Page

Firstly, the marketer needs to login to the system. Figure 3.54 shows the login page for marketer.

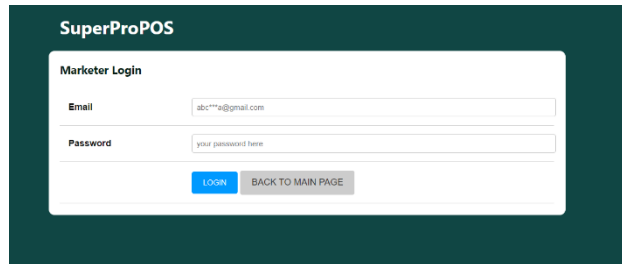


Figure 3.54 UI Login for Marketer

Once marketer successfully login to the system, the system displays the homepage of the system. Figure 3.55 shows the homepage of marketer.

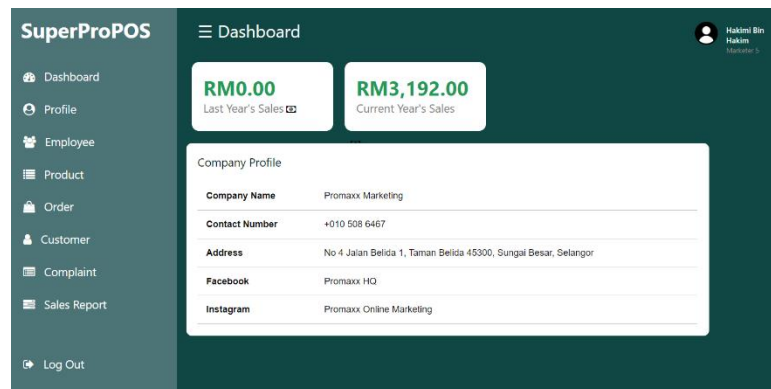


Figure 3.55 UI Homepage for Marketer

The marketer then is able to view the products list and details. Figure 3.56 and Figure 3.57 below shows the product menu for marketer.

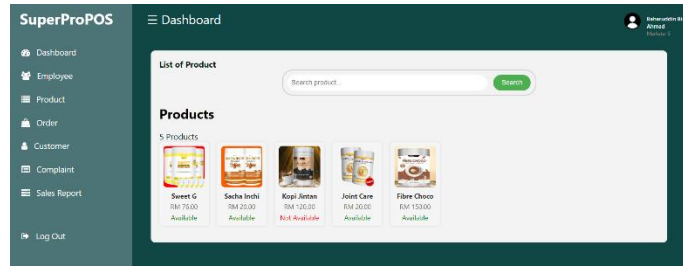


Figure 3.56 UI Product List for Marketer

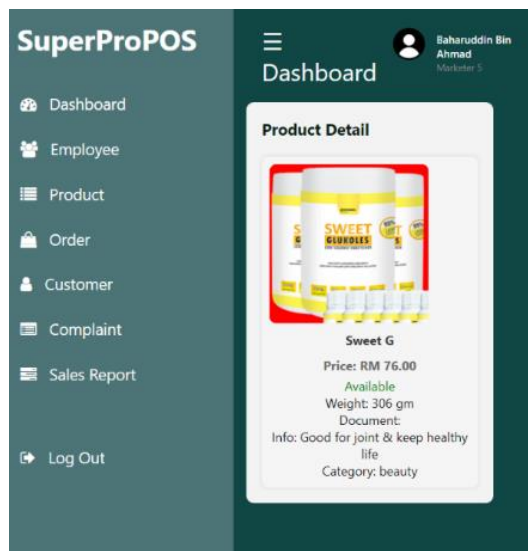


Figure 3.57 UI Product Detail for Marketer

The marketer also can view order list and order detail. Figure 3.58 and figure 3.59 below shows the interface of order list and order detail for marketer.

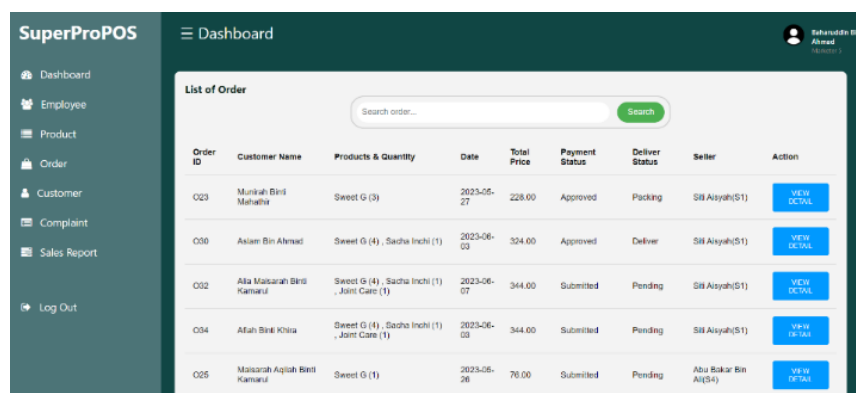


Figure 3.58 UI Order List for Marketer

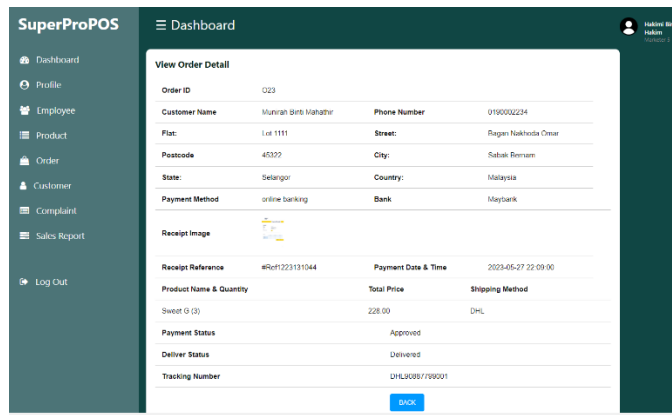


Figure 3.59 View Order Detail for Marketer

Marketer also can view the customer list and their details. Figure 3.60 and figure 3.61 shows the customer list for marketer.

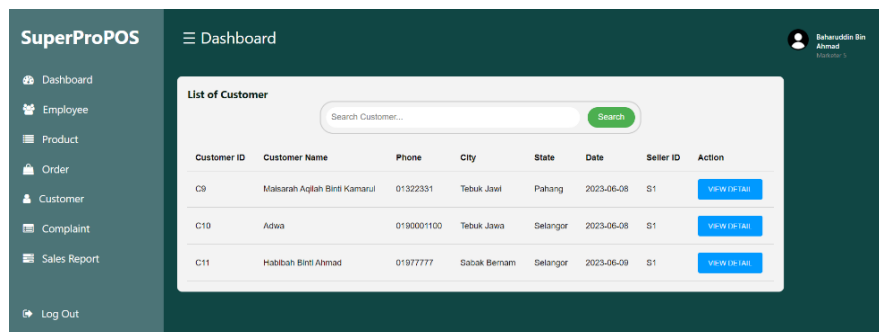


Figure 3.60 UI Customer List for Marketer

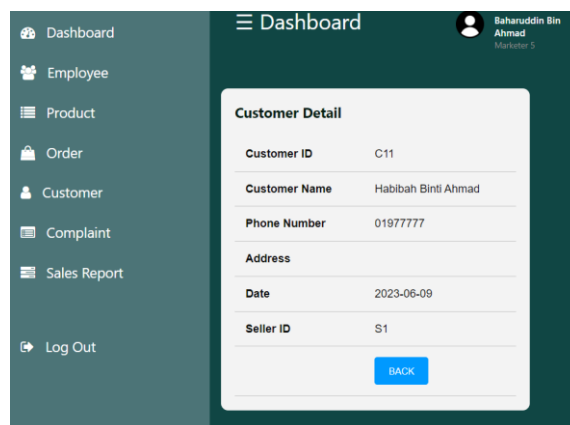


Figure 3.61 UI Customer Detail for Marketer

The marketer also can view the complaint list and details. Figure 3.62 and figure 3.63 shows the complaint list and complaint detail for marketer.

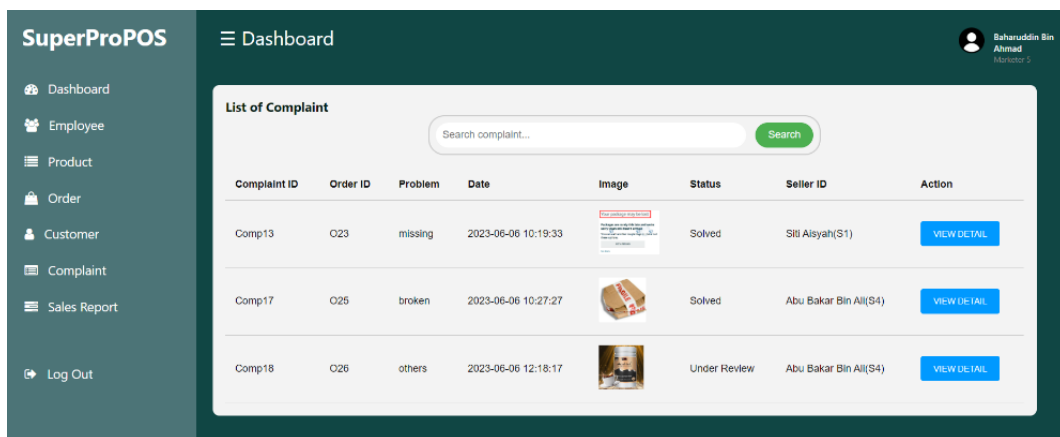


Figure 3.62 UI Complaint List for Marketer

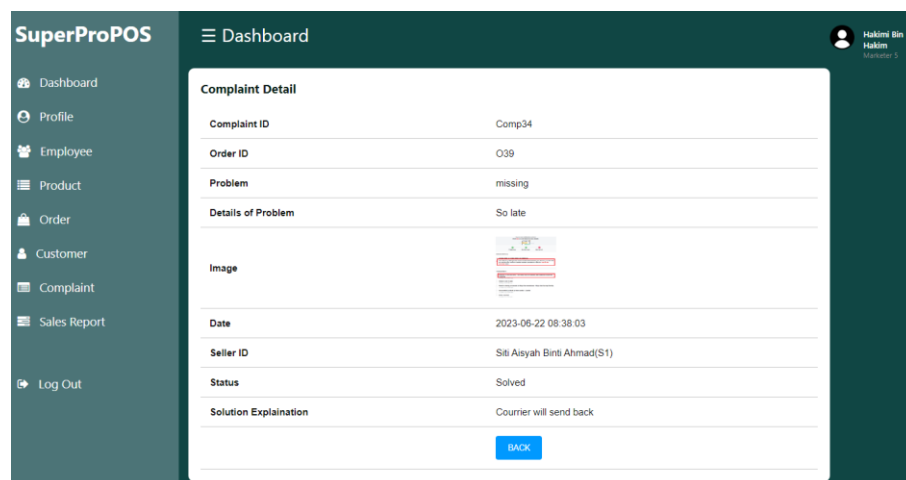


Figure 3.63 View Complaint Detail for Marketer

The marketer also can view the Seller List and their Performance. Figure 3.64 and Figure 3.65 shows the assigned seller for marketer.

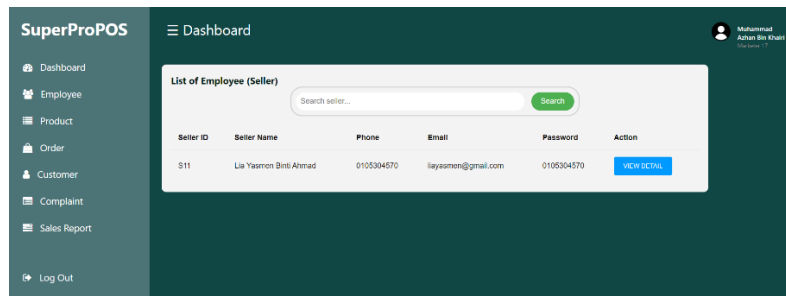


Figure 3.64 UI Seller List for Marketer

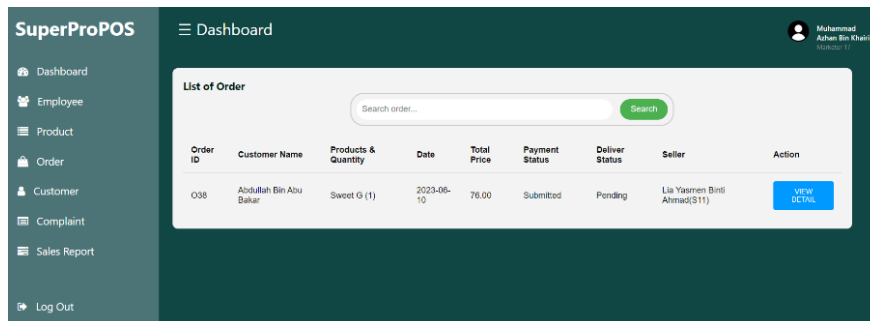


Figure 3.65 List Order of Seller for Marketer

Marketer also can update their profile. Figure 3.66 shows the interface of update profile for marketer.

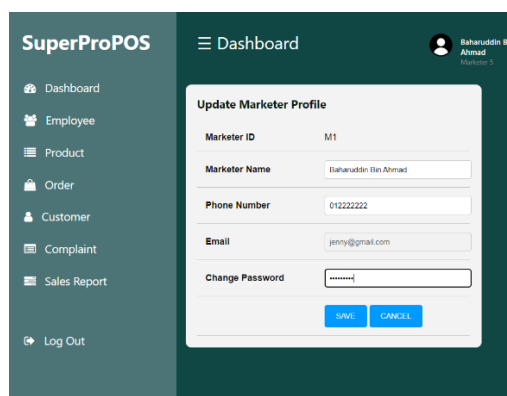


Figure 3.66 UI Update Marketer Profile for Marketer

Lastly, the marketer is able to view the generated report of the seller performance in the system. Figure 3.67 until figure 3.69 shows the sales report for the marketer.

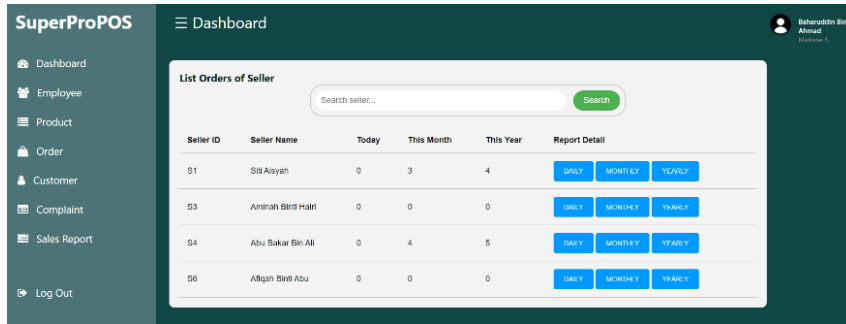


Figure 3.67 UI List of Seller Sales Report for Marketer

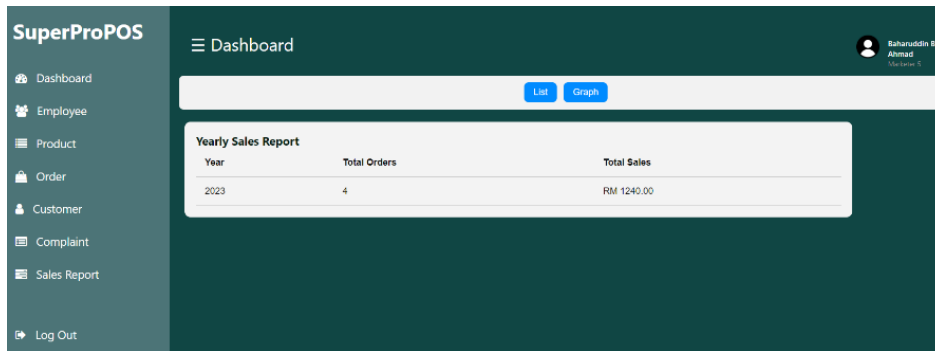


Figure 3.68 UI List Sales Report for Marketer

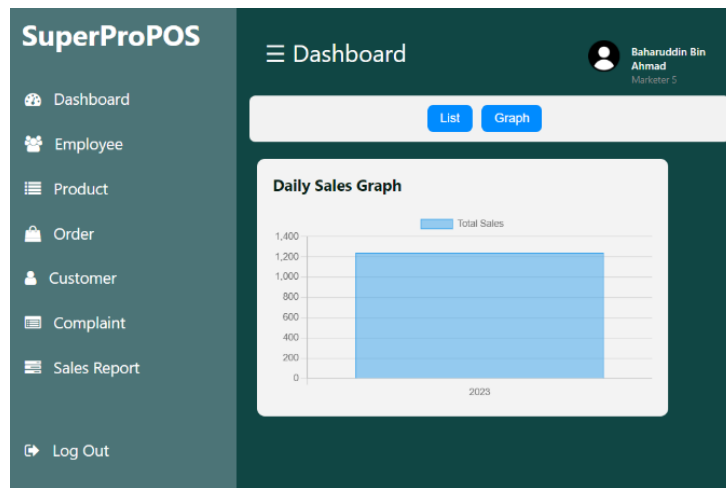


Figure 3.69 UI Graph Sales Report for Marketer

### 3.6.4 Warehouse Staff Page

Firstly, the warehouse staff needs to login to the system. Figure 3.70 shows the login interface for warehouse staff.

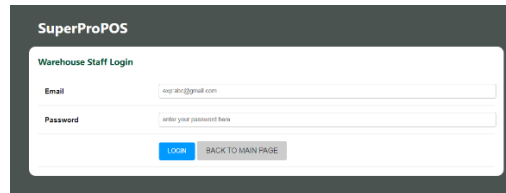


Figure 3.70 UI Login for Warehouse Staff

Once the warehouse staff successfully login to the system, the system then displays the homepage for warehouse staff. Figure 3.71 shows the homepage for warehouse staff.

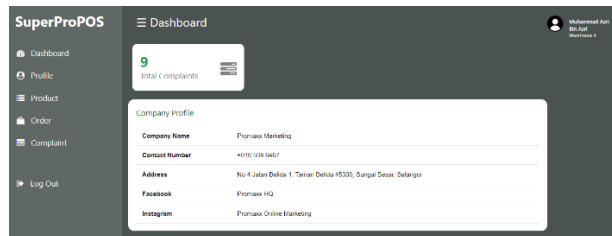


Figure 3.71 UI Homepage for Warehouse staff

The warehouse staff can go to the product list menu to view and update the stock of the products. Figure 3.72 and 3.73 shows the product list and product details.



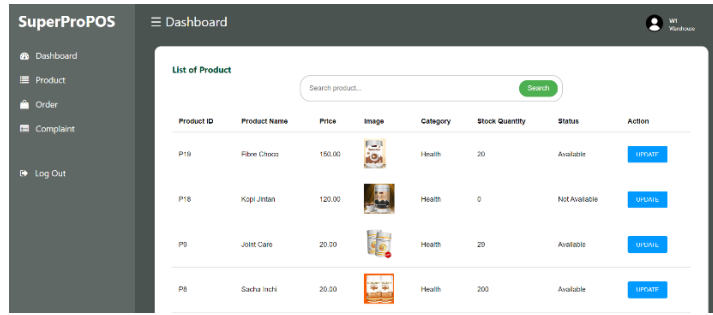


Figure 3.72 UI Product List for Warehouse staff

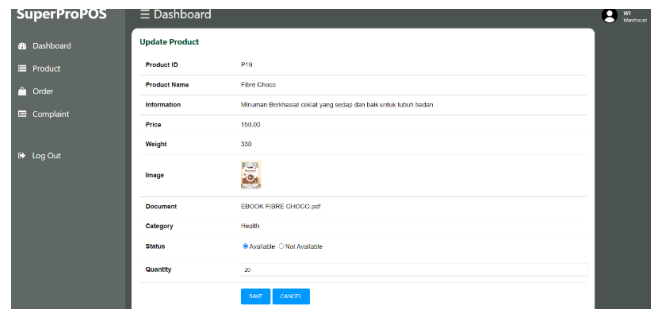


Figure 3.73 UI Update Product for Warehouse staff

The warehouse staff also can view and update the status and tracking number of the order list and print the order detail. Figure 3.74 and figure 3.75 shows the order list and detail for warehouse staff.

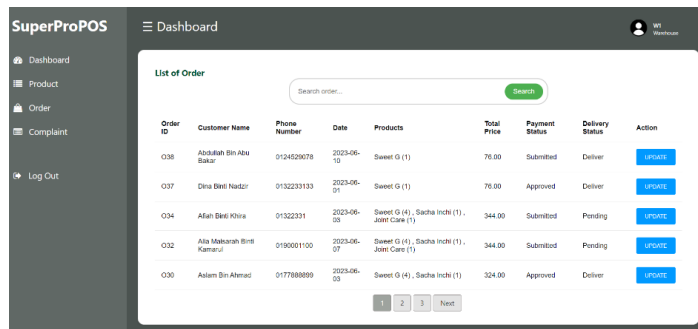


Figure 3.74 UI Order List for Warehouse staff

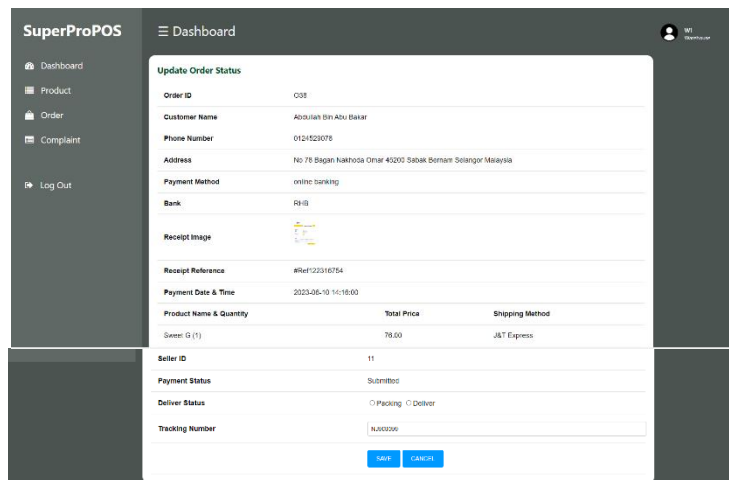


Figure 3.75 UI Order Detail for Warehouse staff

The warehouse staff can view and update the status of the complaint list and details. Figure 3.76 and 3.77 shows the complaint list and details for warehouse staff.

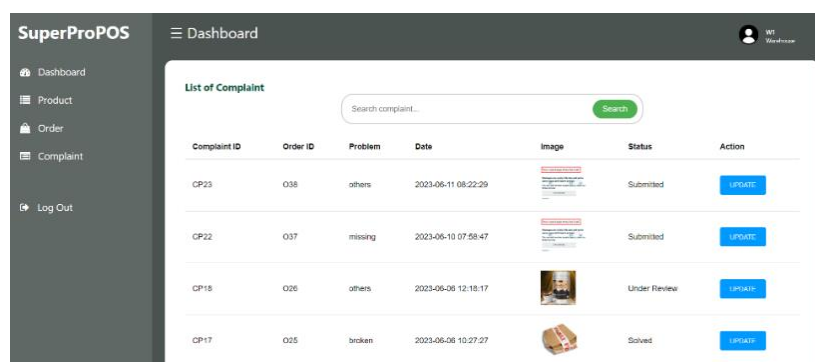


Figure 3.76 UI Complaint List for Warehouse staff

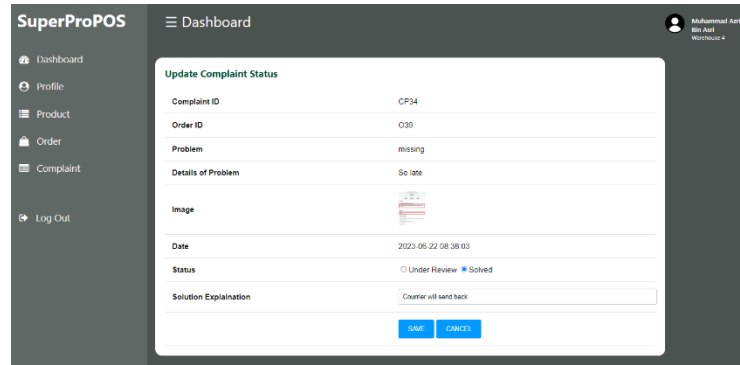


Figure 3.77 UI Complaint Detail for Warehouse staff

Lastly, the warehouse staff can update their profile. Figure 3.78 shows the interface of update profile for warehouse staff.

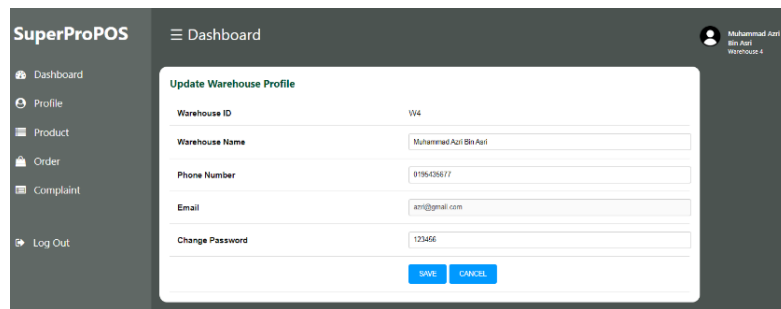


Figure 3.78 UI Update Warehouse staff Profile for Warehouse staff

### 3.7 Testing Plan

Testing strategy that is used to do the testing plan is by using the User Acceptance Testing (UAT). The UAT is done to test the condition of the SuperProPOS: Product Ordering System environment. Each function is tested to ensure that it is free error and produce expected result. Below in Table 3.13 shows the UAT form that can be test by few testers. The formal of UAT form also can be referred in Appendix C.

Table 3.13 UAT of SuperProPOS: Product Ordering System

NO	Module	Activities	Status	Comments
1.	Login	Login to system	Pass/Fail	
2.	Register Employee	Register seller, marketer and warehouse staff	Pass/Fail	
3.	Manage User Profile	View, delete and update admin, seller, marketer and warehouse profile	Pass/Fail	
4.	Manage Customer	View and update customer profile	Pass/Fail	
5.	Manage Product Detail	Add, view, delete and update product information	Pass/Fail	
6.	Manage Order	Add, view and update order information	Pass/Fail	
7.	Manage Complaint	Add, view and update complaint information	Pass/Fail	
8.	Generate Report	View sales report	Pass/Fail	

### **3.8 Potential Use of Proposed Solution**

During this time, a lot of business company run their business by hiring a worker to make sales for their business. Most of the business companies market their product by using the *Facebook Ads*, *Instagram Ads* and *Google Ads*. Marketer and Seller is needed to make sure that the business can run smoothly in dealing with customer from the *Facebook Ads*, *Instagram Ads* and *Google Ads*. So, the SuperProPOS: Product Ordering System is useful for this business company to manage their business online. The system allows the seller to make an order easily for customer, hence the marketer can easily access and view the performance of sales from their assigned seller. The admin also can add and update the product information to the system to let the seller make an order from the products. The complaint for order also helps the business company to improve and understand the seller's customer problem and enhance their business performance. Lastly, the sales report also generated by the system can help the business company aware about their business sales.

### **3.9 Gantt Chart**

The Gantt Chart was utilized to ensure that the project was completed within the designated timeline. The development of the SuperProPOS: Product Ordering System Gantt Chart commenced with requirement planning, where all project requirements were gathered. Subsequently, the user design phase detailed the system's design. Following that, the rapid construction phase involved the complete development of the system. Finally, the transition phase commenced after the development was finished, and the entire system was thoroughly tested.

For reference, the Gantt Chart figure of the SuperProPOS: Product Ordering System phase can be found in Appendix D.

## **CHAPTER 4**

### **IMPLEMENTATION, RESULT AND DISCUSSION**

#### **4.1 Introduction**

Chapter 4 discusses the development, implementation, and testing results of the SuperProPOS: Product Ordering System, which was developed using the waterfall methodology. This web-based system is designed for a business company involved in product sales and caters to various user roles such as admin, marketer, seller, and warehouse staff, who collectively manage customer orders. The system requirements were gathered, and the system model was designed by incorporating insights from existing research. The development process was carried out and the system was tested extensively on the client side. Additionally, the system was successfully deployed to the FK INDAH server. Thorough testing was conducted to identify and address any potential errors or issues promptly.

#### **4.2 Implementation Process**

The implementation process of the SuperProPOS: Product Ordering System discusses about the developer environment and the system functionality in detail.

##### **4.2.1 Developer Environment**

The developer environment was setup first in order to allow the process of developing the SuperProPOS: Product Ordering System to run smoothly. The setup is being done in localhost of the developer laptop in the folder of xampp/htdocs/superpropos. The folder for the SuperProPOS: Product Ordering System is called “superpropos”. Inside the superpropos folder contained the ApplicationLayer

folder, BusinessServiceLayer folder, css folder, images folder and info folder. Figure 4.1 and figure 4.2 below shows the folder of detail of superpropos inside the xampp/htdocs folder.

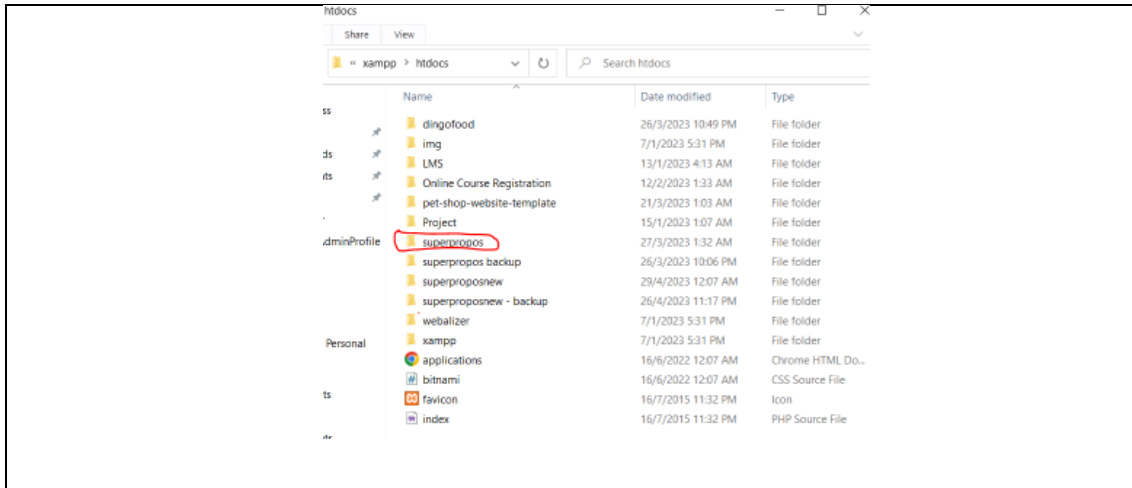


Figure 4.1 Folder of superpropos Inside xampp/htdocs (Localhost)

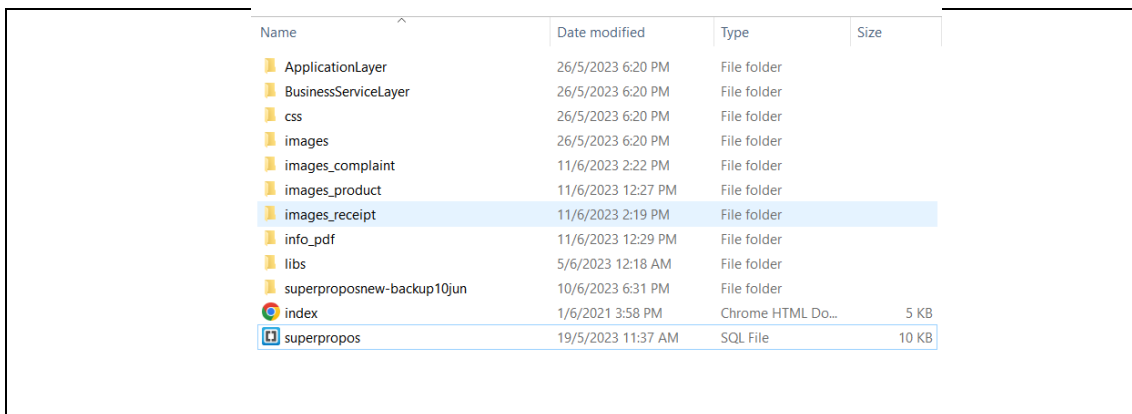


Figure 4.2 Details Folder Inside of superpropos Folder

The ApplicationLayer and BusinessServiceLayer is created in order to apply the MVC method to the SuperProPOS: Product Ordering System. ApplicationLayer is specified for the View of the MVC, in other words, it is folder for the code of the interface of the system. While the BusinessServiceLayer is specified for the Model and Controller of the system.



The ApplicationLayer folder contain a few folders that is being categorised based on users' interfaces. The folder is being named as ManageAdmin which is for Admin interfaces, ManageMarketer which is for the Marketer interfaces, ManageSeller which is for the Seller interfaces and ManageWarehouse which is for the Warehouse staff interfaces. Figure 4.3 below shows the detail of the ApplicationLayer folder.

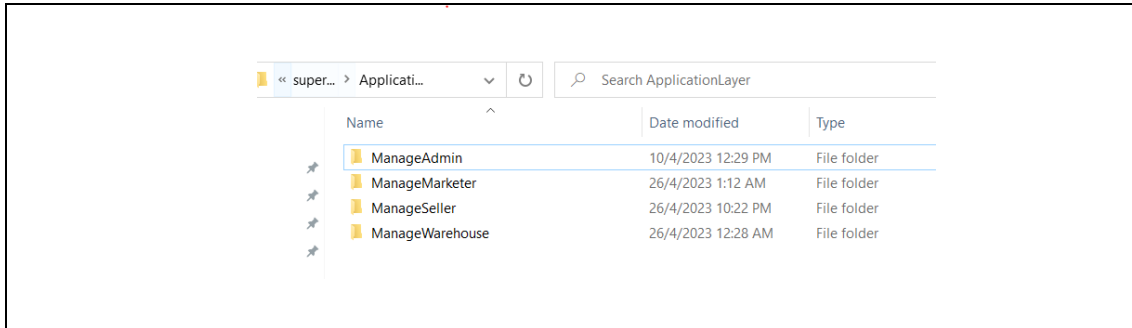


Figure 4.3 Details of ApplicationLayer Folder

All of the code for the interface of admin in the SuperProPOS: Product Ordering System is developed under the ManageAdmin folder. Figure 4.4 below shows the code file of the interface for admin.

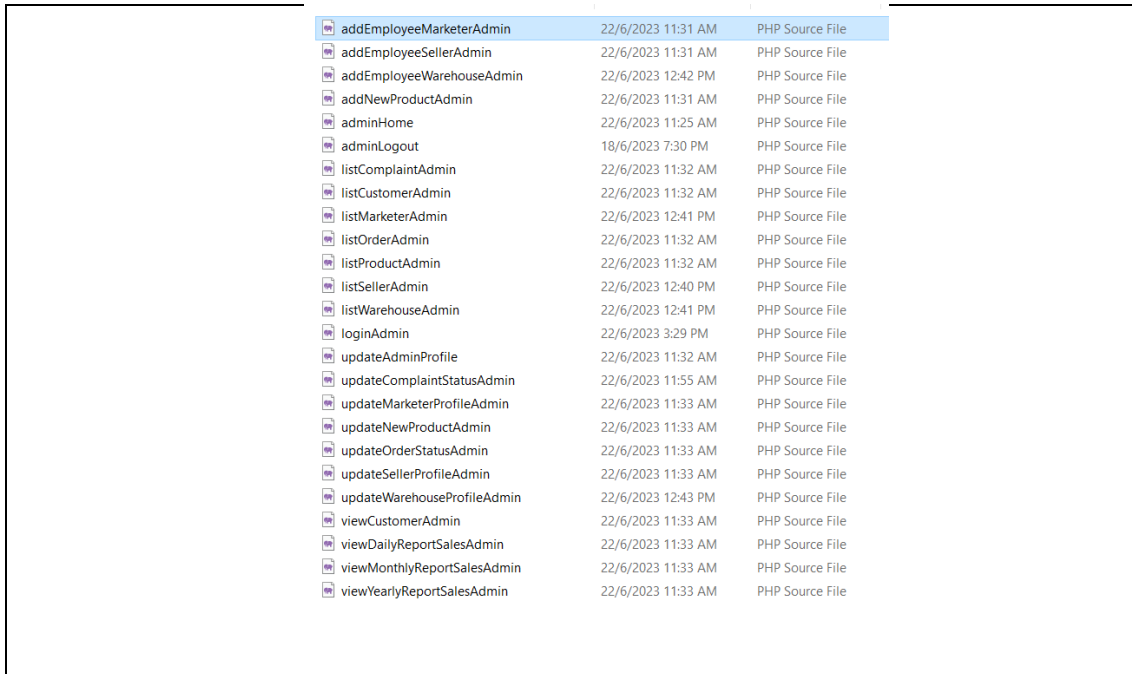


Figure 4.4 Details of ManageAdmin Folder

Next, the code for the interface of marketer in the SuperProPOS: Product Ordering System is developed under the ManageMarketer folder. Figure 4.5 below shows the code file of the interface for marketer.



















	listComplaintMarketer	22/6/2023 12:28 PM	PHP Source File
	listCustomerMarketer	22/6/2023 11:34 AM	PHP Source File
	listOrderMarketer	22/6/2023 11:34 AM	PHP Source File
	listProductMarketer	22/6/2023 11:34 AM	PHP Source File
	listSalesMarketer	22/6/2023 11:35 AM	PHP Source File
	listSellerMarketer	22/6/2023 11:35 AM	PHP Source File
	marketerHome	22/6/2023 11:35 AM	PHP Source File
	marketerLogin	22/6/2023 3:29 PM	PHP Source File
	marketerLogout	18/6/2023 8:19 PM	PHP Source File
	updateProfileMarketer	22/6/2023 11:35 AM	PHP Source File
	viewComplaintStatusMarketer	22/6/2023 11:58 AM	PHP Source File
	viewCustomerMarketer	22/6/2023 11:35 AM	PHP Source File
	viewDailySaleMarketer	22/6/2023 11:35 AM	PHP Source File
	viewMonthlySaleMarketer	22/6/2023 11:35 AM	PHP Source File
	viewOrderStatusMarketer	22/6/2023 11:35 AM	PHP Source File
	viewProductMarketer	22/6/2023 11:35 AM	PHP Source File
	viewSellerProfileMarketer	22/6/2023 11:36 AM	PHP Source File
	viewYearlySaleMarketer	22/6/2023 11:37 AM	PHP Source File

Figure 4.5 Details of ManageMarketer Folder

Then, the code for the interface of seller in the SuperProPOS: Product Ordering System is developed under the ManageSeller folder. Figure 4.6 below shows the code file of the interface for seller.

























Name	Date modified	Type	
	addComplaintSeller	22/6/2023 11:52 AM	PHP Source File
	addCustomerSeller	22/6/2023 11:38 AM	PHP Source File
	addOrderSeller	22/6/2023 11:38 AM	PHP Source File
	cartOrderSeller	22/6/2023 11:38 AM	PHP Source File
	cartSeller	22/6/2023 11:38 AM	PHP Source File
	checkoutSeller	22/6/2023 11:38 AM	PHP Source File
	header	5/6/2023 7:31 PM	PHP Source File
	listComplaintSeller	22/6/2023 12:23 PM	PHP Source File
	listCustomerSeller	22/6/2023 11:38 AM	PHP Source File
	listOrderSeller	22/6/2023 11:38 AM	PHP Source File
	listProductSeller	22/6/2023 11:38 AM	PHP Source File
	sellerHome	22/6/2023 11:39 AM	PHP Source File
	sellerLogin	22/6/2023 3:29 PM	PHP Source File
	sellerLogout	18/6/2023 11:40 PM	PHP Source File
	updateComplaintStatusSeller	22/6/2023 12:20 PM	PHP Source File
	updateCustomerSeller	22/6/2023 11:39 AM	PHP Source File
	updateOrderStatusSeller	22/6/2023 2:23 PM	PHP Source File
	updateProfileSeller	22/6/2023 11:39 AM	PHP Source File
	viewComplaintStatusSeller	22/6/2023 12:21 PM	PHP Source File
	viewDailyReportSeller	22/6/2023 11:39 AM	PHP Source File
	viewMonthlyReportSeller	22/6/2023 11:40 AM	PHP Source File
	viewOrderDetailSeller	22/6/2023 11:40 AM	PHP Source File
	viewProductSeller	22/6/2023 11:40 AM	PHP Source File
	viewYearlyReportSeller	22/6/2023 11:40 AM	PHP Source File

Figure 4.6 Details of ManageSeller Folder

Later, the code for the interface of warehouse staff in the SuperProPOS: Product Ordering System is developed under the ManageWarehouse folder. Figure 4.7 below shows the code file of the interface for warehouse staff.











Name	Date modified	Type
 listComplaintWh	22/6/2023 11:41 AM	PHP Source File
 listOrderWh	22/6/2023 11:41 AM	PHP Source File
 listProductWh	22/6/2023 11:41 AM	PHP Source File
 updateComplaintStatusWh	22/6/2023 11:56 AM	PHP Source File
 updateOrderStatusWh	22/6/2023 11:41 AM	PHP Source File
 updateProductWh	22/6/2023 11:41 AM	PHP Source File
 updateProfileWh	22/6/2023 11:41 AM	PHP Source File
 whHome	22/6/2023 11:41 AM	PHP Source File
 whLogin	22/6/2023 3:29 PM	PHP Source File
 whLogout	18/6/2023 7:08 PM	PHP Source File

Figure 4.7 Details of ManageWarehouse Folder

As for the BusinessServiceLayer folder, it contains a few folders that is being categorised based on controller, model and library of the system. The folder is being named as “controller” which is for the controller to handle all users input and updating the model and view of the system accordingly. While another folder is named “model” which is for the model to helps the system to manage the system state and consistency of that data in the system. Another folder is called “libs” which contained the file of code for connecting to the database. Figure 4.8 below shows the details inside of the BusinessServiceLayer.




Name	Date modified	Type
 controller	1/5/2023 12:16 AM	File folder
 libs	27/4/2023 11:47 PM	File folder
 model	1/5/2023 12:17 AM	File folder

Figure 4.8 Details of BusinessServiceLayer Folder

Figure 4.9 below shows the details of the controller inside the “controller” folder.

Name	Date modified	Type
adminController	22/6/2023 2:36 PM	PHP Source File
complaintController	22/6/2023 2:59 PM	PHP Source File
customerController	22/6/2023 2:59 PM	PHP Source File
marketersController	22/6/2023 4:26 PM	PHP Source File
orderController	22/6/2023 3:01 PM	PHP Source File
productController	22/6/2023 3:02 PM	PHP Source File
reportController	22/6/2023 3:13 PM	PHP Source File
sellerController	22/6/2023 3:07 PM	PHP Source File
warehouseController	22/6/2023 3:09 PM	PHP Source File

Figure 4.9 Details of Controller Folder

Figure 4.10 below shows the details of the model inside the “model” folder.

Name	Date modified	Type
adminModel	15/1/2023 11:56 AM	PHP Source File
complaintModel	28/4/2023 12:47 AM	PHP Source File
customerModel	29/4/2023 1:33 AM	PHP Source File
customerProfileModel	11/1/2023 12:38 AM	PHP Source File
marketerModel	28/4/2023 12:42 AM	PHP Source File
orderModel	27/1/2023 2:45 AM	PHP Source File
productModel	28/4/2023 12:47 AM	PHP Source File
reportModel	11/1/2023 12:38 AM	PHP Source File
sellerModel	27/4/2023 3:02 AM	PHP Source File
warehouseModel	28/4/2023 12:47 AM	PHP Source File

Figure 4.10 Details of Model Folder

Figure 4.11 below shows the details of the library inside the “libs” folder.

Name	Date modified	Type
database	27/4/2023 11:46 PM	PHP Source File

Figure 4.11 Details of Library Folder

The SuperProPOS: Product Ordering System is using MySQL database as the database of the system. It is connected to the system by using the XAMPP and use the phpMyAdmin to manage the database. Figure 4.12 below shows the connected MySQL database for the SuperProPOS: Product Ordering System called “superpropos” database.

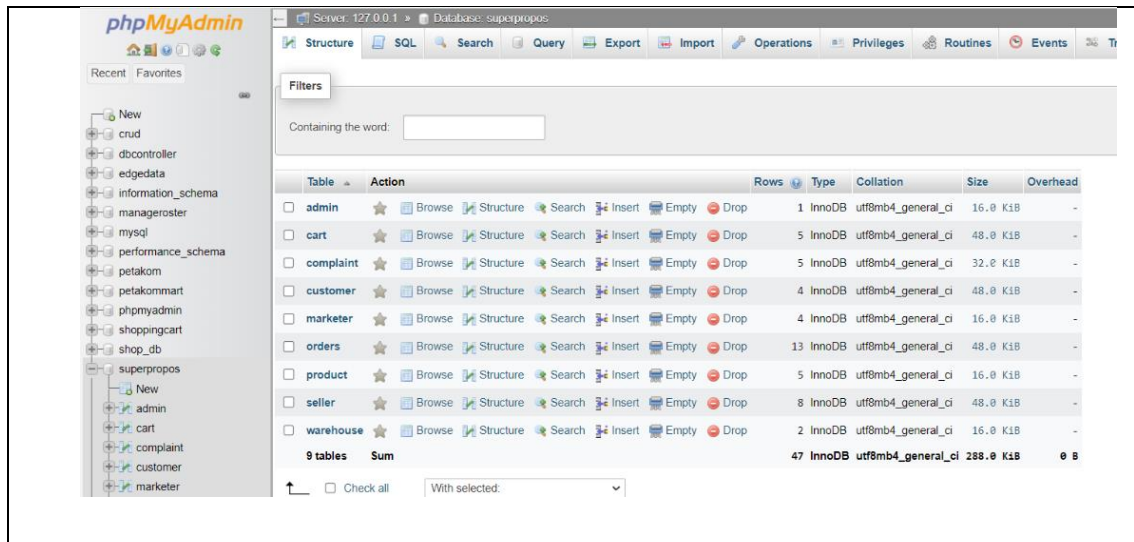


Figure 4.12 Database of SuperProPOS: Product Ordering System

The language that was used in developing this system are HTML, PHP, CSS and Javascript. The code was written by using Visual Studio Code IDE. The system have been deployed to the FK INDAH server and can be referred to this link: <https://indah.ump.edu.my/CB20009/superproposnew/ApplicationLayer/indexMain.php>

## 4.2.2 System Functionality

The SuperProPOS: Product Ordering System functionality consist of login, register employee, manage user profile, manage product, manage customer, manage order, manage complaint and generate report. The detail of the functionality is explained in the subchapter 4.2.2.1 until subchapter 4.2.2.8 below.

### 4.2.2.1 Login

The login function allows the admin, seller, marketer and warehouse staff to login and gain access to the SuperProPOS: Product Ordering System. Figure 4.13 until figure 4.16 below shows the login page for admin, seller, marketer and warehouse staff.

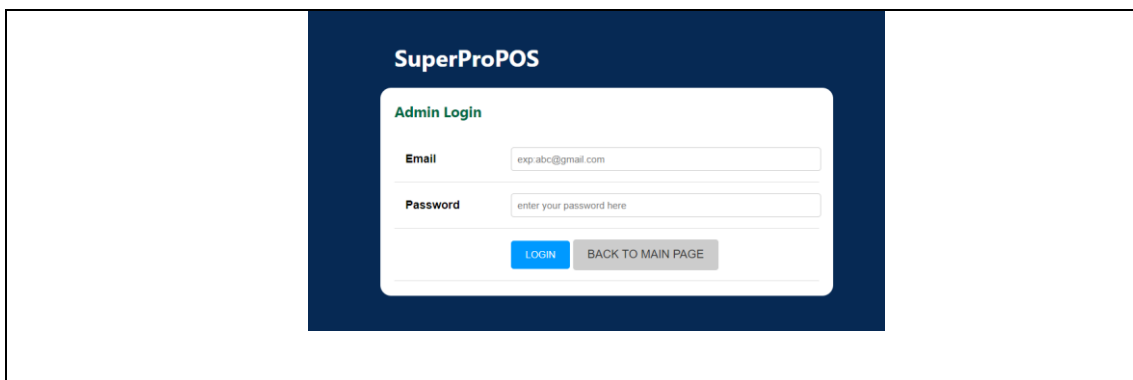
The screenshot shows the 'Admin Login' page of the SuperProPOS system. The page has a dark blue header with the 'SuperProPOS' logo. Below the header is a white login form with a dark blue border. The form contains two input fields: 'Email' with the placeholder 'exp.abc@gmail.com' and 'Password' with the placeholder 'enter your password here'. At the bottom of the form are two buttons: a blue 'LOGIN' button and a grey 'BACK TO MAIN PAGE' button.

Figure 4.13 Login Page for Admin

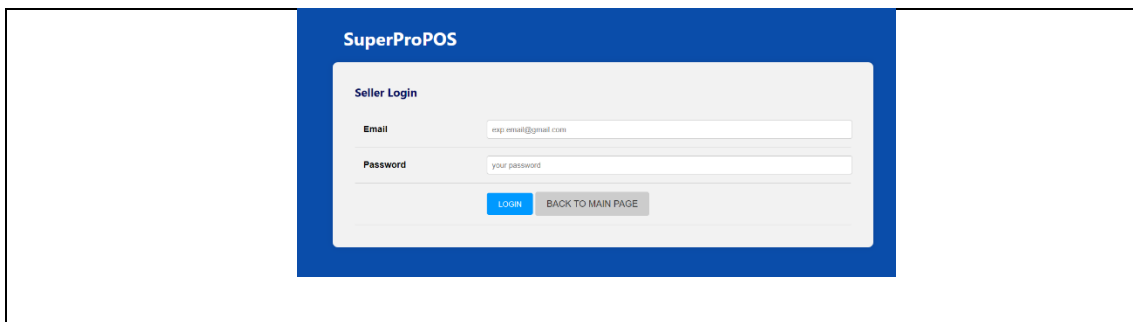
The screenshot shows the 'Seller Login' page of the SuperProPOS system. The page has a blue header with the 'SuperProPOS' logo. Below the header is a white login form with a blue border. The form contains two input fields: 'Email' with the placeholder 'esp\_email@gmail.com' and 'Password' with the placeholder 'your password'. At the bottom of the form are two buttons: a blue 'LOGIN' button and a grey 'BACK TO MAIN PAGE' button.

Figure 4.14 Login Page for Admin

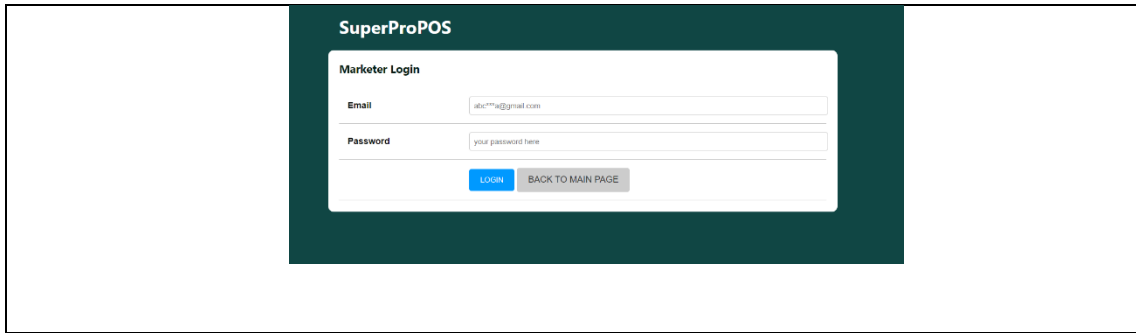


Figure 4.15 Login Page for Marketer

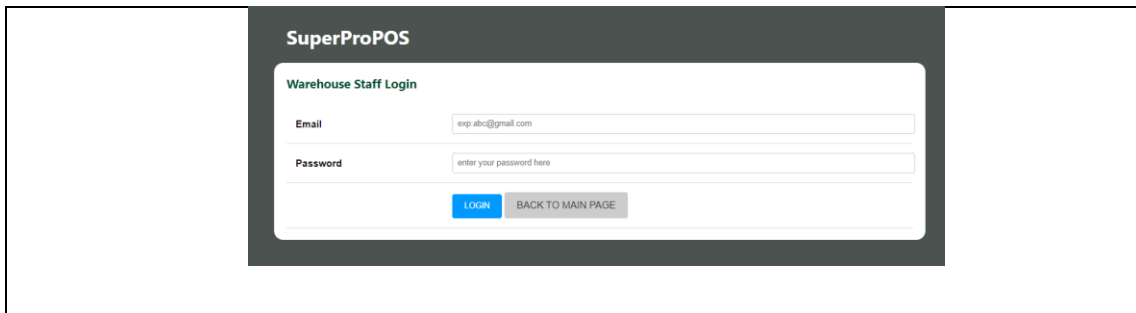


Figure 4.16 Login Page for Warehouse staff

#### 4.2.2.2 Register Employee

Register Employee functionality was handled by Admin in order to register the seller, marketer and warehouse staff to the SuperProPOS: Product Ordering System. The admin needs to choose Employee menu, then the system shows the list of Employee for seller first and the admin can choose the position of employee like seller, marketer or warehouse staff. The admin needs to click the Register New Seller in Seller menu, Register Marketer in Marketer menu and Register Warehouse staff in Warehouse staff menu in order to allow the admin to register seller, marketer and warehouse staff. Figure 4.17 until figure 4.20 shows the pages for register employee in the system.

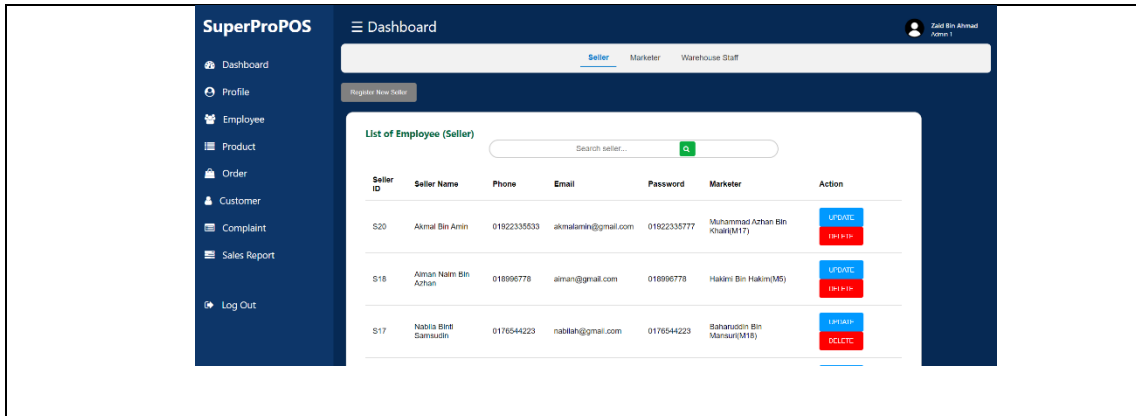


Figure 4.17 List of Seller for Admin

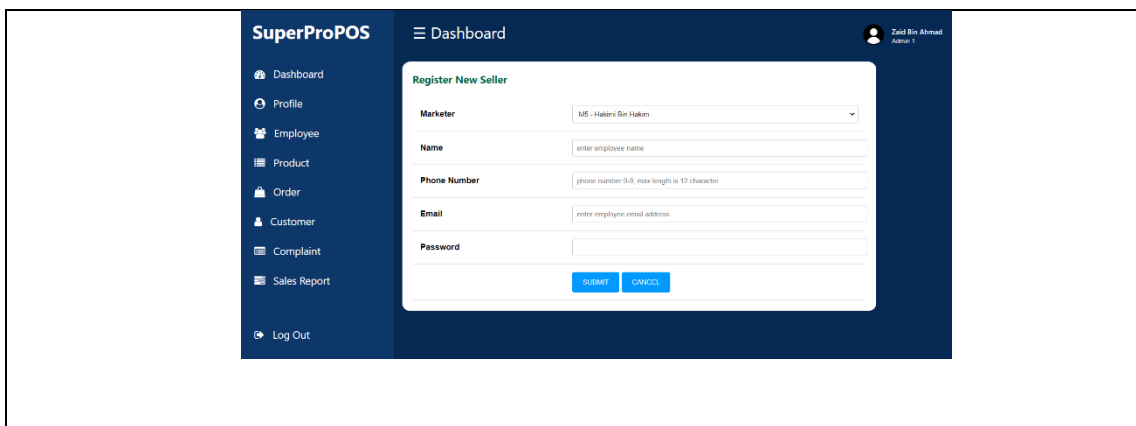


Figure 4.18 Register New Seller Page for Admin

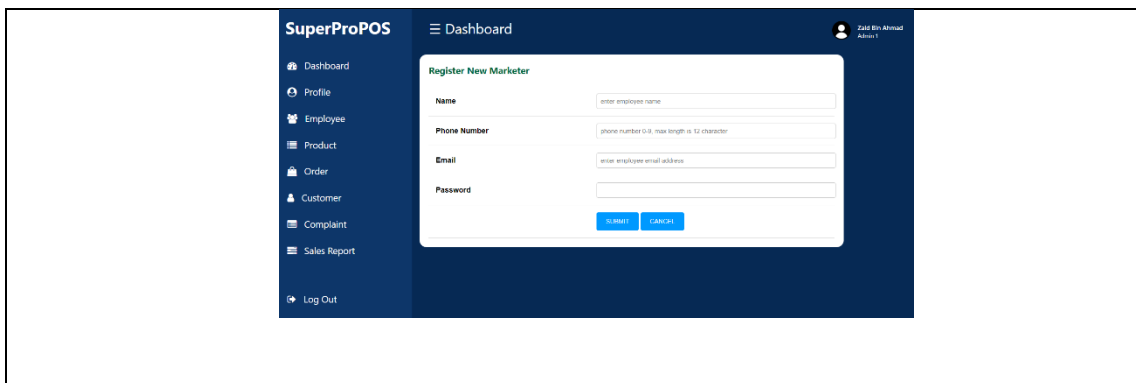


Figure 4.19 Register New Marketer Page for Admin



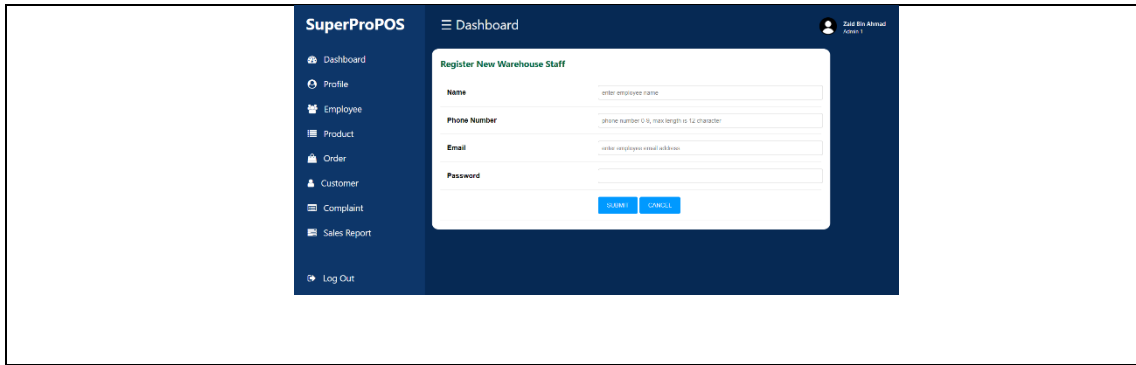


Figure 4.20 Register New Warehouse staff Page for Admin

In order to allow the register employee module to be functioning well, controller of each user like seller, marketer and warehouse staff are needed. The controller code was stated below.

This is sellerController that allow the admin to add, update and delete the seller to be functioning in the system.

```
<?php

include("../BusinessServiceLayer/libs/database.php");

// add seller profile
class sellerController {

public function addSeller() {
if (isset ($_POST['add'])){
$S_Name=$_POST['S_Name'];
$S_PhoneNum=$_POST['S_PhoneNum'];
$S_Email=$_POST['S_Email'];
$S_Password=$_POST['S_Password'];
$Marketer_Id=$_POST['Marketer_Id'];

// Additional code to sanitize and validate inputs

$query = "INSERT INTO seller VALUES('', '$S_Name', '$S_PhoneNum',
'$S_Email', '$S_Password', '$Marketer_Id')";
$conn = mysqli_connect('localhost', 'root', '', 'superpropos');
```

```

// Check if email already exists in the database
$checkQuery = "SELECT * FROM seller WHERE S_Email='$S_Email'";
$result = mysqli_query($conn, $checkQuery);

if (mysqli_num_rows($result) > 0) {
    echo "Error: Email already exists";
    return; // Exit the function if email already exists
}

if (mysqli_query($conn, $query)) {
    echo "<script type='text/javascript'>
window.location='../ApplicationLayer/ManageAdminProfile/listSellerAdmin.php' </script>";
} else {
    echo "Error: " . $query . "<br>" . mysqli_error($conn);
}
}
}

//update seller profile
public function updateSeller() {
    if (isset($_POST['update'])) {
        $Seller_Id = $_POST['Seller_Id'];
        $S_Name = $_POST['S_Name'];
        $S_PhoneNum = $_POST['S_PhoneNum'];
        $S_Email = $_POST['S_Email'];
        $S_Password = $_POST['S_Password'];
        $Marketer_Id= $_POST['Marketer_Id'];

        $query = "UPDATE seller SET S_Name='$S_Name',
S_PhoneNum='$S_PhoneNum', S_Email='$S_Email', S_Password='$S_Password',
Marketer_Id='$Marketer_Id' WHERE Seller_Id='$Seller_Id'";
        $conn = mysqli_connect('localhost', 'root', '',
'superpropos');

        if (mysqli_query($conn, $query)) {
            echo "<script type='text/javascript'>
window.location='../ApplicationLayer/ManageAdminProfile/listSellerAdmin.php' </script>";
        } else {
            echo "Error: " . $query . "<br>" . mysqli_error($conn);
        }
    }
}

```

```

    }
}

//delete seller
public function deleteSeller($Seller_Id) {
    $conn = mysqli_connect('localhost', 'root', '', 'superpropos');

    if (isset($_POST['delete'])){
        $Seller_Id = mysqli_real_escape_string($conn, $Seller_Id);

        $query = "DELETE FROM seller WHERE Seller_Id='$Seller_Id'";

        if (mysqli_query($conn, $query)) {
            echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageAdminProfile/listSellerAdmin.php' </script>";
        } else {
            echo "Error: " . $query . "<br>" . mysqli_error($conn);
        }

        mysqli_close($conn);
    }
}

}
}

$seller = new sellerController();
if(isset($_POST['update'])){
    $seller->updateSeller();
}

if(isset($_POST['add'])){
    $seller->addSeller();
}

if (isset($_POST['delete'])) {
    $Seller_Id = $_POST['Seller_Id'];
    $seller->deleteSeller($Seller_Id);
}

?>

```

This is marketersController that allow the admin to add, update and delete the marketer to be functioning in the system.

```
<?php

include("../../BusinessServiceLayer/libs/database.php");

// add marketer profile
class marketersController {

public function addMarketer() {
if (isset ($_POST['add'])){
$M_Name=$_POST['M_Name'];
$M_PhoneNum=$_POST['M_PhoneNum'];
$M_Email=$_POST['M_Email'];
$M_Password=$_POST['M_Password'];

$query = "INSERT INTO marketer
VALUES('','$M_Name','$M_PhoneNum','$M_Email','$M_Password')";
$conn = mysqli_connect('localhost', 'root', '', 'superpropos');

if (mysqli_query($conn, $query)) {

echo "<script type='text/javascript'>
window.location='../../ApplicationLayer/ManageAdminProfile/listMarketer
Admin.php' </script>";

} else {
echo "Error: " . $query . "<br>" . mysqli_error($conn);
}
}
}

//update marketer profile
public function updateMarketer() {
if (isset ($_POST['update'])){
$Marketer_Id=$_POST['Marketer_Id'];
$M_Name=$_POST['M_Name'];
$M_PhoneNum=$_POST['M_PhoneNum'];
$M_Email=$_POST['M_Email'];
```

```

    $M_Password=$_POST['M_Password'];

    $query = "UPDATE marketer SET M_Name='$M_Name',
M_PhoneNum='$M_PhoneNum', M_Email='$M_Email', M_Password='$M_Password'
WHERE Marketer_Id='$Marketer_Id'";
    $conn = mysqli_connect('localhost', 'root', '', 'superpropos');
    if (mysqli_query($conn, $query)) {

        echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageAdminProfile/listMarketer
Admin.php' </script>";

    } else {
        echo "Error: " . $query . "<br>" . mysqli_error($conn);
    }
}

//delete marketer
public function deleteMarketer($Marketer_Id) {
    $conn = mysqli_connect('localhost', 'root', '', 'superpropos');

    if (isset ($_POST['delete'])){
        $Marketer_Id = mysqli_real_escape_string($conn, $Marketer_Id);

        $query = "DELETE FROM marketer WHERE
Marketer_Id='$Marketer_Id'";

        if (mysqli_query($conn, $query)) {
            echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageAdminProfile/listMarketer
Admin.php' </script>";
        } else {
            echo "Error: " . $query . "<br>" . mysqli_error($conn);
        }

        mysqli_close($conn);
    }
}
}

```

```

}

$marketer = new marketersController();
if(isset($_POST['update'])){
    $marketer->updateMarketer();
}

if(isset($_POST['add'])){
    $marketer->addMarketer();
}

if (isset($_POST['delete'])) {
    $Marketer_Id = $_POST['Marketer_Id'];
    $marketer->deleteMarketer($Marketer_Id);
}

?>

```

This is warehouse Controller that allow the admin to add, update and delete the warehouse staff to be functioning in the system.

```

<?php

include("../BusinessServiceLayer/libs/database.php");

// add warehouse staff profile
class warehouse staffController {

public function addWarehouse staff() {
if (isset ($_POST['add'])){
$W_Name=$_POST['W_Name'];
$W_PhoneNum=$_POST['W_PhoneNum'];
$W_Email=$_POST['W_Email'];
$W_Password=$_POST['W_Password'];

$query = "INSERT INTO warehouse staff
VALUES('','$W_Name','$W_PhoneNum','$W_Email','$W_Password)";
$conn = mysqli_connect('localhost', 'root', '', 'superpropos');

```

```

if (mysqli_query($conn, $query)) {

    echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageAdminProfile/listWarehous
e staffAdmin.php' </script>";

} else {
    echo "Error: " . $query . "<br>" . mysqli_error($conn);
}
}

//update warehouse staff profile
public function updateWarehouse staff() {
if (isset ($_POST['update'])){
    $Warehouse staff_Id=$_POST['Warehouse staff_Id'];
    $W_Name=$_POST['W_Name'];
    $W_PhoneNum=$_POST['W_PhoneNum'];
    $W_Email=$_POST['W_Email'];
    $W_Password=$_POST['W_Password'];

    $query = "UPDATE warehouse staff SET W_Name='$W_Name',
W_PhoneNum='$W_PhoneNum', W_Email='$W_Email', W_Password='$W_Password'
WHERE Warehouse staff_Id='$Warehouse staff_Id'";
    $conn = mysqli_connect('localhost', 'root', '', 'superpropos');
    if (mysqli_query($conn, $query)) {

        echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageAdminProfile/listWarehous
e staffAdmin.php' </script>";

    } else {
        echo "Error: " . $query . "<br>" . mysqli_error($conn);
    }
}

//delete warehouse staff

```

```

public function deleteWarehouse staff($Warehouse staff_Id) {
    $conn = mysqli_connect('localhost', 'root', '', 'superpropos');

    if (isset ($_POST['delete'])){
        $Warehouse staff_Id = mysqli_real_escape_string($conn,
$Warehouse staff_Id);

        $query = "DELETE FROM warehouse staff WHERE Warehouse
staff_Id='$Warehouse staff_Id'";

        if (mysqli_query($conn, $query)) {
            echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageAdminProfile/listWarehous
e staffAdmin.php' </script>";
        } else {
            echo "Error: " . $query . "<br>" . mysqli_error($conn);
        }

        mysqli_close($conn);
    }
}

$warehouse staff = new warehouse staffController();
if(isset($_POST['update'])){
    $warehouse staff->updateWarehouse staff();
}

if(isset($_POST['add'])){
    $warehouse staff->addWarehouse staff();
}

if (isset($_POST['delete'])) {
    $Warehouse staff_Id = $_POST['Warehouse staff_Id'];
    $warehouse staff->deleteWarehouse staff($Warehouse staff_Id);
}

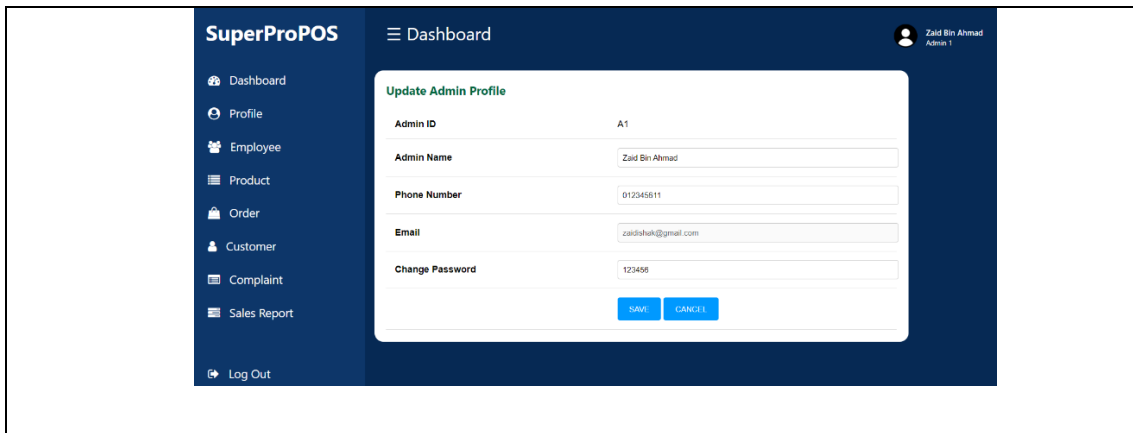
?>

```



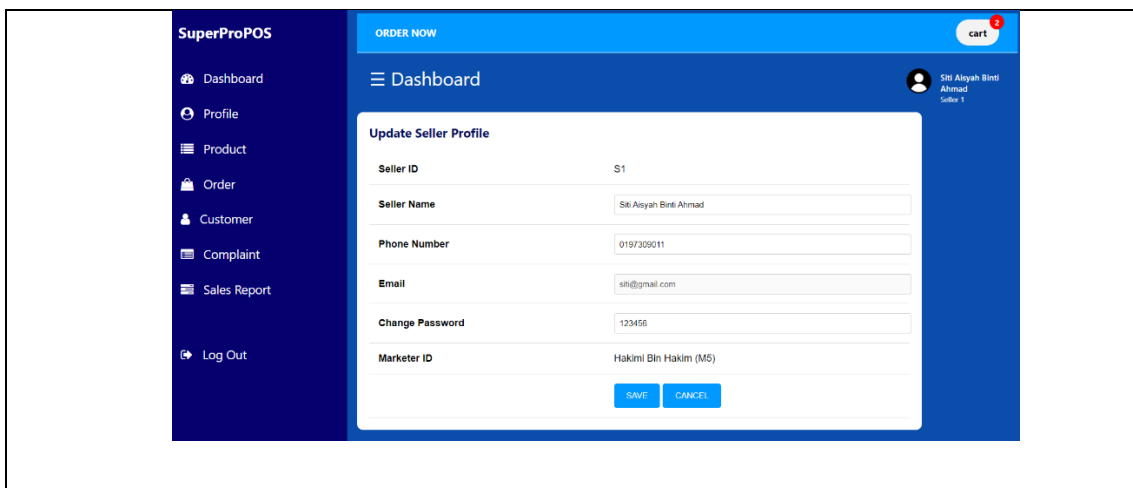
### 4.2.2.3 Manage User Profile

Module Manage User Profile allow the users of the system to manage their own profile. The admin, seller, marketer and warehouse staff are allowed to update their profile in the SuperProPOS: Product Ordering System. Figure 4.21 until figure 4.24 below shows the update profile page for Admin, Seller, Marketer and Warehouse staff.



The screenshot displays the 'Update Admin Profile' interface. On the left is a dark blue sidebar with navigation options: Dashboard, Profile, Employee, Product, Order, Customer, Complaint, Sales Report, and Log Out. The main content area has a dark blue header with 'SuperProPOS' and 'Dashboard' text, and a user profile for 'Zaid Bin Ahmad Admin 1'. The central form is titled 'Update Admin Profile' and contains the following fields: Admin ID (A1), Admin Name (Zaid Bin Ahmad), Phone Number (012345611), Email (zaidshah@gmail.com), and Change Password (123456). At the bottom of the form are 'SAVE' and 'CANCEL' buttons.

Figure 4.21 Update Profile Page for Admin



The screenshot displays the 'Update Seller Profile' interface. On the left is a dark blue sidebar with navigation options: Dashboard, Profile, Product, Order, Customer, Complaint, Sales Report, and Log Out. The main content area has a blue header with 'SuperProPOS' and 'Dashboard' text, and a user profile for 'Siti Aisyah Binti Ahmad Seller 1'. The central form is titled 'Update Seller Profile' and contains the following fields: Seller ID (S1), Seller Name (Siti Aisyah Binti Ahmad), Phone Number (0197309011), Email (siti@gmail.com), Change Password (123456), and Marketer ID (Hakimi Bin Hakim (MS)). At the bottom of the form are 'SAVE' and 'CANCEL' buttons.

Figure 4.22 Update Profile Page for Seller

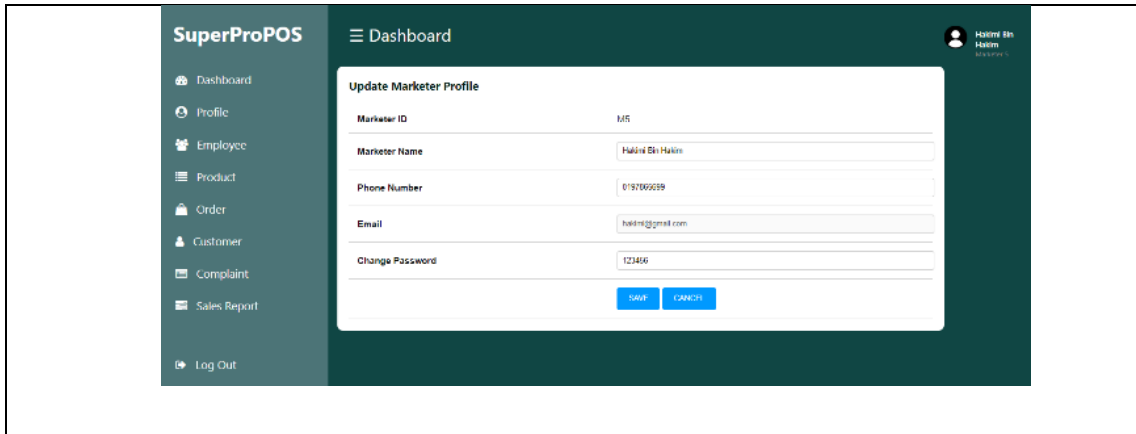


Figure 4.23 Update Profile Page for Marketer

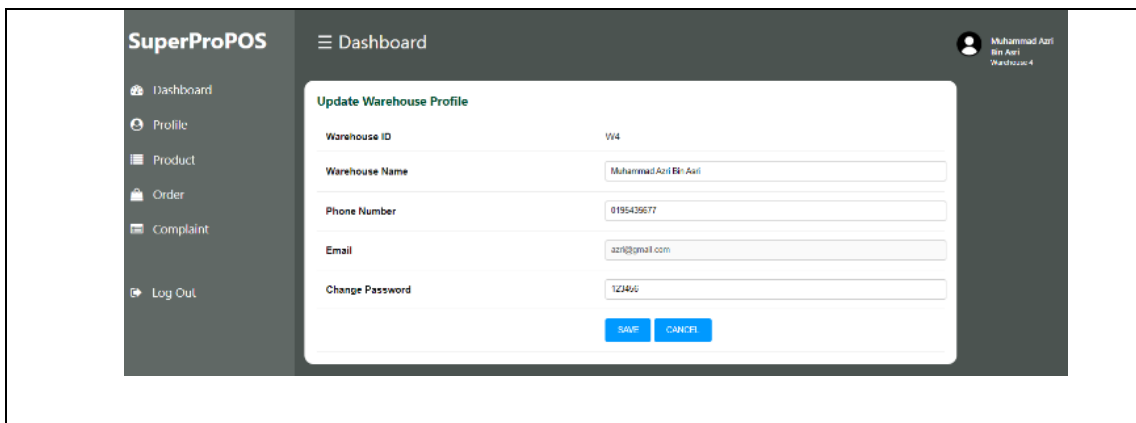
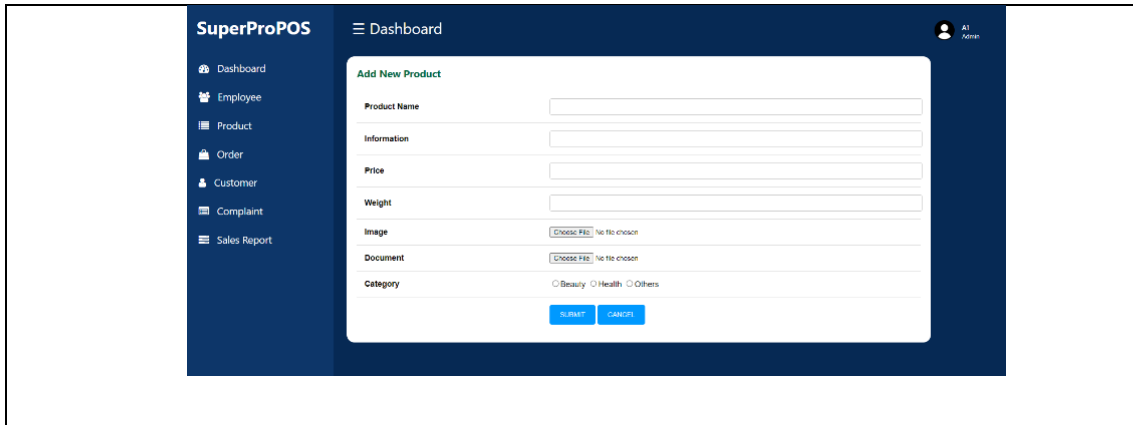


Figure 4.24 Update Profile Page for Warehouse staff

#### 4.2.2.4 Manage Product

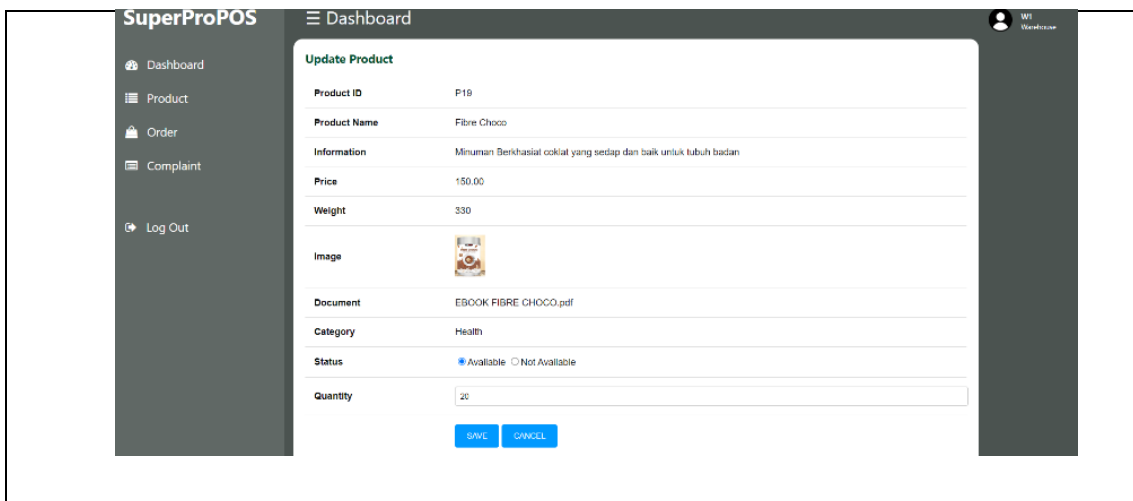
Manage Product module allow the admin to add new product information to the system by product name, information, price, weight, image, document and category. Figure 4.25 below shows the Add New Product page for Admin.



The screenshot shows the 'Add New Product' form in the SuperProPOS system. The interface is dark-themed with a blue sidebar on the left containing navigation options: Dashboard, Employee, Product, Order, Customer, Complaint, and Sales Report. The main content area is titled 'Add New Product' and contains several input fields: Product Name, Information, Price, Weight, Image (with a 'Choose File' button and 'No file chosen' text), Document (with a 'Choose File' button and 'No file chosen' text), and Category (with radio buttons for Beauty, Health, and Others). At the bottom of the form are two buttons: 'SUBMIT' and 'CANCEL'.

Figure 4.25 Add New Product Page for Admin.

Once the admin success adds the product information to the system. The warehouse staff needs to update the product status and quantity of product in the system. Figure 4.26 below shows the update product page for Warehouse staff.



The screenshot shows the 'Update Product' form in the SuperProPOS system. The interface is dark-themed with a dark sidebar on the left containing navigation options: Dashboard, Product, Order, Complaint, and Log Out. The main content area is titled 'Update Product' and displays product details in a table-like format: Product ID (P19), Product Name (Fibre Choco), Information (Minuman Berkhasiat coklat yang sedap dan baik untuk tubuh badan), Price (150.00), Weight (330), Image (a small product image), Document (EBOOK FIBRE CHOCO.pdf), Category (Health), Status (radio buttons for Available and Not Available, with 'Available' selected), and Quantity (20). At the bottom of the form are two buttons: 'SAVE' and 'CANCEL'.

Figure 4.26 Update Product Page for Warehouse staff

Once the Warehouse staff update the status and quantity of product to the system. The marketer and seller are able to view the product information in the system. Figure 4.27 and figure 4.28 below shows the list of product page for marketer and seller.

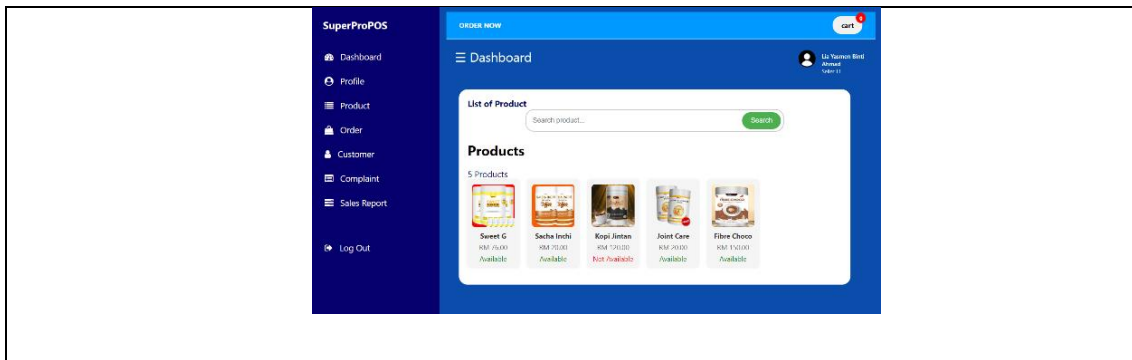


Figure 4.27 List Product Page for Seller

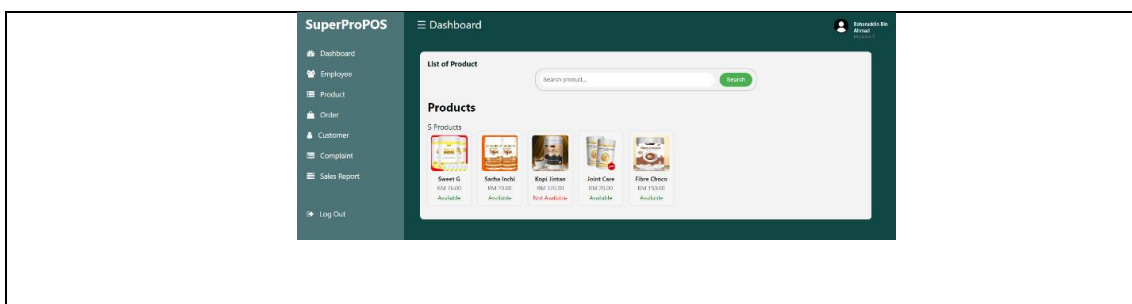


Figure 4.28 List of Product Page for Marketer

Once the seller or marketer click the product image, the product displayed the product information in detail. Figure 4.29 below shows the product detail page looks like.

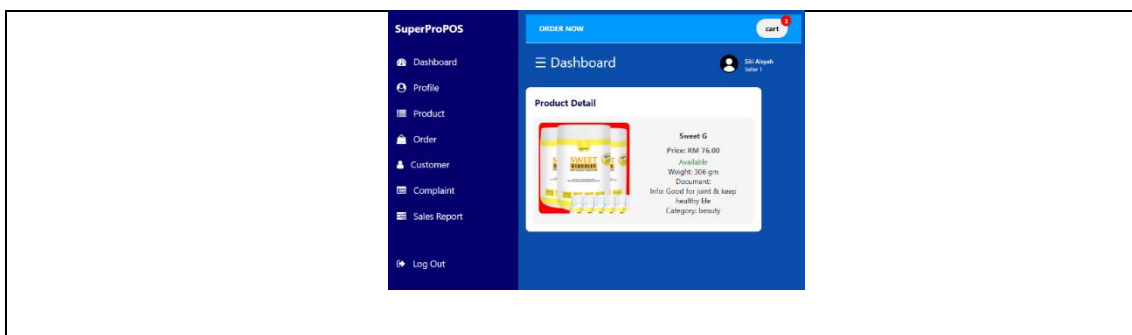


Figure 4.29 Product Detail Page for Seller

#### 4.2.2.5 Manage Customer

Manage Customer module allowed the seller to add customer information to the system. The seller is allowed to add, update and delete the customer information. Figure 4.30 until figure 4.32 below shows the manage customer pages for Seller.

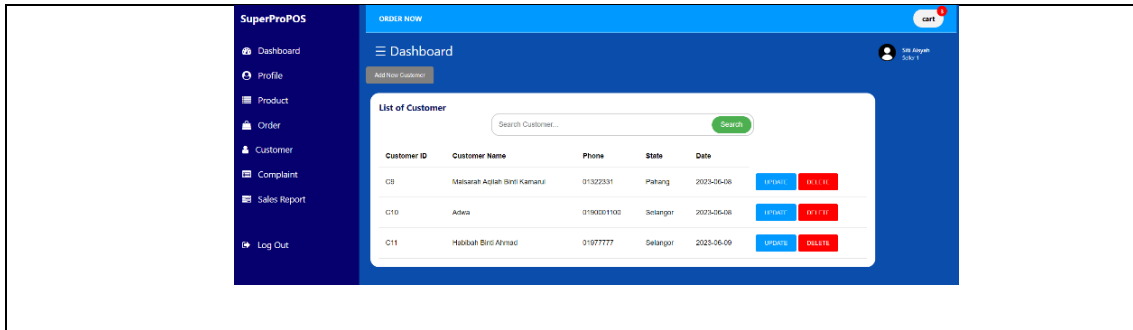


Figure 4.30 List Customer for Seller

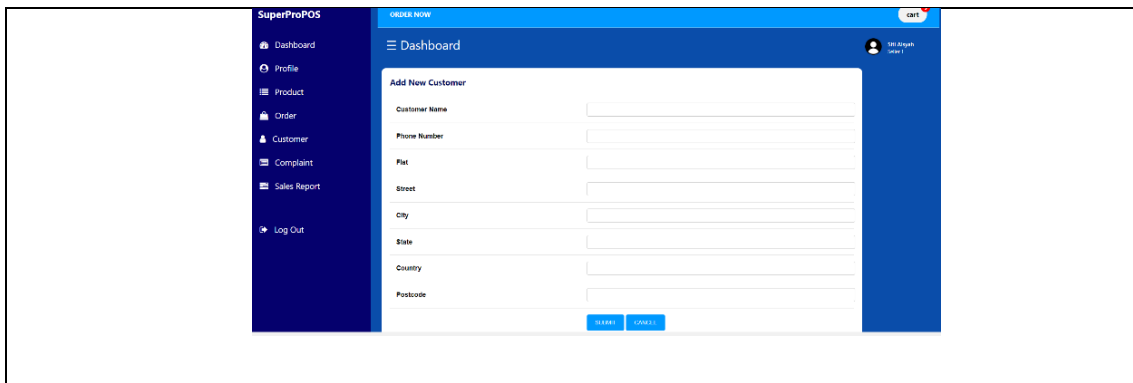


Figure 4.31 Add New Customer Page for Seller

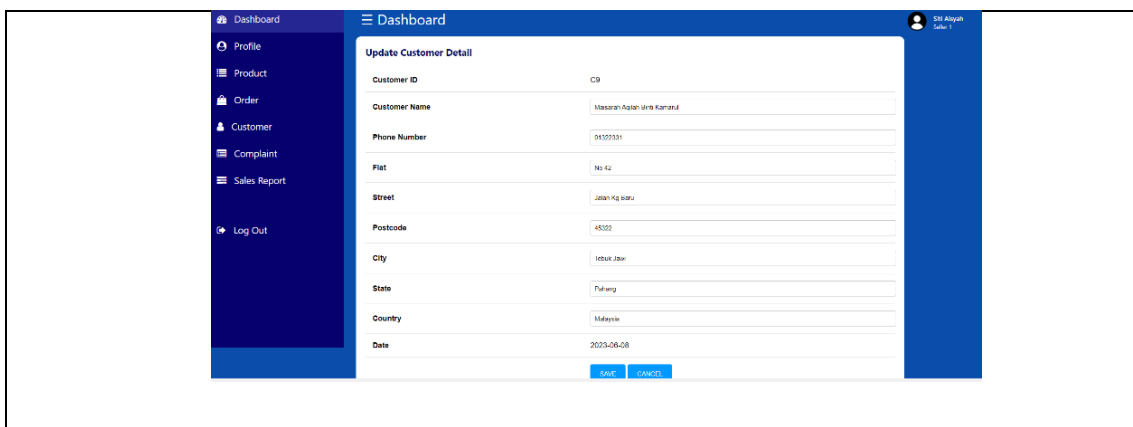


Figure 4.32 Update Customer Page for Seller

In order to allow the function of the customer to be manageable, customerController was done. Below is the code for customerController.

```
<?php

include("../../BusinessServiceLayer/libs/database.php");

// add customer profile
class customerController {

public function addCustomer() {
if (isset ($_POST['add'])){
$C_Name=$_POST['C_Name'];
$C_PhoneNum=$_POST['C_PhoneNum'];
$C_Flat=$_POST['C_Flat'];
$C_Street=$_POST['C_Street'];
$C_City=$_POST['C_City'];
$C_State=$_POST['C_State'];
$C_Country=$_POST['C_Country'];
$C_Postcode=$_POST['C_Postcode'];
$C_Date= date('Y-m-d H:i:s');
$sellerID=$_POST['Seller_Id'];

$query = "INSERT INTO customer
VALUES('','$C_Name','$C_PhoneNum','$C_Flat',
'$C_Street','$C_City','$C_State','$C_Country','$C_Postcode','$C_Date',
$sellerID)";
$conn = mysqli_connect('localhost', 'root', '', 'superpropos');

if (mysqli_query($conn, $query)) {

echo "<script type='text/javascript'>
window.location='../../ApplicationLayer/ManageSeller/listCustomerSeller
.php' </script>";

} else {
echo "Error: " . $query . "<br>" . mysqli_error($conn);
}
}
}
```

```

//update customer profile
public function updateCustomer()
{
    if (isset($_POST['update'])) {
        $Customer_Id = $_POST['Customer_Id'];
        $C_Name = $_POST['C_Name'];
        $C_PhoneNum = $_POST['C_PhoneNum'];
        $C_Flat = $_POST['C_Flat'];
        $C_Street = $_POST['C_Street'];
        $C_City = $_POST['C_City'];
        $C_State = $_POST['C_State'];
        $C_Country = $_POST['C_Country'];
        $C_Postcode = $_POST['C_Postcode'];
        $C_Date = $_POST['C_Date'];
        $sellerID = $_POST['Seller_Id'];

        $query = "UPDATE `customer` SET `C_Name`='$C_Name',
`C_PhoneNum`='$C_PhoneNum', `C_Flat`='$C_Flat', `C_Street`='$C_Street',
`C_City`='$C_City', `C_State`='$C_State', `C_Country`='$C_Country',
`C_Postcode`='$C_Postcode' WHERE `Customer_Id`='$Customer_Id' AND
`Seller_Id`='$sellerID'";
        $conn = mysqli_connect('localhost', 'root', '',
'superpropos');

        if (mysqli_query($conn, $query)) {
            echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageSeller/listCustomerSeller
.php' </script>";
        } else {
            echo "Error: " . $query . "<br>" .
mysqli_error($conn);
        }
    }
}

//delete customer
public function deleteCustomer($Customer_Id) {
    $conn = mysqli_connect('localhost', 'root', '', 'superpropos');

    if (isset ($_POST['delete'])){
        $Customer_Id = mysqli_real_escape_string($conn, $Customer_Id);
    }
}

```

```

        $query = "DELETE FROM customer WHERE
Customer_Id='$Customer_Id'";

        if (mysqli_query($conn, $query)) {
            echo "<script type='text/javascript'>
window.location='../..//ApplicationLayer/ManageAdminProfile/listCustomer
Admin.php' </script>";
        } else {
            echo "Error: " . $query . "<br>" . mysqli_error($conn);
        }

        mysqli_close($conn);
    }
}

$customer = new customerController();
if(isset($_POST['update'])){
    $customer->updateCustomer();
}

if(isset($_POST['add'])){
    $customer->addCustomer();
}

if (isset($_POST['delete'])) {
    $Customer_Id = $_POST['Customer_Id'];
    $customer->deleteCustomer($Customer_Id);
}

?>

```



#### 4.2.2.6 Manage Order

Manage Order module is work as a module that allow the seller to make order for customer. The customer needs to add product to cart and fill the order form in order to make the order submission successful. The admin then updated the status of the payment for the orders to allow the warehouse staff to pack the order and update the delivery status of the orders. Figure 4.33 below shows the process of manage order module.

The seller needs to click the “Add To Cart” in the page below to add the selected product to cart before submit an order form.

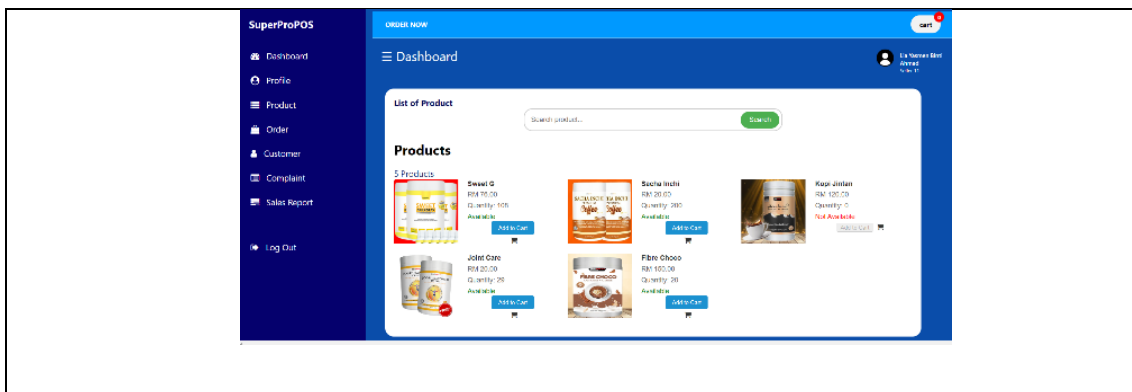


Figure 4.33 Add Order Page for Seller

Once the seller clicks the “cart” button, the system then displayed the Add To Cart page for seller as shown in figure 4.34 below.

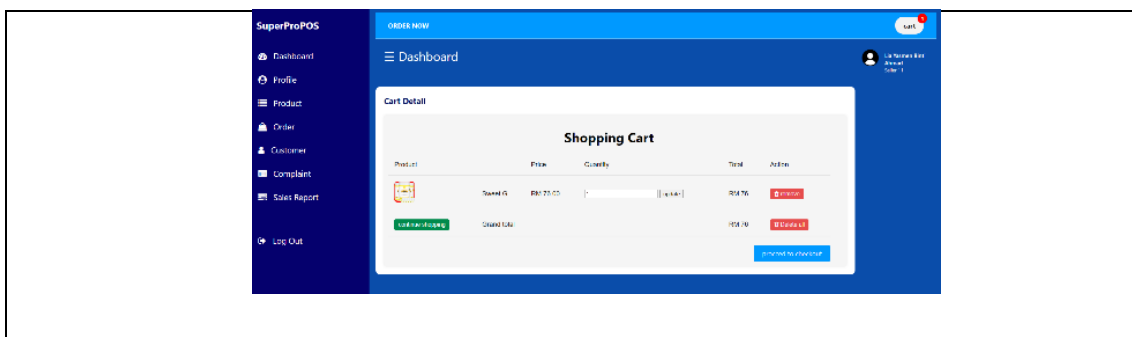


Figure 4.34 Add to Cart Pages for Seller.

Once the seller click add to cart button, the system then displays the Check-Out Order Form for seller as shown in figure 4.35 below.

The screenshot shows a 'Complete Your Order' form with the following fields and values:

- Order ID: 049
- Customer Name: Habibah Binti Ahmad
- Phone Number: 0190001100
- Address: No 42 Kg Simpang 4 65200 Sabak Bernam Selangor Malaysia
- Payment Method: ATM/CDM Machine
- Bank: Maybank
- Receipt Image: [Image placeholder]
- Receipt Reference: #Ref1223131331
- Payment Date & Time: 2023-06-23 10:53:00
- Product Name & Quantity: Sacha Inchi (1), Teh Jintan HBam (1), Neuro Up Tablet (1), Madu Propolis (1)
- Total Price: 290.00
- Shipping Method: Ninja Van
- Seller: Siti Aisyah Binti Ahmad (S1)
- Payment Status:  Approved  Disapproved
- Deliver Status: Pending
- Tracking Number: [Empty field]

Figure 4.35 Check Out Order Form Page for Seller.

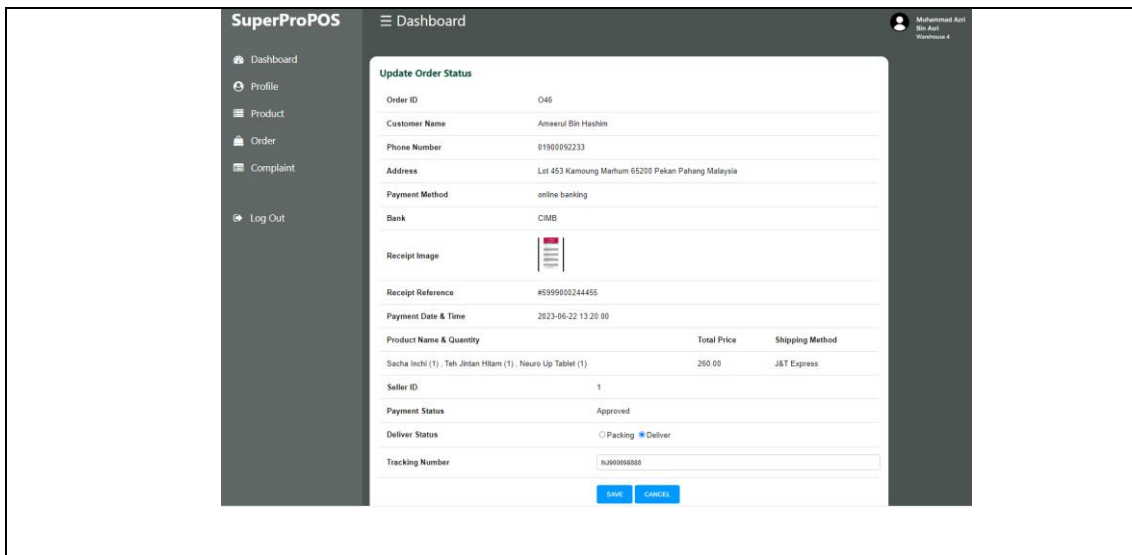
Once the order submitted, the admin then verify the payment status from the bank receipt information given by updating the Payment Status approved or disapproved.as shown in figure 4.36 below.

The screenshot shows an 'Update Order Status' form with the following details:

- Order ID: 049
- Customer Name: Habibah Binti Ahmad
- Phone Number: 0190001100
- Address: No 42 Kg Simpang 4 65200 Sabak Bernam Selangor Malaysia
- Payment Method: ATM/CDM Machine
- Bank: Maybank
- Receipt Image: [Image placeholder]
- Receipt Reference: #Ref1223131331
- Payment Date & Time: 2023-06-23 10:53:00
- Product Name & Quantity: Sacha Inchi (1), Teh Jintan HBam (1), Neuro Up Tablet (1), Madu Propolis (1)
- Total Price: 290.00
- Shipping Method: Ninja Van
- Seller: Siti Aisyah Binti Ahmad (S1)
- Payment Status:  Approved  Disapproved
- Deliver Status: Pending
- Tracking Number: [Empty field]

Figure 4.36 Update Order Payment Status Page for Admin.

Once the payment had been approved by admin, the warehouse staff then update the delivery status of the order as shown in figure 4.37 below:



The screenshot shows the 'Update Order Status' page in the SuperProPOS system. The page is titled 'Update Order Status' and contains the following information:

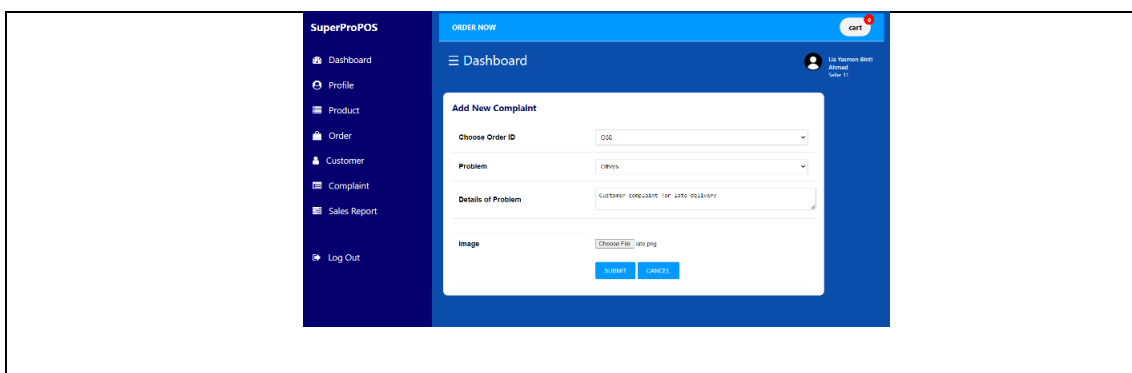
- Order ID: 046
- Customer Name: Ameerul Bin Hashim
- Phone Number: 01980992333
- Address: Lot 453 Kampong Marhum 65200 Pekan Pahang Malaysia
- Payment Method: online banking
- Bank: CIMB
- Receipt Image: [Image placeholder]
- Receipt Reference: #599900244455
- Payment Date & Time: 2023-06-22 13:20:00
- Product Name & Quantity: Sacha Inchi (1), Teh Jantan Hitam (1), Neuro Up Tablet (1)
- Total Price: 200.00
- Shipping Method: J&T Express
- Seller ID: 1
- Payment Status: Approved
- Deliver Status:  Packing  Deliver
- Tracking Number: [Input field with value: NJ99008888]

At the bottom of the form, there are 'SAVE' and 'CANCEL' buttons.

Figure 4.37 Update Delivery Status Page for Warehouse staff.

#### 4.2.2.7 Manage Complaint

Manage Complaint module allowed the seller to make a complaint of the orders that the seller have made for their customers. The seller needs to choose the existing orders that have been made and fill the complaint form. Figure 4.38 below shows the complaint form of orders for seller.



The screenshot shows the 'Add New Complaint' page in the SuperProPOS system. The page is titled 'Add New Complaint' and contains the following information:

- Choose Order ID: [Dropdown menu with value: 000]
- Problem: [Dropdown menu with value: others]
- Details of Problem: [Text area with value: customer complaint for late delivery]
- Image: [Image upload field with value: Choose File]

At the bottom of the form, there are 'SUBMIT' and 'CANCEL' buttons.

Figure 4.38 Add Complaint Form Page for Seller.

The admin and warehouse staff then can update the status of the complaint of the orders as shown in figure 4.39 and figure 4.40 below.

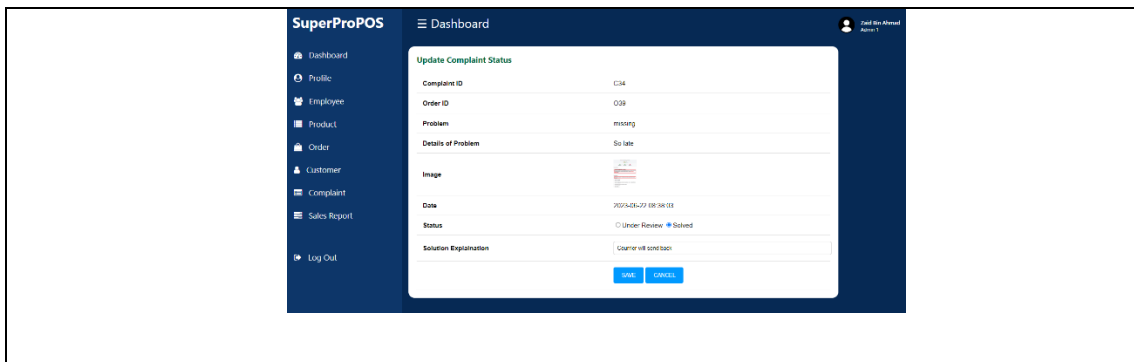


Figure 4.39 Update Complaint Status Page for Admin.

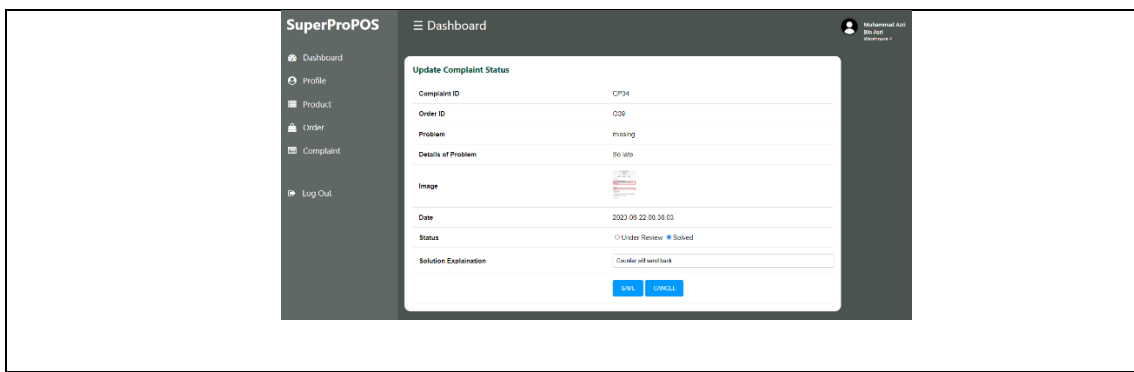


Figure 4.40 Update Complaint Status Page for Warehouse staff.

Then the marketer can view the complaint that have been made by their seller. Figure 4.41 below shows the list of complaint made by the marketer's seller.

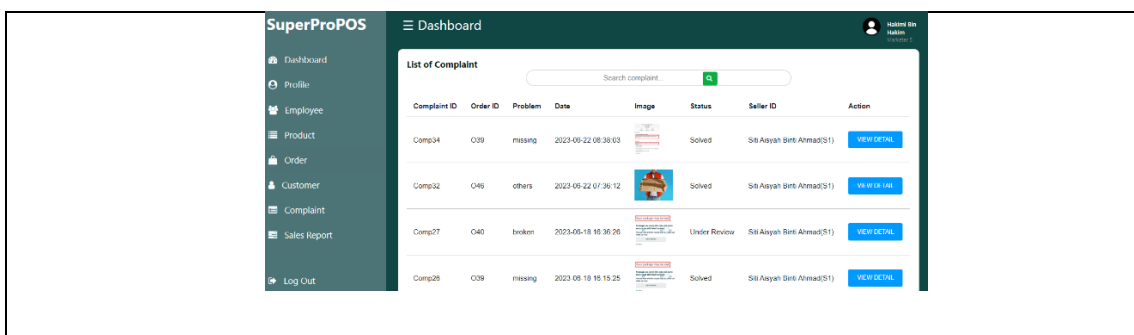


Figure 4.41 List of Seller Complaint Page for Marketer

#### 4.2.2.8 Generate Report

Generate Report module allowed the system to generate the sales report of orders for admin, seller and marketer. The admin is able to view the overall orders sales by daily, monthly and yearly sales. While the seller is able to view their sales individually by daily, monthly and yearly. Marketer also can view their assign seller sales report by daily, monthly and yearly. Figure 4.42 until figure 4.45 below shows the generate report module details.

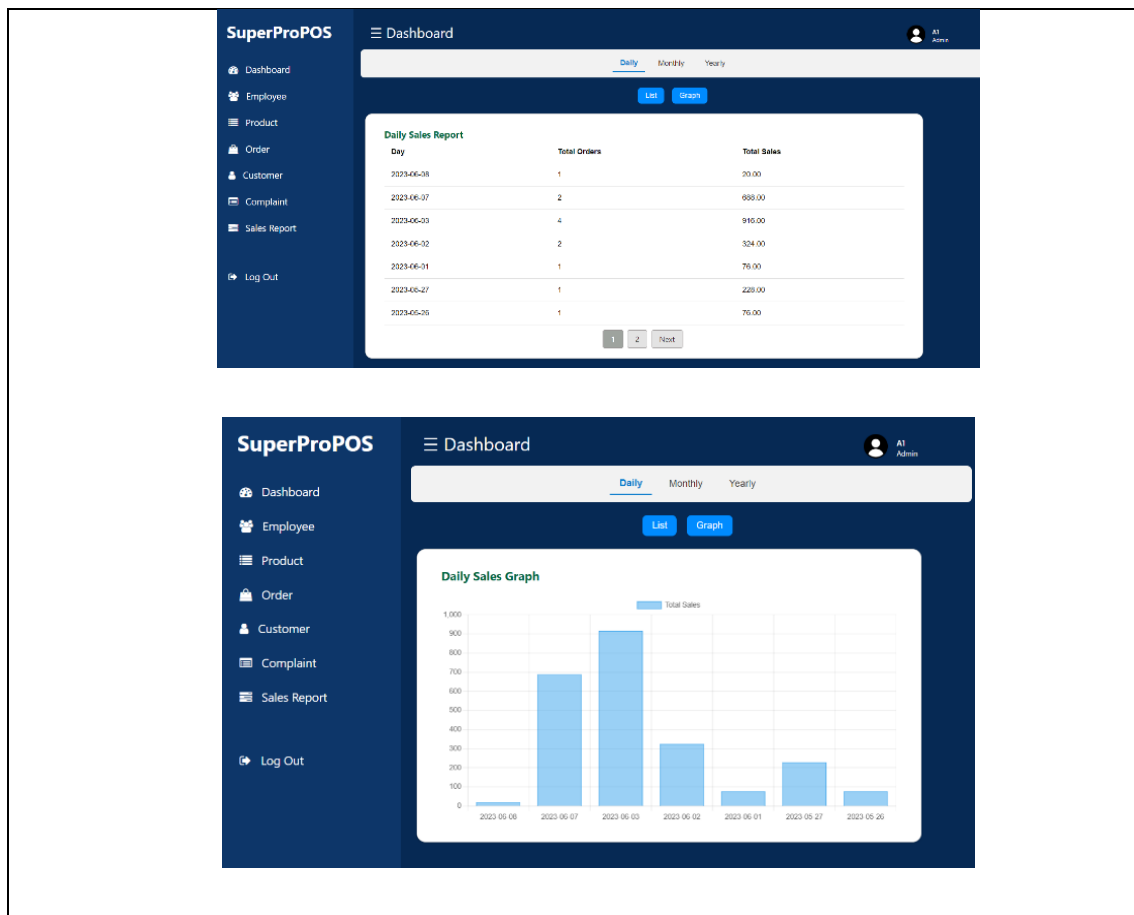


Figure 4.42 Daily Sales Report Page for Admin.

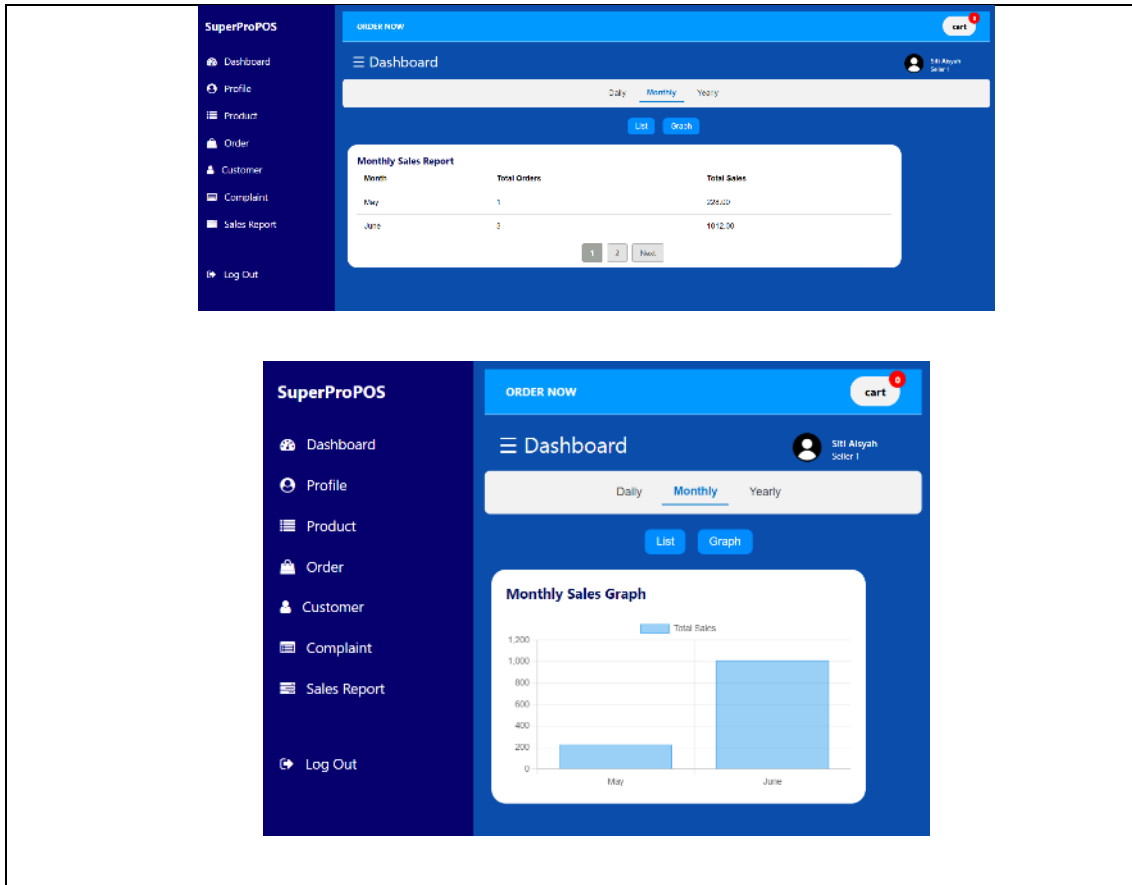


Figure 4.43 Monthly Sales Report for Seller.

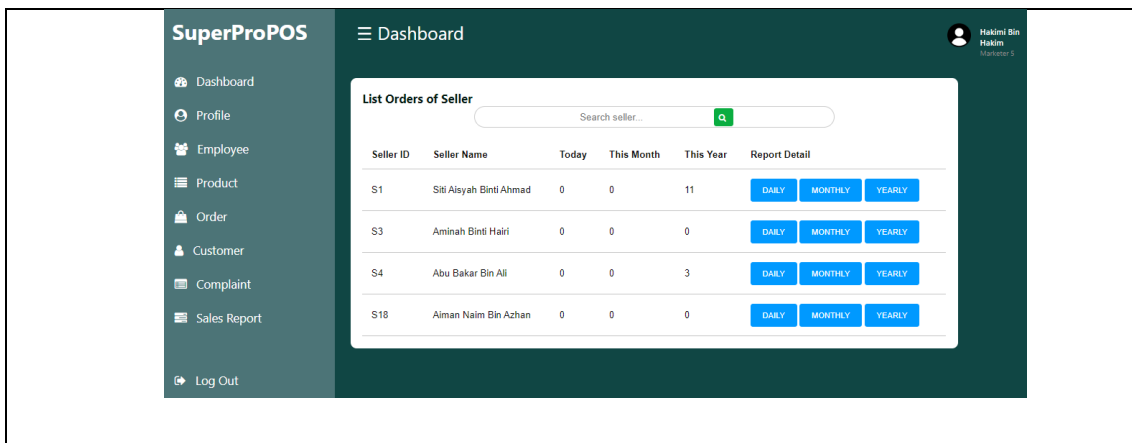


Figure 4.44 List of Marketer's Seller Sales Page for Marketer.

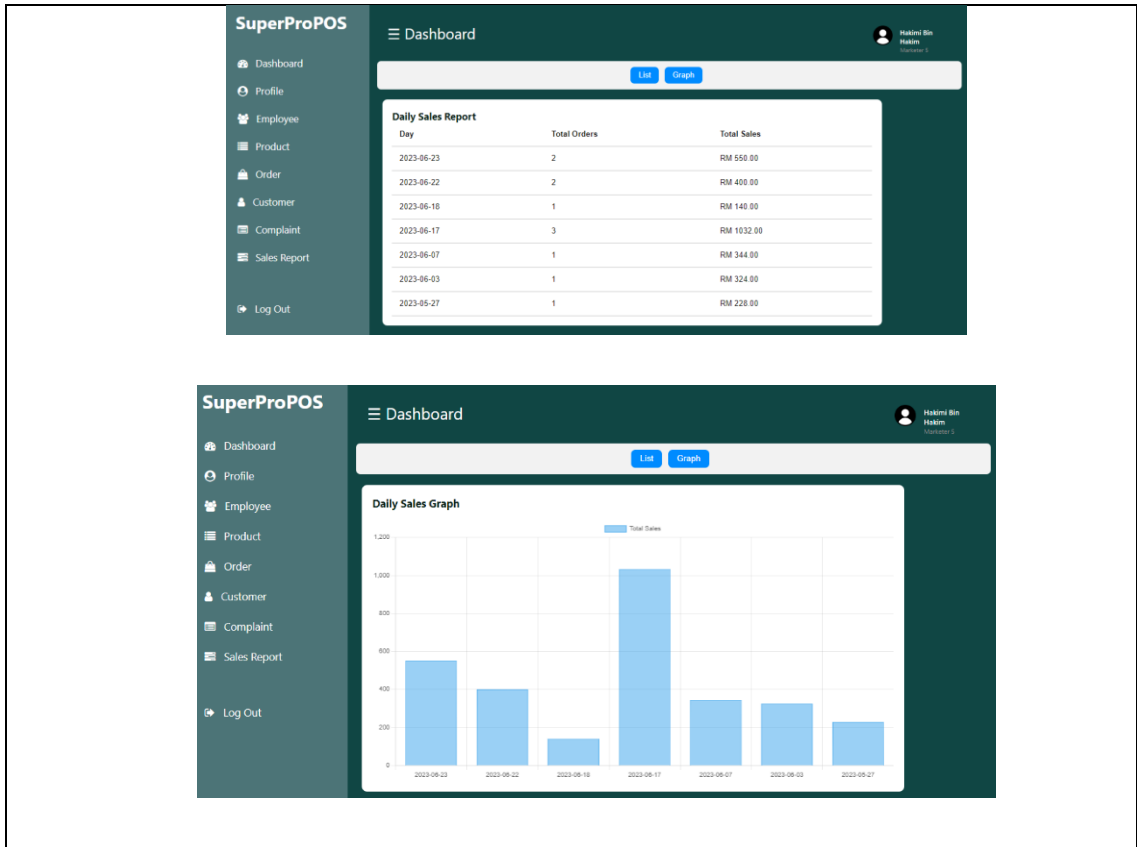


Figure 4.45 Marketer’s Seller Daily Sales Report Page for Marketer.

### **4.3 System Testing**

The SuperProPOS: Product Ordering System testing was conducted at Promaxx Marketing, a business company. User Acceptance Testing Forms were distributed to four users with different roles: Admin, Seller, Marketer, and Warehouse Staff. These forms were crucial for gathering valuable feedback regarding the SuperProPOS: Product Ordering System.

User Acceptance Testing (UAT) was employed to thoroughly assess every function within the SuperProPOS: Product Ordering System. It involved distributing the UAT forms to potential users of the system. The outcomes of the UAT indicated that all tested functions have successfully passed. The detailed results of these UAT can be found in Appendix C.

The formal approval of the User Acceptance Testing by the Client, as documented in Appendix C, reaffirmed that the SuperProPOS: Product Ordering System successfully meets the requirements and expectations of its users. The positive feedback received from the UAT further validates the system's efficiency and effectiveness in seamlessly facilitating business operations at Promaxx Marketing.



## 4.4 Result Discussion

The SuperProPOS: Product Ordering System is expected to be completed before the due of the deadline. The result from the developing the system following the objectives that have been mentioned before, all of the objectives successfully achieved. It is measure by the User Acceptance Testing result given by the client as mentioned in Appendix C.

As stated in the first objective above which is to study the current method of ordering product in an improvised system. All the findings are gathered and resulting to develop the SuperProPOS: Product Ordering System as result of an improvised system.

The second objective was met which the development of SuperProPOS: Product Ordering System was done. Eight modules were developed which are login, register employee, manage user profile, manage product, manage customer, manage order, manage complaint and generate report. Figure 4.46 until figure 4.57 below are some the results of the development for SuperProPOS: Product Ordering System.

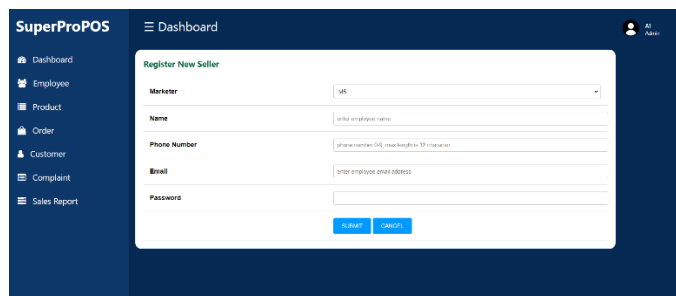
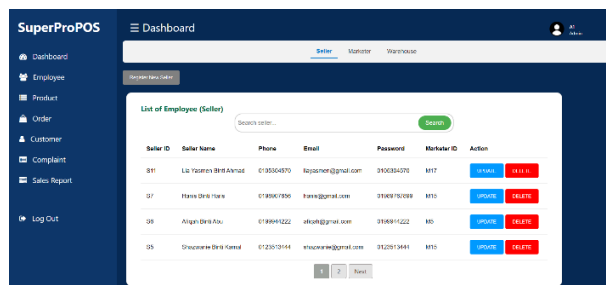


Figure 4.46 Register New Seller Page for Admin



Seller ID	Seller Name	Phone	Email	Password	Marketer ID	Action
811	Lili Yuliana Binti Anand	0193504870	liliponnet@gmail.com	9100334576	8117	<button>Update</button> <button>Delete</button>
827	Hana Dinda Hana	0189061836	hanah@gmail.com	914691761089	8110	<button>Update</button> <button>Delete</button>
828	Aliper Dinda Ayu	0189911222	aliper@gmail.com	9146911222	810	<button>Update</button> <button>Delete</button>
835	Shucorena Binti Kemal	0123573144	shucorena@gmail.com	9123519441	8110	<button>Update</button> <button>Delete</button>

Figure 4.47 List of Seller Page for Admin

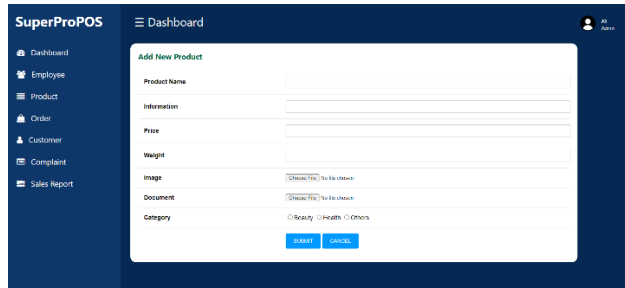


Figure 4.48 Add New Product Page for Admin

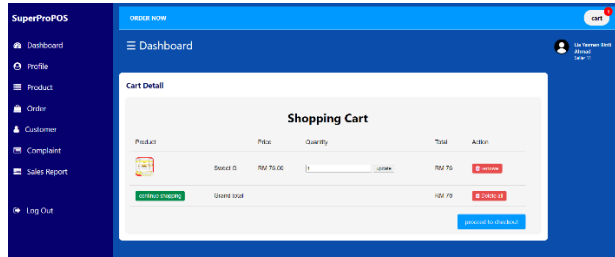


Figure 4.49 Cart Pages for Seller

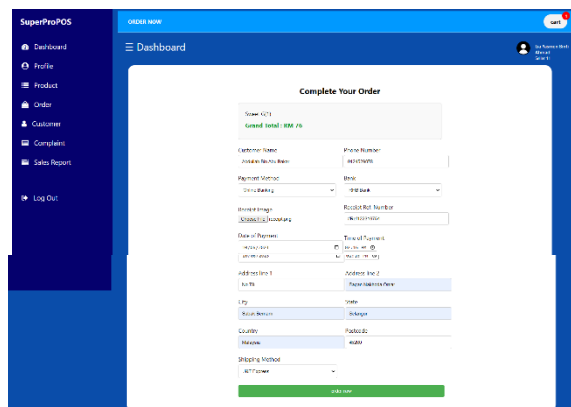


Figure 4.50 Check Out Order Form Page for Seller

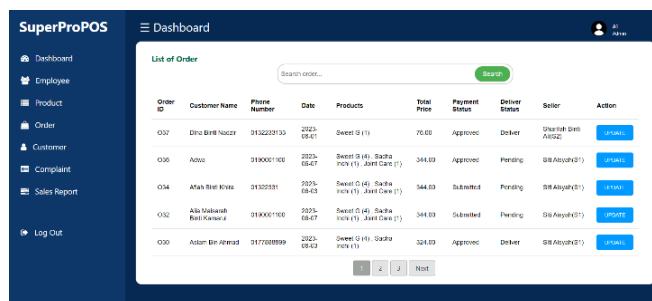


Figure 4.51 List Order Page for Admin

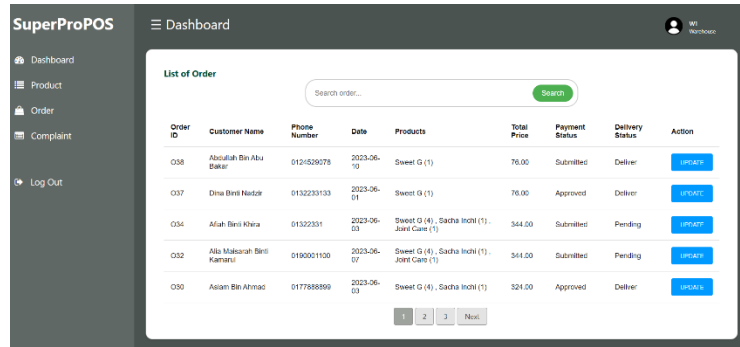


Figure 4.52 List Order Page for Warehouse staff

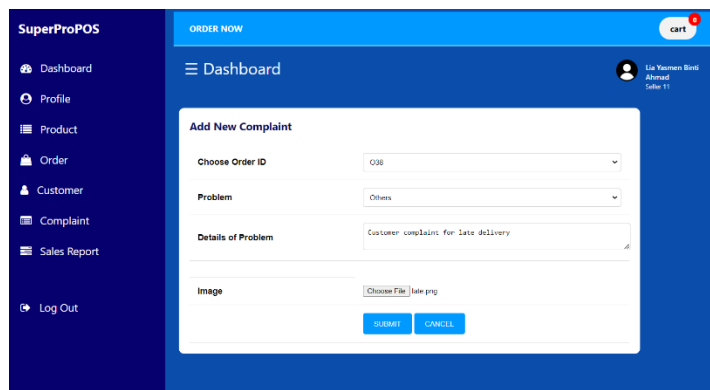


Figure 4.53 Add Complaint Form Page for Seller

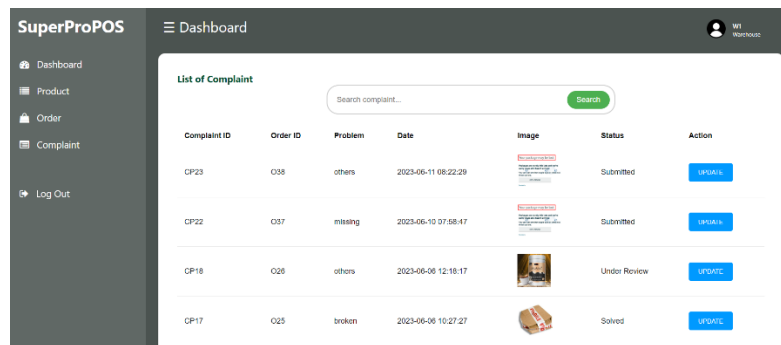


Figure 4.54 List Complaint Page for Warehouse staff

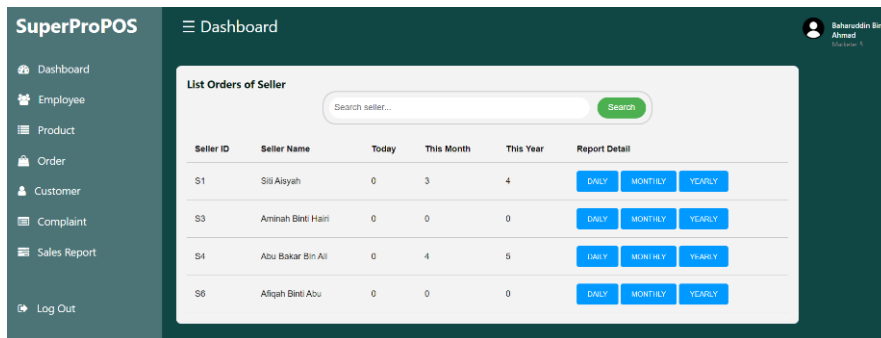


Figure 4.55 List Seller's Sales Report Page for Marketer

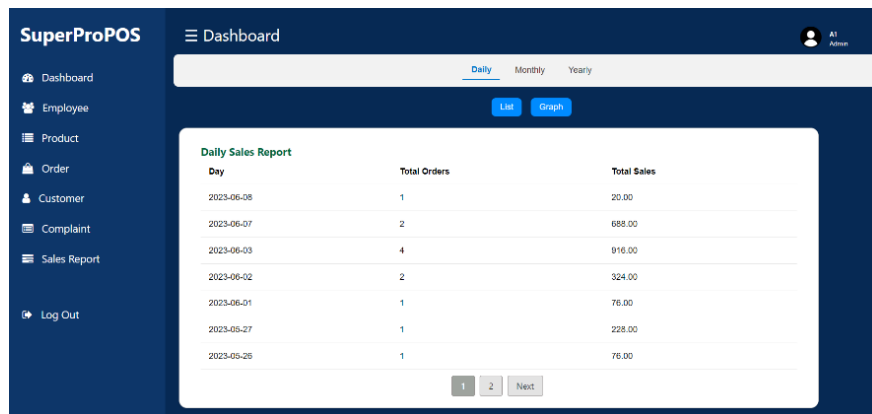


Figure 4.56 Daily Report Sales Page for Admin

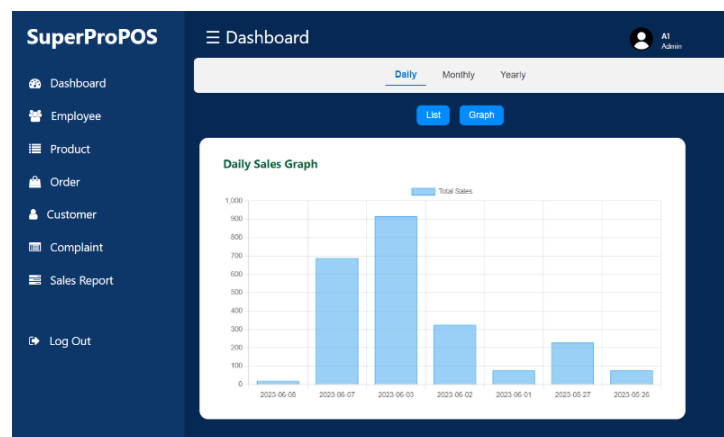


Figure 4.57 Daily Report Sales Graph Page for Admin

The completion of development for SuperProPOS: Product Ordering System project has been evaluated by the client company which is Promaxx Marketing. This evaluation has met the third objective of this project. The evaluation had been done by using a formal User Acceptance Testing form that had been approved by the Owner of Promaxx Marketing business company. The formal result of User Acceptance Testing form by the client can be found in Appendix C.

## CHAPTER 5

### CONCLUSION

#### 5.1 INTRODUCTION

This chapter provides a summary of the SuperProPOS: Product Ordering System project, specifically focusing on the development process of a product ordering system suitable for e-commerce businesses. It presents an overview of the achieved objectives and the implemented solutions for previously identified problems. In the context of e-commerce companies, having an efficient and effective product ordering system is crucial for smooth business management. This system encompasses various functions such as login, register employee, manage user profile, manage product detail, manage order, manage customer, manage complaint and generate report.

To develop this system, Visual Studio Code and XAMPP Control Panel were utilized as software tools, while the programming languages employed were HTML, PHP, CSS, and JavaScript. The Waterfall Methodology was employed to ensure effective project timeline management. The system's effectiveness and functionality were thoroughly tested and evaluated by employees of Promaxx Marketing Company. The evaluation process resulted in positive feedback from the employees, affirming that the system successfully achieved its objectives. Additionally, it should be noted that the system was successfully deployed on the FK INDAH server and can be found in this link: <https://indah.ump.edu.my/CB20009/superproposnew/ApplicationLayer/indexMain.php>.

## **5.2 System Constraint**

The constraint that happened in the developing of the project are:

- Time limitation

The time given was limited so it requires an effective time management, prioritization of tasks and efficient allocation of resources to ensure the system is developed and deployed within the specified timeframe in order to finish 7 module of the SuperProPOS: Product Ordering System.

- Code Error

Coding is important in developing the system. Some error had occurred during the scripting of the code. The error has caused the system to display unexpected output and not functioning well. When the error happened, a guidance from an expert was needed in order to solve the problem.

## **5.3 Future Work**

The future work of the SuperProPOS: Product Ordering System can be enhanced for e-commerce business company.

- Developer can make an additional modules and content, such as manage commission of seller.
- Developer may make some changes on manage order modules that allow the module to also been used by customer to make an order.
- Developer may add print sales report function in the system.

## REFERENCES

1. Handrianto, Y., & Sanjaya, B. (2020). Model Waterfall Dalam Rancang Bangun Sistem Informasi Pemesanan Produk Dan Outlet Berbasis Web. *Jurnal Inovasi Informatika*, 5(2), 153–160. <https://doi.org/10.51170/jii.v5i2.66>
2. Beyari, H. (2021). RECENT E-COMMERCE TRENDS AND LEARNINGS FOR E-COMMERCE SYSTEM DEVELOPMENT FROM A QUALITY PERSPECTIVE. *International Journal for Quality Research*, 15(3), 797–810. <https://doi.org/10.24874/IJQR15.03-07>
3. Bulman, M. (2017). SDLC - Waterfall Model. *The Independent*, May.
4. Kramer, M. (2018). Best Practices in Systems Development Lifecycle: an Analyses Based on the Waterfall Model. *Review of Business & Finance Studies*, 9(1), 77–84. Retrieved from [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3131958](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3131958).
5. Software Testing Help. (2019). What is SDLC Waterfall Model? *Software Testing Help*, 1–7.
6. bio-asli.com. (n.d.). Laman Utama | BIO-ASLI. <https://www.bio-asli.com/mobile/default.asp?id=A0001>
7. Obsess Cosmetics. (n.d.). <https://obsesscosmetics.com/>
8. D’Herbs Terlajak Laris. (n.d.). Dherbs. <https://dherbs.easy.co/>
9. This article is written by plus2net.com team. <https://www.plus2net.com>. (2000, February 5). *Plus2net.com*. Group by command in date field to list number of records in day wise. [https://www.plus2net.com/sql\\_tutorial/date-group-by.php](https://www.plus2net.com/sql_tutorial/date-group-by.php)
10. Kumar, A. (2022, August 17). *How to get sales reports from the database using PHP*. PHPGurukul. <https://phpgurukul.com/how-to-get-sales-reports-from-the-database-using-php/>
11. Adams, D. (2023, March 17). *Shopping cart system with PHP and mysql*. CodeShack. <https://codeshack.io/shopping-cart-system-php-mysql/>
12. *Simple PHP shopping cart*. Phppot. (n.d.). <https://phppot.com/php/simple-php-shopping-cart/>
13. *PHP mysql crud application*. TutorialRepublic. (n.d.). <https://www.tutorialrepublic.com/php-tutorial/php-mysql-crud-application.php>



## **APPENDICES**

Appendix A: Software Requirement Specification (SRS) of SuperProPOS: Product Ordering System

Version

1

SUPERPROPOS: PRODUCT ORDERING SYSTEM

---

Faculty of Computing

# Software Requirement Specification (SRS)

# Table of Contents

1.0	PRODUCT DESCRIPTION	1
	1.1 User Characteristics	
	1.2 Constraints	
2.0	INTERFACES REQUIREMENTS	1
	2.1 User Interface	
	2.2 Hardware Interface	
	2.3 Software Interface	
3.0	SOFTWARE PRODUCT FEATURES	2
	3.1 Use Case Name	
	3.2 Use Case Name	
4.0	REQUIREMENTS TRACEABILITY	3
5.0	SYSTEM REQUIREMENTS APPROVAL	5
	Appendix B-1	
	Appendix B-2	

## 1.0 PRODUCT DESCRIPTION

SuperProPOS: Product Ordering System is a system that help a business company to manage their ordering products information and operation. The system's end-user are admin, seller, marketer and warehouse staff.

### 1.1 User Characteristics

Table 1: User Characteristics

User	Education level	Background experience
Admin	Bachelor's degree	Must have a high level of experience in using IT Technology.
Seller	Diploma	Must have a medium level of experience in using IT Technology.
Marketer	Diploma	Must have a medium level of experience in using IT Technology.
Warehouse staff	Diploma	Must have a medium level of experience in using IT Technology.

### 1.2 Constraints

- The admin, marketer, seller and warehouse staff must login to the system in order to get an access to the system's functions.
- The system should be available for 24 hours of all time.
- The system should allow only one email address for each user to register in the system.
- The system must have internet connection to allow the system to functioning well.
- The system should have continuous integration to deploy features and fix bugs without downtime.
- The system interface should be user friendly and free or error.
- The system should be accessible for all type of web browser.

## 2.0 INTERFACE REQUIREMENTS

This topic discusses about the detailed description of all inputs and outputs of the SuperProPOS: Product Ordering System.

### 2.1 User Interface

User Interface Name	Description
Login interface	Admin, Seller, Marketer and warehouse staff shall insert email and password to login.
Homepage	System displays the homepage once Admin, Seller, Marketer and Warehouse staff success to login to system.
Register Employee	Admin should be able to register seller, marketer and warehouse staff to the system.
Manage User Profile	Seller, Marketer and Warehouse staff should be able to update their profile. Admin and marketer shall be able to view sellers list. Admin also should be able to view marketer and warehouse staff list.
Manage Product Detail	Admin should be able to add, update and delete product list. Warehouse staff should be able to update the product stock quantity. Seller and Marketer should be able to view the products.

Manage Order	Seller should be able to add and edit order. Admin and Warehouse staff shall be able to update the status of order. Marketer is able to view the order list of assigned seller.
Manage Complaint	Seller shall be able to make complaint and edit complaint. Admin and Warehouse staff shall be able to update the complaint status. Marketer is able to view the complaint.
Generate Report	The systems generated report should display the sales report.

## 2.2 Hardware Interface

### Hardware:

Asus Laptop = Windows 11, Intel i7, 16GB of RAM

Purpose = As a tool for the documentation of the SuperProPOS: Product Ordering System.

## 2.3 Software Interface

The system is developed using the PHP, CSS and JavaScript. The platform that is used to run the code is VS Code. While the platform to connect the database is PhpMyAdmin and XAMPP. Draw.io also used for designing the diagram for each document. Microsoft Word is used as a tool for writing the document.

### 3.0 SOFTWARE PRODUCT FEATURES

#### 3.1 Login

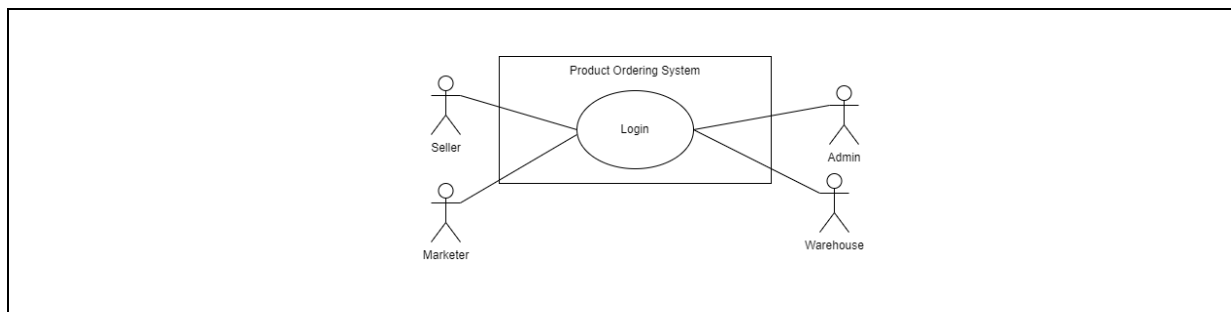


Figure 1: Login

Table 2: Use Case Description for Login

Use Case ID	POS-UC001
Use Case Name	Login
Brief Description	This use case is initiated by admin, seller, marketer and warehouse staff. It allowed the admin, seller, marketer and warehouse staff to login into the SuperProPOS: Product Ordering System to access the functionality of the system.
Actor	Admin, Seller, Marketer and Warehouse staff
Pre-Condition	<ul style="list-style-type: none"> <li>The admin, seller, marketer and warehouse staff already have a registered account in the system.</li> <li>The admin, seller, marketer and warehouse staff must have internet connection.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>The use case starts when the admin, seller, marketer and warehouse staff go to the login menu.</li> <li>The system displays the login page.</li> </ol>



	<ol style="list-style-type: none"> <li>3. The admin, seller, marketer and warehouse staff insert their email and password.</li> <li>4. The admin, seller, marketer and warehouse staff click &lt;&lt;LOGIN&gt;&gt; button.</li> <li>5. The system validates the email and password information. <b>[E1: Invalid Email or Password]</b></li> <li>6. The admin, seller, marketer and warehouse staff successfully login to system.</li> <li>7. The system displayed the homepage screen.</li> <li>8. The use case end.</li> </ol>
Alternative Flow	None
Exception Flow	<p><b>[E1: Invalid Email or Password]</b></p> <ol style="list-style-type: none"> <li>1. Admin, seller, marketer and warehouse staff insert invalid email or username.</li> <li>2. Admin, seller, marketer and warehouse staff click &lt;&lt;LOGIN&gt;&gt; button.</li> <li>3. The system display error message.</li> <li>4. The use case continues with step 2 from basic flow.</li> </ol>
Post Conditions	The admin, seller, marketer and warehouse staff are able to access to the SuperProPOS: Product Ordering System.
Rules	None
Constraints	None

### 3.2 Register Employee

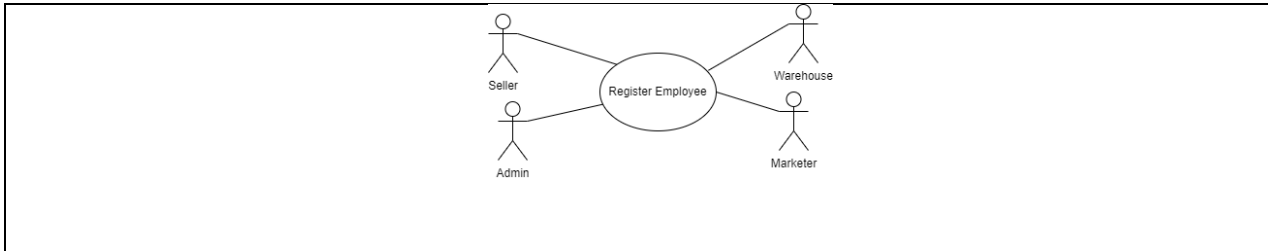


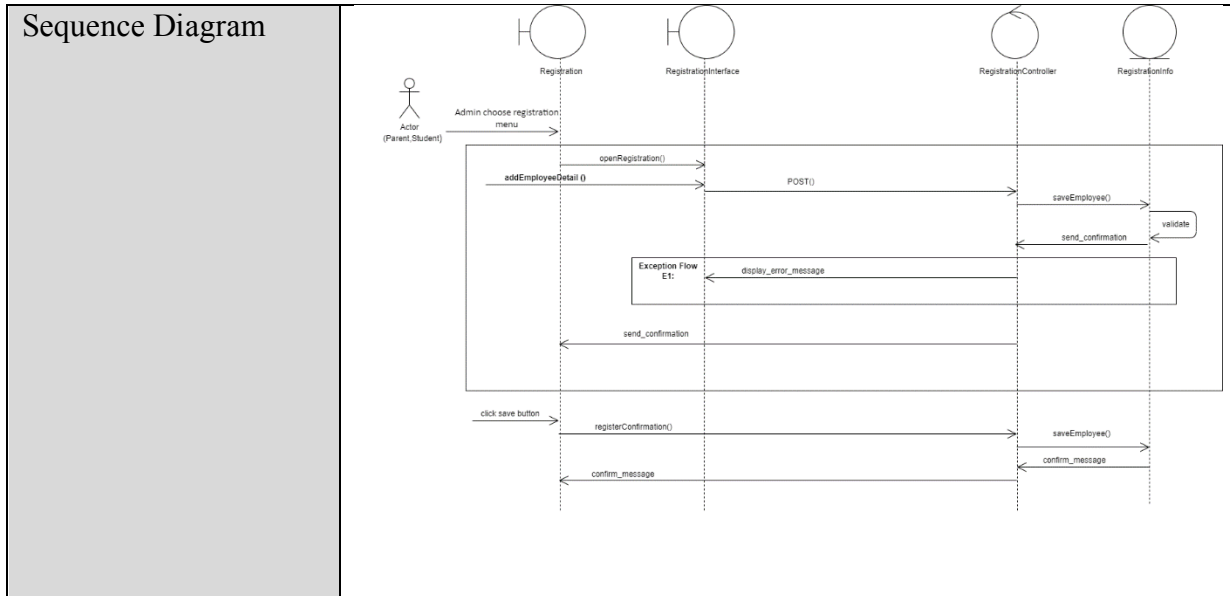
Figure 2: Register Employee

Table 3: Use Case Description for Register Employee

Use Case ID	POS-UC002
Use Case Name	Register Employee
Brief Description	This use case is initiated by admin. It allowed the admin to register employees which are seller, marketer and warehouse staff into the SuperProPOS: Product Ordering System.
Actor	Admin, Seller, Marketer and Warehouse
Pre-Condition	<ul style="list-style-type: none"> <li>• The admin already has a registered account in the system.</li> <li>• The admin already login to the system.</li> <li>• The admin must have internet connection.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>1. The use case starts when the admin goes to employee menu page.</li> <li>2. The system displays the employee list page by category seller, marketer and warehouse staff,</li> <li>3. The admin chooses Register New Seller, Register New Marketer and Register New Warehouse staff.</li> <li>4. The system then displays the registration form page.</li> </ol>

SuperProPOS: Product Ordering System

	<ol style="list-style-type: none"> <li>5. The admin inserts employee information in the registration form. <b>[E1: Email already existed]</b></li> <li>6. The admin clicks &lt;&lt;SUBMIT&gt;&gt; button.</li> <li>7. The system validates the employee information.</li> <li>8. The system displays the employees list information from the database.</li> <li>9. The admin is able to view the list of employee list by clicking on the Employee List menu and choose menus of seller list, marketer list and warehouse staff list.</li> <li>10. The seller, marketer and warehouse are able to view their registered profile account after login to the system.</li> <li>11. The use case ends.</li> </ol>
Alternative Flow	None
Exception Flow	<p>E1: Email Already Existed</p> <ol style="list-style-type: none"> <li>1. The admin inserts a new email for registration.</li> <li>2. The system checks the non-existence of similar email.</li> <li>3. The use case continues to step 6 in basic flow.</li> </ol>
Post Conditions	The seller, marketer and warehouse staff are able to login and access to the SuperProPOS: Product Ordering System.
Rules	None
Constraints	None



### 3.3 Manage User Profile

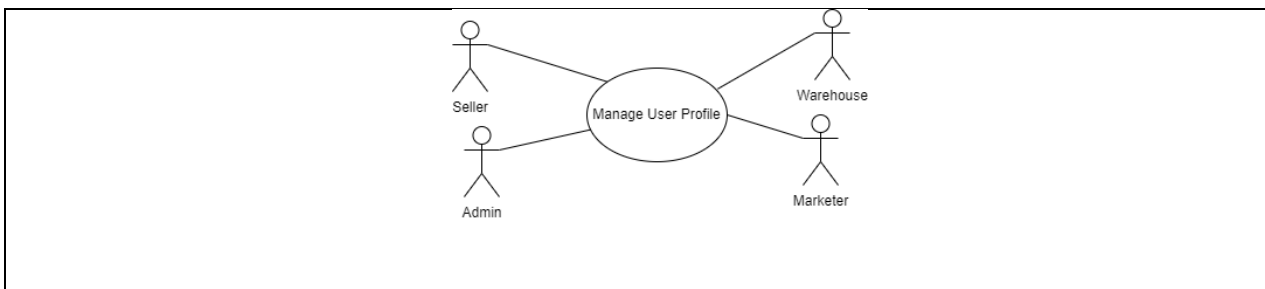


Figure 3: Manage User Profile

Table 4: Use Case Description for Manage User Profile

Use Case ID	POS-UC003
Use Case Name	Manage User Profile
Brief Description	<p>This use case is initiated by seller, admin, marketer, warehouse staff and user. It allowed the admin, seller, marketer and warehouse staff to update their profile detail.</p> <p>The admin also can update and delete seller, marketer and warehouse staff detail from the database.</p> <p>The marketer can view all their assigned seller detail made by the admin.</p>
Actor	Seller, Admin, Marketer and Warehouse staff.
Pre-Condition	<ul style="list-style-type: none"> <li>• The seller, admin, warehouse staff and marketer already have a registered account in the system.</li> <li>• The seller, warehouse staff, admin and marketer already login to the system.</li> <li>• The seller, warehouse staff, admin, marketer and user must have internet connection.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>1. The use case starts when the seller, warehouse staff, marketer and admin go to update profile detail page.</li> <li>2. The system displays the user profile detail.</li> <li>3. The admin, seller, marketer and warehouse staff update their own name, phone number and password details in the system.</li> <li>4. While the admin is able to:             <ol style="list-style-type: none"> <li>a. Update the name, phone number, marketer ID and password of seller, marketer and</li> </ol> </li> </ol>

	<p>warehouse staff profile detail. [<b>A1: Update Employee Profile</b>].</p> <p><b>b.</b> Delete seller, marketer and warehouse staff profile detail [<b>A2: Delete Employee Profile</b>]</p> <ol style="list-style-type: none"> <li>5. The seller, warehouse staff, marketer and admin click &lt;&lt;SAVE&gt;&gt; button.</li> <li>6. The system validates the user's profile information.</li> <li>7. The system displays successful message to the seller, marketer, warehouse staff and admin.</li> <li>8. The system list all the employees profile details from the database.</li> <li>9. The marketer and admin are able to view the updated sellers profile details by click on the Seller List menu.</li> <li>10. The admin able to view the updated marketer profile details by click on the Marketer List menu.</li> <li>11. The admin able to view the updated warehouse staff profile details by click on the Warehouse staff List menu.</li> <li>12. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1: Update Employee Profile</b></p> <ol style="list-style-type: none"> <li>1. The admin chooses the employee profile detail that he/she already done it before this.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The admin clicks &lt;&lt;UPDATE&gt;&gt; button.</li> <li>3. The admin updates the employee profile detail that he/she key-in before this.</li> <li>4. The use case continues to step 5 in basic flow.</li> </ol> <p><b>A2: Delete Employee Profile</b></p> <ol style="list-style-type: none"> <li>1. The admin chooses the employee that he/she already done it before this.</li> <li>2. The admin clicks &lt;&lt;DELETE&gt;&gt; button.</li> <li>3. The use case continues to step 5 in basic flow.</li> </ol>
Exception Flow	None
Post Conditions	The list of employee profile detail is successfully updated.
Rules	None
Constraints	None

### 3.4 Manage Product Detail

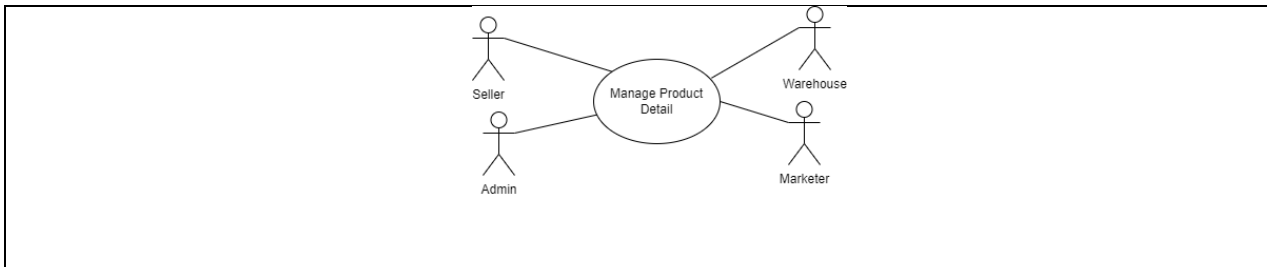


Figure 6: Use Case Manage Product Detail

Table 5: Use Case Description for Manage Product Detail

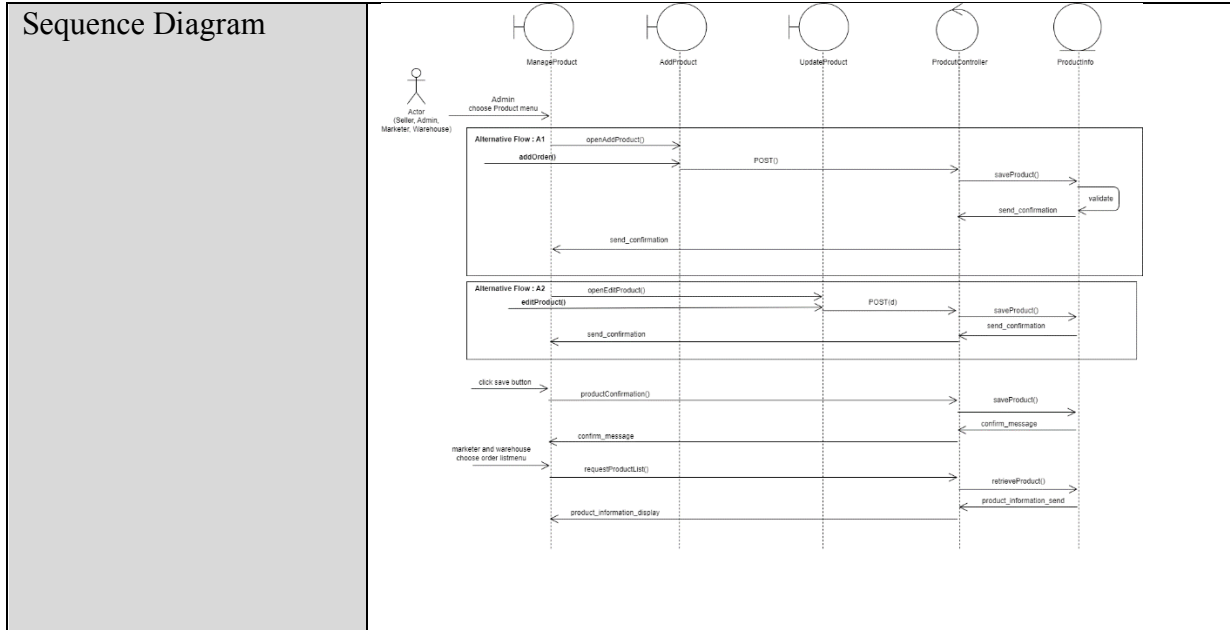
Use Case ID	POS-UC004
Use Case Name	Manage Product Detail
Brief Description	<p>This use case is initiated by admin, seller, marketer warehouse staff and user. It allowed the admin to add a product by categories. The admin also can update and delete the product information.</p> <p>The warehouse staff can update the quantity and status of product stock in the product information.</p> <p>The marketer and seller can view all the products information from the database.</p>
Actor	Admin, Warehouse staff, Seller and Marketer.
Pre-Condition	<ul style="list-style-type: none"> <li>• The admin, warehouse staff, seller and marketer already have a registered account in the system.</li> <li>• The admin, warehouse staff, seller and marketer already login to the system.</li> <li>• The admin, warehouse staff, seller, user and marketer must have internet connection.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>1. The use case starts when the admin goes to product list menu page.</li> </ol>



	<ol style="list-style-type: none"><li>2. The system displays the product list detail.</li><li>3. The admin is able to:<ol style="list-style-type: none"><li>a. Add new product by name, information, price, weight, image, document and category to system [<b>A1: Add Product</b>]</li><li>b. Delete product from the system [<b>A2: Delete Product</b>].</li><li>c. Update product's name, information, price, weight, image, document and category in the system [<b>A3: Update Product</b>]</li></ol></li><li>4. The admin clicks &lt;&lt;SAVE&gt;&gt; button.</li><li>5. The system validates the products information.</li><li>6. The warehouse staff is able to update the quantity and status of product stocks in the database.</li><li>7. The system validates the products information.</li><li>8. The system displays successful message to the warehouse staff and admin.</li><li>9. The seller and marketer are able to view the updated products details by click on the Product menu.</li><li>10. The system list all the products details from the database.</li><li>11. The use case ends.</li></ol>
--	--

Alternative Flow	<p><b>A1: Add Product</b></p> <ol style="list-style-type: none"> <li>1. The admin goes to add new product menu.</li> <li>2. The admin inserts the new product by name, information, price, weight, image, document and category to system.</li> <li>3. The admin clicks &lt;&lt;SUBMIT&gt;&gt; button.</li> <li>4. The use case continues to step 4 in basic flow.</li> </ol> <p><b>A2: Delete Product</b></p> <ol style="list-style-type: none"> <li>1. The admin chooses the product that he/she already done it before this.</li> <li>2. The admin clicks &lt;&lt;DELETE&gt;&gt; button.</li> <li>3. The use case continues to step 4 in basic flow.</li> </ol> <p><b>A3: Update Product</b></p> <ol style="list-style-type: none"> <li>1. The admin chooses the product detail that he/she already done it before this.</li> <li>2. The admin clicks &lt;&lt;UPDATE&gt;&gt; button.</li> <li>3. The admin updates the product detail that he/she key-in before this.</li> <li>4. The use case continues to step 4 in basic flow.</li> </ol>
Exception Flow	None
Post Conditions	The list of products is successfully updated.
Rules	None
Constraints	None

# SuperProPOS: Product Ordering System



### 3.5 Manage Order

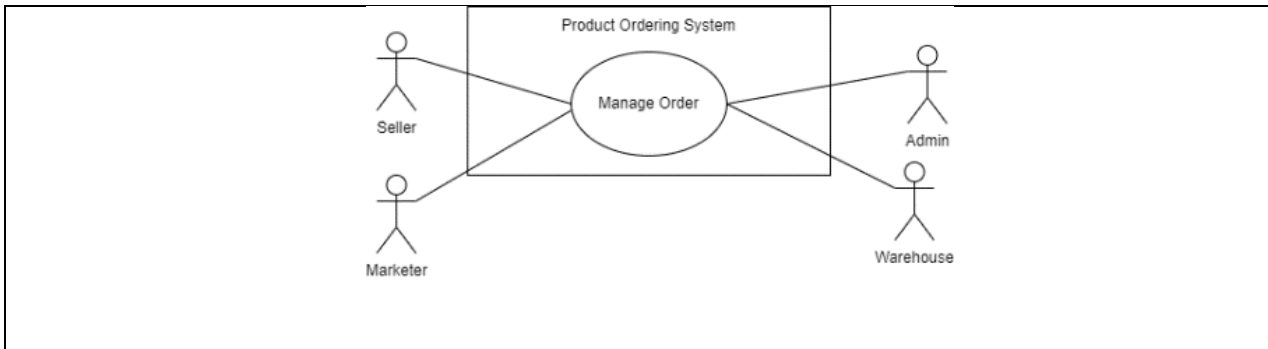


Figure 5: Manage Order

Table 6: Use Case Description for Manage Order

Use Case ID	POS-UC005
Use Case Name	Manage Order
Brief Description	<p>This use case is initiated by seller, admin, marketer and warehouse staff. It allowed the seller to make and order for the product. The seller also can edit the order information.</p> <p>The admin can view and update the order payment status from the order list.</p> <p>The warehouse staff can view and update the order status and tracking number after the payment has been approved by admin from the order list.</p> <p>The marketer can view all the order list information from their assigned seller from the database.</p>
Actor	Admin, Warehouse staff, Seller and Marketer.
Pre-Condition	<ul style="list-style-type: none"> <li>• The admin, warehouse staff, seller and marketer already have a registered account in the system.</li> <li>• The admin, warehouse staff, seller and marketer already login to the system.</li> </ul>

	<ul style="list-style-type: none"> <li>• The admin, warehouse staff, seller and marketer must have internet connection.</li> </ul>
<p>Basic Flow</p>	<ol style="list-style-type: none"> <li>1. The use case starts when the seller goes to Order Now menu page.</li> <li>2. The system displays the products to be ordered by seller.</li> <li>3. The seller clicks &lt;&lt;ADD TO CART&gt;&gt; button on the product.</li> <li>4. The system submits the product to the cart.</li> <li>5. The seller clicks the &lt;&lt;CART&gt;&gt; button to go to the cart page.</li> <li>6. The system displays the chosen products details and its total price and payment that need to be made.</li> <li>7. The seller is able to             <ol style="list-style-type: none"> <li>a. Update quantity of products in cart. [<b>A1: Update Cart</b>]</li> <li>b. Delete the selected product from cart. [<b>A2: Delete Cart</b>]</li> </ol> </li> <li>8. The seller then clicks on the &lt;&lt;Proceed to Check Out&gt;&gt; button.</li> <li>9. The system displays the total price, total products and order form.</li> <li>10. The seller is able to:</li> </ol>

	<ul style="list-style-type: none"> <li>a. Add order details choose existing customer details, payment method and upload the receipt image to the system [<b>A3: Add Order</b>]</li> <li>b. Update order details in the system. [<b>A4: Edit Order</b>]</li> </ul> <ol style="list-style-type: none"> <li>11. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</li> <li>12. The system validates the order information.</li> <li>13. The system displays successful message to the seller.</li> <li>14. The admin is able to update the status of order payment in the database.</li> <li>15. The warehouse staff is able to update the delivery status and tracking number of order after admin approved the payment in the database.</li> <li>16. The seller, marketer, warehouse staff and admin are able to view the updated order details by click on the Order List menu.</li> <li>17. The system list all the order details from the database.</li> <li>18. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1: Update Cart</b></p> <ol style="list-style-type: none"> <li>1. The seller inserts the quantity of selected products.</li> <li>2. The seller clicks the &lt;&lt;UPDATE&gt;&gt; button.</li> <li>3. The use case continues to use case 8 in basic flow.</li> </ol>

	<p><b>A2: Delete Cart</b></p> <ol style="list-style-type: none"><li>1. The seller chooses the existing products in cart.</li><li>2. The seller clicks &lt;&lt;DELETE&gt;&gt; button.</li><li>3. The use case continues to use case 8 in basic flow.</li></ol> <p><b>A3: Add Order</b></p> <ol style="list-style-type: none"><li>1. The seller chooses the existing customer from the list. <b>[A3.1 Add New Customer]</b></li><li>2. The seller inserts the payment method and upload the receipt image to the system.</li><li>3. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</li><li>4. The use case continues to step 11 in basic flow.</li></ol> <p><b>A4: Edit Order</b></p> <ol style="list-style-type: none"><li>1. The seller chooses the order detail that he/she already done it before this.</li><li>2. The seller clicks &lt;&lt;EDIT&gt;&gt; button.</li><li>3. The seller edits the order's payment receipt image that he/she key-in before this.</li><li>4. The use case continues to step 11 in basic flow.</li></ol> <p><b>A3.1 Add New Customer</b></p> <ol style="list-style-type: none"><li>1. The seller clicks on the &lt;&lt;+NEW CUSTOMER&gt;&gt; button.</li></ol>
--	---

	<ol style="list-style-type: none"> <li>2. The seller inserts customer name, phone number and address to the system.</li> <li>3. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</li> <li>4. The use case continues to step 1 in alternative flow A3.</li> </ol>
Exception Flow	None
Post Conditions	The list of order is successfully updated.
Rules	None
Constraints	None
Sequence Diagram	<pre> sequenceDiagram     actor Actor as Actor (Seller, Admin, Marketer, Warehouse)     participant ManageOrder     participant AddOrder     participant UpdateOrder     participant OrderController     participant OrderInfo      Actor-&gt;&gt;ManageOrder: Seller chooses Shop menu     activate ManageOrder     alt Alternative Flow: A1         ManageOrder-&gt;&gt;AddOrder: openShop()         activate AddOrder         AddOrder-&gt;&gt;OrderController: POST()         activate OrderController         OrderController-&gt;&gt;OrderInfo: saveOrder()         activate OrderInfo         OrderInfo-&gt;&gt;OrderInfo: validate()         OrderInfo--&gt;&gt;OrderController: send_confirmation         deactivate OrderInfo         OrderController--&gt;&gt;AddOrder: send_confirmation         deactivate OrderController         AddOrder--&gt;&gt;ManageOrder: send_confirmation         deactivate AddOrder     else Alternative Flow: A2         ManageOrder-&gt;&gt;UpdateOrder: openEditOrder()         activate UpdateOrder         UpdateOrder-&gt;&gt;OrderController: POST()         activate OrderController         OrderController-&gt;&gt;OrderInfo: saveOrder()         activate OrderInfo         OrderInfo--&gt;&gt;OrderController: send_confirmation         deactivate OrderInfo         OrderController--&gt;&gt;UpdateOrder: send_confirmation         deactivate OrderController         UpdateOrder--&gt;&gt;ManageOrder: send_confirmation         deactivate UpdateOrder     end     deactivate ManageOrder     Actor-&gt;&gt;OrderController: click save button     activate OrderController     OrderController-&gt;&gt;ManageOrder: orderConfirmation()     deactivate OrderController     OrderController-&gt;&gt;OrderInfo: saveOrder()     activate OrderInfo     OrderInfo--&gt;&gt;OrderController: confirm_message     deactivate OrderInfo     OrderController--&gt;&gt;ManageOrder: confirm_message     deactivate OrderController     Actor-&gt;&gt;OrderController: admin, marketer and warehouse choose order list menu     activate OrderController     OrderController-&gt;&gt;ManageOrder: requestOrderList()     deactivate OrderController     OrderController-&gt;&gt;OrderInfo: retrieveOrder()     activate OrderInfo     OrderInfo--&gt;&gt;OrderController: order_information_send     deactivate OrderInfo     OrderController--&gt;&gt;ManageOrder: order_information_display     deactivate OrderController     </pre>



### 3.6 Manage Complaint

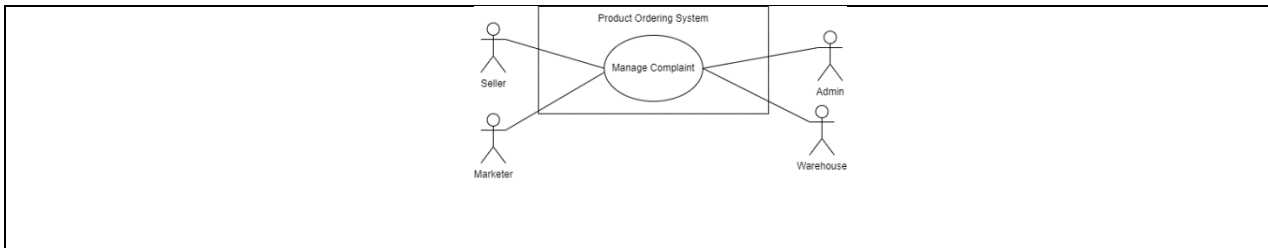


Figure 6: Manage Complaint

Table 7: Use Case Description for Manage Complaint

Use Case ID	POS-UC006
Use Case Name	Manage Complaint
Brief Description	<p>This use case is initiated by seller, admin, warehouse staff and marketer. It allowed the seller to add a complaint for the order in the system. The seller also can edit the complaint information.</p> <p>The admin and warehouse staff can view and update the status of complaint in the complaint information.</p> <p>The marketer can view the complaint information of their assigned seller from the database.</p>
Actor	Admin, Warehouse staff, Seller and Marketer.
Pre-Condition	<ul style="list-style-type: none"> <li>• The admin, warehouse staff, seller and marketer already have a registered account in the system.</li> <li>• The admin, warehouse staff, seller and marketer already login to the system.</li> <li>• The admin, warehouse staff, seller and marketer must have internet connection.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>1. The use case starts when the seller goes to complaint menu page.</li> <li>2. The system displays the complaint list detail.</li> </ol>

	<ol style="list-style-type: none"> <li>3. The seller is able to:             <ol style="list-style-type: none"> <li>a. Add new complaint by order id, problem, details of problem and image in the system. <b>[A1: Add Complaint]</b></li> <li>b. Edit the complaint problem, details of problem and image in the system <b>[A2: Edit Complaint]</b></li> </ol> </li> <li>4. The seller clicks &lt;&lt;SAVE&gt;&gt; button.</li> <li>5. The system validates the complaint information.</li> <li>6. The system displays successful message to the seller.</li> <li>7. The admin and warehouse staff are able to update the status of complaint in the database.</li> <li>8. The marketer is able to view the complaint list made by their assigned sellers.</li> <li>9. The system list all the complaint details from the database.</li> <li>10. The use case ends.</li> </ol>
Alternative Flow	<p><b>A1: Add Complaint</b></p> <ol style="list-style-type: none"> <li>1. The seller goes to Add New Complaint menu.</li> <li>2. The seller inserts the order id, problem, details of problem and image in the system.</li> <li>3. The seller clicks &lt;&lt;SUBMIT&gt;&gt; button.</li> <li>4. The use case continues to step 4 in basic flow.</li> </ol>

	<p><b>A2: Edit Complaint</b></p> <ol style="list-style-type: none"> <li>1. The seller chooses the complaint detail that he/she already done it before this.</li> <li>2. The seller clicks &lt;&lt;EDIT&gt;&gt; button.</li> <li>3. The seller edits the complaint detail that he/she key-in before this.</li> <li>4. The use case continues to step 4 in basic flow.</li> </ol>
Exception Flow	None
Post Conditions	The list of complaints is successfully updated.
Rules	None
Constraints	None
Sequence Diagram	<pre> sequenceDiagram     actor Actor as Actor (Seller, Admin, Manager, Warehouse)     participant ManageComplaint     participant AddComplaint     participant EditComplaint     participant ComplaintController     participant ComplaintInfo      Actor-&gt;&gt;ManageComplaint: Seller choose Complaint menu     activate ManageComplaint     ManageComplaint-&gt;&gt;AddComplaint: openAddComplaint()     activate AddComplaint     AddComplaint-&gt;&gt;ComplaintController: POST()     activate ComplaintController     ComplaintController-&gt;&gt;ComplaintInfo: saveComplaint()     activate ComplaintInfo     ComplaintInfo-&gt;&gt;ComplaintController: validate     deactivate ComplaintInfo     ComplaintController-&gt;&gt;AddComplaint: send_confirmation     deactivate ComplaintController     AddComplaint-&gt;&gt;ManageComplaint: send_confirmation     deactivate AddComplaint     deactivate ManageComplaint      %% Alternative Flow A2     ManageComplaint-&gt;&gt;EditComplaint: openEditComplaint()     activate EditComplaint     EditComplaint-&gt;&gt;ComplaintController: POST()     activate ComplaintController     ComplaintController-&gt;&gt;ComplaintInfo: saveComplaint()     activate ComplaintInfo     ComplaintInfo-&gt;&gt;ComplaintController: send_confirmation     deactivate ComplaintInfo     ComplaintController-&gt;&gt;EditComplaint: send_confirmation     deactivate ComplaintController     EditComplaint-&gt;&gt;ManageComplaint: send_confirmation     deactivate EditComplaint     deactivate ManageComplaint      %% Final Steps     Actor-&gt;&gt;ManageComplaint: click save button     activate ManageComplaint     ManageComplaint-&gt;&gt;ComplaintController: complaintConfirmation()     activate ComplaintController     ComplaintController-&gt;&gt;ComplaintInfo: saveComplaint()     activate ComplaintInfo     ComplaintInfo-&gt;&gt;ComplaintController: confirm_message     deactivate ComplaintInfo     ComplaintController-&gt;&gt;Actor: confirm_message     deactivate ComplaintController     Actor-&gt;&gt;ManageComplaint: admin, manager and warehouse choose complain list menu     activate ManageComplaint     ManageComplaint-&gt;&gt;ComplaintController: requestComplaintList()     activate ComplaintController     ComplaintController-&gt;&gt;ComplaintInfo: retrieveComplaint()     activate ComplaintInfo     ComplaintInfo-&gt;&gt;ComplaintController: complaint_information_send     deactivate ComplaintInfo     ComplaintController-&gt;&gt;Actor: product_information_display     deactivate ComplaintController     deactivate ManageComplaint     </pre>

### 3.7 Generate Report

Table 8: Use Case Description for Generate Report

Use Case ID	POS-UC007
Use Case Name	Generate Report
Brief Description	<p>This use case describes the process of generating sales reports for different user roles within the system.</p> <p>Sellers, marketers, and admins have the ability to view sales reports based on their specific access rights and requirements.</p> <p>The reports can be generated on a daily, monthly, or yearly basis.</p>
Actor	Admin, Seller and Marketer.
Pre-Condition	<ul style="list-style-type: none"> <li>• The admin, seller and marketer already have a registered account in the system.</li> <li>• The admin, seller and marketer already login to the system.</li> <li>• The admin, seller and marketer must have internet connection.</li> <li>• Sales data must be available and up to date in the system.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>1. The use case starts when the seller, admin and marketer go to Sales Report menu page.</li> <li>2. The system presents options to generate sales reports by daily, monthly, or yearly intervals.</li> </ol>

	<ol style="list-style-type: none"><li>3. The seller is able to:<ol style="list-style-type: none"><li>a. selects the desired time interval (daily, monthly, or yearly).</li><li>b. The system retrieves the sales data for the selected interval and generates a sales report specific to the Seller's orders within that interval.</li><li>c. The Seller views and analyses the generated sales report.</li></ol></li><li>4. The marketer is able to:<ol style="list-style-type: none"><li>a. Choose to view their assigned seller Sales Report by selects the desired time interval (daily, monthly, or yearly).</li><li>b. The system retrieves the sales data for the selected interval and generates a sales report specific to the Marketer's assigned Sellers' orders within that interval.</li><li>c. The Marketer views and analyses the generated sales report.</li></ol></li><li>5. The admin is able to:<ol style="list-style-type: none"><li>a. Selects the desired time interval (daily, monthly, or yearly).</li></ol></li></ol>
--	--

	<p>b. The system retrieves the sales data for the selected interval and generates an overall sales report based on all Sellers' orders within that interval.</p> <p>c. The admin views and analyses the generated sales report.</p> <p>6. The use case ends.</p>
Alternative Flow	None
Exception Flow	None
Post Conditions	<p>The sales report is displayed to the respective user (Seller, Marketer, or Admin) based on their access rights and specific requirements.</p> <p>The user can analyse the sales report to gain insights into their sales performance within the selected time interval.</p>
Rules	None
Constraints	None

#### 4.0 REQUIREMENT TRACEABILITY

This section shall contain:

- a. Traceability from each software unit identified in this SRS of the system requirements allocated to it.
- b. Traceability from each system requirement for the software units to which it is allocated.

Table 4: Requirement Traceability

Requirement	Description
POS-UC001	Login  Provide capability of users to login in the system
POS-UC002	Register Employee  Provides the capability of admin to register seller, marketer and warehouse staff to system.
POS-UC003	Manage User Profile  Provide the capability of seller, marketer warehouse staff and admin to update their profile. The marketer also can view the seller's profile. The admin can view, update and delete marketer, seller and warehouse staff profile.
POS-UC004	Manage Product Detail  Provide the capability of admin to add and update the product detail. The warehouse staff also can update the product stock detail. The seller and marketer can view the product list.
POS-UC005	Manage Order

SuperProPOS: Product Ordering System

	<p>Provide the capability of seller to add and edit the order detail. The warehouse staff and admin also can update the order status. The marketer can view the order list.</p>
POS-UC006	<p>Manage Complaint</p> <p>Provide the capability of seller to add and update the complaint detail. The warehouse staff and admin also can update the status complaint. The marketer can view the complaint list.</p>
POS-UC007	<p>Generate Report</p> <p>Provide the capability of admin, seller and marketer to view the sales report performance by daily, monthly and yearly in list and graph format.</p>



**5.0 SYSTEM REQUIREMENTS APPROVAL**

	<b>Name</b>	<b>Date</b>
<p><b>Verified by:</b></p> <p><i>Rina.</i> _____</p> <p>Developer</p>	RINA BINTI ISHAK	15 FEBRUARY 2023
<p><b>Approved by:</b></p> <p><i>Zaid.</i> _____</p> <p>Client</p>	ZAID BIN ISHAK	15 FEBRUARY 2023

SuperProPOS: Product Ordering System

Appendix B: Software Design Document

**Version**

**1**

**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

---

Faculty of Computing

**Software Design  
Document (SDD)**

# Table of Contents

1.0	DATA DICTIONARY	1
2.0	PRELIMINARY DESIGN	2
	2.1 System Architecture	
	2.1.1 Static Organization	
	2.1.2 Dynamic Organization	
3.0	DETAILED DESIGN	4
	3.1 Package Name	
4.0	SYSTEM DESIGN APPROVAL	6

## 1.0 DATA DICTIONARY

Define set of tables that provides information about the database. The data dictionary example is shows in Table 1.1.

**Table 1.1: Data Dictionary for table SuperProPOS: Product Ordering System.**

### 1.1 admin

Field Name	Description	Data Type	Constraint
Admin_Id	Admin ID	INT (10)	PK
A_Name	Admin Name	VARCHAR (200)	
A_PhoneNum	Admin Phone Number	VARCHAR (11)	
A_Email	Admin Email	VARCHAR (100)	
A_Password	Admin Password	VARCHAR (20)	

### 1.2 seller

Field Name	Description	Data Type	Constraint
Seller_Id	Seller ID	INT (10)	PK
Marketer_Id	Marketer ID	INT (10)	FK
S__Name	Seller Name	VARCHAR (200)	
S_PhoneNumber	Seller Phone Number	VARCHAR (11)	
S_Email	Seller Email	VARCHAR (100)	

### 1.3 marketer

Field Name	Description	Data Type	Constraint
Marketer_Id	Marketer ID	INT (10)	PK
M_Name	Marketer Name	VARCHAR (200)	
M_PhoneNum	Marketer Phone Number	VARCHAR (11)	
M_Email	Marketer Email	VARCHAR (100)	
M_Password	Marketer Password	VARCHAR (20)	

### 1.4 warehouse

Field Name	Description	Data Type	Constraint
Warehouse staff_Id	Warehouse staff ID	INT (10)	PK
W__Name	Warehouse staff Name	VARCHAR (200)	
W_PhoneNum	Warehouse staff Phone Number	VARCHAR (12)	
W__Email	Warehouse staff Email	VARCHAR (100)	

**1.5 customer**

Field Name	Description	Data Type	Constraint
Customer_Id	Customer ID	INT (10)	PK
Seller_Id	Seller ID	INT (10)	FK
C_Name	Customer Name	VARCHAR (200)	
C_PhoneNum	Customer Phone Number	VARCHAR (11)	
C_Flat	Customer Flat Address	VARCHAR (255)	
C_Street	Customer Street Address	VARCHAR (255)	
C_City	Customer City Address	VARCHAR (255)	
C_State	Customer State Address	VARCHAR (255)	
C_Country	Customer Country Address	VARCHAR (255)	
C_Postcode	Customer Postcode Address	VARCHAR (255)	
C_Date	Customer Information Date Inserted	VARCHAR (10)	

**1.6 product**

Field Name	Description	Data Type	Constraint
Product_Id	Product ID	INT (11)	PK
P_Name	Product Name	VARCHAR (200)	
P_Info	Product Info	VARCHAR (500)	
P_Price	Product Price	DECIMAL (7,2)	
P_Weight	Product Weight	FLOAT	
P_Image	Product Image	VARCHAR (500)	
P_Doc	Product Document	VARCHAR (500)	
P_Status	Product Status	VARCHAR (20)	
P_Category	Product Category	VARCHAR (20)	
P_Stock	Product Stock	INT (11)	

**1.7 orders**

Field Name	Description	Data Type	Constraint
Order_Id	Order ID	INT (255)	PK
Seller_Id	Seller ID	INT (12)	FK
C_Name	Customer Name	VARCHAR (255)	
C_PhoneNum	Customer Phone Number	VARCHAR (255)	
O_PayMethod	Payment Method	VARCHAR (255)	
O_Bank	Bank Name	VARCHAR (255)	
O_ReclImage	Receipt Image	VARCHAR (255)	
O_RefRec	Receipt Reference	VARCHAR (255)	
O_Date	Receipt Payment Date	DATE	
O_Time	Receipt Payment Time	TIME	
C_Flat	Customer Flat Address	VARCHAR (255)	
C_Street	Customer Street Address	VARCHAR (255)	
C_City	Customer City Address	VARCHAR (255)	
C_State	Customer State Address	VARCHAR (255)	
C_Country	Customer Country Address	VARCHAR (255)	
C_Postcode	Customer Postcode Address	VARCHAR (255)	
total_products	Total Product Ordered	VARCHAR (255)	
total_price	Total Price Paid	DECIMAL (7,2)	
O_ShipMethod	Order Shipment Method	VARCHAR (255)	



O_Status	Order Payment Status	VARCHAR (255)	
O_DeliverStatus	Order Delivery Status	VARCHAR (255)	
O_Tracking	Order Tracking Number	VARCHAR (255)	

### 1.8 complaint

Field Name	Description	Data Type	Constraint
Complaint_Id	Complaint ID	VARCHAR (10)	PK
Order_Id	Order_ID	VARCHAR (10)	FK
Seller_Id	Seller ID	VARCHAR (10)	FK
Comp_Prob	Complaint Problem	VARCHAR (20)	
Comp_Info	Complaint Info	VARCHAR (200)	
Comp_Date	Complaint Date	DATE	
Comp_Image	Complaint Image	VARCHAR (500)	

### 1.9 cart

Field Name	Description	Data Type	Constraint
Cart_Id	Cart ID	INT (11)	PK
Seller_Id	Seller ID	INT (255)	FK
P_Name	Product Name	VARCHAR (255)	
P_Price	Product Price	DECIMAL (7,2)	
P_Image	Product Image	VARCHAR (255)	
P_Quantity	Product Quantity	VARCHAR (255)	

## 2.0 PRELIMINARY DESIGN

### 2.1 System Architecture

This paragraph identifies the internal organizational the SuperProPOS: Product Ordering System (POS). The relationship among system subsystem will be described.

#### 2.1.1 Static Organization

Figure 2.1 shows the static organization of SuperProPOS: Product Ordering System. It consists of:

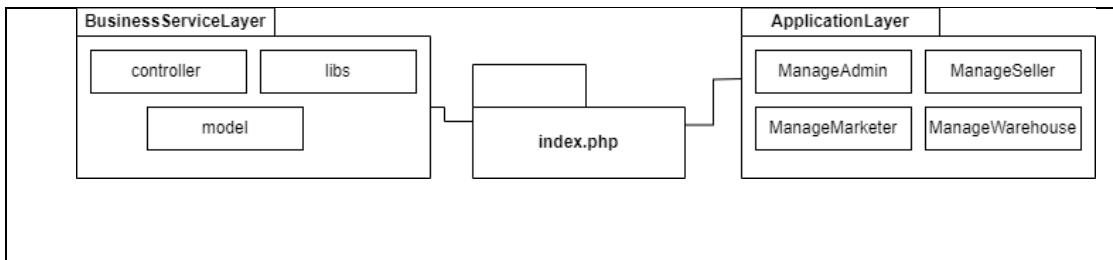


Figure 2.1: Static Organization of SuperProPOS: Product Ordering System

This section describes the detail for each subsystem/package.

#### 1. ManageAdmin

This package is responsible to for all function for admin when interact with SuperProPOS: Product Ordering System. This package consists of the following classes or unit:

- a. addEmployeeMarketerAdmin Class
- b. addEmployeeSellerAdmin Class
- c. addEmployeeWarehouse staffAdmin Class
- d. addNewProductAdmin Class
- e. adminHome Class
- f. adminLogout Class
- g. listComplaintAdmin Class
- h. listCustomerAdmin Class
- i. listMarketerAdmin Class
- j. listOrderAdmin Class
- k. listProductAdmin Class
- l. listSellerAdmin Class

- m. listWarehouse staffAdmin Class
- n. loginAdmin Class
- o. updateAdminProfile Class
- p. updateComplaintStatusAdmin Class
- q. updateMarketerProfileAdmin Class
- r. updateNewProductAdmin Class
- s. updateOrderStatusAdmin Class
- t. updateSellerProfileAdmin Class
- u. updateWarehouse staffrProfileAdmin Class
- v. viewCustomerAdmin Class
- w. viewDailyReportSalesAdmin Classs
- x. viewMonthlyReportSalesAdmin Class
- y. viewYearlyReportSaleAdmin Class

## **2. ManageMarketer**

This package is responsible to for all function for admin when interact with SuperProPOS: Product Ordering System. This package consists of the following classes or unit:

- a. listComplaintMarketer Class
- b. listCustomerMarketer Class
- c. listOrderMarketer Class
- d. listProductMarketer Class
- e. listSalesMarketer Class
- f. marketerHome Class
- g. marketerLogin Class
- h. marketerLogout Class
- i. updateProfileMarketer Class
- j. viewComplaintStatusMarketer Class
- k. viewCustomerMarketer Class
- l. viewDailySaleMarketer Class
- m. viewMonthlySaleMarketer Class
- n. viewOrderStatusMarketer Class
- o. viewProductMarketer Classs

- p. viewSellerProfileMarketer Class
- q. viewYearlySaleMarketer Class

### **3. ManageSeller**

This package is responsible to for all function for seller when interact with SuperProPOS: Product Ordering System. This package consists of the following classes or unit:

- a. addComplaintSeller Class
- b. addCustomerSeller Class
- c. addOrderSeller Class
- d. cartOrderSeller Class
- e. cartSeller Class
- f. checkoutSeller Class
- g. header Class
- h. listComplaintSeller Class
- i. listCustomerSeller Class
- j. listOrderSeller Class
- k. listProductSeller Class
- l. sellerHome Class
- m. sellerLogin Class
- n. sellerLogout Class
- o. updateComplaintStatusSeller Class
- p. updateCustomerSeller Class
- q. updateOrderStatusSeller Class
- r. updateProfileSeller Class
- s. viewDailyReportSeller Class
- t. viewMonthlyReportSeller Class
- u. viewProductSeller Class
- v. viewYearlyReportSeller Class

#### **4. ManageWarehouse**

This package is responsible to for all function for warehouse staff when interact with SuperProPOS: Product Ordering System. This package consists of the following classes or unit:

- a. listComplaintWH Class
- b. listOrderWh Class
- c. listProductWh Class
- d. updateComplaintStatusWh Class
- e. updateOrderStatusWh Class
- f. updateProductWh Class
- g. updateProfileWh Class
- h. whHome Class
- i. whLogin Class
- j. whLogout Class

#### **5. controller**

- a. adminController Class
- b. complaintController Class
- c. customerController Class
- d. marketersController Class
- e. warehouse staffController Class
- f. orderController Class
- g. productController Class
- h. reportController Class
- i. sellerController Class

#### **6. libs**

- a. database Class

#### **7. model**

- a. admin Class
- b. seller Class

## SuperProPOS: Product Ordering System

- c. marketer Class
- d. warehouse staff Class
- e. cart Class
- f. orders Class
- g. product Class
- h. complaint Class
- i. customer Class

#### 4.0 SYSTEM DESIGN APPROVAL

	Name	Date
<p><b>Verified by:</b></p> <p><i>Rina.</i> _____</p> <p>Developer</p>	RINA BINTI ISHAK	15 FEBRUARY 2023
<p><b>Approved by:</b></p> <p><i>Zaid.</i> _____</p> <p>Client</p>	ZAID BIN ISHAK	15 FEBRUARY 2023

SuperProPOS: Product Ordering System

Appendix C: User Acceptance Test (UAT) Form



Version

1

SUPERPROPOS: PRODUCT ORDERING SYSTEM

---

Faculty of Computing

# User Acceptance Test (UAT)

## Table of Contents

1.0	TESTING REPORT	1
2.0	SYSTEM TESTING APPROVAL	2

**SUPERPROPOS: PRODUCT ORDERING SYSTEM****1.0 TESTING REPORT**

The purpose of this section is to outline the User Acceptance Testing (UAT) process for the system. Approval of this testing implies that reviewers are confident that following the execution of the test plan, the resulting system will be considered fully-tested and eligible for implementation.

The chosen user will go through each of the instructions in the user manual. Any errors or problems found by the user must be noted on this form. The form is also needs to be signed by the user after the test is finished.

**1.1 Login**

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Admin login to the system</b>	Email:zaid@gmail.com Password: 0123456789	Succefully login	Succefully login	Pass	Good
<b>Seller login to system</b>	Email: <a href="mailto:liyasmn@gmail.com">liyasmn@gmail.com</a> Password: 0105304570	Succefully login	Succefully login	Pass	Good
<b>Marketer login to system</b>	Email: <a href="mailto:azhankhir@gmail.com">azhankhir@gmail.com</a> Password: 0173045649	Succefully login	Succefully login	Pass	Good
<b>Warehouse login to system</b>	Email: <a href="mailto:azri@gmail.com">azri@gmail.com</a> Password: 0123999977	Succefully login	Succefully login	Pass	Good

**1.2 Registration**

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Register seller</b>	Marketer Id:17 Name: Lia Yasmn Binti Ahmad.	Successfully registered	Successfully registered	Pass	Good

**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

	Phone Number: 0105304570 Email: <a href="mailto:liyasmn@gmail.com">liyasmn@gmail.com</a> Password: 0105304570				
<b>Register marketer</b>	Name: Muhammad Azhan Bin Khairi Phone Number: 0173045649 Email: <a href="mailto:azhankhir@gmail.com">azhankhir@gmail.com</a> Password: 0173045649	Successfully registered	Successfully registered	Pass	Good
<b>Register warehouse</b>	Name: Muhammad Azri Bin Ismail Phone Number: 0123999977 Email: <a href="mailto:azri@gmail.com">azri@gmail.com</a> Password: 0123999977	Successfully registered	Successfully registered	Pass	Good

**1.3 Manage Employee Profile**

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Update seller profile</b>	Marketer Id:17 Name: Lia Yasmen Binti Ahmad. Phone Number:	Updated in database	Updated in database	Pass	Able to update

**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

	0105304570 Email: <a href="mailto:liyasmn@gmail.com">liyasmn@gmail.com</a> Password: 0105304570				
<b>Update marketer profile</b>	Name: Muhammad Azhan Bin Khairi Phone Number: 0173045649 Email: <a href="mailto:azhankhir@gmail.com">azhankhir@gmail.com</a> Password: 0173045649	Updated in database	Updated in database	Pass	Able to update
<b>Update warehouse profile</b>	Name: Muhammad Azri Bin Ismail Phone Number: 0123999977 Email: <a href="mailto:azri@gmail.com">azri@gmail.com</a> Password: 0123999977	Updated in database	Updated in database	Pass	Able to update
<b>Update admin profile</b>	Name: Zaid Bin Ishak Phone Number: 0123456789 Email: <a href="mailto:zaid@gmail.com">zaid@gmail.com</a> Password: 0123456789	Updated in database	Updated in database	Pass	Able to update
<b>Delete seller</b>	None	Deleted from database	Deleted from	Pass	Able to delete

**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

			database database		
<b>Delete marketer</b>	None	Deleted from database	Deleted from database	Pass	Able to delete
<b>Delete warehouse</b>	None	Deleted from database	Deleted from database	Pass	delete

**1.4 Manage Product**

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Add product information</b>	Name: Fibre Choco Info: Minuman Berkhasiat coklat yang sedap dan baik untuk tubuh badan Price: 150.00 Weight:330 Image: fibre-choco.png Document: EBOOK FIBRE CHOCO.pdf Status: Available Category: Health Quantity: 20	Successfully added	Successfully added	Pass	Product Image and Product Document are needed. If not insert, message will ask to insert.

**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

<b>Update product information</b>	Name: Fibre Choco Info: Minuman Berkhasiat coklat yang sedap dan baik untuk tubuh badan Price: 150.00 Weight:330 Image: fibre-choco.png Document: EBOOK FIBRE CHOCO.pdf Status: Available Category: Health Quantity: 20	Updated in database	Updated in database	Pass	Good
<b>Delete product information</b>	-	Deleted from database	Deleted from database	Pass	Good
<b>View product list</b>	Name: Fibre Choco Info: Minuman Berkhasiat coklat yang sedap dan baik untuk tubuh badan Price: 150.00 Weight:330 Image: fibre-choco.png Document:	System display product list	System display product list	Pass	Good

**SUPERPROPOS:PRODUCT ORDERING SYSTEM**

	EBOOK FIBRE CHOCO.pdf Status: Available Category: Health Quantity: 20				
<b>View product information</b>	Name: Fibre Choco Info: Minuman Berkhasiat coklat yang sedap dan baik untuk tubuh badan Price: 150.00 Weight:330 Image: fibre-choco.png Document: EBOOK FIBRE CHOCO.pdf Status: Available Category: Health Quantity: 20	System display product detail	System display product detail	Pass	Good

**1.5 Manage Order**

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Add product to cart</b>	Product: Sweet G Price: 76.00 Quantity: 1	Product added to cart	Product added to cart	Pass	Okay
<b>Update</b>	Product: Sweet G	Product	Product	Pass	Okay



**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

<b>product in cart</b>	Price: 152.00 Quantity: 2	updated in cart	updated in cart		
<b>Delete product in cart</b>	-	Product deleted in cart	Product deleted in cart	Pass	Okay
<b>Add order form</b>	Name: Dina Binti Nadzir Phone: 0132233133 Pay method: online banking Bank: Ambank Receipt Image: receipt.png Rec. Reference: #Ref122316754 Date: 2023-06-10 Time: 14:16:00 Address: No 90 Jalan Tebu, Sabak Bernam, 45200, Selangor Malaysia	Successful add order	Successful add order	Pass	Okay. Maybe can add option for extra delivery payment.
<b>Update order information</b>	Name: Dina Binti Nadzir Phone: 0132233133 Pay method: online banking Bank: Ambank Receipt Image: receipt.png Rec. Reference:	Order information updated	Order information updated	Pass	okay

**SUPERPROPOS:PRODUCT ORDERING SYSTEM**

	#Ref122316754 Date: 2023-06-10 Time: 14:16:00 Address: No 90 Jalan Tebu, Sabak Bernam, 45200, Selangor Malaysia				
<b>View order list</b>	Name: Dina Binti Nadzir Phone: 0132233133 Pay method: online banking Bank: Ambank Receipt Image: receipt.png Rec. Reference: #Ref122316754 Date: 2023-06-10 Time: 14:16:00 Address: No 90 Jalan Tebu, Sabak Bernam, 45200, Selangor Malaysia	Order list displayed	Order list displayed	Pass	okay

### 1.6 Manage Payment

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Update receipt image</b>	Receipt Image: receipt.png	Succefully update receipt image	Succefully update receipt	Pass	Okay. Maybe can make payment link that can be generated.
<b>Update receipt reference</b>	Rec. Reference: #Ref122316754	Succefully update receipt reference	Succefully update receipt reference	Pass	okay
<b>Update receipt date</b>	Date: 2023-06-10	Succefully update receipt date	Succefully update receipt date	Pass	okay
<b>Update receipt time</b>	Time: 14:16:00	Succefully update receipt time	Succefully update receipt time	Pass	okay

### 1.7 Manage Complaint

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Add complaint information</b>	Order Id: O34 Problem: broken Image:lost.png	Successfully added complaint	Successfully added complaint	Pass	
<b>Update complaint information</b>	Order Id: O34 Problem: broken Image:lost.png Status: Under Review	Successfully updated complaint information	Successfully updated complaint	Pass	

**1.2 Manage Customer**

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>Add customer information</b>	Name: Dina Binti Nadzir Phone: 0132233133 Address: No 90 Jalan Tebu, Sabak Bernam, 45200, Selangor Malaysia	Successfully added customer	Successfully added customer	Pass	
<b>Update customer information</b>	Name: Dina Binti Nadzir Phone: 0132233133 Address: No 90 Jalan Tebu, Sabak Bernam, 45200, Selangor Malaysia	Successfully updated customer	Successfully updated customer	Pass	
<b>View customer information</b>	Name: Dina Binti Nadzir Phone: 0132233133 Address: No 90 Jalan Tebu, Sabak Bernam, 45200, Selangor Malaysia	Customer detail displayed in system	Customer detail displayed in system	Pass	

---



**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

**1.8 Generate Report**

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
<b>View Daily Sales</b>	-	List and graph of daily sales displayed	List and graph of daily sales displayed	Pass	May change to variety of color for the graph
<b>View Monthly Sales</b>	--	List and graph of monthly sales displayed	List and graph of monthly sales displayed	Pass	May change to variety of color for the graph
<b>View Yearly Sales</b>		List and graph of yearly sales displayed	List and graph of yearly sales displayed	Pass	May change to variety of color for the graph

**SUPERPROPOS: PRODUCT ORDERING SYSTEM**

**2.0 SYSTEM TESTING APPROVAL**

	<b>Name</b>	<b>Date</b>
<p><b>Verified by:</b></p>  <hr/> <p>Developer</p>	RINA BINTI ISHAK	12/06/2023
<p><b>Approved by:</b></p>  <hr/> <p>Client</p>	ZAID BIN ISHAK	12/06/2023

Name: Zaid Bin Ishak  
 Position:  Admin     Seller     Marketer     Warehouse

NO	Module	Activities	Status	Comments
1.	Login	Login to system	Pass	
2.	Register Employee	Register seller, marketer and warehouse	Pass	
3.	Manage User Profile	View, delete and update admin, seller, marketer and warehouse staff profile	Pass	
4.	Manage Customer	View and update customer profile	Pass	Only can view customer
5.	Manage Product Detail	Add, view, delete and update product information	Pass	
6.	Manage Order	Add, view and update order information	Pass	Cannot delete order
7.	Manage Complaint	Add, view and update complaint information	Pass	
8.	Generate Report	View sales report	Pass	Need improvement

Figure 1.1 User Acceptance Testing Form by Admin

Name: Muhd Azhan Bin Khairi  
 Position:  Admin  Seller  Marketer  Warehouse

NO	Module	Activities	Status	Comments
1.	Login	Login to system	Pass	
2.	Register Employee	Register seller, marketer and warehouse	Fail	Not available (admin did it)
3.	Manage User Profile	View, delete and update admin, seller, marketer and warehouse staff profile	Pass	Able to update own profile and can view seller assign to me
4.	Manage Customer	View and update customer profile	Pass	Only can view seller's customer
5.	Manage Product Detail	Add, view, delete and update product information	Pass	Can view product details only
6.	Manage Order	Add, view and update order information	Pass	Can view my seller order only
7.	Manage Complaint	Add, view and update complaint information	Pass	Can view my seller complaint only
8.	Generate Report	View sales report	Pass	Can view my seller sales report only

Figure 1.2 User Acceptance Testing Form by Marketer



SuperProPOS: Product Ordering System

Name: Lia Yasmien Binti Abdul Hassan  
 Position:  Admin  Seller  Marketer  Warehouse

NO	Module	Activities	Status	Comments
1.	Login	Login to system	Pass	
2.	Register Employee	Register seller, marketer and warehouse	Fail	Not available (admin did it)
3.	Manage User Profile	View, delete and update admin, seller, marketer and warehouse staff profile	Pass	Able to update own profile.
4.	Manage Customer	Add, view and update customer profile	Pass	
5.	Manage Product Detail	Add, view, delete and update product information	Pass	View product only
6.	Manage Order	Add, view and update order information	Pass	
7.	Manage Complaint	Add, view and update complaint information	Pass	
8.	Generate Report	View sales report	Pass	

Figure 1.3 User Acceptance Testing Form by Seller

Name: Muhd Azwan Bin Kamal  
 Position:  Admin  Seller  Marketer  Warehouse

NO	Module	Activities	Status	Comments
1.	Login	Login to system	Pass	
2.	Register Employee	Register seller, marketer and warehouse	Fail	Not available (admin did it)
3.	Manage User Profile	View, delete and update admin, seller, marketer and warehouse staff profile	Pass	Able to update own profile.
4.	Manage Customer	Add, view and update customer profile	Fail	Not available
5.	Manage Product Detail	Add, view, delete and update product information	Pass	Can update product stock and availability.
6.	Manage Order	Add, view and update order information	Pass	Can update order delivery status and tracking number only
7.	Manage Complaint	Add, view and update complaint information	Pass	Can update complaint status and explanation only
8.	Generate Report	View sales report	Fail	Not available

Figure 1.4 User Acceptance Testing Form by Warehouse Staff

# SuperProPOS: Product Ordering System

## Appendix D: Gantt Chart for SuperProPOS: Product Ordering System

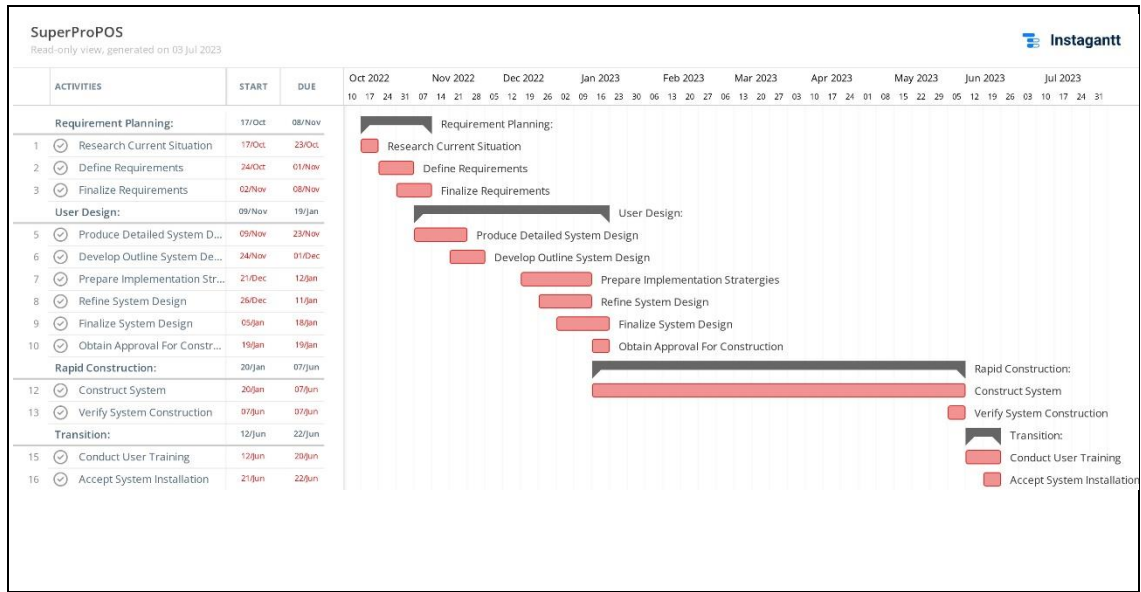


Figure 1.0 Gantt Chart of SuperProPOS: Product Ordering System Phase



