

# Megasura Point of Sales and Financial Management System

Mohamed Hazizi Hamdan<sup>1</sup>, Nurezayana Zainal<sup>1\*</sup>

<sup>1</sup> *Fakulti Sains Komputer dan Teknologi Maklumat,*

*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

\*Corresponding Author: [nurezayana@uthm.edu.my](mailto:nurezayana@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.070>

## Article Info

Received: 30 July 2024

Accepted: 18 June 2025

Available online: 30 June 2025

## Keywords

Point of Sales, Financial,  
Management

## Abstract

The Megasura Enterprise is a franchised store of Lesong Mart that is based on Muslim and halal products. The financial and point-of-sales management system aims to replace an existing client-based system with a web-based system which is more accessible as well as remove any use of physical data stores for financial management that includes calculating the amount of profit and sales, calculating the tax payment and manage inventory with invoicing and provide a smooth operation for the staff of Megasura Enterprise that removes any loading time to ensure high customer satisfaction during each purchase transaction. The goal of this project would be to design and develop a financial and point of sales management system using an object-oriented approach on a web application by using Waterfall methodology as well as to test the developed system using alpha, beta, performance testing, and user acceptance tests. The system is developed in the PHP programming language, Laravel Framework and the data is stored on a MySQL database that is hosted on a web service. From this application, the staff has a better user experience compared to the previous system and indirectly improves the store customer experience. In conclusion, the developed system achieved the objective of this proposed solution and helped in solving the problem faced by the organization.

## 1. Introduction

The Megasura Enterprise is a Lesong Mart franchise based on Muslim and halal products. As of the writing of this proposal, the Megasura Enterprise only has a single asset of store based on Ligamas Batang Kali, Hulu Selangor, Selangor.

Currently, Megasura Enterprise only rents a single lot store on Ligamas Batang Kali, where it faces intense competition from other businesses offering similar halal and Muslim product concepts. Additionally, a franchised store with a similar idea is situated less than 10 kilometers away from Megasura Enterprise. Megasura Enterprise paid for approximately RM 15,000 worth of monthly operating costs, including the cost of transportation, staff salaries, water bills, and electricity bills. The company concentrated mostly on the electricity bill because it uses the most energy when operating freezers and air conditioners.

Growing economic expansion leads to the emergence of new consumer products, which raises questions about their uniqueness and halal status. It also increases consumer spending on education. Every day, every consumer chooses not to purchase halal products. This is shown especially during the post-COVID-19 era where the majority of consumers tend to make more thoughtful purchases [1]. Due to the high product rotation and staffing levels, this resulted in an influx of consumers that overwhelmed Megasura Enterprise, which limited the time for financial management.

Thus, this project proposed to develop the new system of Megasura Point of Sales and Financial Management System and cutting down on the amount of time spent monitoring the enterprise's financial data would be another factor. This will reduce the amount of time needed for weekly or even monthly financial management, making it more comfortable.

This paper is organized into four sections. Section 1 explains the project introduction. Section 2 discusses related work. Section 3 describes the methodology used in the project and discusses each of its phases. Section 4 describes the results and discussions. Lastly, Section 5 concludes the project.

## 2. Related Work

In this section, the domain background, the financial management system, and the result of the comparative analysis are discussed.

### 2.1 Domain Background

The current system for Megasura Enterprise is in dire need of coping with a large number of customers, especially during the post-Covid-19 era. The main objective of Megasura Enterprise would be to make sure customer satisfaction is on par with the major retailers like Lotus, AEON, and Eonsave. Currently, for the current system, each of the transactions from customers with the system took some of the loading time for the system to calculate the total prices and store the purchasing data. This will lessen the customer's satisfaction as customers do not like to be kept waiting for some trivial matter. Other than that, to improve accessibility, the information for the transaction and stock needs to be easily accessible by the staff, manager, and owner of the store. As of now, the current medium of communication is through WhatsApp group only. This process also applies to the information regarding the new stock arrival.

### 2.2 Financial Management System

Financial management for a retail store involves a comprehensive approach to planning, organizing, and controlling the store's financial activities with the overarching goal of ensuring profitability, liquidity, and long-term financial stability. One critical aspect is budgeting and planning, where the store engages in sales forecasting to estimate future revenue and allocates funds for various operating expenses through expense budgeting. Efficient cash flow management is pivotal, necessitating a delicate balance in inventory management to prevent overstocking or stockouts and strategic negotiation with suppliers to optimize payment terms. Profitability analysis is crucial, with regular reviews of gross margin and break-even points to assess financial health accurately. Financial reporting, including income statements, balance sheets, and cash flow statements, provides insights into the store's financial performance over specific periods.

### 2.3 Comparative Analysis

This section provides a comparative analysis of three existing applications which are Synergy Software Solution Sdn Bhd [6], Odoo [7], and Loyverse [8]. Table 1 shows the comparison between the three existing systems and the proposed system. The compared features include a system platform, managed user profile, multiple device accessibility, tax SST calculation, real-time stock levels, and real-time sales analytics.

**Table 1** System Comparison

Features/System	Synergy Software Solution Sdn Bhd	Odoo	Loyverse	Megasura Point of Sales and Financial Management System
Platform	Client-based	Web-based	Mobile Application	Web-based
Manage User Profile	No	No	Yes	Yes
Multiple Device Accessibility	No	Yes	Yes	Yes
Tax and SST Calculation	Yes	No	No	Yes
Real-time stock levels	No	No	No	Yes
Real-time sales analytic	No	No	No	Yes

In summary, the proposed system offers a comprehensive set of features, including managing user profiles, multiple device accessibility, tax SST calculation, real-time stock levels, and real-time sales analytics. The comparison shows that the proposed system is improved from the existing systems.

### 3. Methodology

For the development of the Megasura Point of Sales and Financial Management System, the waterfall methodology is chosen. The waterfall methodology is the most suited for this proposed system development because the system requirement is crystal clear from the beginning [9]. The waterfall model is a sequence of phases that need the completion of the previous phase before moving on to the next phase. By current technology standards, the Waterfall model may seem old and no longer efficient for today's business world [10], but for the proposed system that does not have a lot of resources and plenty of time, the Waterfall methodology is the most suited. The prototype methodology can be referred to in Appendix A.

#### 3.1 Phases in Software Development

#### 3.2 Requirement Gathering and Analysis Phase

The requirement Gathering and Analysis phase is the initial stage where project requirements are identified, documented, and analyzed before any development work begins. This phase involves several key activities, including the use of various elicitation techniques to gather comprehensive and accurate requirements.

The elicitation technique used to gather information:

- Observation – To understand the existing system, identify inefficiencies, and areas of improvement.
- Interview - Interview with the stakeholders of the proposed system to gather details about user needs.
- Gathering of Financial Documentation – Gain an understanding of the financial requirements, standards and organization needs.

##### 3.2.1 System Requirement Analysis

In this section, user requirements, functional requirements, and non-functional requirements will be presented and discussed. The system requirement analysis addresses the specific functionalities to be implemented in the proposed system. Table 2 shows the user requirements of the proposed system. Table 3 shows the functional requirements of the proposed system. Lastly, table 4 shows the non-functional requirements of the proposed system.

**Table 2** *User requirements of the proposed system*

User	Requirements
Staff	Enable to login to the system, the only user to use a point of sales system services, manage all of the inventory, and access to view the dashboard.
Manager	Enable to login to the system, manage all of the inventory, the only type of user to access the sales and inventory report, and access to view the dashboard.
Admin	Enable to login to the system, and the only user that is able to manage all of the user profiles.

**Table 3** *Functional requirements of the proposed system*

Modules	Requirements
Login	The system shall allow all users to enter account credentials to log in to access the system.
Manage User Profile	The system shall be able to allow the admin to view, create, and delete an account.
Point of Sales (POS)	The system shall be able to allow staff to manage all transactions with the customer's product purchasing as well as enter any incoming and outgoing transaction that does not involve the customer i.e. stock purchasing. The staff can use customer membership as a sales discount.

**Table 3:** (cont)

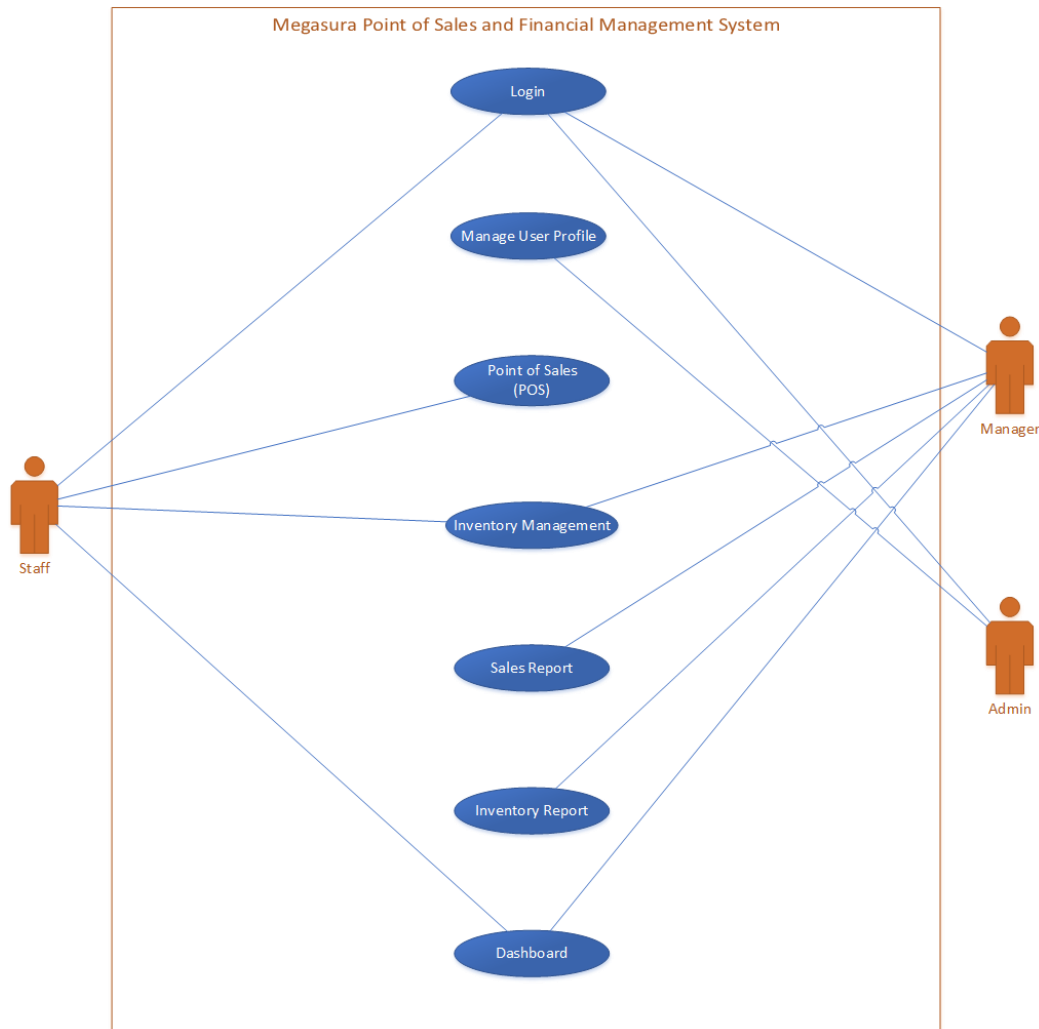
Inventory Management	The system shall be able to allow staff and managers to add, edit, or remove products as well as edit the quantity of the product which is also recorded as incoming and outgoing products.
Sales Report	The system shall be able to allow the manager can view the sales report based on the performance of sales daily, track the profit of the store daily, and manage the tax calculation based on the sales of the store.
Inventory Report	The system shall be able to allow the manager to view the sales report based on the performance of sales of each product, monitor stock levels to avoid any stock shortage and create an invoice based on the incoming stock that calculates the amount to pay to the distributor by the manager.
Dashboard	The system shall be able to allow staff and managers to track the stock levels and profit sales performance in real-time and view the working staff that operates the POS service.

**Table 4** *Non-functional requirements of the proposed system*

Modules	Requirements
Security	All of the credentials of the user’s account should be stored in the database using a secure hash algorithm, the password of the user’s account creation should be re-entered twice to prevent mistakes from the user and any data operation use a parameterized query to avoid SQL injection.
Operational	The system should be able to function on any of the web-browsers platforms i.e. desktop, tablet, mobile smartphone.
Performance	Any data operation of the system should not be longer than 5 seconds.

### 3.2.2 Use Case Diagram

Fig. 2 shows the use case diagram of the system. The figure includes the seven modules which are Login, Manage User Profile, Point of Sales (POS), Inventory Management, Sales Report, Inventory Report, and Dashboard. The login module is available for all users for the users to access the system. Manage User Profile is an exclusive module for the manager. Only the manager can add, and update the user’s profile. The point of sales module can only be accessed by the staff since the staff is the only user that manages transactions with the customers. Inventory Reports and Inventory Management can be accessed by both the staff and the manager meanwhile the sales report can only be accessed by the manager since only the manager can access the financial information of the store. The dashboard modules can be accessed by both the staff and manager to view the real-time information of the store.



**Fig. 2** Use Case Diagram of the Proposed System

### 3.3 Design Phase

The design phase follows the Requirement Gathering and Analysis phase and is focused on transforming the specified requirements into a detailed blueprint for the software system. This phase is typically divided into two main components: Architectural Design and User Interface Design.

### 3.4 Development and Implementation Phase

The Development and Implementation phase involves the actual creation of the system based on the requirements gathered in the earlier stages. During this phase, developers write code based on the defined requirements and design specifications. Once the code is developed, it undergoes unit testing to verify individual components' functionality. Simultaneously, performance testing is carried out to ensure the system's ability to perform efficiently and reliably under various scenarios.

### 3.5 Testing Phase

The testing phase occurs after the Development and Implementation phases, forming a crucial step to ensure the quality and functionality of the developed system. This phase involves systematic testing processes to identify and rectify defects before the final product is deployed. The testing phase in Waterfall methodology typically includes the stages of Alpha Testing, Beta Testing, and User Acceptance Testing (UAT).

### 3.6 Deployment Phase

This phase involves several key processes, including scheduling with stakeholders, installation of the developed system, and testing to ensure a smooth deployment. Before the deployment process begins, it is crucial to schedule and coordinate activities with stakeholders. This involves communicating the deployment timeline, outlining the activities that will take place, and addressing any concerns or expectations stakeholders may have. Effective communication during this phase is essential to ensure that all parties are aligned with the deployment plan.

### 3.7 Documentation Phase

In the Waterfall methodology, the Documentation phase is the final stage of the software development life cycle. This phase involves the creation and organization of comprehensive documentation that captures various aspects of the system, its design, implementation, and usage. User manuals are part of the user documentation that is created to guide end-users on how to interact with and use the system. These manuals provide step-by-step instructions, explanations of features, and troubleshooting information. The goal is to empower users to make effective use of the system without requiring extensive technical knowledge.

## 4. Analysis and Design

This chapter explains the analysis and design of the proposed system. In Section 4.1, the general system architecture of the proposed system is presented. Section 4.2 discusses the system requirement analysis, including user, functional, and non-functional requirements. Section 4.3 discusses the Use case diagrams of the proposed system. The domain class diagram will be discussed in Section 4.4, while Section 4.5 will discuss the design of the interface and schema table of the database.

### 4.1 Implementation

Chapter 5 explains the implementation and testing phases for the developed system. Chapter 5.2 explains the tools and methods used for system development and implementation and Chapter 5.3 explains the testing to access the functionality of the developed module.

#### 4.1.1 Software for System Development

This section will discuss the software used to develop the web based application of Megasura Point of Sales and Financial Management System. Laravel is used as the framework and visual studio code editor to develop the web based application. Fig. 3 shows the .env file of the Laravel and a snip of Visual Studio code editor that has been loaded with Laravel environment. Fig. 4 (a) shows the interface for Laragon which is used for the system localhost. Fig 4 (b) shows the snip of HeidiSQL that has been used for database manager.

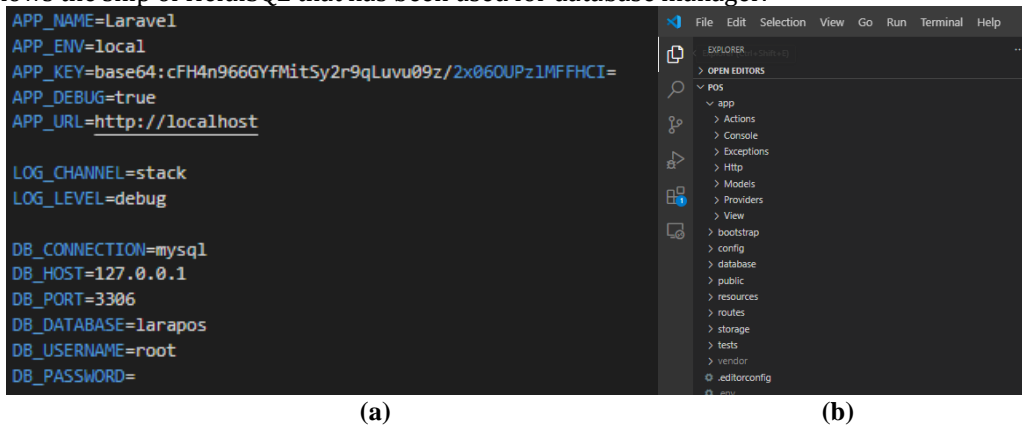
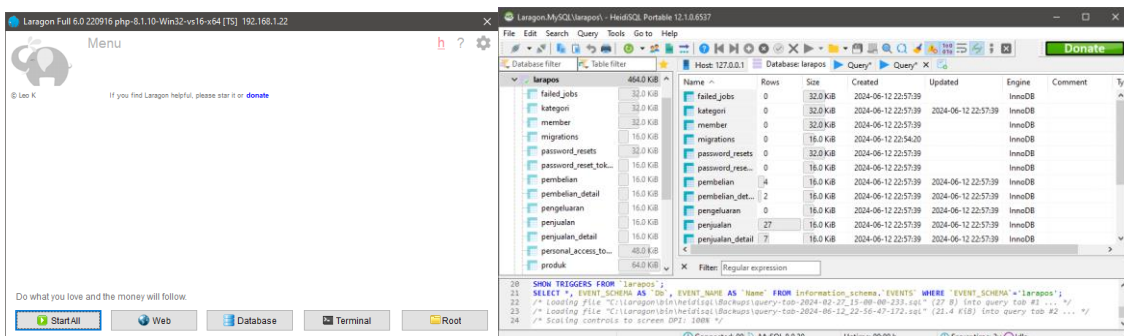


Fig. 3 Software for System Development (a) Laravel env. File; (b) Visual Studio Laravel Environment



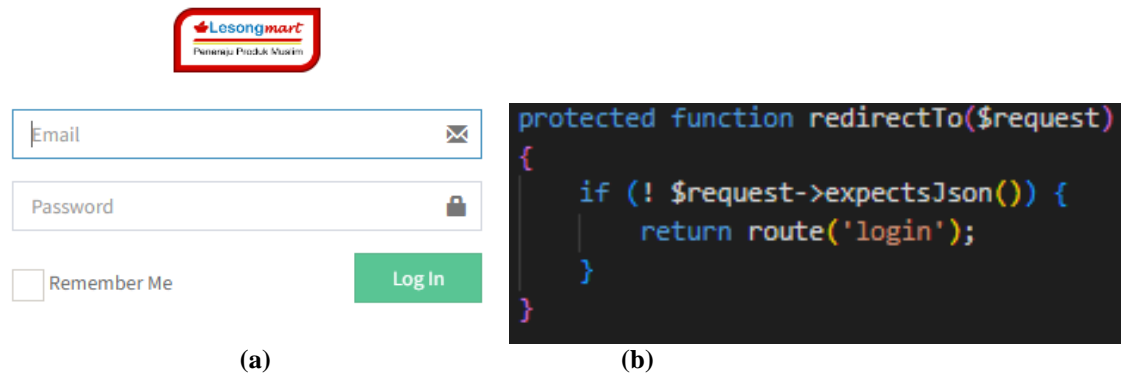
(a) (b)  
**Fig. 4** Software for System Development (a) Laragon Interface; (b) HeidiSQL Database Management

## 4.1.2 Functional Module Development

This section describes the development of functional modules in a system. A program code is provided to aid clarification.

### 4.1.2.1 Login Module

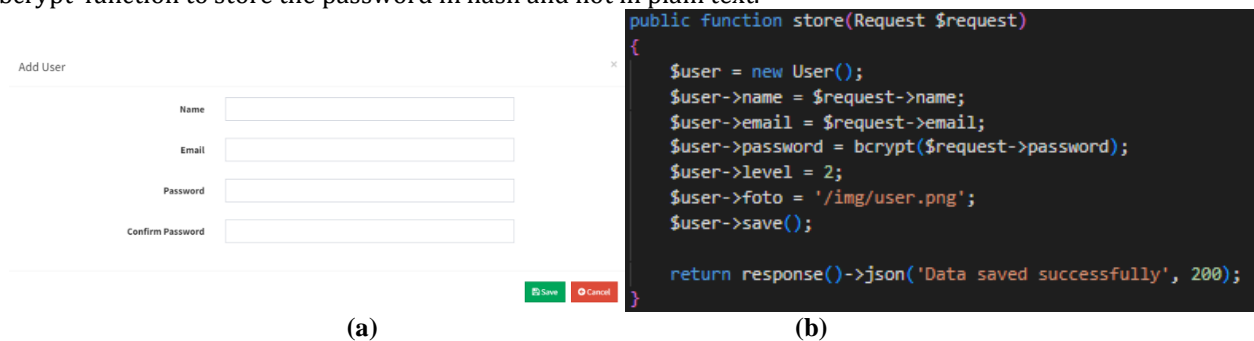
Fig. 5 show the server-side coding and user interface for the user Login. Fig. 5 (a) shows the interface for the Login where the users need to enter their email and password as the credentials. The Login has a function of remember me where when enabled, the system will pre-load the email and the user only needs to enter only for the password. Next Fig. 5 (b) shows the server-side coding for the Login. The Laravel framework automatically creates the Login functionality. Fig. 5 (b) shows the coding to redirect the user after the login credentials have been validated.



**Fig. 5** Interface for Login Modules (a) Login Interface; (b) Server-side Coding

### 4.1.2.2 Manage User Profile Module

Based on Fig. 6 (a), the user enters the name, email, and password and confirms the password that will be stored in the database using the 'store' function the Fig. 6(b). Then, the password will be stored cryptically using the 'bcrypt' function to store the password in hash and not in plain text.



**Fig. 6** Interface for Manage User Profile Module (a) Manage User Profile Interface; (b) Server-side coding

### 4.1.2.3 Point of Sales (POS) Module

Fig. 7 shows the Point of Sales interface that has the functionality to retrieve product data and calculate the total amount based on the selected product with each quantity. Fig. 7 also shows the functionality to add a discount toward the product based on manual input or based on the customer membership. Next, Fig. 8 shows the coding to store the saved transaction of Point of Sales in the database.

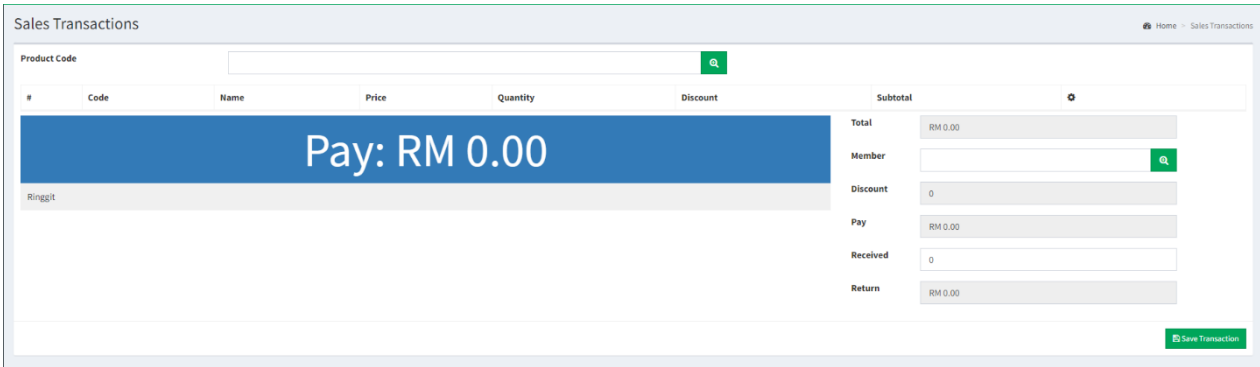


Fig. 7 Interface for Point of Sales

```
public function store(Request $request)
{
    $penjualan = Penjualan::findOrFail($request->id_penjualan);
    $penjualan->id_member = $request->id_member;
    $penjualan->total_item = $request->total_item;
    $penjualan->total_harga = $request->total;
    $penjualan->diskon = $request->diskon;
    $penjualan->bayar = $request->bayar;
    $penjualan->diterima = $request->diterima;
    $penjualan->update();

    $detail = PenjualanDetail::where('id_penjualan', $penjualan->id_penjualan)->get();
    foreach ($detail as $item) {
        $item->diskon = $request->diskon;
        $item->update();

        $produk = Produk::find($item->id_produk);
        $produk->stok -= $item->jumlah;
        $produk->update();
    }

    return redirect()->route('transaksi.selesai');
}
```

Fig. 8 Server-side Coding for Point of Sales

#### 4.1.2.4 Inventory Management Module

Fig. 9 shows the interface of Inventory Management that has the functionality of creating new products, and deleting or updating existing products. The Inventory Management also has the functionality of printing the barcode that has been generated by the system. Next, Fig. 10 shows the coding for storing a newly added product in the database.

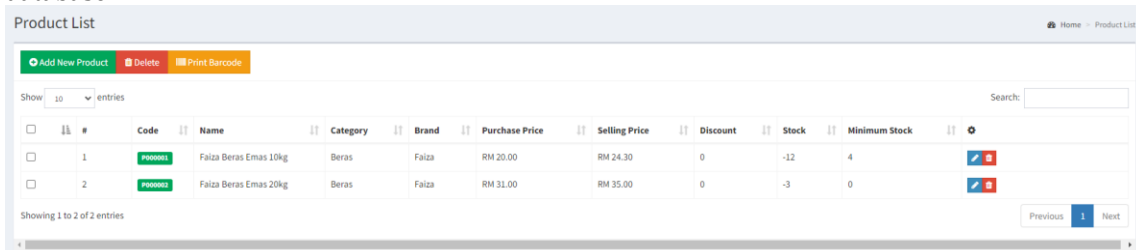


Fig. 9 Interface for Inventory Management

```
public function store(Request $request)
{
    if ($request->has('barcode_produk') && $request->barcode_produk != "") {
        $request->merge(['kode_produk' => $request->barcode_produk]);
    } else {
        $produk = Produk::latest()->first() ?? new Produk();
        $request->merge(['kode_produk' => 'P' . tambah_nol_didepan((int)$produk->id_produk + 1, 6)]);
    }

    $produk = Produk::create($request->all());

    return response()->json('Data saved successfully', 200);
}
```

Fig. 10 Server-side coding for Inventory Management Module

#### 4.1.2.5 Sales Report Module

Fig. 11 (a) shows the interface of the Sales Report Module. On this interface, the report is displayed as a table based on the sales that have been made. The details on each sale will be displayed when the user clicks on the eye icon.

Next, Fig. 11 (b) shows the coding to retrieve the data of the Sales Report from the database. The data of the sales report is based on the product, total sales and the product prices.

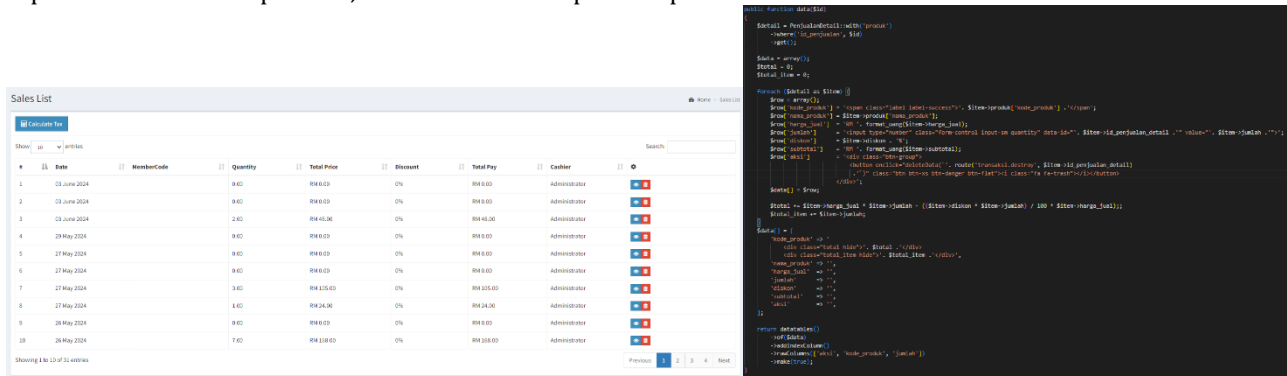


Fig. 11 Interface for Sales Report (a) Sales Report Interface; (b) Server-side coding

### 4.1.2.6 Inventory Report Module

Fig. 12 shows the interface of the Inventory Report that displays the performance of sales based on the selected product. The performance is displayed using the chart to better represent the performance of a certain product in a specific timeframe. Next, Fig. 13, shows the coding to retrieve data from the database. The data is based on the summation of the product sales from a specific timeframe.

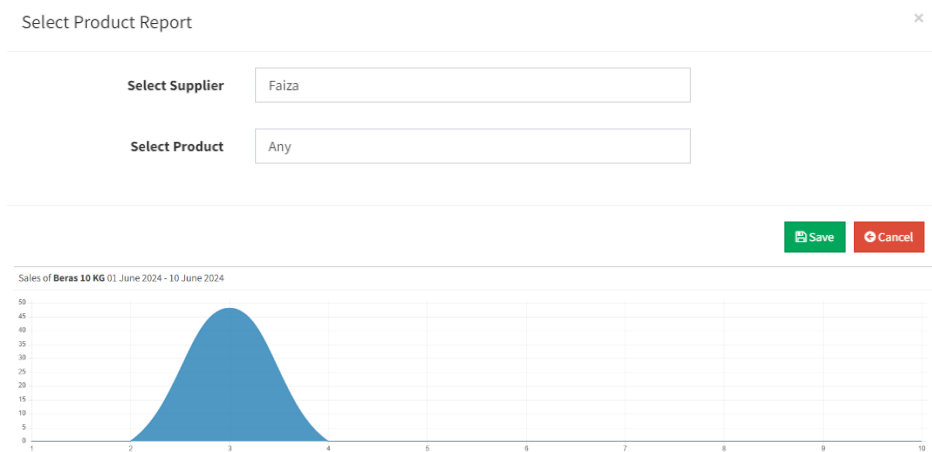


Fig. 12 Interface for Inventory Report



Fig. 13 Server-side coding for Inventory Report

### 4.1.2.7 Dashboard

Fig. 14 (a) shows the interface of the dashboard that will display the essential information for the admin of the system. The dashboard will display the total number of categories, products, suppliers, and the total amount of sales, expenses, and purchases including the staff that actively logged on to the system. The chart displays the total amount of sales that have been deducted by expenses and purchases. Next, Fig. 14 (b), shows the coding to retrieve data from the database. The chart data is based on the first day of the month until the current date of the system.

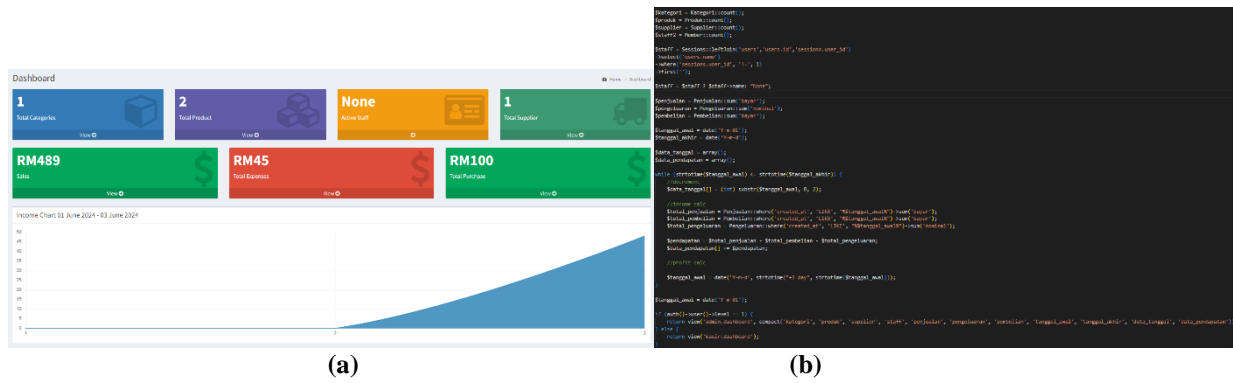


Fig. 14 Interface for Dashboard Module (a) Dashboard Interface: (b) Dashboard Server-side Coding

## 4.2 Testing

In this section, system testing will be carried out by using the User Acceptance Test (UAT) method to access the functionality of each module.

### 4.2.1 Functional Testing

The test cases for each module in the system are displayed in Table 5. For each module, there are a total of seven test cases. This test aims to confirm if the administrator can create an account, and log in, and if the system will prevent login attempts if the wrong credentials are entered. Table 5 demonstrates that each of the tests was passed.

Table 5 Overall Test Results

Test Cases	Requirements ID	Description	Status
<b>TC_100 REQ_100 Login</b>			
TC_100_01	REQ_101	The system shall allow the user to login by using the credentials username and password	PASS
TC_100_02	REQ_102	The system must verify the user's credentials (username and password) against stored data.	PASS
TC_100_03	REQ_103	The user's credentials (username, password) are invalid, the system must display an error.	FAIL
TC_100_04	REQ_104	The user's credentials (username, password) are invalid, the system will not permit registration.	PASS
<b>TC_200 REQ_200 Manage User Profile</b>			
TC_200_01	REQ_201	Admin should have the option to deactivate their profiles temporarily or permanently.	
TC_200_02	REQ_202	Admin must be able to create a new profile with the required information.	PASS
TC_200_03	REQ_203	Profile loading times should be below 2 seconds to ensure a responsive user experience.	PASS
TC_200_04	REQ_204	Editing and saving profile changes should occur within 3 seconds.	PASS
TC_200_05	REQ_205	User profile data must be encrypted during transmission to and from the MySQL server.	PASS
TC_200_06	REQ_206	Profile data should be stored with high integrity to prevent corruption or loss.	PASS
<b>TC_300 REQ_300 Point of Sales</b>			
TC_300_01	REQ_301	The POS system must generate clear and comprehensive receipts for each transaction, including details like items purchased, prices, taxes, and the total amount, to provide customers with accurate records of their purchases.	PASS
TC_300_02	REQ_302	The system should process transactions within a specified time frame (e.g., 2 seconds per transaction) to ensure a seamless and efficient checkout experience for customers.	PASS
TC_300_03	REQ_303	The system must be scalable to accommodate an increasing number of products, users, and transactions, ensuring consistent performance as the business grows.	PASS

**Table 5: (cont)**

TC_300_04	REQ_304	The POS system must be reliable, ensuring minimal downtime. It should recover gracefully from failures, and transactions should not be lost or corrupted.	PASS
TC_300_05	REQ_305	All data, including sales records, inventory levels, and financial information, must be accurate. The system should implement validation checks and data integrity measures to prevent errors.	PASS
<b>TC 400</b>	<b>REQ 400</b>	<b>Inventory Management</b>	
TC_400_01	REQ_401	Users should be able to add new products to the inventory and remove outdated or discontinued items.	PASS
TC_400_02	REQ_402	Data retrieval for inventory queries should occur within 3 seconds.	PASS
TC_400_03	REQ_403	The system must maintain a high level of accuracy in tracking inventory levels and transactions.	PASS
TC_400_04	REQ_404	The system should provide clear and helpful error messages in case of user input errors or system failures.	PASS
<b>TC 500</b>	<b>REQ 500</b>	<b>Sales Report</b>	
TC_500_01	REQ_501	The system must accurately retrieve and present sales data, ensuring precision in calculations and representations.	PASS
TC_500_02	REQ_502	The sales report must reflect real-time data, updating dynamically as new sales transactions occur.	PASS
TC_500_03	REQ_503	The system should generate and display the sales report within a maximum response time of 3 seconds under normal operating conditions.	PASS
TC_500_04	REQ_504	The system must handle an increasing volume of sales data without significant degradation in performance. It should be scalable to accommodate growth.	PASS
TC_500_05	REQ_505	The sales report feature must be reliable, minimizing the occurrence of system failures or errors during regular usage.	PASS
TC_500_06	REQ_506	The sales report interface should be intuitive and user-friendly, requiring minimal training for users to navigate and interpret the presented data.	PASS
<b>TC 600</b>	<b>REQ 600</b>	<b>Inventory Report</b>	
TC_600_01	REQ_601	The inventory report must reflect real-time data to provide accurate stock information.	PASS
TC_600_02	REQ_602	The inventory report interface should be user-friendly and intuitive.	PASS
TC_600_03	REQ_603	Users should have the capability to generate inventory reports for offline analysis.	PASS
TC_600_04	REQ_604	The system should respond to user requests within an acceptable timeframe.	PASS
<b>TC 700</b>	<b>REQ 700</b>	<b>Dashboard</b>	
TC_700_01	REQ_701	The dashboard should provide real-time updates for key metrics and data to ensure users have the most current information.	PASS
TC_700_02	REQ_702	The dashboard should have a response time of less than 2 seconds for standard operations to provide a smooth and responsive user experience.	PASS
TC_700_03	REQ_703	Large datasets should load efficiently, and the dashboard should display data promptly to prevent long loading times that could impact user experience.	PASS

A total of 7 modules with 32 test cases had been tested in this system. Table 6 shows the overall test results.

**Table 6 Overall Test Results**

Test Cases	Total Test Cases	Total Success	Total Failed
TC_100	4	3	1
TC_200	6	6	0
TC_300	5	5	0
TC_400	4	4	0

**Table 6: (cont)**

TC_500	6	6	0
TC_600	4	4	0
TC_700	3	3	0
Total	32	31	1

One test case, TC\_100\_03, which corresponds to requirement REQ\_100, was the only one to fail when the total test results for the system’s eight modules were examined. When the user enters an invalid email and password, supposedly the system was required to display feedback of error message. During testing, the system will turn into a blank screen when the user enters an invalid login credential. This problem will be looked into, repaired and fix will be redeployed to fully satisfy the requirement.

### 4.2.2 User Acceptance Test

The User Acceptance Test (UAT) is an important phase in the software development life cycle. During the phase of UAT, the stakeholders of the system validate the system functionality, usability and overall experience with the system requirement. The stakeholders of the system get the proper training and exposure before using the system in the production environment. Alpha testing is carried out in a controlled setting by stakeholders. Before the system is made available to a larger audience, the main objective is to verify its functionality and make sure it satisfies the requirements. An overview of the alpha testing carried out by stakeholders is shown in Table 6. The form for alpha testing user acceptance test can refer to Appendix B.

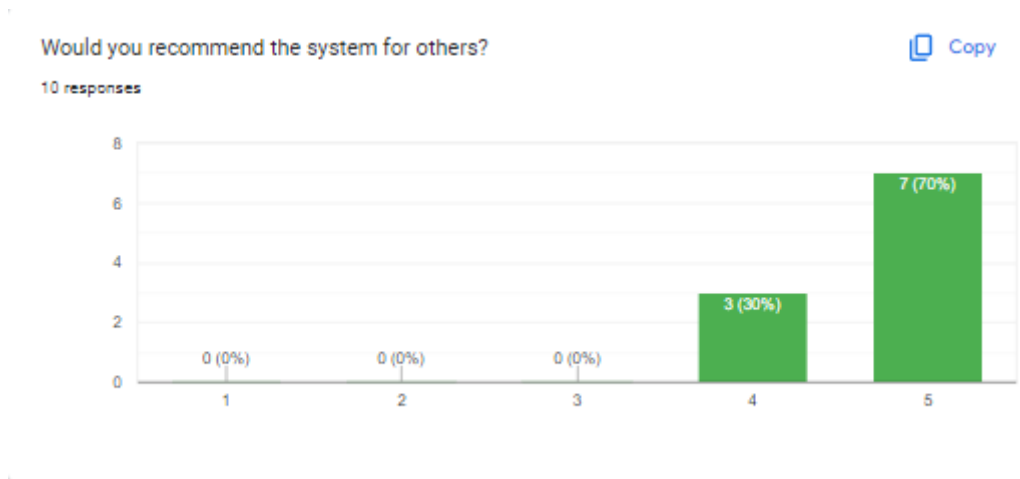
**Table 7 Alpha Testing Summary**

No	Acceptance Criteria Module	Test Result	Pass Percentage (%)
1	Login	PASS	98
2	Manage User Profile	PASS	100
3	Point of Sales	PASS	100
4	Inventory Management	PASS	100
5	Sales Report	PASS	100
6	Inventory Report	PASS	100
7	Dashboard	PASS	100

According to Table 7, alpha testing resulted in an almost 100% pass rate as the only setback was the Login mentioned on the test case TC\_100\_03, as certified by shareholder Titik Suraya, owner of Megasura Enterprise. User Acceptance Testing provides a means of verifying that the system is appropriate for the administrative functions of the store. To successfully implement the system, cater to user needs, and make any last-minute adjustments, the feedback gathered during this phase is essential. Beta Testing follows the original Alpha Testing phase and is an important step in the User Acceptance Testing process. During Beta Testing, the emphasis switches from internal testing to getting feedback from a larger group of genuine users under realistic usage situations. Table 7 shows the detailed breakdown from the survey evaluation.

**Table 7 Positive Value Analysis**

Question	Positive Value	Test Value Score	Percentage (%)
1	45	50	90
2	46	50	92
3	47	50	94
4	46	50	92
5	47	50	94
6	47	50	94
7	46	50	92
8	47	50	94
9	48	50	96
10	47	50	94
11	44	50	88
12	47	50	94
13	47	50	94
Total	605	650	92.9

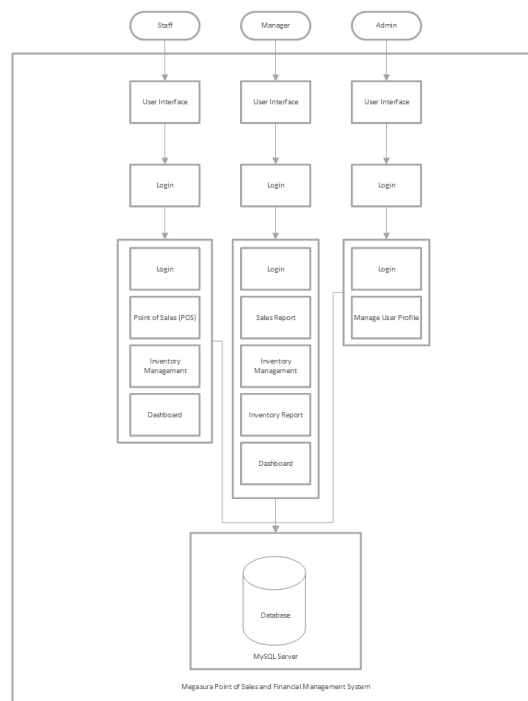


**Fig. 15** Example Response of Overall Evaluation

Fig. 15 shows an example response from the last question of overall system recommend ability. As from the positive value analysis from Table 7, the total positive value of 605 and the maximum value of 650 gives the overall percentage of 92.9%. However, the results from the respondents do not paint the full picture. The result is only based on the initial testing of the system. There is still a need for improvement after the system has been thoroughly used by the users.

### 4.3 General System Architecture

Fig. 16 shows the system architecture of the Megasura Point of Sales and Financial Management System. The system is web-based and will be deployed on the web servers. There are three types of users for this system, which are staff, manager and admin.



**Fig. 16** System Design Diagram of the Proposed System

### 4.4 Use Case Specification

A use case specification is a concise document outlining the interactions between users and a system to achieve specific goals. It includes the activity diagram and sequence diagram which are shown in Appendix A and Appendix B respectively. The goal is to provide a clear understanding of how users will interact with the system to accomplish specific tasks. The use case specification serves as a blueprint for system design and development.

### 4.5 Domain Class Diagram

The domain class diagram illustrates the conceptual model in database modelling as shown in Fig. 17. The domain class diagram includes the user, user type, product, product type, transaction, product transaction, payment type, changelog, stockage, and vendor of the database tables. All staff as users, can manage many Umrah packages and feedback. A user can be multiple types of users. A product can be multiple types of product. Each transaction can have multiple types of payments. Each transaction can have multiple products as well as each stockage of stock. This is done by having product transactions as an intermediary class.

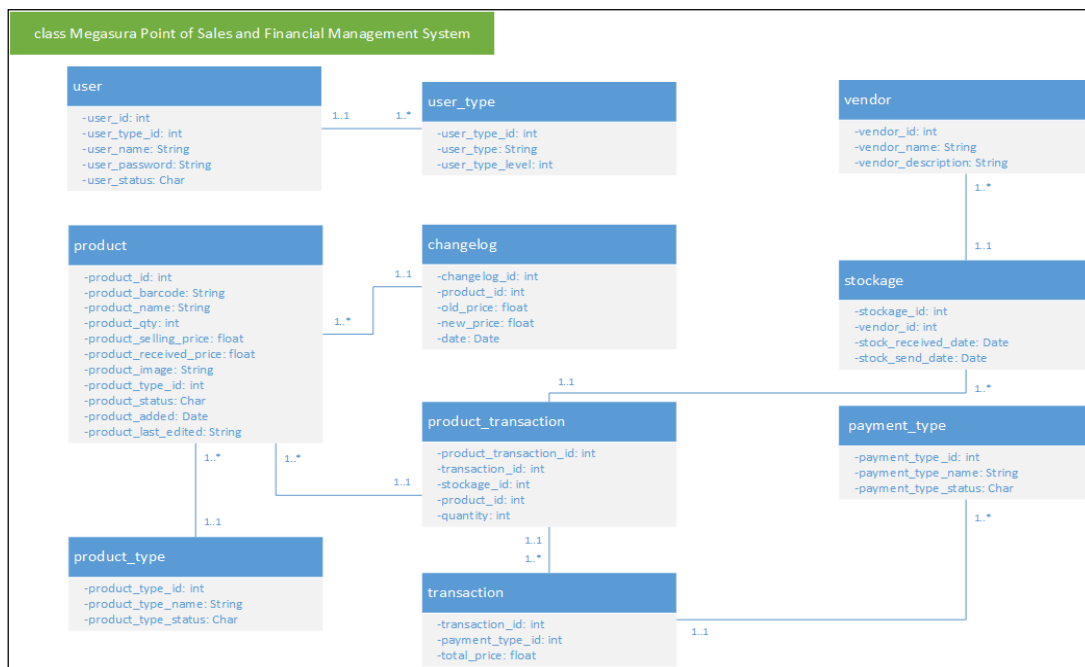


Fig. 17 Domain Class Diagram of the Proposed System

### 4.6 Interface Design

In this section, each interface of the system module will be shown. The interfaces shown in each system module are wireframes that represent the illustration of each page interface. Fig. 18 shows the interface for user login modules. Fig. 19 shows the interface for managing user profile modules. Fig. 20 shows the interface for point-of-sales modules. Fig. 21 shows the interface for inventory management modules. Fig. 22 shows the interface for sales report modules. Fig. 23 shows the inventory report modules.

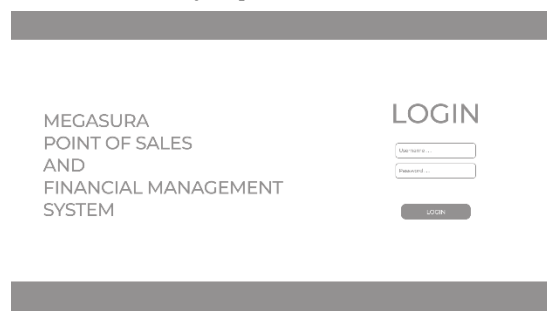
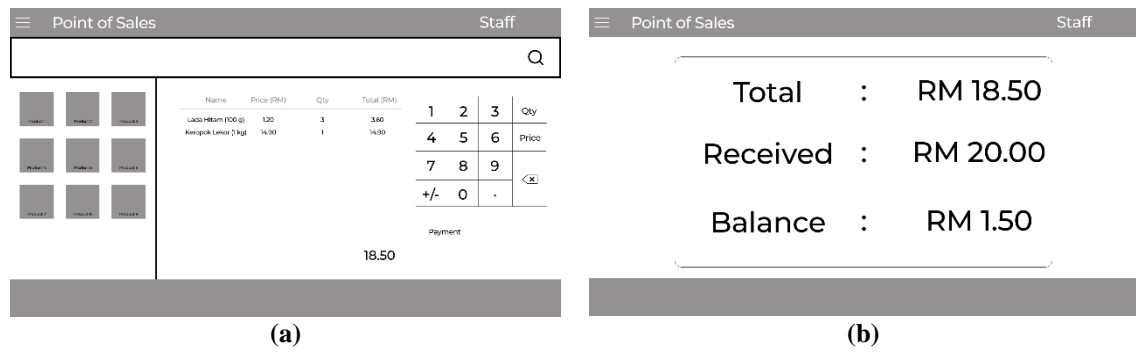


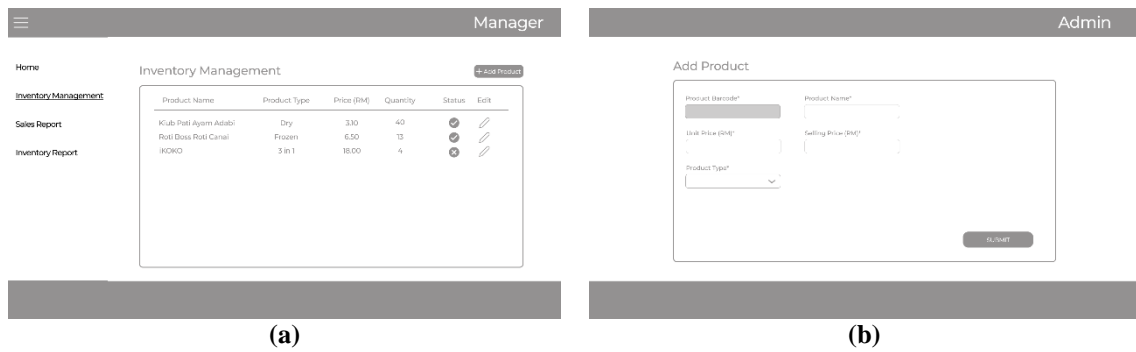
Fig. 18 Interface for login modules; Login Page



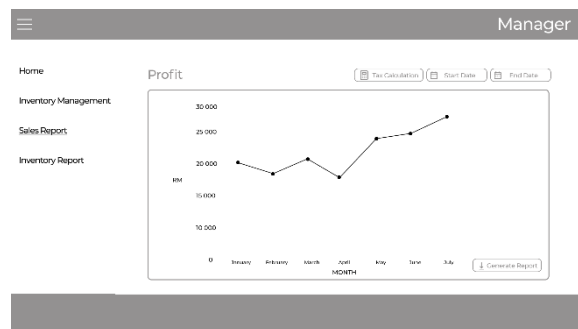
**Fig. 19** Interface for Manage User modules (a) View User; (b) Add User



**Fig. 20** Interface for Point of Sales modules (a) Point of Sales; (b) Cash Transaction



**Fig. 21** Interface for Inventory Management modules (a) Inventory Product Details; (b) Add New Product



**Fig. 22** Interface for Sales Report modules; Profit Statistic



**Fig. 23** Interface for Inventory Report modules; Vendor Sales Statistic

## 5. Conclusion

The Megasura Financial and Point-of-Sales Management System project has been completed, achieving the design, development, and comprehensive testing objectives.

For the first objective, the system was designed based on an object-oriented approach, which included identifying key entities, defining classes with their attributes and methods, and establishing relationships between these classes. This approach was meticulously documented using UML diagrams and tables, ensuring that the system architecture is modular, reusable, and maintainable. This object-oriented design facilitated the successful implementation of the financial and point-of-sales functionalities, providing a robust foundation for future enhancements and maintenance.

Moving to the second objective, the system was developed as a web-based platform using modern web development technologies. PHP was employed for server-side scripting, MySQL for data storage, and the entire development process was managed in a local environment using Laragon. The system's web-based nature ensures accessibility from various devices, enhancing the user experience. Key functionalities such as sales processing, inventory management, financial reporting, and tax calculation were seamlessly integrated into the web application. This accomplishment ensures that the system is efficient, scalable, and capable of handling the operational needs of Megasura Enterprise.

Finally, for the third objective, extensive testing methodologies were applied to validate the system's performance and usability. The testing phase included functional testing to ensure all requirements were met, followed by alpha testing with internal stakeholders to validate user acceptance. Subsequently, beta testing was conducted with actual users, collecting feedback through surveys and usability assessments. The positive feedback from beta testers confirmed the system's usability and effectiveness. Any issues identified during testing were promptly addressed, ensuring the system's readiness for deployment. These thorough testing processes guaranteed that the system met the expectations of stakeholders and end-users, ensuring high levels of satisfaction and reliability. Overall, the successful completion of these objectives demonstrates that the Megasura Financial and Point-of-Sales Management System is well-designed, robustly developed, and rigorously tested, ready to enhance the operational efficiency and user experience for Megasura Enterprise.

## Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of the paper.

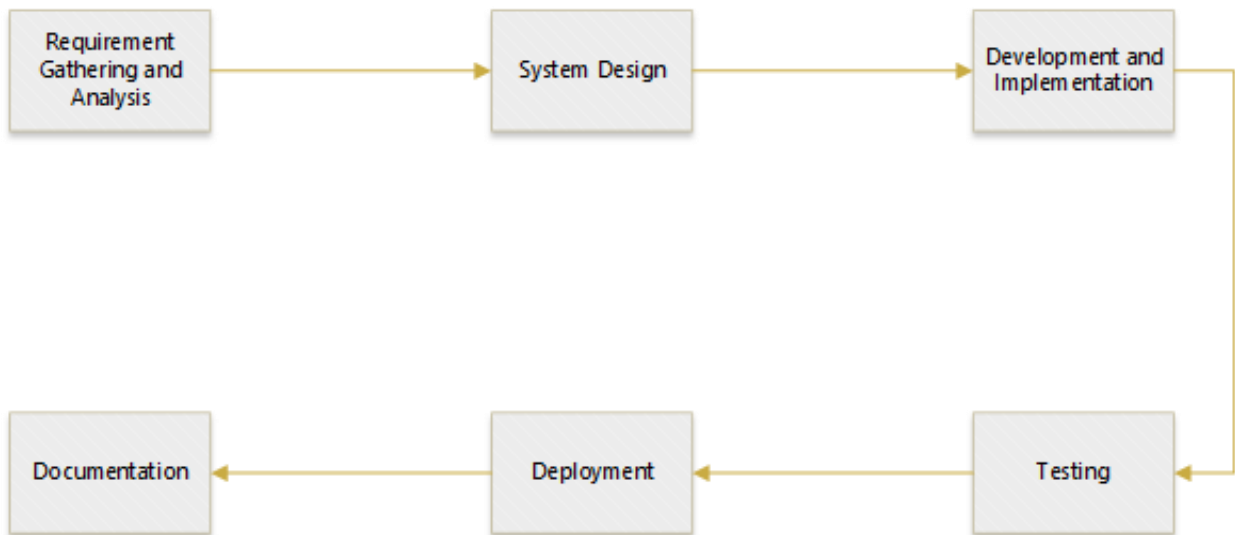
## Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

## References

- [1] Orîndaru, A., Popescu, M., Căescu, Ş., Botezatu, F., Florescu, M., & Runceanu-Albu, C. (2021). Leveraging COVID-19 Outbreak for Shaping a More Sustainable Consumer Behavior. *Sustainability*, 13, 5762. <https://www.mdpi.com/2071-1050/13/11/5762>
- [2] PHP. (n.d.). PHP: Hypertext Preprocessor. Retrieved December 1, 2023, from <https://www.php.net/>
- [3] Bootstrap. (n.d.). Overview. Retrieved December 1, 2023, from <https://getbootstrap.com/docs/5.3/about/overview/>
- [4] jQuery. (n.d.). Overview. Retrieved December 1, 2023, from <https://jquery.com/>
- [5] Li, L., & Qi, D. (2010). The Construction of Enterprises' Financial Management System Based on ASP. *International Journal of Biometrics*, 5, 225. <https://www.ccsenet.org/journal/index.php/ijbm/article/view/8513>
- [6] Synergy Software Solutions Sdn Bhd. (n.d.). Overview. Retrieved December 1, 2023, from <https://synergysoftware.com.my/about/>
- [7] Odoo. (n.d.). Overview. Retrieved December 1, 2023, from <https://www.odoo.com/page/about-us>
- [8] Loyverse. (n.d.). Overview. Retrieved December 1, 2023, from <https://loyverse.com/about>
- [9] Eason, O. K. (2016). Information systems development methodologies transitions: An analysis of waterfall to agile methodology. <https://scholars.unh.edu/cgi/viewcontent.cgi?article=1288&context=honors>
- [10] Aroral, H. K. (2021). Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis. *International Journal of Applied Business and Management Studies*, 6(1), 91-99. [http://www.ijabms.com/wp-content/uploads/2021/05/05\\_ARORAL\\_PB.pdf](http://www.ijabms.com/wp-content/uploads/2021/05/05_ARORAL_PB.pdf)

APPENDIX A



APPENDIX B

User Acceptance Test (UAT): Megasura Point of Sales and Financial Management System

System Admin User: Titik Suraya

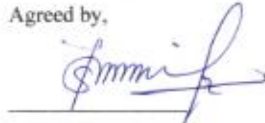
Business Name: Megasura Enterprise

Position: Owner

No	Acceptance Criteria	Test Result		Comment
		Pass	Fail	
1	I can view daily transactions without having to physically attend to the store.	/		
2	The dashboard allows me to easily monitor the current total sales and view which staff is using the POS system on a given day.	/		
3	I can easily generate reports of products and sales and view any shortages of products.	/		
4	As an owner, I can easily add and remove staff from the system.	/		
5	The product can be easily managed by adding, removing, or updating without having to be in the store.	/		
6	I can easily access the system while the staff is performing point-of-sales transactions (multiple simultaneous user sessions) without having any performance degradability.	/		
7	Every page response time does not exceed 2 seconds.	/		

I, Titik Suraya hereby declare that the information provided is true and correct.

Agreed by,



Name: TITIK SURAYA HJ ARSHAD

Date:

