CEMENT COMPANY LOGISTIC MANAGEMENT SYSTEM (CCLMS)

WONG YI XUAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CEMENT COMPANY LOGISTIC MANAGEMENT SYSTEM
(CCLMS)

WONG YI XUAN

This report is submitted in partial fulfilment of the requirement for the Bachelor of Computer Science (Database Management) with Honours

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2024

DECLARATION

I hereby declare that this project report entitled

CEMENT COMPANY LOGISTIC MANAGEMENT SYSTEM (CCLMS)

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT: _____ Date: 25/9/2024
(WONG YI XUAN)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR: _____ Date: 25/9/2024
(PROFESOR MADYA TS. DR. NURUL AKMAR BINTI EMRAN)

DEDICATION

I dedicate this work to my supervisor, lecturers, my family, and the employees of Unipertiwi Sdn. Bhd. for their unwavering support, guidance, and encouragement throughout this journey.

ABSTRACT

Cement Company Logistic Management System (CCLMS) is a web-based digital solution designed to manage the operations of a cement company, addressing challenges associated with manual processes in account creation, order management, and logistics coordination. Utilizing a three-tier architecture built with ASP.NET for the presentation and business logic layers, and SQL Server for the data layer, the system streamlines and enhances operational efficiency. The primary users of this system are admins, logistics partners, and customers. Key modules encompass order placement, order processing, order tracking, as well as billing and payment functionalities. The project introduces its background, problem statement, objectives, scope, significance, and expected outcomes, establishing the need for a digital solution to improve operational transparency and efficiency. It outlines the methodology and planning, employing the Database Life Cycle (DBLC) methodology to ensure systematic progress. The project's aim is to create a user-friendly system that enhances productivity, reduces operational costs, and improves customer satisfaction, thereby contributing to the company's growth and competitiveness.

ABSTRAK

Sistem Pengurusan Logistik Syarikat Simen (CCLMS) adalah penyelesaian digital berasaskan web yang direka untuk menguruskan operasi syarikat simen, menangani cabaran yang berkaitan dengan proses manual dalam penciptaan akaun, pengurusan pesanan, dan koordinasi logistik. Menggunakan seni bina tiga lapisan yang dibina dengan ASP.NET untuk lapisan persembahan dan logik perniagaan, serta SQL Server untuk lapisan data, sistem ini menyelaraskan dan meningkatkan kecekapan operasi. Pengguna utama sistem ini adalah pentadbir, rakan logistik, dan pelanggan. Modul utama merangkumi penempatan pesanan, pemprosesan pesanan, penjejakan pesanan, serta fungsi pengebilan dan pembayaran. Projek ini memperkenalkan latar belakangnya, pernyataan masalah, objektif, skop, kepentingan, dan hasil yang diharapkan, dengan menegaskan keperluan untuk penyelesaian digital bagi meningkatkan ketelusan dan kecekapan operasi. Ia menggariskan metodologi dan perancangan, menggunakan metodologi Kitaran Hayat Pangkalan Data (DBLC) untuk memastikan kemajuan yang sistematik. Matlamat projek ini adalah untuk mencipta sistem mesra pengguna yang meningkatkan produktiviti, mengurangkan kos operasi, dan meningkatkan kepuasan pelanggan, sekali gus menyumbang kepada pertumbuhan dan daya saing syarikat.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CCLMS – Cement Company Logistic Management System

DBLC – Database Life Cycle

DFD – Data Flow Diagram

GUI – Graphical User Interface

DDL – Data Definition Language

DML – Data Manipulate Language

ERD – Entity Relationship Diagram

PO-Purchase Order

PO Num-Purchase Order Number

SO-Sales Order

SO Num-Sales Order Number

DO-Delivery Order

POD-Proof of Delivery

CHAPTER 1

INTRODUCTION

1.1 Project Background

Unipertiwi Sdn. Bhd. serves as a construction, specializing in providing cement for a diverse array of projects, including commercial buildings, industrial facilities, road construction and housing development. However, the company's current operational processes heavily rely on manual intervention. For instance, customers are required to initiate orders by contacting the administrative team via telephone. Subsequently, the administrative staff must then coordinate with logistics companies to arrange for order collection. This dependence on manual efforts not only introduces the possibility of human errors but also poses challenges in terms of efficiency and scalability. Thus, there is a pressing need for the implementation of an automated system to streamline Unipertiwi Sdn. Bhd.'s business workflow, enhancing accuracy and optimizing resource utilization.

The system is designed with a focus on three primary users: the admin, the customer, and the logistic company. The admin holds the authority to approve and manage customer orders, allocate delivery tasks to logistic companies, and generate invoices for customers upon order completion. Customers, on the other hand, are able to place orders and track their status directly through the system interface. For logistic companies, they can view assigned delivery orders and facilitates the assignment of appropriate lorries and drivers for efficient order fulfillment.

1.2 Problem Statements

a) The current logistics operations lack efficiency, leading to delays in order processing, suboptimal inventory management, and increased operational costs.
b) The absence of reporting functions to visualize the company sales performance.
c) The current invoicing and billing processes heavily rely on manual efforts, resulting in delays, errors, and a lack of agility in financial transactions.

1.3 Objectives

a) To streamline order processing, inventory management, and delivery logistics to improve overall operational efficiency.
b) To implement an automated reporting function to generate sales report.
c) To implement an automated invoicing and billing system to reduce manual efforts, minimize errors, and expedite financial transactions.

1.4 Scope

Figure 1.1 shows the structured chart of CCLMS. The scope encompasses three users: cement company admin, customer and logistic. For the Cement Company Admin, the system allows them to create and update customer accounts while setting parameters such as credit limits. Admins also manage partnerships with logistic companies, overseeing account details and their roles in the order fulfillment process. Additionally, the admin is responsible for managing orders, including approving orders, updating their status, assigning them to logistic companies, and overseeing the issuance of invoices and receipts. Cement management is another critical area, where admins can add and update cement products and track price changes.

For customers, the system facilitates the submission of new orders, where customers can specify the type and quantity of cement, as well as recipient details. The system also allows customers to submit payment evidence, which is verified and processed, ensuring that their orders are fulfilled smoothly. Customers can track the status of their orders at any time. Additionally, they can view and download invoices and receipts related to their orders, ensuring they have all the necessary documentation for their transactions.

In the logistic module, logistic companies have the tools to efficiently manage delivery orders assigned to them. They can view these orders and are responsible for assigning drivers and lorries to ensure that deliveries are completed on time. This module streamlines the logistics process, helping to optimize routes and improve the overall efficiency of the company's delivery operations. Overall, the CCLMS is designed to cover all aspects of the company's operations, ensuring a seamless and integrated workflow from order placement to delivery.

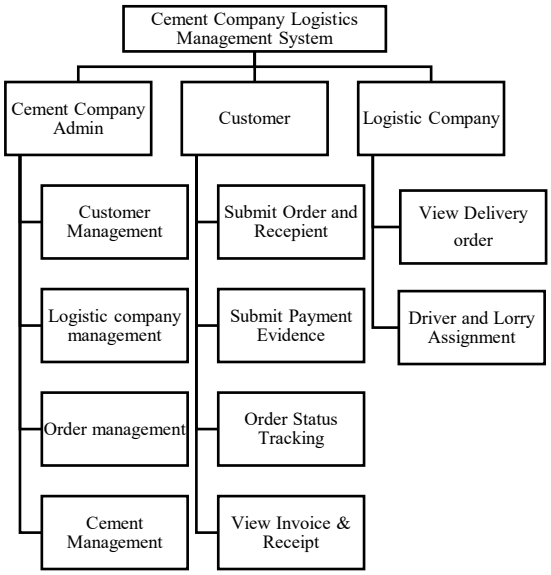


Figure 1.1: Structured Chart

1.5 Project Significance

The implementation of an automated system for Unipertiwi Sdn. Bhd. holds immense significance, promising a transformative impact on the company's operations and overall efficiency. By streamlining the current manual workflow, the system will reduce ordering processing times, facilitate driver and lorry assignment of logistic company, and automate the invoice and billing process.

1.6 Expected Outcomes

Output 1: Automated generation of purchase order, sales order, delivery order.

Output 2: Automated generation of invoice, receipt.

Output 3: Driver and lorry assignment time table.

1.7 Conclusion

In this chapter, the background of the project is being discussed in detail. The problem statement is figured out and is aimed to be solved based on the objectives listed. The scope of system modules and users are also discussed to provide a better solution to overcome the challenges and problems. The system is expected to have 3 modules which are cement company admin, customer and logistic. The system will be developed web based as the users can use the system whenever there is a connection.

CHAPTER 2

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

This chapter describes the development methodology utilized for the project. While several System Development Life Cycle (SDLC) approaches, such as waterfall, agile, and spiral models, can be applied to build and test software effectively, the Database Life Cycle (DBLC) was chosen for this project. The DBLC offers a structured approach through essential stages: initial database analysis, design, implementation and data loading, training and assessment, and long-term operations and maintenance. A key characteristic of the DBLC is its ongoing nature, where tasks such as monitoring, updates, and maintenance continue throughout the database's life, well beyond the initial setup. In summary, the DBLC manages the database's entire lifecycle.

2.2 Project Methodology

The Database Life Cycle (DBLC) methodology was selected for the development of the CCLMS. The DBLC consists of six key phases: initial database study, database design, implementation and loading, testing and evaluation, and ongoing operations, maintenance, and evaluation.

2.2.1 Database Initial Study

In this phase, the existing manual business process is analyzed to identify the reasons for its inefficiencies. To gain a deeper understanding and evaluate the current system's workflow, an unstructured interview was conducted on 1st March 2024 with Ms Gigi Pang, the director of the cement company and Ms Chu Lee Fang, the staff of one of the partner logistic companies to understand business flow of the ordering process.

From the unstructured interview, the problems and limitations of the current system were identified, leading to the creation of objectives that form the fundamental requirements for the new system. An analysis was then conducted to differentiate between the ideal and the achievable by acknowledging the system's scope and boundaries. The scope outlines the

design's extent based on operational needs, such as defining the necessary data structures, data types, the number of entities, and the physical size of the database.

2.2.2 Database Design

The second phase centers on crafting the database model to fulfill the data requirements that support the system's functionality and goals. This phase is crucial for ensuring the final solution meets the system's specifications. It is divided into four key sub-phases: conceptual design, DBMS selection, logical design, and physical design. In the conceptual design stage, an entity-relationship diagram (ERD) is created, and normalization is applied according to the system's specifications. Microsoft SQL Server was chosen as the DBMS for this system. In the logical design phase, data definitions and business logic were outlined, and a relational data model was established to define the necessary data and queries. Finally, the physical design stage organized attributes from the logical model into detailed tables, columns, indexes, sequences, and constraints, reflecting the core business rules and data relationships.

2.2.3 Implementation and Loading

In this phase, the DBMS software which is SQL Server Management Studio (SSMS) is installed in the computer, and connected with Microsoft SQL Server in the local system. Next, the database is created using Data Definition Language (DDL) based on the Entity-Relationship Diagram (ERD) produced. Data Manipulation Language (DML) is then utilized to populate data into the database tables. The website is developed using the ASP.NET framework, with C# serving as the programming language for the backend.

2.2.4 Testing and evaluation

This phase consists of three sub-phases: testing the database, fine-tuning the database, and evaluating the database and its application programs to ensure data integrity, security, and performance.

Test the Database: During this sub-phase, the database is tested to verify data integrity and security, ensuring the system's performance. Data integrity is enforced by the DBMS through the use of primary keys, foreign keys, unique constraints, and other mechanisms. Data security is tested by evaluating password protection, privileges, and access rights. Unit testing is performed to check individual functions like registration, login, and admin features by inputting sample data to assess database connections and error messages, ensuring correct data types are entered. System testing is then conducted to validate the business processes and ensure the system operates without errors.

Fine-Tune the Database: If the database shows errors or does not meet the required data specifications, it is fine-tuned to improve performance and ensure it aligns with the data requirements.

Evaluate the Database and Its Application Programs: Once the database has been tested and fine-tuned, an evaluation process is carried out. This involves broader system testing to confirm that all components function together correctly and meet the data requirements.

2.2.5 Operation

Upon completing the evaluation process, the system transitions into a fully operational information system. Users will then be able to utilize the system to handle essential tasks such as data entry, data management, and generating reports, facilitating the required information flow.

2.2.6 Maintenance and Evolution

The database maintenance phase includes routine activities like index maintenance, table optimization, user management, and password updates, along with regular backups and restoration in the event of a failure. As the system and organization grow, new requirements or changes may emerge, prompting adaptive maintenance. This could involve adding new fields or creating additional tables to enhance system performance and address evolving needs.

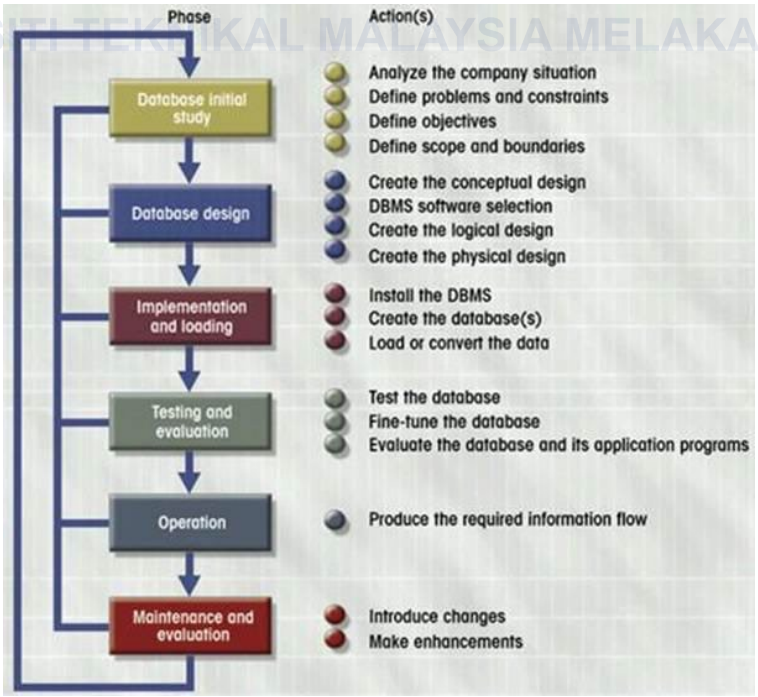


Figure 2.1:Database Life Cycle Illustration

2.3 Project Schedule and Milestones

Table 2.1: Gantt Chart of Cement Company Logistic Management System (CCLMS)

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Database Initial Study															
Database design															
Implementation and loading															
Testing and evaluation															

Table 2.2: Project Milestones

Milestones	Expected Outcomes	Date
Interview Ms Gigi Pang & Ms Chu Lee Fang	The business flow of the cement company and partnered logistic company.	10/3/2024
Problems identification and analysis	Problem statement Objective Flow chart of the current and proposed system Requirement and module of proposed system.	18/3/2024
Conceptual design of the proposed system	DFD and ERD of the proposed system Business rules Data Definition Language Data Manipulation Language Normalization and query	5/4/2024
Implementation of the proposed system	Database environment set up Graphical User Interface (GUI) Source code Procedure and trigger.	15/4/2024
Testing of the proposed system functionality	Test plan Test case Test result and analysis Solution solving of error message from testing process	31/5/2024
Completed documentation and log book	PSM 1 final report Project Demonstration	24/6/2024
Project Demonstration	Final Presentation	24/6/2024

2.4 Conclusion

This chapter outlines the chosen methodology that guided the system's development, detailing the methods used and the processes involved in each phase of the Database Life Cycle (DBLC), as well as the rationale for selecting this methodology. The DBLC was selected for this project because it encompasses six stages: initial database study, database design, implementation and loading, training and evaluation, and operations and maintenance. Additionally, the maintenance phase will persist throughout the system's operation to ensure its smooth functioning. A more comprehensive technical analysis of the overall system will be provided in the next chapter.

CHAPTER 3

ANALYSIS

3.1 Introduction

This chapter focuses on the analysis phase of the project, where the system is broken down into sub-modules to gather factual data, understand processes, identify problems, and recommend viable solutions for improvement. This includes examining the current business processes of the cement company, collecting operational data, analyzing data flow, pinpointing user bottlenecks, and devising strategies to ease their workload. The chapter will emphasize comprehending the existing business operations, recognizing weaknesses in the recruitment process, and proposing enhancements for the new system to meet user needs.

3.2 Problem Analysis

Figure 3.1 shows the current manual business workflow at Unipertiwi Sdn. Bhd. begins with customers contacting the administrative team to place their orders. This process, while traditional, is prone to delays and miscommunications due to its reliance on phone calls or other manual forms of communication. Once orders are received, admin will check the stock and decide whether or not to approve customer order. This manual approach not only introduces inefficiencies but also increases the likelihood of errors in order processing and fulfilment.

Following order approval, the administrative team manually contacts logistics partners to send out delivery orders, either through face-to-face interaction or via messaging platforms like WhatsApp. Additionally, logistics personnel must manually check availability, assign drivers and vehicles, and execute. Submitting the prove of delivery (POD) through face-to-face interaction or via WhatsApp to the administrative team introduces an additional manual step, exacerbating inefficiencies in the delivery workflow. This is compounded by the fact that admins often make mistakes due to the necessity of cross-referencing the delivery order number with the received POD for approval.

In the final stage of the current workflow, administrative staff issue invoices in paper format, contributing to administrative burden and potential delays in payment processing. Customers, in turn, are required to make payments and provide proof of payment via manual

channels (WhatsApp), prolonging transaction times. The issuance of paper receipts further extends the administrative overhead, completing a cumbersome billing and invoicing cycle. This manual approach not only hampers efficiency but also impedes financial transparency and accuracy in the billing process.

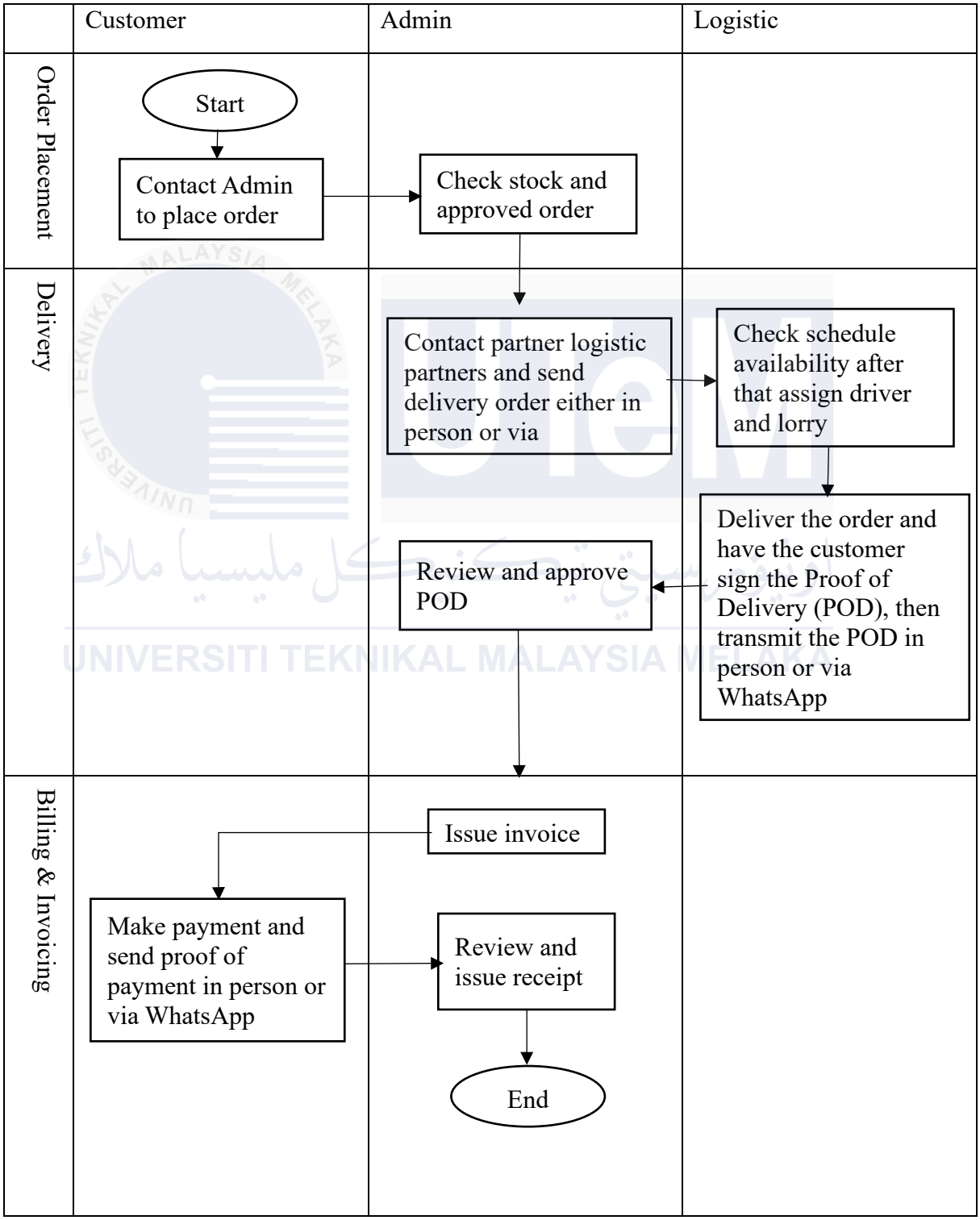


Figure 3.1: Current Business Flow

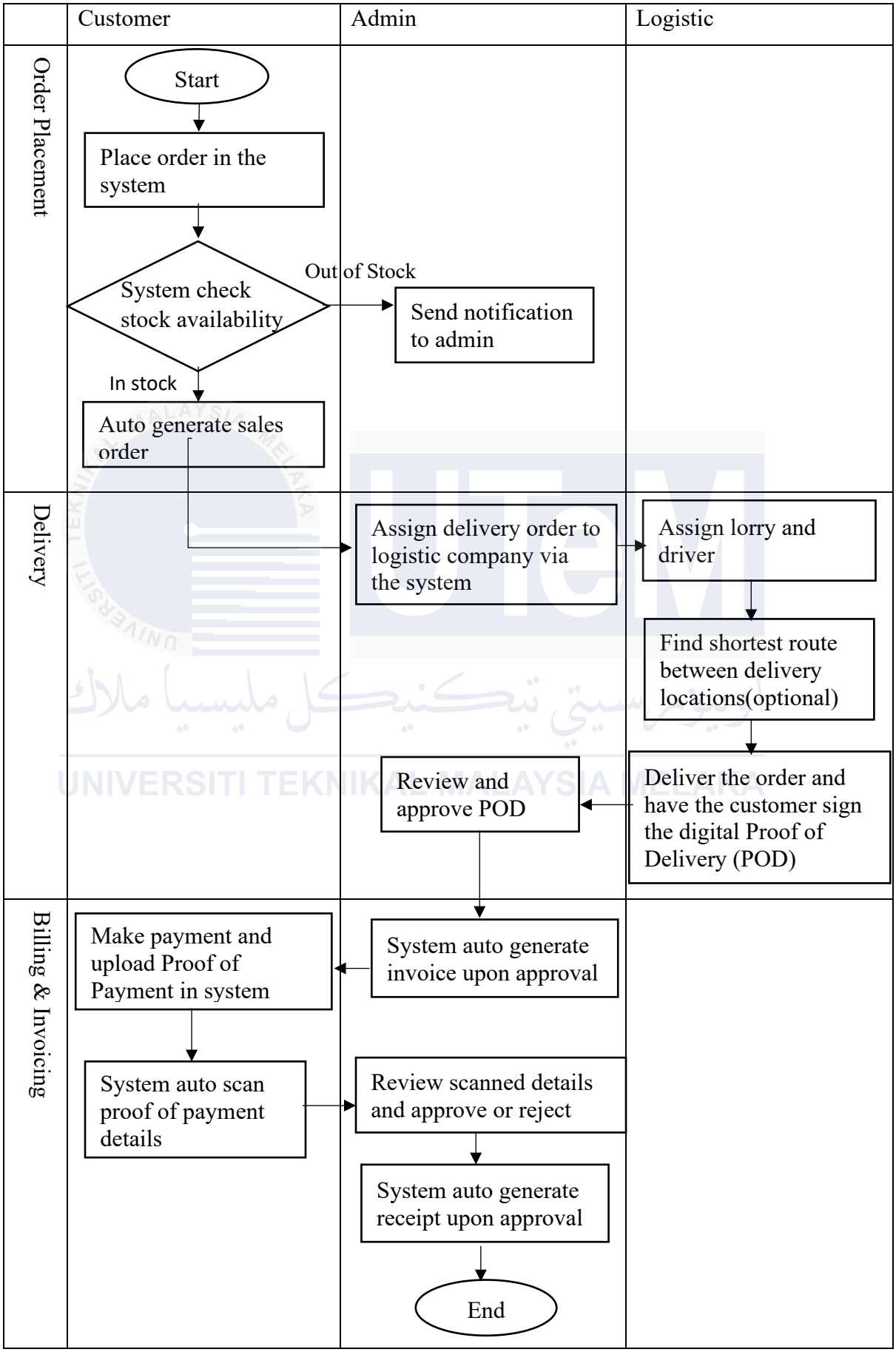


Figure 3.2: Proposed Improvement

3.3 Requirement Analysis

3.3.1 Functional Requirement

a) Cement Company Admin

- Admin can create new customer accounts by entering required details such as name, contact information, and address.
- Admin can set and update the credit limit for each customer.
- System should enforce credit limit constraints during order processing.
- Admin can create accounts for partnered logistic companies by entering necessary details such as company name, contact information, and address.
- Admin can view and approve customer orders.
- Admin can update the status of orders (e.g., pending, approved, shipped, delivered, cancelled).
- Admin can assign approved orders to partnered logistic companies for delivery.
- System can generate and issue invoices to customers upon order approval.
- System can generate and issue receipts upon payment confirmation.
- Admin can add new cement products and update existing cement product details.
- System should maintain a history of price changes for each cement product.

b) Customer

- Customers can place orders for cement products by selecting desired products and quantities.
- Customers can upload proof of payment for their orders.
- Customers can view the status of their orders.
- Customers can view and download invoices for their orders.
- Customers can view and download receipts upon payment confirmation.

c) Logistic Company

- Logistic company users can view the list of orders assigned to their company for delivery.
- Logistic company users can assign drivers and lorries to each delivery order.
- Logistic company users can upload proof of delivery.

3.3.2 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a graphical representation that illustrates the processes or activities within a system and how data moves between those functions. For this project, a context diagram is first created to show the external entities involved and the data flows originating or ending with them. Following this, a DFD fragment is constructed based on the project requirements, detailing the flow of data through various processes, interactions with external entities, and connections to data stores. Figures 3.3 to 3.7 provide these diagrams for the Cement Company Logistics Management System (CCLMS).

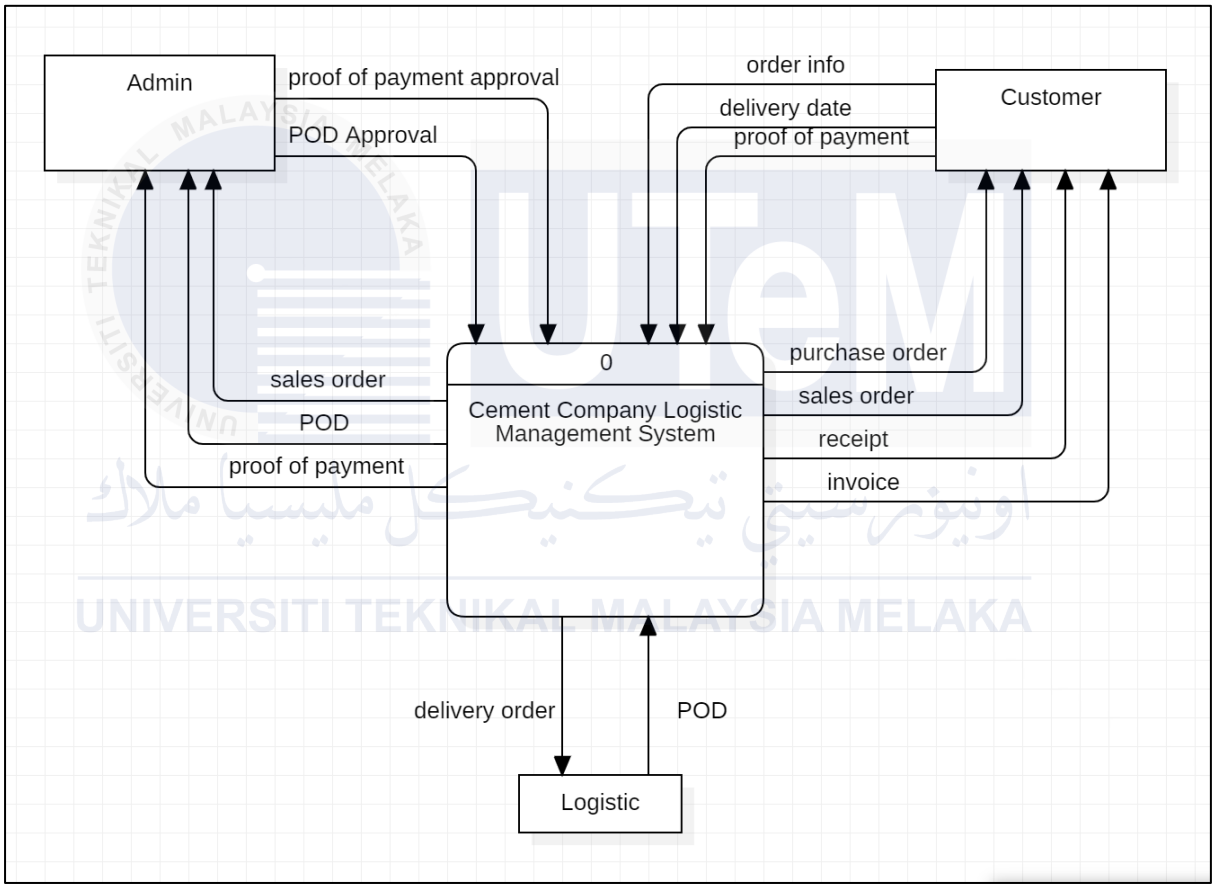


Figure 3.3: Context Diagram

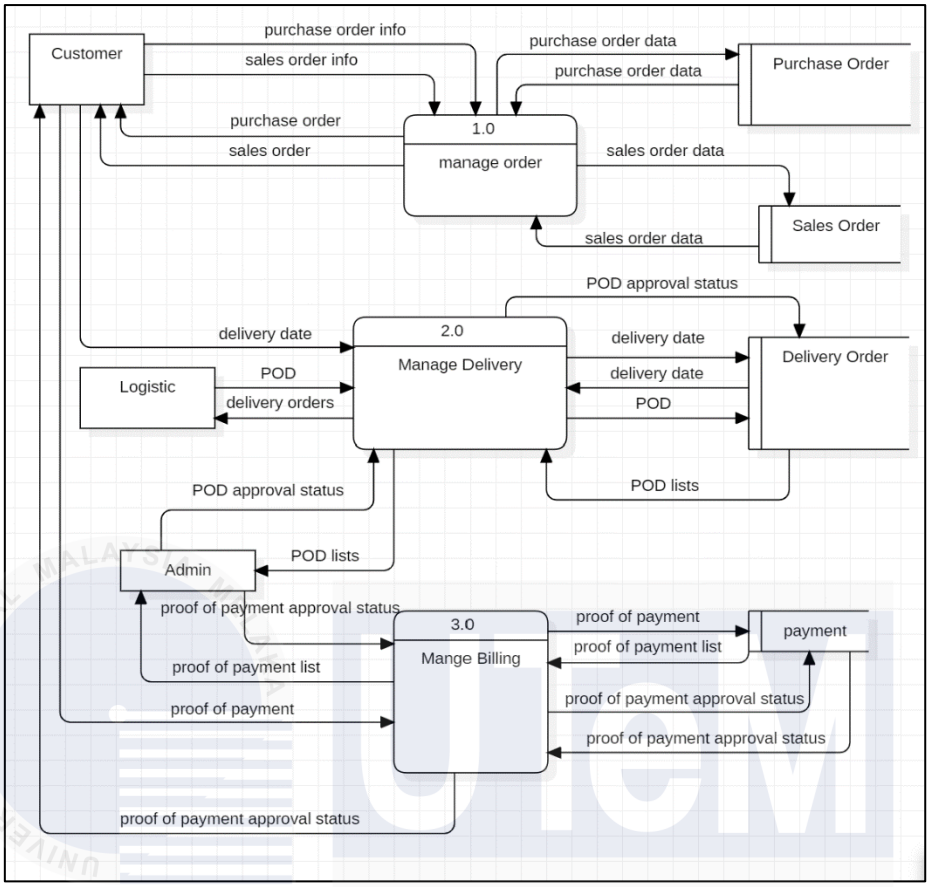


Figure 3.4: DFD Level 1

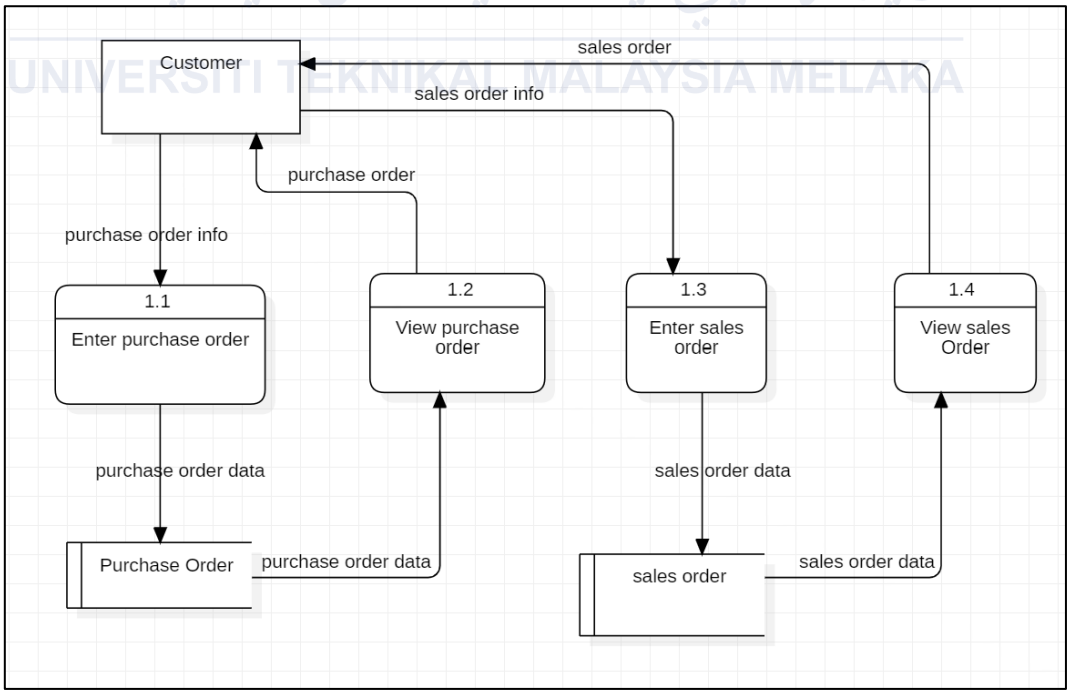


Figure 3.5: DFD Level 2-Manage Order

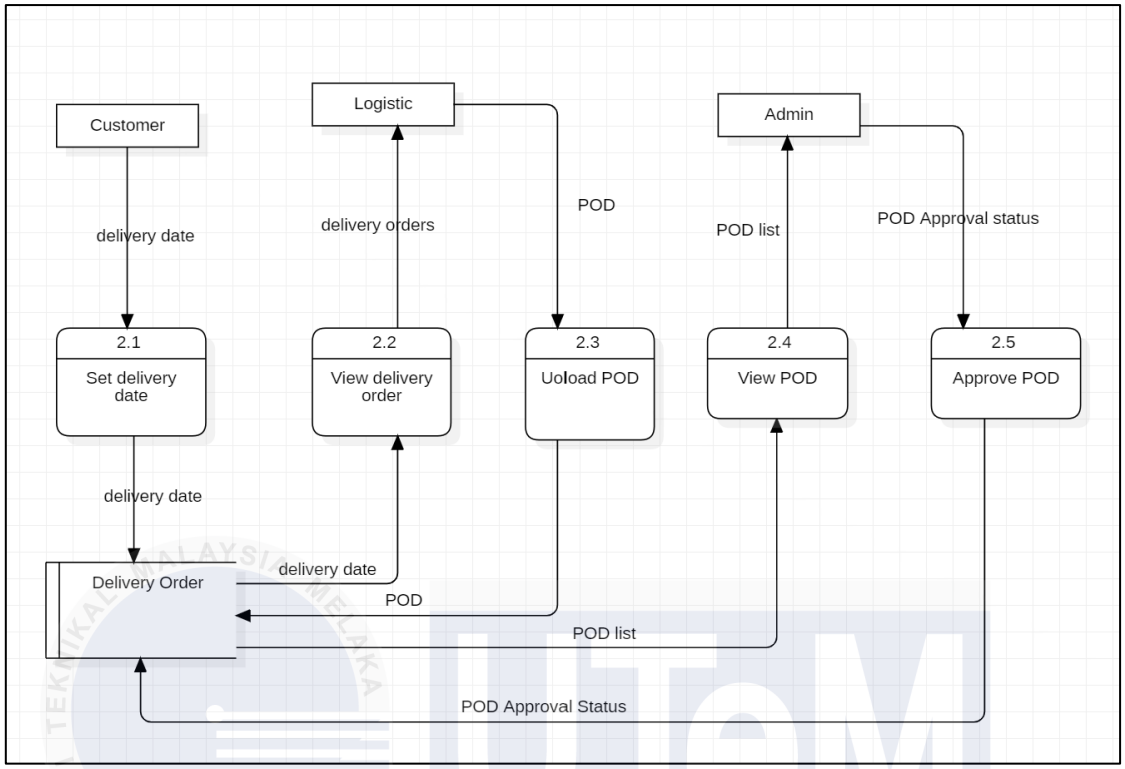


Figure 3.6: DFD Level 2-Manage Delivery

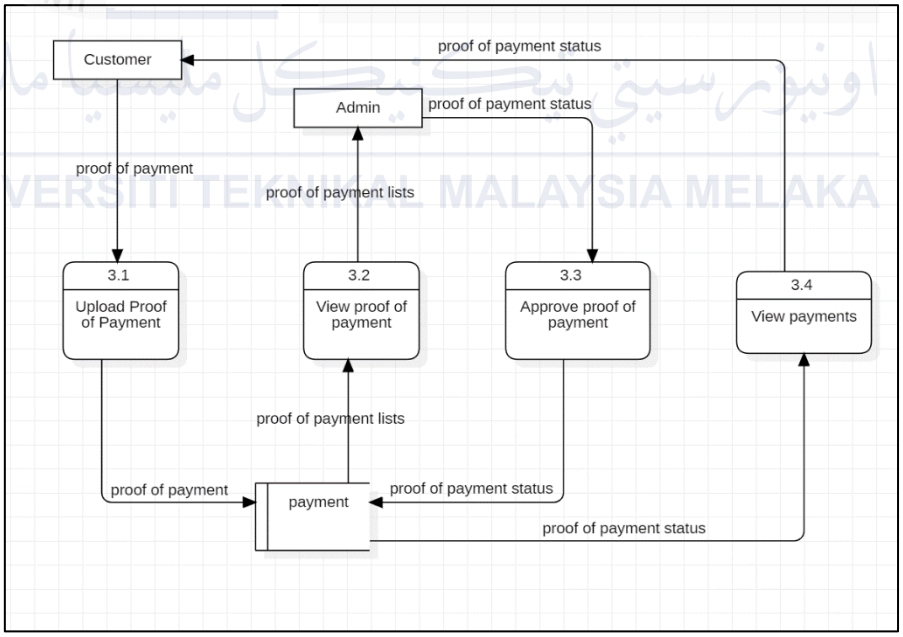


Figure 3.7: DFD Level 2-Manage Billing

3.3.3 Non-functional Requirement

Table 3.1: Non-Functional Requirement

Aspect	Requirement
Performance	-The system must ensure high reliability to maintain consistent operations. -Clear and appropriate error messages should be provided to notify users when a request cannot be processed.
Security	-The system should authenticate users with a valid username and password and determine user access levels based on their roles.
Usability	- The system should have a user-friendly interface that is intuitive and easy for users to learn and operate. -The system should have an easy navigation and access to key functionalities.
Reliability	-The calculation of order and billing documents must be correct -The dashboard sales should be accurate base on the data in database

3.3.4 Software Requirement

Table 3.2: Software Requirement

Software	Description
Microsoft Visual Studio	Usage: Integrated Development Environment (IDE) for developing the web application. Purpose: Used for writing, debugging, and deploying ASP.NET code. It provides tools

	and features to enhance productivity and streamline the development process.
SQL Server Management Studio (SSMS)	Purpose: Used for designing, managing, and querying the database. It helps in creating and managing database schemas, tables, stored procedures, and performing database maintenance tasks.
ASP.NET	Usage: Web framework for building web applications and services with .NET. Purpose: Used for creating the web interfaces and handling HTTP requests. It allows the development of dynamic, data-driven web applications with robust security, performance, and scalability features.
PyCharm	Usage: Integrated Development Environment (IDE) for Python development. Purpose: Used for developing Python endpoints and scripts.

3.3.5 Hardware Requirement

Table 3.3: Hardware Requirement

Hardware	Description
Laptop	To develop the website.

3.4 Conclusion

Flow charts illustrating the current (as-is) and proposed (to-be) systems have been presented to clarify the concept of system flow. The context diagram provides a high-level overview of the entire system to be developed, while the data flow diagram (DFD) breaks down the flow of data within the system's processes. The DFD details how data moves through inputs and outputs, associated with each entity and process. In conclusion, Chapter 3 has summarized the investigation of the current system, highlighted areas for improvement, and established a conceptual framework for the new system.

CHAPTER 4

DESIGN

4.1 Introduction

In the design phase, the architecture of the proposed system is established, focusing on creating a design that meets the agreed application requirements. The requirements developed during the analysis phase are refined and expanded to encompass all specified functions of the application. This chapter will cover the system architecture design, database design, and graphical user interface (GUI) design, ensuring that the final product aligns with user needs and system objectives.

4.2 System Architecture Design

Figure 4.1 shows the three-tier architecture in ASP.NET is a well-established software architecture pattern that organizes an application into three main logical layers: Presentation Tier, Business Tier, and Data Tier. This structured approach helps in organizing code, improving scalability, and making the application easier to maintain and update.

The Presentation Tier is the topmost layer of the application, dedicated to displaying the user interface and handling user interactions. It consists of HTML, CSS, JavaScript, and ASP.NET Web Forms or MVC views, providing a platform for users to interact with the application. This tier is responsible for rendering data to the user, capturing user inputs, and facilitating communication with the Business Tier.

The Business Tier, also known as the middle tier or logic tier, serves as the core processing unit of the application, where the business logic is implemented. This layer contains business logic classes, services, and controllers that execute the business rules and workflows of the application. It validates data, orchestrates operations by interacting with both the Presentation Tier and the Data Tier, and ensures the business rules are adhered to. The Business Tier relies on tools like ASP.NET Controllers.

The Data Tier, also known as the database layer, is responsible for data storage, retrieval, and management. This tier includes the SQL Server database, tables, stored procedures, and Entity Framework models. It manages database operations such as Create, Read, Update, and

Delete (CRUD), ensuring data integrity and security. The Data Tier provides the necessary data to the Business Tier as required and is managed using tools like Microsoft SQL Server, SQL Server Management Studio (SSMS) and Entity Framework (EF).

Adopting a three-tier architecture provides several benefits. Firstly, it ensures a clear separation of concerns, with each tier having distinct responsibilities, making the system more modular and easier to manage. Secondly, it enhances scalability, as each tier can be scaled independently to handle increased loads. Thirdly, it improves maintainability, allowing changes in one tier (such as UI updates in the Presentation Tier) without affecting the others, making the system easier to update. Lastly, it promotes reusability, as business logic and data access code can be reused across different parts of the application or even in different applications.

Implementing this architecture for the cement company management system will result in a well-organized, scalable, and maintainable application, ensuring robust performance and efficient operations.

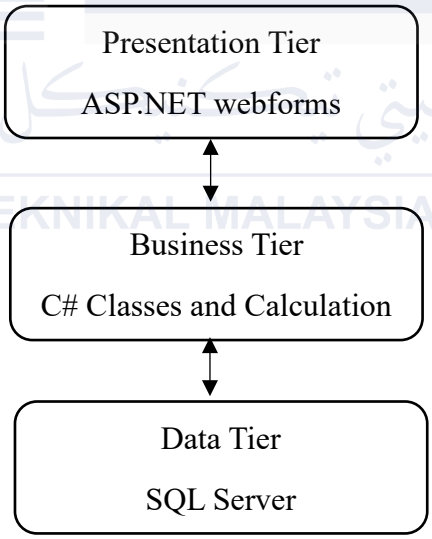


Figure 4.1: Tier Architecture

4.3 Database Design

Database design is the process of defining the structure, organization, and constraints of a database to effectively store, manage, and retrieve data. It involves creating a detailed data model that outlines the entities, attributes, and relationships within the database, ensuring that the design aligns with the specific requirements of the application or system. The design phase typically includes conceptual, logical, and physical design stages, which help to optimize data integrity, reduce redundancy, and enhance overall performance. A well-designed database not only supports current data needs but also allows for scalability and adaptability as requirements evolve over time.

4.3.1 Conceptual Database Design

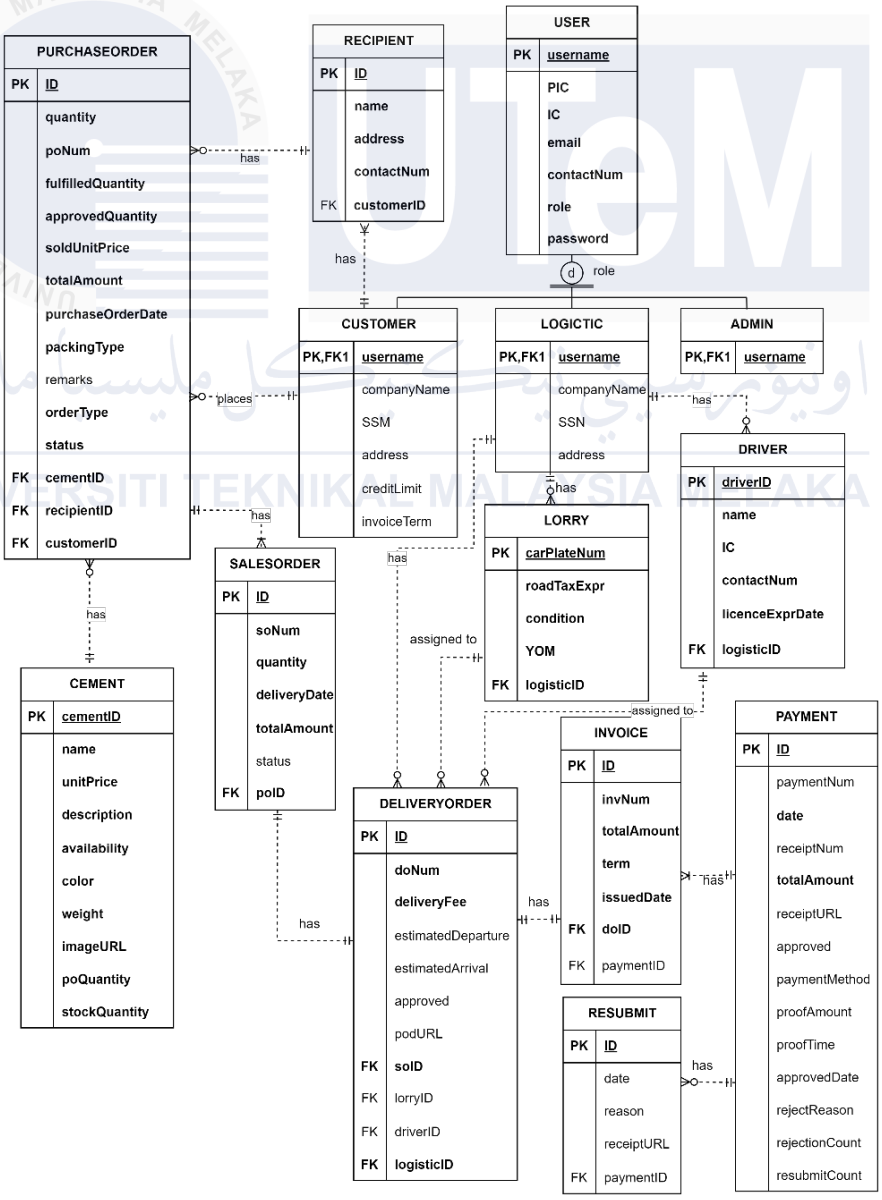


Figure 4.2: ERD

Business Rule

a) Each cement has zero or many purchase orders.
   Each purchase order can has one and only one cement.
b) Each customer places zero or many purchase order.
   Each purchase order can be placed by one and only one customer.
c) Each recipient has zero or many purchase orders.
   Each purchase order can has one and only one recipient.
d) Each purchase order has one or many sales order.
   Each sales order has one and only one purchase order.
e) Each sales order has one and only one delivery order.
   Each delivery order has one and only one sales order.
f) Each logistic has zero or many delivery order.
   Each delivery order has one and only one logistic.
g) Each logistic has zero or may drivers.
   Each driver has one and only one logistic.
h) Each logistic has zero or may lorries.
   Each lorry has one and only one logistic.
i) Each logistic has zero or many delivery orders.
   Each delivery order has one and only one logistic.
j) Each lorry can be assigned to zero or many delivery order.
   Each delivery order has one and only one lorry.
k) Each driver can be assigned to zero or many delivery order.
   Each delivery order has one and only one driver.
l) Each delivery order has one and only one invoice.
   Each invoice has one and only one delivery order.
m) Each payment has one or many invoices.
   Each invoice has one and only one payment.
n) Each payment has zero or many resubmits.
   Each resubmit has one and only one payment.

4.3.2 Logical Design

Logical design is the second part of database design. In this phase, conceptual design is translated into logical design that is in relational model. In relational model, entities are translated into tables. Data dictionary consists of the structure design of the tables following the data types of the targetted DBMS (MS SQL Server).

4.3.2.1 Data Dictionary

Table 4.1: Data Dictionary of User Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
Username	Unique identifier used by the user to log in.	nvarchar(max)	Yes	PK	
Password	Secret key for user authentication.	nvarchar(30)	Yes		
companyName	Name of the customer's company.	nvarchar(100)			
creditLimit	Maximum allowable credit for a customer.	float			
invoiceTerm	Payment terms of invoices.	int			
address	Location of the customer company.	nvarchar(200)			
SSN	Social Security Number of the company.	nvarchar(30)			
PIC	Person In Charge at the company.	nvarchar(100)	Yes		
IC	Identity Card number of the PIC.	nvarchar(14)	Yes		
contactNum	Phone number of PIC.	nvarchar(20)	Yes		
email	Email address of PIC.	nvarchar(80)	Yes		
role	User's role (admin, customer, logistic)	nvarchar(10)	Yes		

Table 4.2: Data Dictionary of Cement Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for each cement product.	int	Yes	PK	
Name	Name of the cement product.	nvarchar(50)	Yes		
Description	Detailed information about the cement product.	nvarchar(max)	Yes		
unitPrice	Price per unit of the cement.	Float	Yes		
Color	Color of the cement product.	nvarchar(20)	Yes		
Weight	Weight of the cement product per unit.	float	Yes		
availability	Availability status of the cement product (Yes or No).	nvarchar(10)	Yes		
ImageURL	URL of the image representing the cement product.	nvarchar(max)	Yes		
poQuantity	Quantity of cement ordered in purchase orders.	Int	Yes		
stockQuantity	Quantity of cement available in stock.	int	Yes		

Table 4.3: Data Dictionary of Recipient Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Name of the recipient.	int	Yes	PK	
Name	Name of the recipient.	nvarchar(max)	Yes		
Address	Address of the recipient.	nvarchar(100)	Yes		
contactNum	Contact number of the recipient.	nvarchar(250)	Yes		
customerID	Unique identifier for the customer associated with the recipient.	nvarchar(20)	Yes	FK	User

Table 4.4: Data Dictionary of Purchase Order Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for the order.	int	Yes	PK	
poNum	Purchase order number.	nvarchar(16)	Yes		
Quantity	Total quantity of cement ordered.	int	Yes		
fulfilledQuantity	Quantity of cement that has been fulfilled (delivered).	int	Yes		
ApprovedQuantity	Quantity of cement that has been approved for fulfillment.	int	Yes		
soldUnitPrice	Unit price at which the cement is sold.	float	Yes		

totalAmount	Total amount for the order calculated as approved quantity multiplied by sold unit price.	float	Yes		
orderDate	Date when the order was placed.	Datetime 2(7)	Yes		
packingType	Type of packing for the order, either palletized or non-palletized.	nvarchar(20)	Yes		
Remarks	Additional remarks about the order.	nvarchar(max)			
orderType	Type of order, either single or bulk.	nvarchar(20)	Yes		
Status	Current status of the order.	nvarchar(30)	Yes		
cementID	Unique identifier for the type of cement ordered.	int	Yes	FK	Cement
customerID	Unique identifier for the customer placing the order.	nvarchar(450)	Yes	FK	User
recipientID	Unique identifier for the recipient of the order.	int	Yes	FK	Recipient

Table 4.5: Data Dictionary of Lorry Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
carPlateNum	The license plate number of the lorry.	nvarchar(450)	Yes	PK	
roadTaxExprDate	The expiration date of the lorry's road tax.	Datetime2(7)	Yes		
Condition	The current condition of the lorry.	Nvarchar(30)			
YOM	Year of Manufacture of the lorry.	Int	Yes		
logisticID	Unique identifier for the logistic company that owns or operates the lorry.	Nvarchar(450)	Yes	FK	User

Table 4.6: Data Dictionary of Driver Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for the lorry driver.	int	Yes	PK	
Name	Full name of the lorry driver.	Nvarchar(100)	Yes		
IC	Identification card number of the lorry driver.	Nvarchar(14)	Yes		
contactNum	Contact number of the lorry driver.	Nvarchar(20)	Yes		
licenseExprdate	Expiration date of the lorry driver's license.	Datetime2(7)	Yes		

logisticID	Unique identifier for the logistic company that employs the lorry driver.	Nvarchar(450)	Yes	FK	User

Table 4.7: Data Dictionary of Sales Order Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for the sales order.	Int	Yes	PK	
soNum	Sales order number associated with the sales order.	Nvarchar(18)	Yes		
Quantity	Total quantity of cement ordered.	Int	Yes		
totalAmount	Total monetary amount for the sales order.	Float	Yes		
Status	Current status of the sales order.	30	Yes		
deliveryDate	Scheduled date for the delivery of the order.	Datetime2(7)	Yes		
poID	Identifier for the related purchase order.	int	Yes	FK	Purchase Order

Table 4.8: Data Dictionary of Delivery Order Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for the	int	Yes	PK	

	delivery order.				
doNum	Delivery order number associated with the delivery order.	Nvarchar(18)	Yes		
deliveryFee	Fee charged for the delivery service.	Float	Yes		
estimatedDeparture	Estimated departure date and time for the delivery.	Datetime2(7)			
estimatedArrival	Estimated arrival date and time for the delivery.	Datetime2(7)			
approved	Approval status of the proof of delivery.	Nvarchar(8)			
podURL	URL to the proof of delivery image.	Nvarchar(max)			
soID	Identifier for the related sales order.	int	Yes	FK	Sales Order
logisticID	Identifier for the logistic company handling the delivery.	Nvarchar(450)	Yes	FK	User
lorryID	Identifier for the lorry assigned to the delivery.	Nvarchar(450)		FK	Lorry
driverID	Identifier for the driver assigned to the delivery.	int		FK	Driver

Table 4.9: Data Dictionary of Invoice Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for the invoice.	int	Yes	PK	
invNum	Invoice number associated with the invoice.	Nvarchar(18)	Yes		
Term	Invoice term agreed upon.	Int	Yes		
date	Date when the invoice was issued.	Datetime2(7)	Yes		
doID	Identifier for the related delivery order.	int	Yes	FK	Delivery Order
paymentID	Identifier for the related payment transaction.	int		FK	Payment

Table 4.10: Data Dictionary of Payment Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for the payment.	Int	Yes	PK	
Date	Date when the payment information was uploaded.	Datetime2(7)	Yes		
totalAmount	Total amount paid.	Float	Yes		
receiptNum	Receipt number associated with the payment.	Nvarchar(16)			
Approved	Boolean indicating if the payment is approved.	Nvarchar(8)			

receiptURL	URL link to the scanned receipt.	Nvarchar(max)			
paymentMethod	Method used for the payment (e.g., credit card, bank transfer).	Nvarchar(40)			
proofAmount	Amount shown in the scanned receipt.	Nvarchar(40)			
proofDate	Date extracted from the scanned receipt.	Nvarchar(40)			
proofTime	Time extracted from the scanned receipt.	Nvarchar(40)			
approvedDate	Date when the payment was approved.	Datetime2(7)			
rejectReason	Reason for rejecting the payment, if applicable.	Nvarchar(200)			
RejectionCount	Number of times the payment has been rejected.	int			
paymentNum	Payment number associated with the transaction.	Nvarchar(16)			
resubmitCount	Number of times the payment has been resubmitted.	int			

Table 4.11: Data Dictionary of Resubmit Table

Attribute	Description	Data Type	Required	PK/FK	Reference Table
ID	Unique identifier for the resubmit.	int	Yes	PK	
reason	Invoice number associated with the invoice.	Nvarchar(200)			
date	Date when the invoice was issued.	Datetime2(7)	Yes		
receiptURL	URL link to the scanned receipt.	Nvarchar(max)	Yes		
paymentID	Identifier for the related payment transaction.	int	Yes	FK	Payment

4.3.2.3 Normalization

Normalization is a database design process focused on minimizing redundancy and enhancing data integrity by organizing a schema into clearly defined tables. Full dependency ensures that non-key attributes rely on the entire primary key, thus preventing partial dependencies. Transitive dependency arises when a non-key attribute depends on another non-key attribute rather than directly on the primary key. Figures 4.3 and 4.4 illustrate the normalization of the Purchase Order.

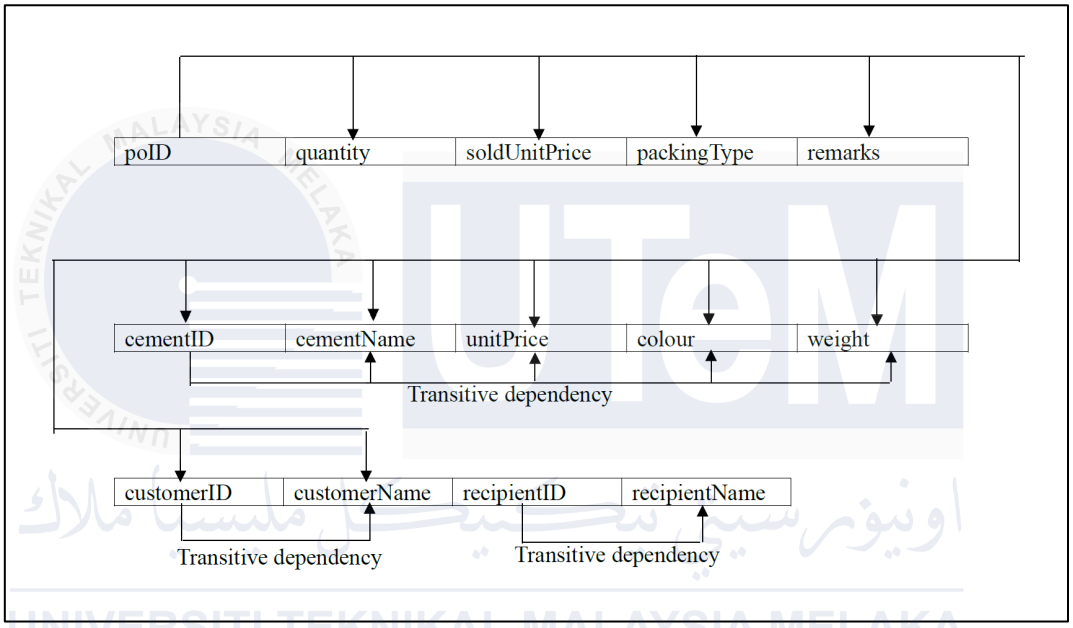


Figure 4.3: 1st and 2nd Normalization of Purchase Order

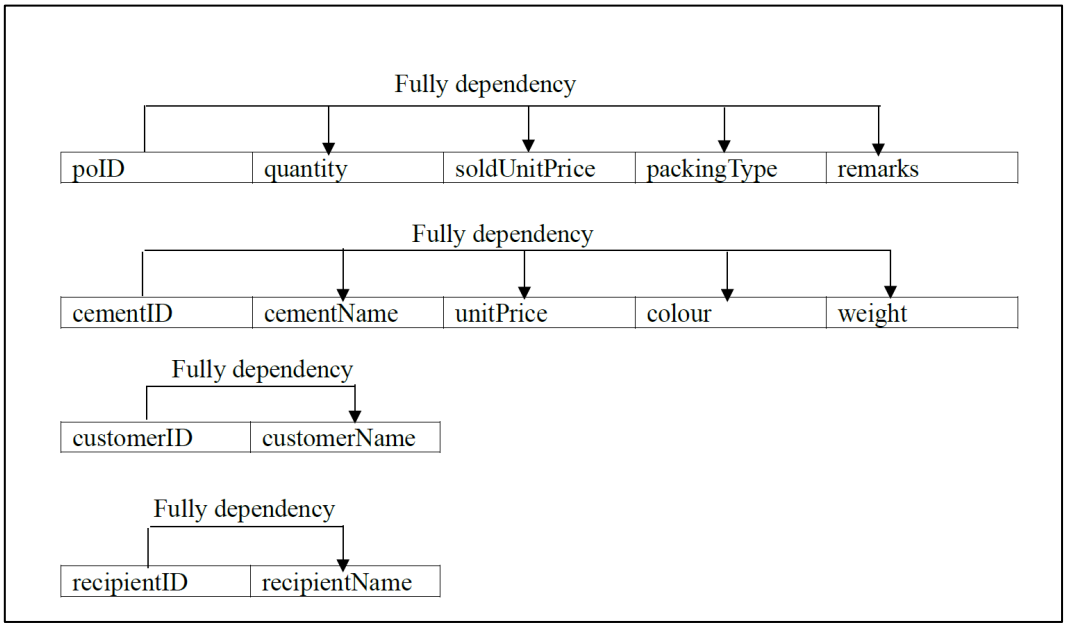


Figure 4.4: 3rd Normalization of Purchase Order

4.3.3 ERD Validation

Table 4.12: ERD Validation

Transaction	Entities Involved
To retrieve all the purchases order along with customer name, item, quantity, total amount and status	User: provide customer name Cement: item Purchase Order: quantity, total amount status
To retrieve all pending assignment sales order along with so number, delivery date, item, quantity, recipient address	Recipient: recipient address Sales Order: so number, delivery date, quantity, status Cement: item
Customer search for sales order using SONumber.	Sales Order: so number
Retrieve Top 5 best selling cements	Purchase Order: total amount Cement: cement name
Generate a list of sales order of certain month with status 'Invoice Issued' along with customer name.	Invoice- issuedDate, paymentID. Sales Order: status User-customer name
Find all lorries registered under the logistic company 'Rapid Logistic Sdn. Bhd.'	Lorry: lorry details User: logistic company name

4.3.4 Query Design

Table 4.12: Query Design

No	User Transaction	SQL Statement
1	To retrieve all the purchases order along with customer name, item, quantity, total amount and status	SELECT po.ID AS PurchaseOrderID, u.Name AS CustomerName, c.Name AS Item, po.Quantity, po.TotalAmount, po.Status FROM PurchaseOrders po JOIN Users u ON po.CustomerID = u.username JOIN

		Cement c ON po.CementID = c.CementID;
2	To retrieve all pending assignment sales order along with so number, delivery date, item, quantity, recipient address	SELECT so.soNum AS SalesOrderNumber, so.deliveryDate AS DeliveryDate, c.Name AS Item, so.Quantity, r.Address AS RecipientAddress FROM SalesOrders so JOIN Cement c ON so.cementID = c.ID JOIN Recipients r ON so.recipientID = r.ID WHERE so.Status = 'approved';
3	Customer search for sales order using SONumber.	SELECT * FROM SalesOrders WHERE soNum = @SONumber;
4	Retrieve Top 5 best selling cements	SELECT TOP 5 c.Name AS CementName, SUM(po.totalAmount) AS TotalSalesAmount FROM PurchaseOrders po JOIN Cement c ON po.cementID = c.ID GROUP BY c.Name ORDER BY TotalSalesAmount DESC;
5	Generate a list of sales order of certain month with status ‘Invoice Issued’ along with customer name.	SELECT so.soNum AS SalesOrderNumber, so.Status, u.Name AS CustomerName, i.issuedDate FROM SalesOrders so JOIN Users u ON so.customerID = u.ID JOIN Invoice i ON so.ID = i.soID WHERE so.Status = 'Invoice Issued' AND MONTH(i.issuedDate) = @Month -- Replace @Month with the specific month (e.g., 6 for June) AND YEAR(i.issuedDate) = @Year; -- Replace @Year with the specific year

6	Find all lorries registered under the logistic company ‘Rapid Logistic Sdn. Bhd.’	SELECT
		  l.carPlateNum,
		  l.roadTaxExprDate,
		  l.Condition,
		  l.YOM,
		  l.logisticID
		FROM
		  Lorries l
		JOIN
		  Users u ON l.logisticID = u.ID
		WHERE
		  u.companyName = 'Rapid Logistic Sdn. Bhd.';

4.3.5 Application of Procedure & Trigger

Table 4.13 Application of Procedure

Procedure	Database Table	Query

Insert	Recipient	CREATE PROCEDURE [dbo].[AddRecipient]
		(
		  @CustomerID NVARCHAR(50),
		  @Name NVARCHAR(100),
		  @Address NVARCHAR(100),
		  @ContactNum NVARCHAR(20)
		)
		AS
		BEGIN
		  INSERT INTO Recipients (CustomerID, name, address, contactNum)
		  VALUES (@CustomerID, @Name, @Address, @contactNum)
		END

Select	Recipient	CREATE PROCEDURE [dbo].[GetRecipients]
		(
		  @CustomerID NVARCHAR(50)
		)
		AS
		BEGIN
		  SET NOCOUNT ON;
		  SELECT * FROM Recipients WHERE CustomerID = @CustomerID;
		END;

Update	Recipient	ALTER PROCEDURE [dbo].[UpdateRecipient]
		(
		  @RecipientID INT,
		  @Name NVARCHAR(100),
		  @Address NVARCHAR(100),
		  @ContactNum NVARCHAR(20)
		)

		AS
		BEGIN
		SET NOCOUNT ON;
		UPDATE Recipients
		SET Name = @Name,
		Address = @Address,
		ContactNum = @ContactNum
		WHERE id=@RecipientID;
		END;

Delete	Recipient	CREATE PROCEDURE DeleteRecipient
		(
		@RecipientID INT
		)
		AS
		BEGIN
		SET NOCOUNT ON;
		DELETE FROM Recipients WHERE Id = @RecipientID;
		END;

Table 4.14 Application of Trigger

Trigger	Database Table	Query	Explanation
After Insert	SalesOrders	ALTER TRIGGER [dbo].[AfterSalesOrderInsert] ON [dbo].[SalesOrders] AFTER INSERT AS BEGIN -- Update PurchaseOrder table UPDATE PurchaseOrders SET approvedQty = approvedQty + 800 WHERE Id IN (SELECT poID FROM inserted); -- Update Cement table UPDATE Cements SET poQuantity=poQuantity-800, stockQuantity = stockQuantity - 800 WHERE Id IN (SELECT c.Id FROM Cements c INNER JOIN PurchaseOrders po ON c.id = po.cementID	This trigger ensures that after each insertion into the SalesOrders table, the related PurchaseOrders and Cements tables are appropriately updated to reflect the changes in approved quantities and stock levels.

		INNER JOIN inserted i ON po.Id = i.poID); END;	

After Insert	PurchaseOrders	ALTER trigger [dbo].[AfterPurchaseOrderInsert] ON [dbo].[PurchaseOrders]
		AFTER INSERT
		AS
		BEGIN
		
		-- Update Cement table
		UPDATE Cements
		SET
		poQuantity = poQuantity + i.quantity
		FROM Cements c
		INNER JOIN inserted i ON c.id = i.cementID;
		END;
			This trigger ensures that the Cements table accurately reflects the updated purchase order quantities whenever new purchase orders are added.

4.4 Graphical User Interface (GUI) Design

Graphical User Interface design is the process of designing the aesthetics and user interaction of the system. The design often focuses on ease-of-use to ensure users can interact with the software easily and intuitively. Graphical user interface design consists of three designs which are navigation design, output design and input design.

4.4.1 Navigation Design

Navigation design is to design the system navigation and how the users can navigate between the screens. Navigation can be image based and text based and is the act of moving from one screen to another. A good navigation design can enable users to spend minimal time to make the right interactions.

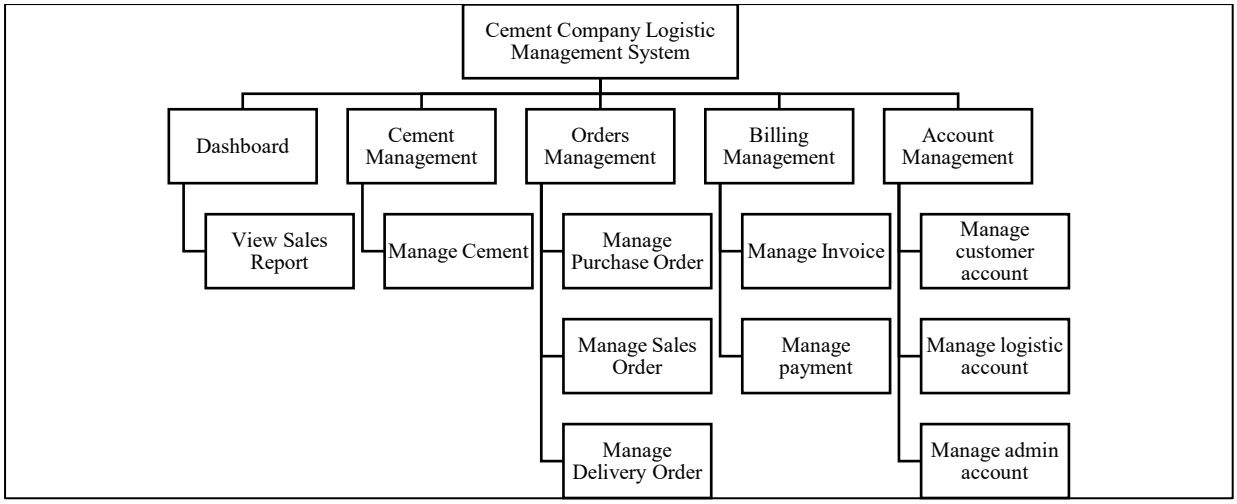


Figure 4.5: Navigation Design of Admin Interface

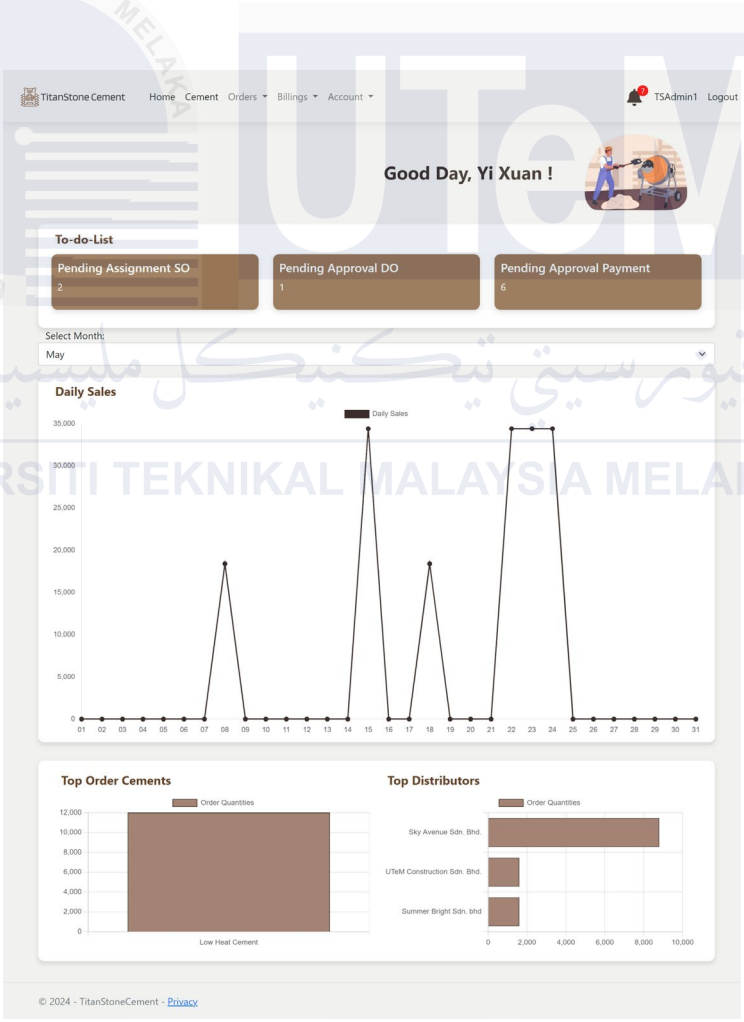


Figure 4.6: Admin Dashboard

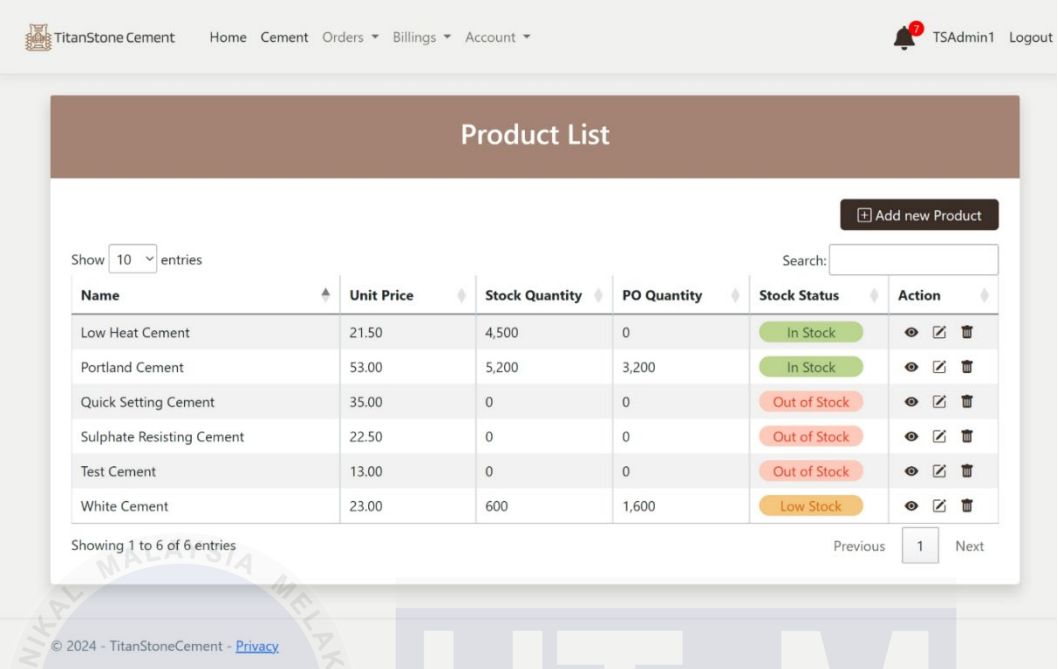


Figure 4.7: Admin Cement Management Interface

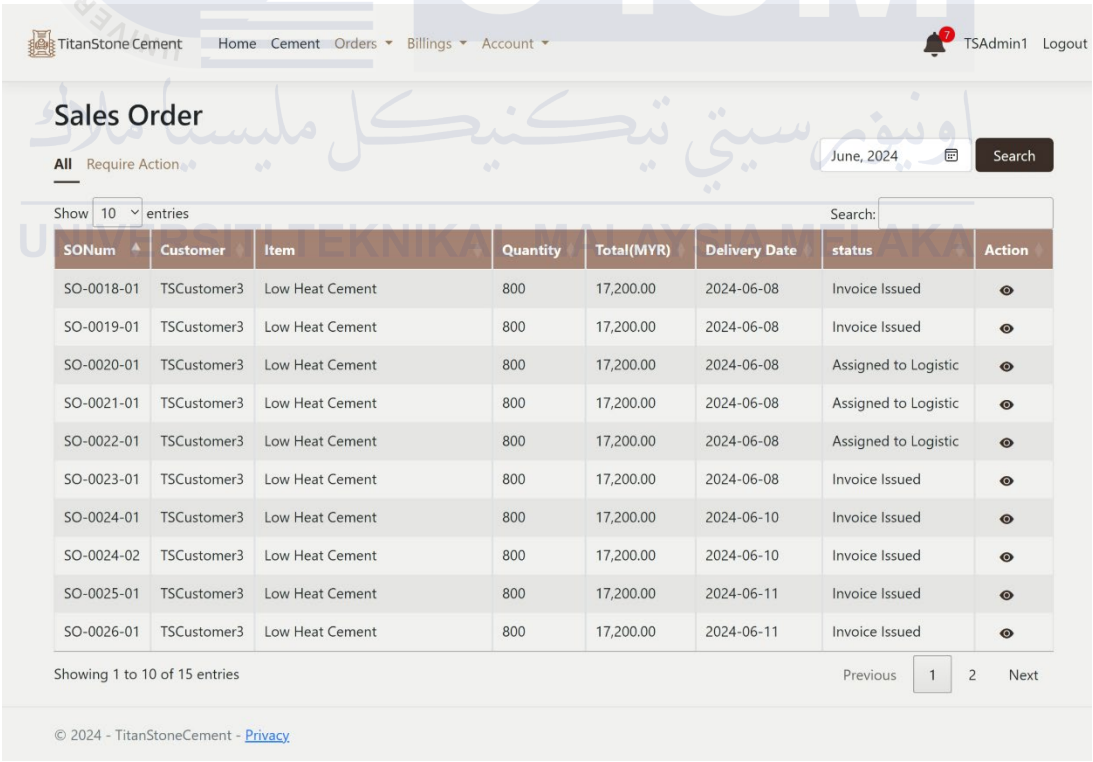


Figure 4.8: Admin Order Management Interface

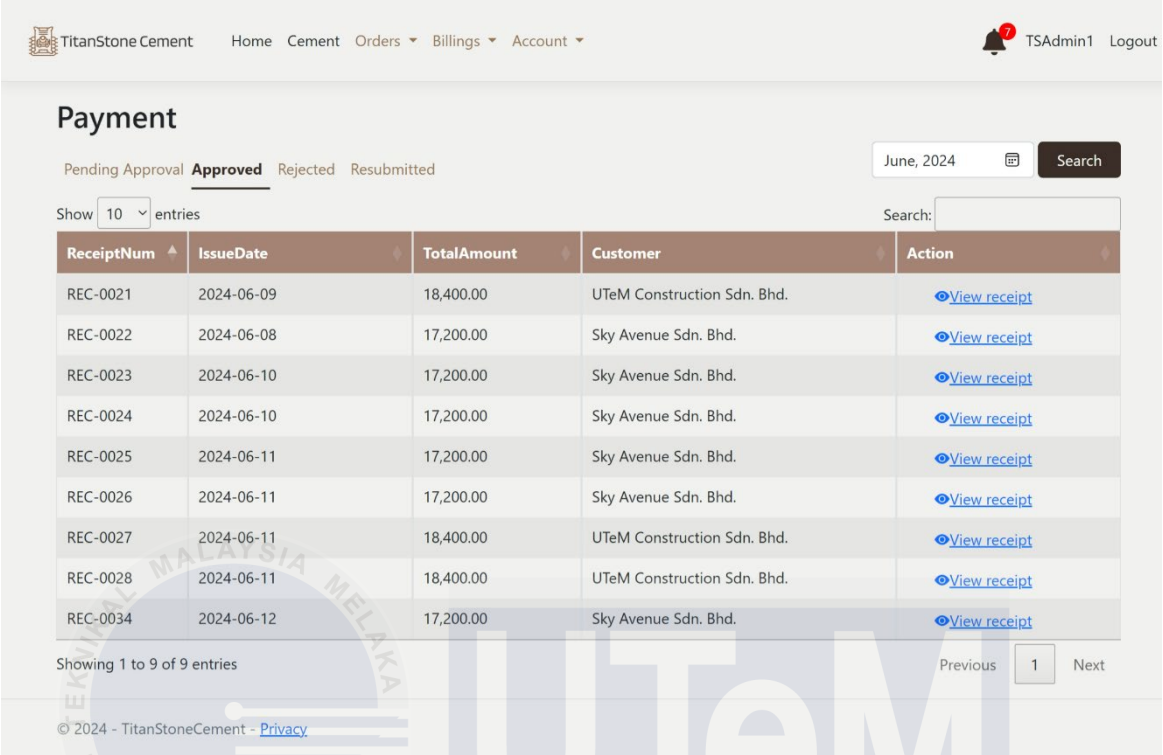


Figure 4.9: Admin Payment Management Interface

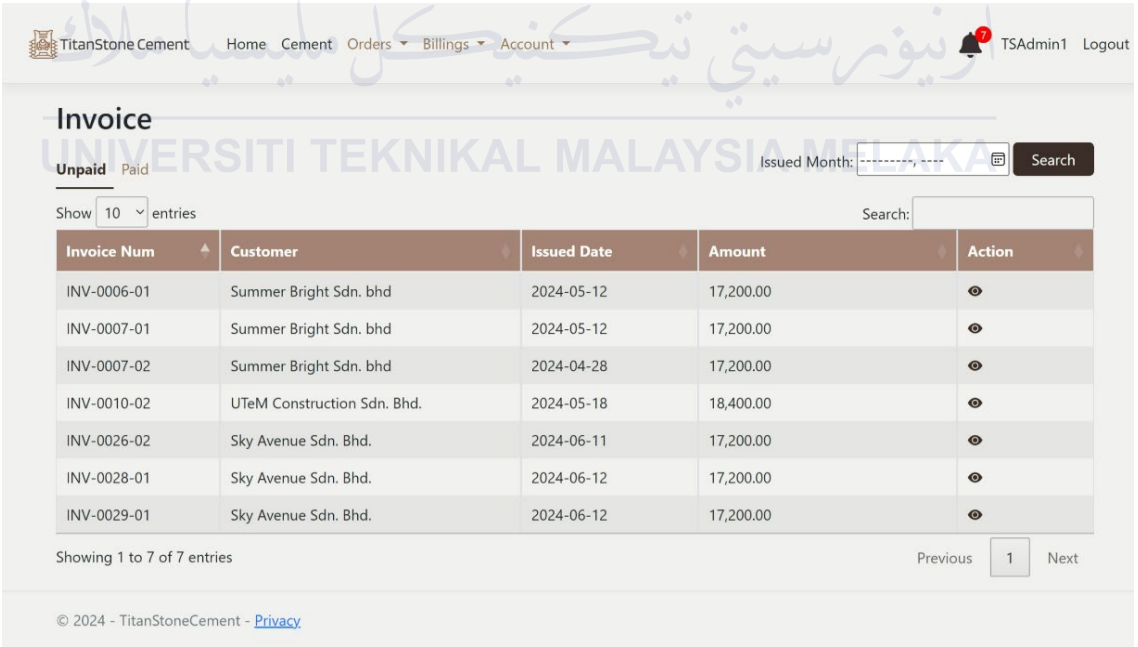


Figure 4.10: Admin Invoice Management Interface

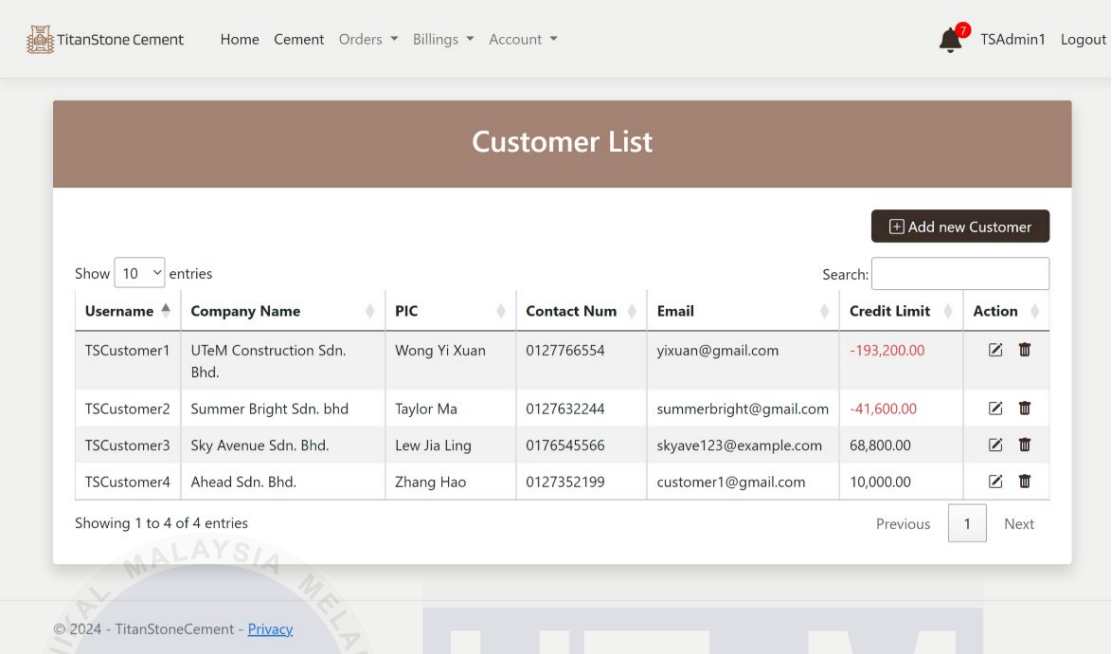


Figure 4.11: Admin Account Management Interface

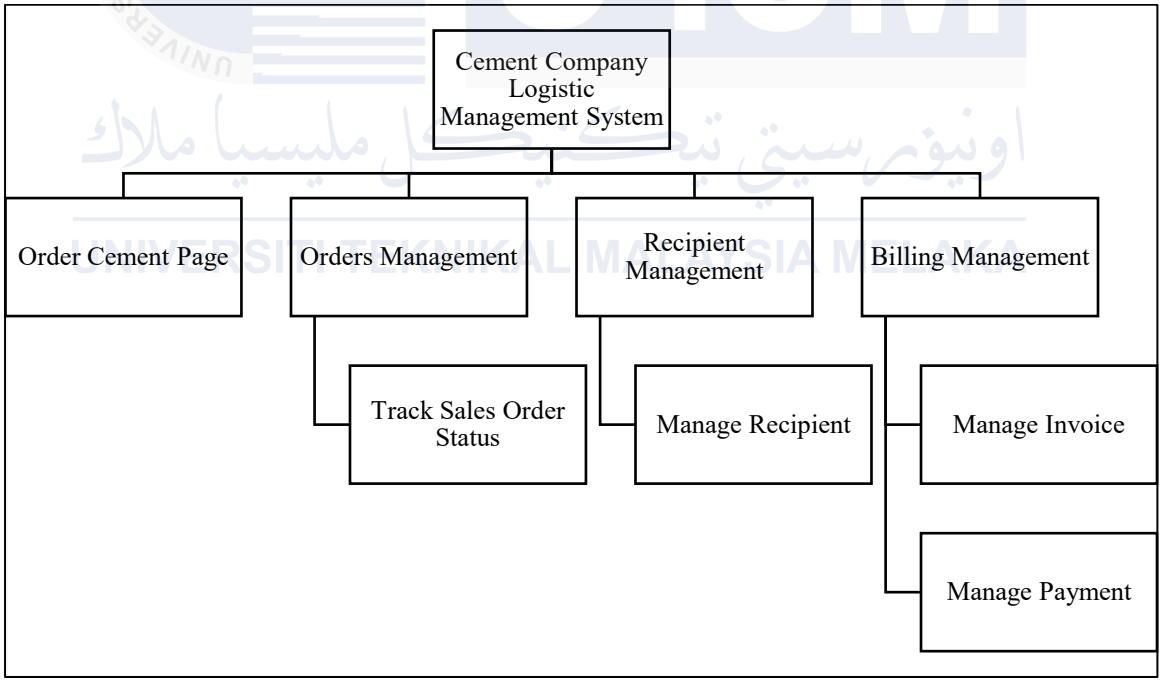


Figure 4.12: Navigation Design of Customer Interface

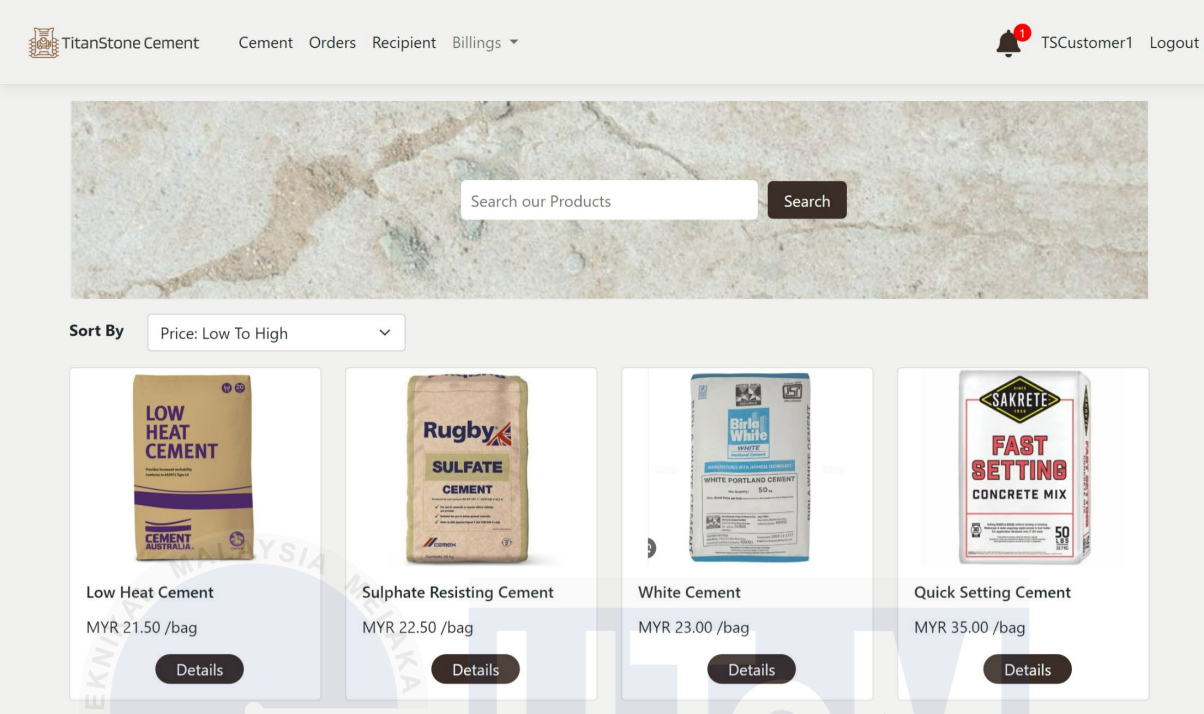


Figure 4.13: Customer Order Cement Page

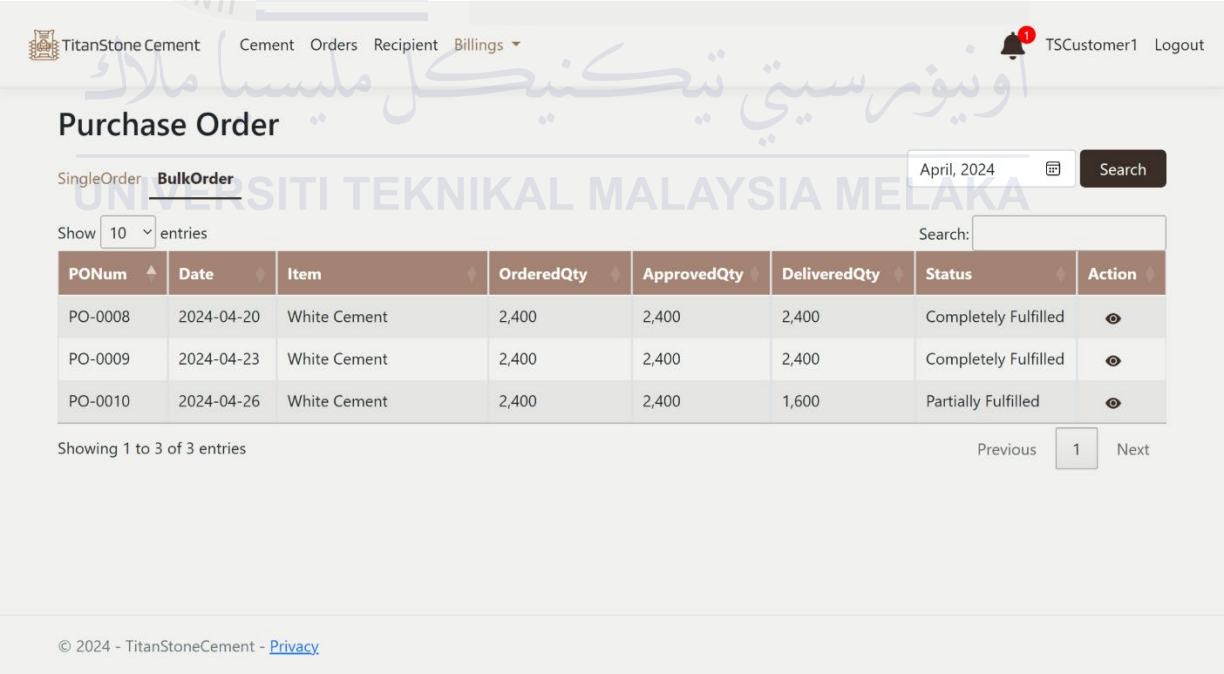


Figure 4.14: Customer Order Management Interface

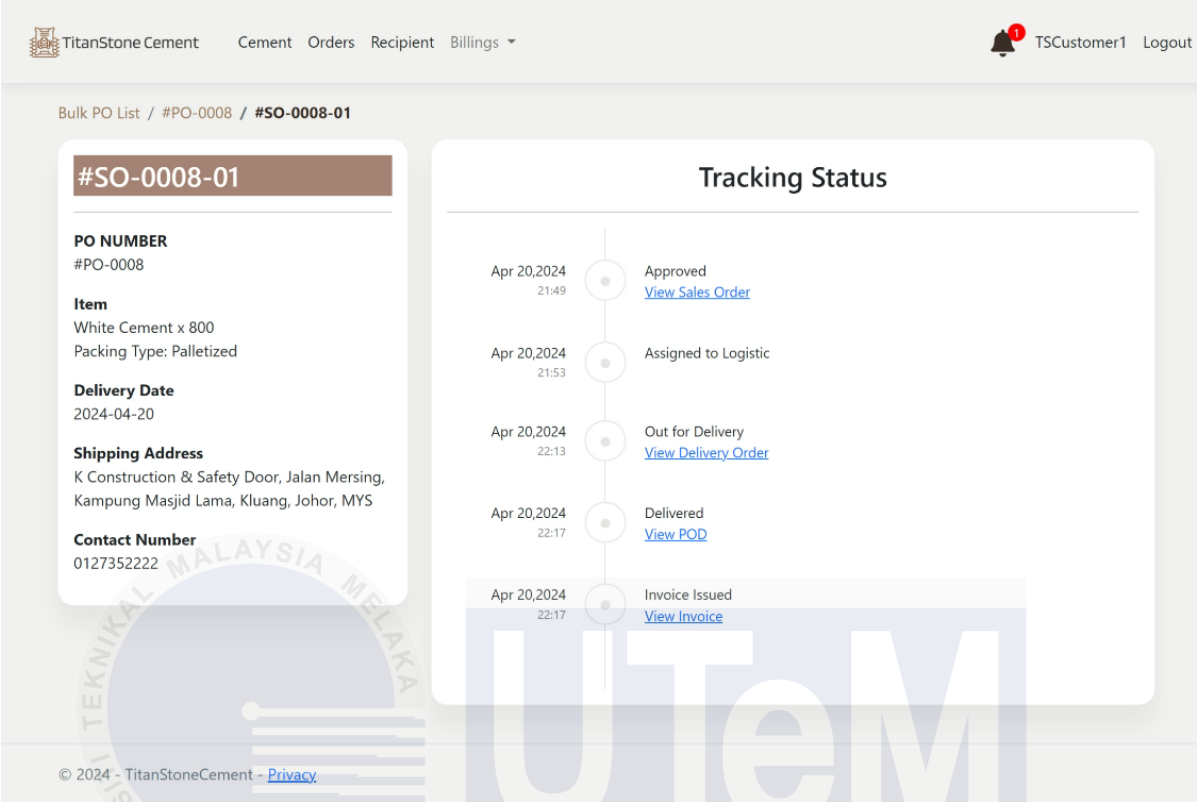


Figure 4.15: Customer Track Order Status Interface

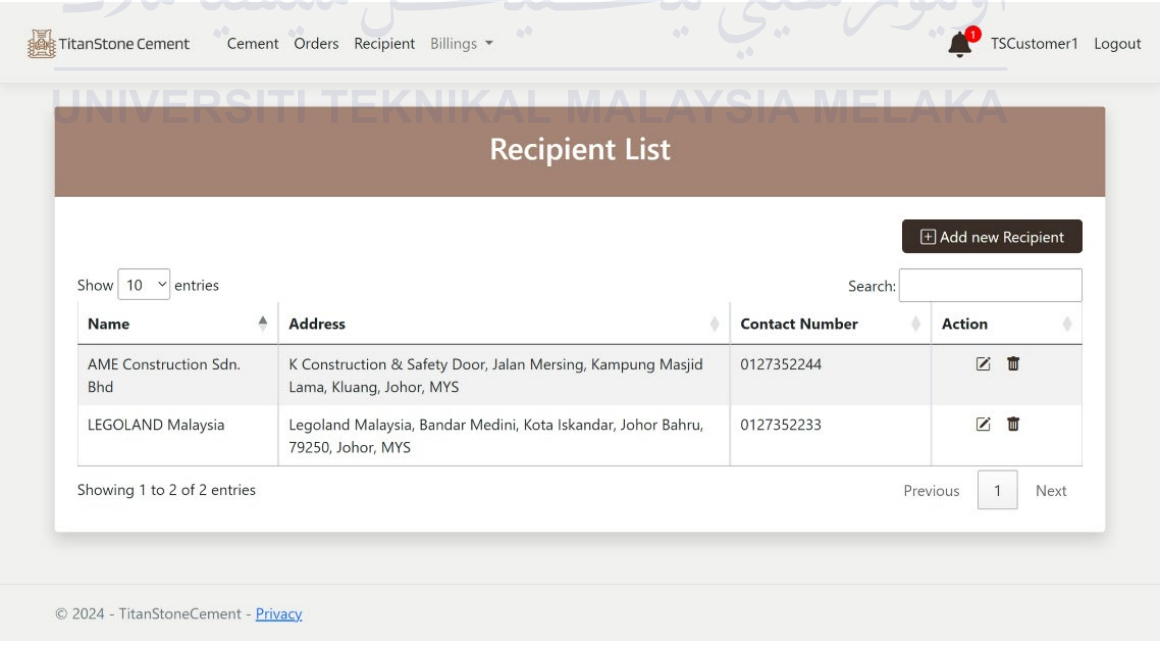


Figure 4.16 Customer Manage Recipient Interface

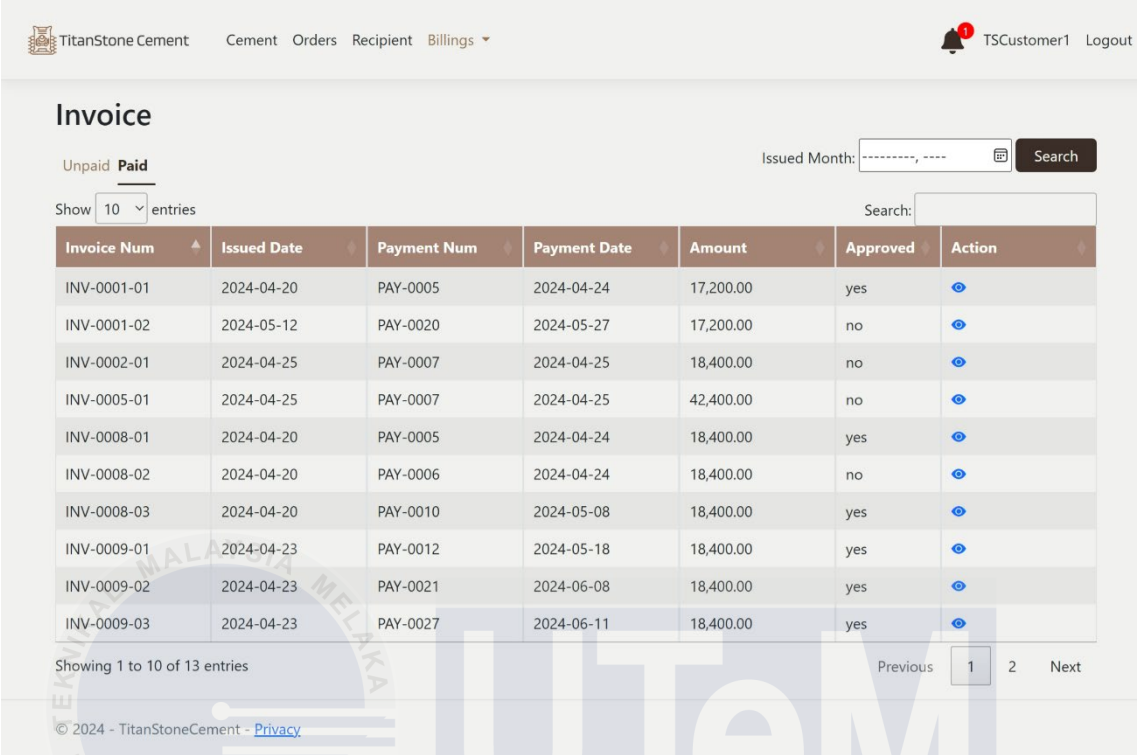


Figure 4.17: Customer Invoice Management Interface

Figure 4.18: Customer Manage Payment Interface

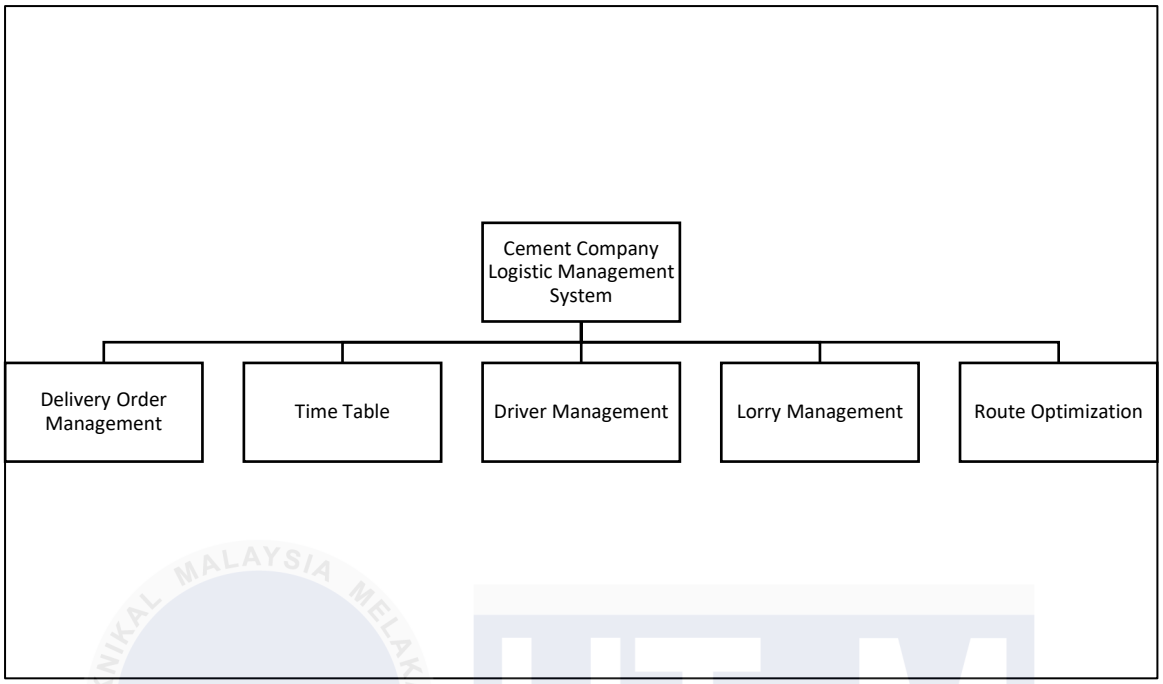


Figure 4.19: Navigation Design of Logistic

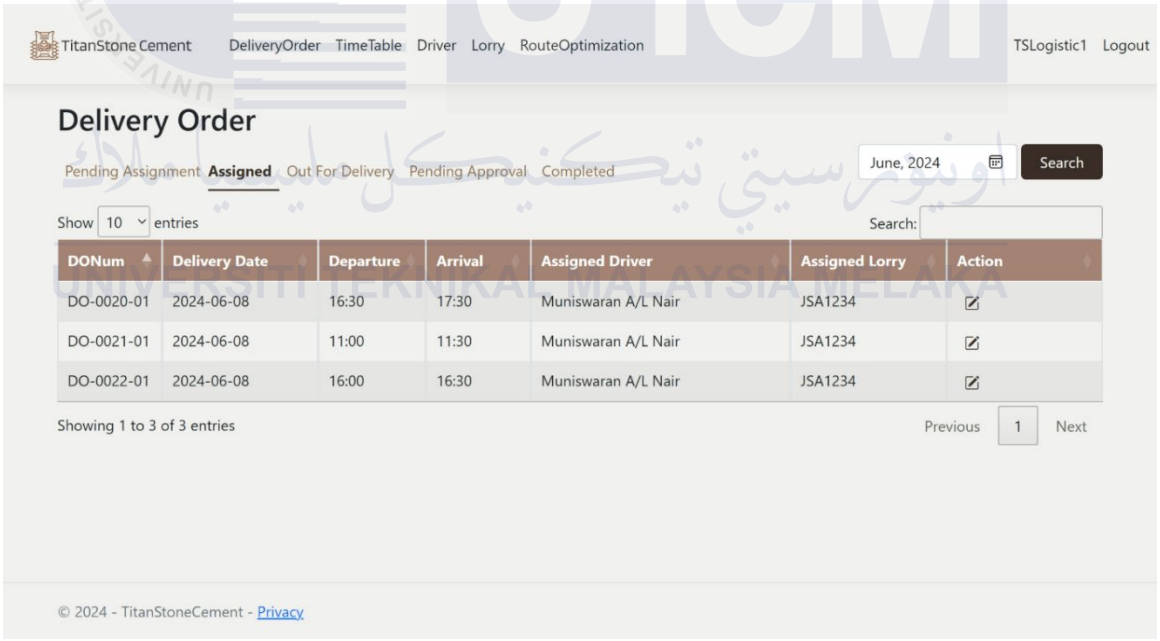


Figure 4.20: Logistic Delivery Order Management Interface

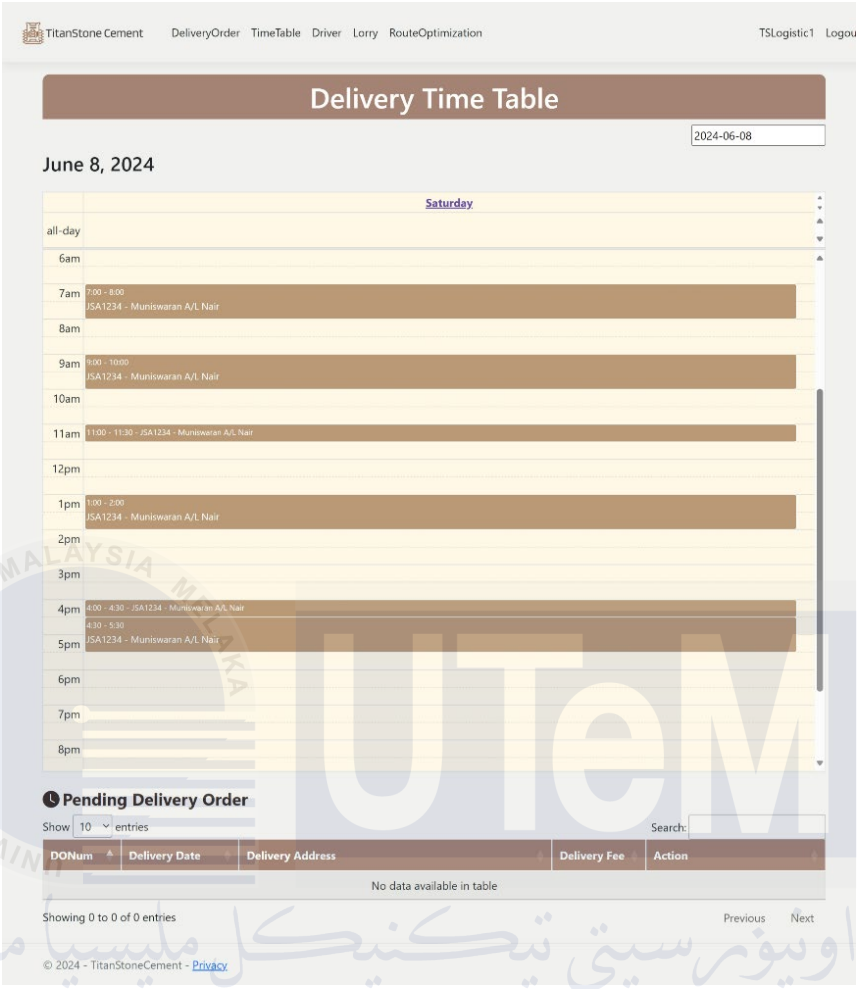


Figure 4.21: Logistic Time table interface

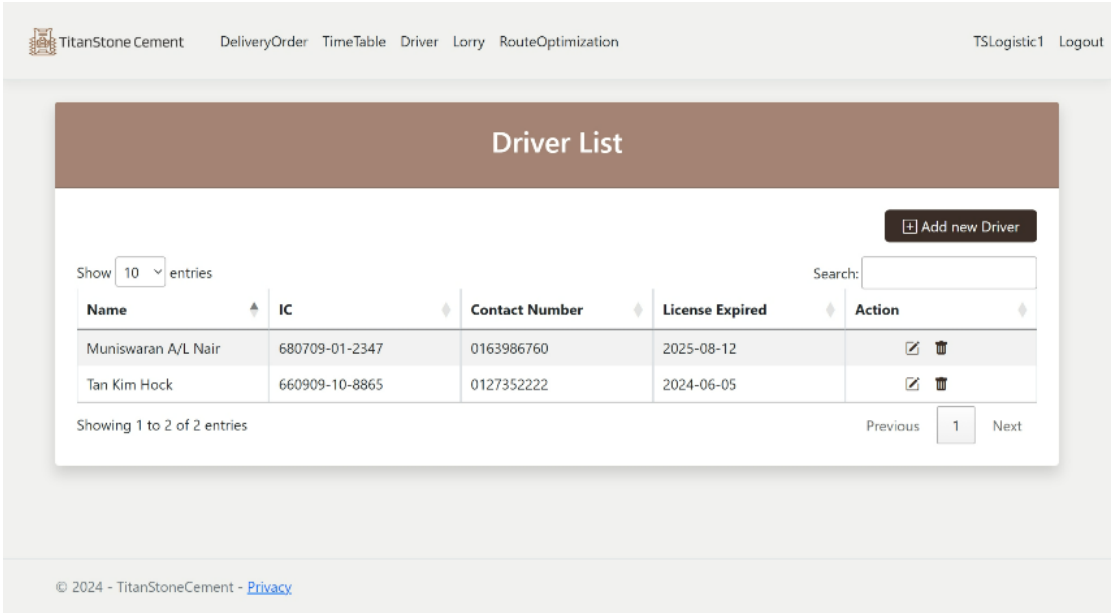


Figure 4.22: Logistic Driver Management Interface

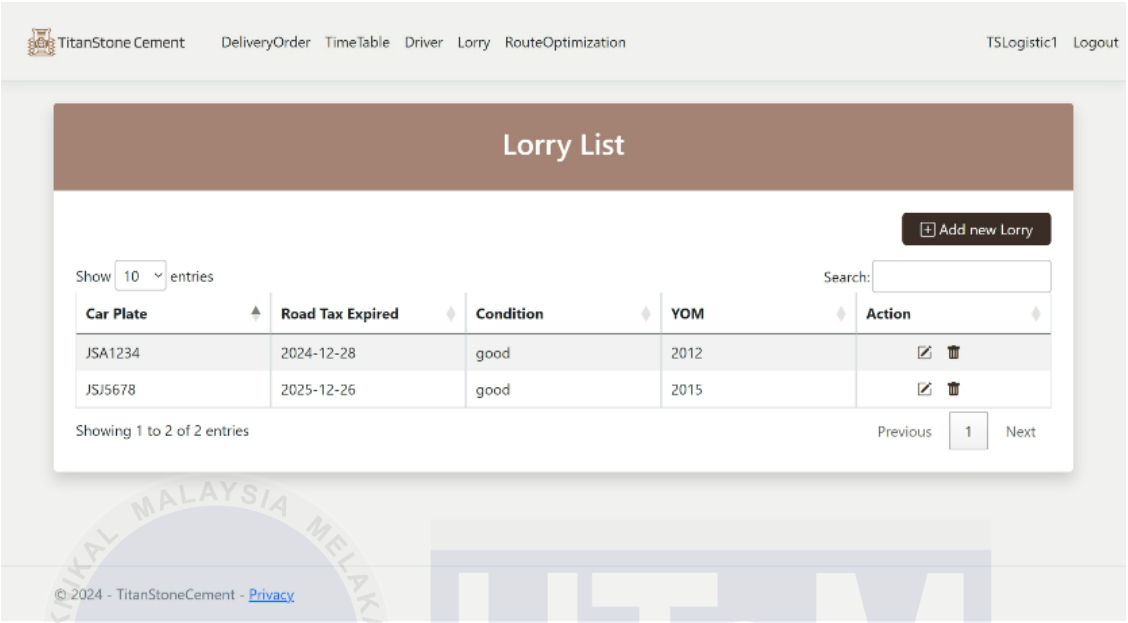


Figure 4.23: Logistic Lorry Management Interface

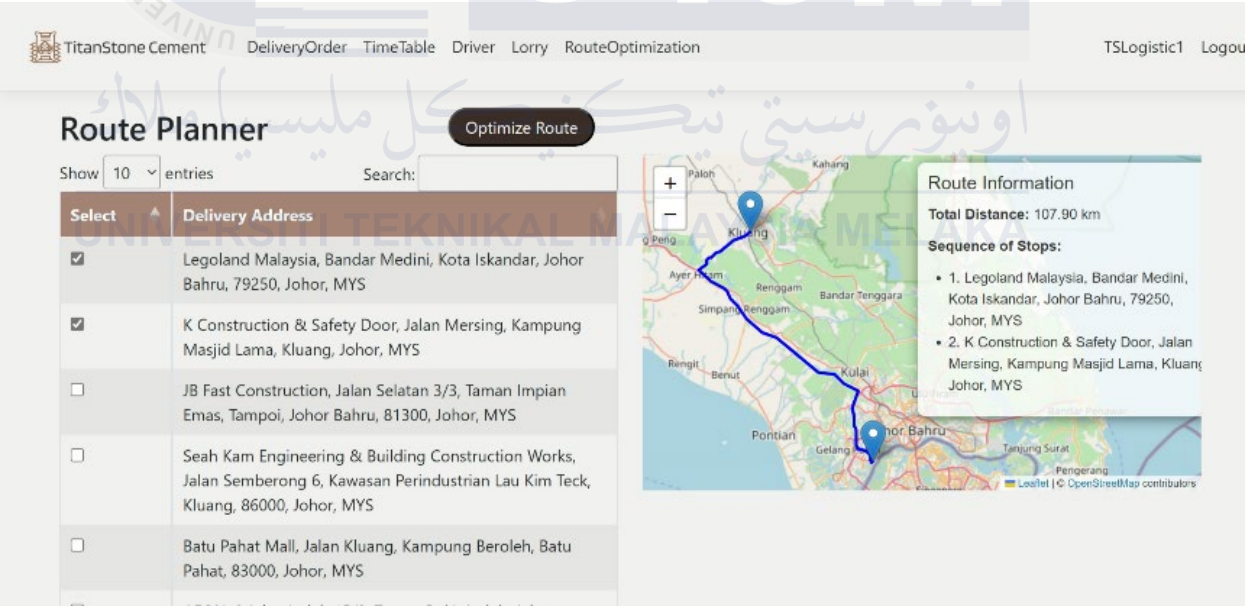


Figure 4.24: Logistic Route Planner Interface

4.5 Conclusion

In conclusion, this chapter has covered the project design, which encompasses the selection of the Database Management System (DBMS) for the physical design, the establishment of business rules for the conceptual design, the creation of the Entity Relationship Diagram, the development of the data dictionary, and the normalization process for the logical design, as well as the design of the Graphical User Interface (GUI). The outputs from this phase will serve as a foundation for the subsequent phases, including implementation, testing, and maintenance.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

In this chapter, the primary focus is on the successful implementation of the previously designed database. The procedures for database installation and configuration will be outlined. SQL Server is installed on Windows 11, and both Data Definition Language (DDL) and Data Manipulation Language (DML) will be implemented during this phase.

5.2 System Development Environment Setup

In CCLMS, the software development environment has to be set up before developing the website. The project consists of two main components which includes the ASP.Net MVC Framework with Microsoft Visual Studio 2022 as Integrated development Environment and Microsoft SQL Server with SQL Server Management Studio.

5.2.1 Step of Installation Setup

Step 1: Install the Microsoft SQL Server. Link: https://www.microsoft.com/en-my/sql-server/sql-server-downloads

Figure 5.1: Website to install Microsoft SQL Server

Step 2: Configure the storage location of log files, media files and resources files.

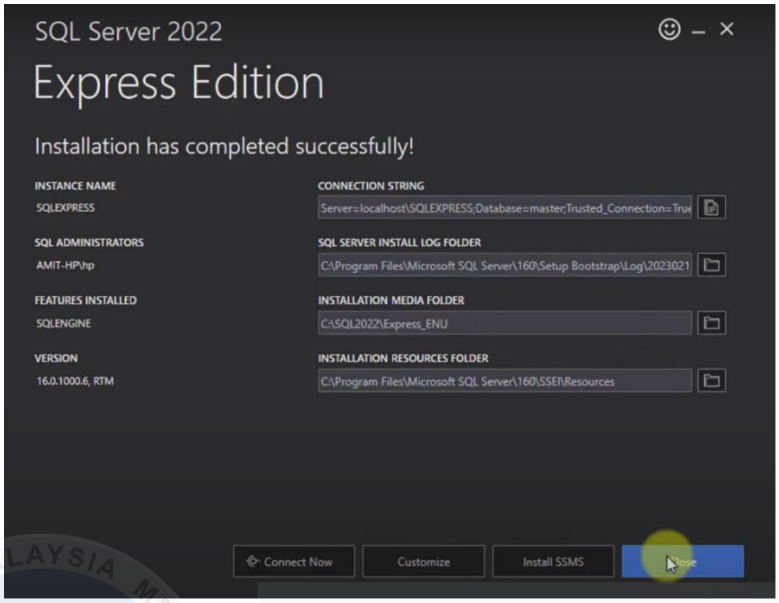


Figure 5.2: Configure the storage location of log files, media files and resources files.

Step 3: Install SQL Server Management Studio (SSMS). Link: https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

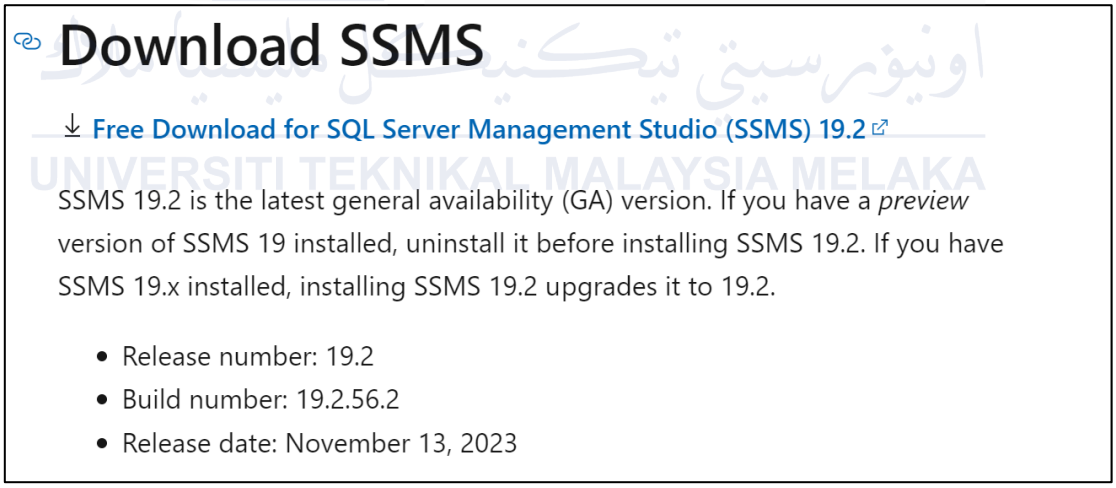


Figure 5.3: Website to install SQL Server Management Studio (SSMS)

Step 4: Run the SQL Server in local operating system.

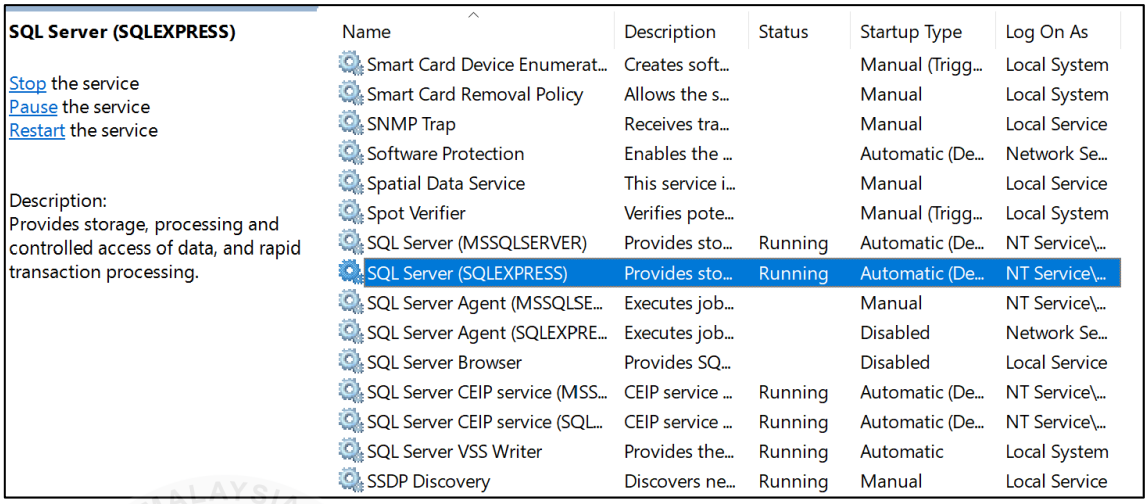


Figure 5.4: Services in local operating system

Step 5: Connect the local SQL Server to the SSMS.

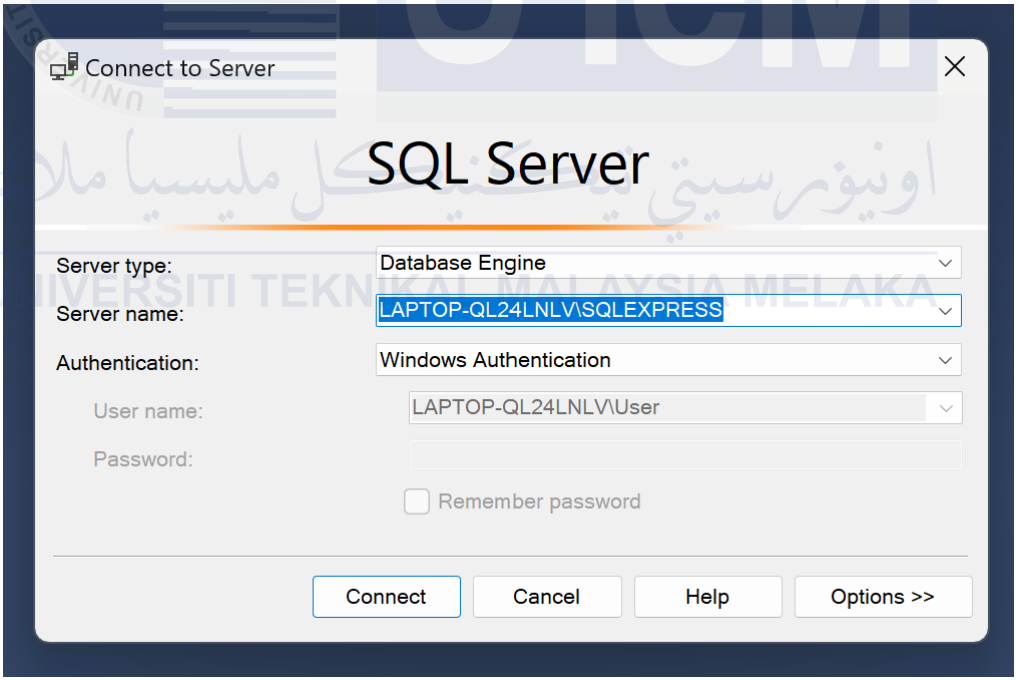


Figure 5.5: Connection of local SQL Server with SSMS

Step 6: Connection successful.

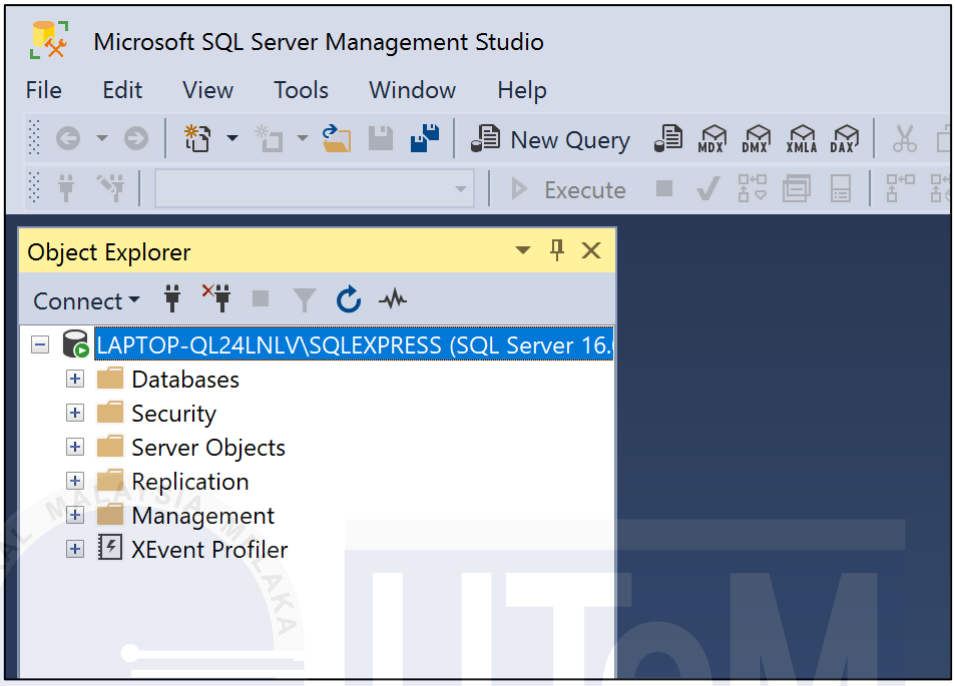


Figure 5.6: Connection of local SQL Server with SSMS is successful

5.3 Database Implementation

In the database implementation phase, the database is utilized to execute various queries, including simple queries, complex queries, aggregate functions, stored procedures, and triggers within the system. The purpose of these queries is to insert, retrieve, validate, and verify the information stored in the database.

5.3.1 Data Definition Language

Data Definition Language (DDL) refers to the SQL commands used to create and manipulate tables in a relational database. DDL statements can be employed to create, alter, and drop database objects, including tables, procedures, and triggers for the Cement Company Logistics Management System (CCLMS). Figures 5.7 to 5.15 display the DDL for each create table command.

```
CREATE TABLE [dbo].[Users](
	[username] [nvarchar](450) NOT NULL,
	[password] [nvarchar](30) NOT NULL,
	[companyName] [nvarchar](100) NULL,
	[SSN] [nvarchar](30) NULL,
	[PIC] [nvarchar](100) NOT NULL,
	[contactNum] [nvarchar](20) NOT NULL,
	[email] [nvarchar](80) NOT NULL,
	[IC] [nvarchar](14) NOT NULL,
	[role] [nvarchar](10) NOT NULL,
	[address] [nvarchar](200) NULL,
	[creditLimit] [float] NULL,
	[invoiceTerm] [int] NULL,
 CONSTRAINT [PK_Users] PRIMARY KEY CLUSTERED
(
	[username] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Figure 5.7: DDL of User Table

```
CREATE TABLE [dbo].[Cements](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[name] [nvarchar](50) NOT NULL,
	[description] [nvarchar](max) NOT NULL,
	[unitPrice] [float] NOT NULL,
	[color] [nvarchar](20) NOT NULL,
	[weight] [float] NOT NULL,
	[availability] [nvarchar](10) NOT NULL,
	[ImageUrl] [nvarchar](max) NULL,
	[PeriodEnd] [datetime2](7) GENERATED ALWAYS AS ROW END HIDDEN NOT NULL,
	[PeriodStart] [datetime2](7) GENERATED ALWAYS AS ROW START HIDDEN NOT NULL,
	[poQuantity] [int] NOT NULL,
	[stockQuantity] [int] NOT NULL,
 CONSTRAINT [PK_Cements] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY],
	PERIOD FOR SYSTEM_TIME ([PeriodStart], [PeriodEnd])
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
WITH
(
SYSTEM_VERSIONING = ON (HISTORY_TABLE = [dbo].[CementsHistory])
)
GO

ALTER TABLE [dbo].[Cements] ADD  DEFAULT ((0)) FOR [poQuantity]
GO

ALTER TABLE [dbo].[Cements] ADD  DEFAULT ((0)) FOR [stockQuantity]
GO
```

Figure 5.8: DDL of Cement Table

```
CREATE TABLE [dbo].[Recipients](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[name] [nvarchar](100) NOT NULL,
	[address] [nvarchar](250) NOT NULL,
	[contactNum] [nvarchar](20) NOT NULL,
	[customerID] [nvarchar](450) NOT NULL,
	[latitude] [float] NULL,
	[longitude] [float] NULL,
 CONSTRAINT [PK_Recipients] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Recipients]  WITH CHECK ADD  CONSTRAINT
[FK_Recipients_Users_customerID] FOREIGN KEY([customerID])
REFERENCES [dbo].[Users] ([username])
GO

ALTER TABLE [dbo].[Recipients] CHECK CONSTRAINT [FK_Recipients_Users_customerID]
GO
```

Figure 5.9: DDL of Recipient Table

```
CREATE TABLE [dbo].[PurchaseOrders](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[poNum] [nvarchar](16) NOT NULL,
	[quantity] [int] NOT NULL,
	[soldUnitPrice] [float] NOT NULL,
	[totalAmount] [float] NOT NULL,
	[orderDate] [datetime2](7) NOT NULL,
	[packingType] [nvarchar](20) NOT NULL,
	[remarks] [nvarchar](max) NULL,
	[orderType] [nvarchar](20) NOT NULL,
	[status] [nvarchar](30) NOT NULL,
	[cementID] [int] NOT NULL,
	[recipientID] [int] NOT NULL,
	[customerID] [nvarchar](450) NOT NULL,
	[fulfilledQty] [int] NOT NULL,
	[approvedQty] [int] NOT NULL,
 CONSTRAINT [PK_PurchaseOrders] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[PurchaseOrders] ADD  DEFAULT ((0)) FOR [fulfilledQty]
GO

ALTER TABLE [dbo].[PurchaseOrders] ADD  DEFAULT ((0)) FOR [approvedQty]
GO

ALTER TABLE [dbo].[PurchaseOrders]  WITH CHECK ADD  CONSTRAINT
[FK_PurchaseOrders_Cements_cementID] FOREIGN KEY([cementID])
REFERENCES [dbo].[Cements] ([Id])
GO

ALTER TABLE [dbo].[PurchaseOrders] CHECK CONSTRAINT
[FK_PurchaseOrders_Cements_cementID]
GO

ALTER TABLE [dbo].[PurchaseOrders]  WITH CHECK ADD  CONSTRAINT
[FK_PurchaseOrders_Recipients_recipientID] FOREIGN KEY([recipientID])
```

Figure 5.10: DDL of PurchaseOrder Table

```
CREATE TABLE [dbo].[Lorries](
	[carPlateNum] [nvarchar](450) NOT NULL,
	[roadTaxExprDate] [datetime2](7) NOT NULL,
	[condition] [nvarchar](30) NULL,
	[YOM] [int] NOT NULL,
	[logisticID] [nvarchar](450) NOT NULL,
 CONSTRAINT [PK_Lorries] PRIMARY KEY CLUSTERED
(
	[carPlateNum] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Lorries]  WITH CHECK ADD  CONSTRAINT
[FK_Lorries_Users_logisticID] FOREIGN KEY([logisticID])
REFERENCES [dbo].[Users] ([username])
GO

ALTER TABLE [dbo].[Lorries] CHECK CONSTRAINT [FK_Lorries_Users_logisticID]
GO
```

Figure 5.11: DDL of Lorry Table

```
CREATE TABLE [dbo].[Drivers](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[name] [nvarchar](100) NOT NULL,
	[IC] [nvarchar](14) NOT NULL,
	[contactNum] [nvarchar](20) NOT NULL,
	[licenseExprDate] [datetime2](7) NOT NULL,
	[logisticID] [nvarchar](450) NOT NULL,
 CONSTRAINT [PK_Drivers] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Drivers]  WITH CHECK ADD  CONSTRAINT
[FK_Drivers_Users_logisticID] FOREIGN KEY([logisticID])
REFERENCES [dbo].[Users] ([username])
GO
```

Figure 5.12: DDL of Driver Table

```
CREATE TABLE [dbo].[SalesOrders](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[soNum] [nvarchar](18) NOT NULL,
	[quantity] [int] NOT NULL,
	[totalAmount] [float] NOT NULL,
	[status] [nvarchar](30) NOT NULL,
	[deliveryDate] [datetime2](7) NOT NULL,
	[poID] [int] NOT NULL,
	[PeriodEnd] [datetime2](7) GENERATED ALWAYS AS ROW END HIDDEN NOT NULL,
	[PeriodStart] [datetime2](7) GENERATED ALWAYS AS ROW START HIDDEN NOT
NULL,
 CONSTRAINT [PK_SalesOrders] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY],
	PERIOD FOR SYSTEM_TIME ([PeriodStart], [PeriodEnd])
) ON [PRIMARY]
WITH
(
SYSTEM_VERSIONING = ON (HISTORY_TABLE = [dbo].[SalesOrdersHistory])
)
GO

ALTER TABLE [dbo].[SalesOrders]  WITH CHECK ADD  CONSTRAINT
[FK_SalesOrders_PurchaseOrders_poID] FOREIGN KEY([poID])
REFERENCES [dbo].[PurchaseOrders] ([Id])
GO

ALTER TABLE [dbo].[SalesOrders] CHECK CONSTRAINT
[FK_SalesOrders_PurchaseOrders_poID]
GO
```

Figure 5.13: DDL of SalesOrder Table

```
CREATE TABLE [dbo].[DeliveryOrders](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[doNum] [nvarchar](18) NOT NULL,
	[deliveryFee] [float] NOT NULL,
	[estimatedDeparture] [datetime2](7) NULL,
	[estimatedArrival] [datetime2](7) NULL,
	[soID] [int] NOT NULL,
	[logisticID] [nvarchar](450) NOT NULL,
	[lorryID] [nvarchar](450) NULL,
	[driverID] [int] NULL,
	[approved] [nvarchar](8) NULL,
	[podURL] [nvarchar](max) NULL,
	[accepted] [nvarchar](8) NULL,
 CONSTRAINT [PK_DeliveryOrders] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[DeliveryOrders]  WITH CHECK ADD  CONSTRAINT
[FK_DeliveryOrders_Drivers_driverID] FOREIGN KEY([driverID])
REFERENCES [dbo].[Drivers] ([Id])
GO

ALTER TABLE [dbo].[DeliveryOrders] CHECK CONSTRAINT
[FK_DeliveryOrders_Drivers_driverID]
GO

ALTER TABLE [dbo].[DeliveryOrders]  WITH CHECK ADD  CONSTRAINT
[FK_DeliveryOrders_Lorries_lorryID] FOREIGN KEY([lorryID])
REFERENCES [dbo].[Lorries] ([carPlateNum])
GO

ALTER TABLE [dbo].[DeliveryOrders] CHECK CONSTRAINT
[FK_DeliveryOrders_Lorries_lorryID]
GO

ALTER TABLE [dbo].[DeliveryOrders]  WITH CHECK ADD  CONSTRAINT
[FK_DeliveryOrders_SalesOrders_soID] FOREIGN KEY([soID])
REFERENCES [dbo].[SalesOrders] ([Id])
GO

ALTER TABLE [dbo].[DeliveryOrders] CHECK CONSTRAINT
[FK_DeliveryOrders_SalesOrders_soID]
GO

ALTER TABLE [dbo].[DeliveryOrders]  WITH CHECK ADD  CONSTRAINT
[FK_DeliveryOrders_Users_logisticID] FOREIGN KEY([logisticID])
REFERENCES [dbo].[Users] ([username])
GO

ALTER TABLE [dbo].[DeliveryOrders] CHECK CONSTRAINT
[FK_DeliveryOrders_Users_logisticID]
GO
```

Figure 5.13: DDL of DeliveryOrder Table

```
CREATE TABLE [dbo].[Invoices](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[invNum] [nvarchar](18) NOT NULL,
	[term] [int] NOT NULL,
	[date] [datetime2](7) NOT NULL,
	[doID] [int] NOT NULL,
	[paymentID] [int] NULL,
 CONSTRAINT [PK_Invoices] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Invoices]  WITH CHECK ADD  CONSTRAINT
[FK_Invoices_DeliveryOrders_doID] FOREIGN KEY([doID])
REFERENCES [dbo].[DeliveryOrders] ([Id])
GO

ALTER TABLE [dbo].[Invoices] CHECK CONSTRAINT [FK_Invoices_DeliveryOrders_doID]
GO

ALTER TABLE [dbo].[Invoices]  WITH CHECK ADD  CONSTRAINT
[FK_Invoices_Payments_paymentID] FOREIGN KEY([paymentID])
REFERENCES [dbo].[Payments] ([Id])
ON DELETE SET NULL
GO

ALTER TABLE [dbo].[Invoices] CHECK CONSTRAINT [FK_Invoices_Payments_paymentID]
GO
```

Figure 5.14: Invoice Table

```
CREATE TABLE [dbo].[Payments](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[date] [datetime2](7) NOT NULL,
	[totalAmount] [float] NOT NULL,
	[receiptNum] [nvarchar](16) NULL,
	[approved] [nvarchar](8) NULL,
	[receiptURL] [nvarchar](max) NULL,
	[paymentMethod] [nvarchar](40) NULL,
	[proofAmount] [nvarchar](40) NULL,
	[proofDate] [nvarchar](40) NULL,
	[proofTime] [nvarchar](40) NULL,
	[approvedDate] [datetime2](7) NULL,
	[rejectReason] [nvarchar](200) NULL,
	[rejectionCount] [int] NOT NULL,
	[paymentNum] [nvarchar](16) NULL,
	[resubmitCount] [int] NOT NULL,
 CONSTRAINT [PK_Payments] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[Payments] ADD  DEFAULT ((0)) FOR [rejectionCount]
GO

ALTER TABLE [dbo].[Payments] ADD  DEFAULT ((0)) FOR [resubmitCount]
GO
```

Figure 5.15: Data Dictionary of Payment Table

```
CREATE TABLE [dbo].[Resubmits](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[reason] [nvarchar](200) NULL,
	[date] [datetime2](7) NOT NULL,
	[paymentID] [int] NOT NULL,
	[receiptURL] [nvarchar](max) NOT NULL,
 CONSTRAINT [PK_Resubmits] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[Resubmits] ADD  DEFAULT (N'') FOR [receiptURL]
GO

ALTER TABLE [dbo].[Resubmits]  WITH CHECK ADD  CONSTRAINT
[FK_Resubmits_Payments_paymentID] FOREIGN KEY([paymentID])
REFERENCES [dbo].[Payments] ([Id])
GO

ALTER TABLE [dbo].[Resubmits] CHECK CONSTRAINT [FK_Resubmits_Payments_paymentID]
GO
```

Figure 5.16: Data Dictionary of Resubmit Table

5.3.2 Data Loading

Data loading is the phase where data is loaded into the database to be used in the future. The data is loaded using Data Manipulation Language (DML), for example INSERT statement. After the user interface is developed, the users can now use the system to insert new data into the database since the system is now connected to the database. Figure 5.16 shows the query to insert new recipient into the database. Table 5.1 shows the sample data of recipient table.

```
INSERT INTO Recipients (CustomerID, name, address, contactNum)
VALUES (@CustomerID, @Name, @Address, @contactNum)
```

Figure 5.17: Insert Statement of Recipient Table

Table 5.1: Sample data of recipient table

ID	customerID	name	address	contactNum
1	1	Future Sdn. Bhd.	No 12, Jalan Kluang Baru 1, Taman Kluang Baru, 86000 Kluang.	0127685566

5.3.3 Data Access

Users can access the data stored in the database using queries such as restricted data queries, join data queries and aggregate function queries. These queries will be discussed later in this section.

5.3.3.1 Restricted Data Queries

Figure 5.17 shows query retrieves all records from the Drivers table where the logisticID matches the specified username.

```
SELECT *
FROM Drivers
WHERE logisticID = 'username';
```

Figure 5.18: Example of restricted select statement

5.3.3.2 Join Data Queries

Figure 5.18 shows example of quey that join four tables: PurchaseOrder, Cement, Customer, Recipient.

```
SELECT *
FROM PurchaseOrders p
LEFT JOIN Cements c ON p.CementID = c.ID
LEFT JOIN Customers cu ON p.CustomerID = cu.ID
LEFT JOIN Recipients r ON p.RecipientID = r.ID
WHERE p.Id = @salesOrder_poID;
```

Figure 5.19: Example of join statement

5.3.3.3 Aggregate Function Queries

Figure 5.19 shows the query counts the number of records in the DeliveryOrders table where the driverID column matches the specified id. The COUNT(*) function is used to return the total number of such records.

```
SELECT COUNT(*)
FROM DeliveryOrders
WHERE driverID = @id;
```

Figure 5.20: Example of aggregate function

5.4 Implementation Status

The implementation status provides an overview of each module, including its description, the duration required to complete it, and the completion date throughout the implementation process.

Table 5.2: Implementation Status

Module	Description	Duration (days)	Completed Date
Customer management	This module allows the admin to manage customer information, including creating and updating customer accounts. It also includes setting customer credit limits.	10/4/2024	12/4/2024

Logistic Company Management	In this module, the admin can manage the details of partnered logistic companies. This includes creating and maintaining logistic company profiles, assigning delivery orders.	13/4/2024	15/4/2024
Order Management	This module handles the entire process of managing orders, from submission to approval. Admins can view and approve orders, update order statuses, and manage related documents such as invoices and receipts.	16/4/2024	24/4/2024
Cement Management	The Cement Management module is responsible for managing the inventory of cement products. Admins can add new cement types, update pricing, track stock levels, and monitor price changes.	25/4/2024	27/4/2024
Submit Order & Recipient	This module allows customers to place orders for cement and specify the recipient details.	28/4/2024	30/4/2024
Submit payment evidence	Customers can use this module to submit evidence of payment, such as bank slips or cheque. This evidence is then verified by the admin before processing the order further.	1/5/2024	10/5/2024
Order status tracking	This module provides customers with real-time updates on the status of their orders.	11/5/2024	25/5/2024
View invoice and receipt	Customers can use this module to view and download invoices and receipts for their completed transactions.	26/5/2024	31/5/2024
View delivery order	This module allows logistic companies to view, reject all delivery orders assigned to them. It provides details such as the destination, required delivery date, and the specific cement products to be delivered.	1/6/2024	5/6/2024
Driver lorry and assignment	Logistic companies use this module to assign or edit drivers and lorries to specific delivery orders.	6/6/2024	8/6/2024

5.5 Conclusion

In summary, this chapter outlines the setup of the development environment for the Cement Company Logistics Management System (CCLMS), detailing the procedures for installing Microsoft SQL Server on a Windows platform. Additionally, the chapter covers the execution of database implementation to facilitate the system's processes. The system is designed based on business logic that incorporates Data Definition Language (DDL), Data Manipulation Language (DML), as well as commands for creating database tables and constraints.

CHAPTER 6

TESTING

6.1 Introduction

Testing is a process aimed at evaluating the functionality of a software application to determine whether the developed software meets the specified requirements and to identify any defects, ensuring that the system is free of errors. In this chapter, verification and validation will be performed on the Cement Company Logistics Management System (CCLMS). The two main goals of testing CCLMS are:

a) To demonstrate CCLMS to the end users has meet its user requirement

b) To discover any bug or faults from the CCLMS with difference test strategy

Additionally, system testing is a crucial phase in the Database Life Cycle (DBLC). The testing phase for the Cement Company Logistics Management System (CCLMS) includes a test plan that encompasses the test organization, test schedule, and test environment.

6.2 Test Plan

A test plan is a technical document that outlines the testing strategy, scope, resources (including manpower, software, and hardware) needed for testing, the test schedule, and the expected deliverables. It provides a comprehensive understanding of the system's workflow and functions, detailing how each component will be tested to assess whether the system operates as designed, identify bugs, and determine its actual limitations.

6.2.1 Test Organization

In the Cement Company Logistics Management System (CCLMS), the test organization consists of three user roles: admin, customer, and logistic. Both functional and non-functional requirements will be tested for each user role. Table 6.1 illustrates how the three users undergo testing based on their specific responsibilities.

Table 6.1: User Responsibilities List

Tester ID	User	Responsibilities
T1	Admin	• Testing the system with follow the test script given • Testing the admin module • Defect and bug detection

T2	Customer	• Testing the system with follow the test script given • Testing the customer module • Defect and bug detection
T3	Logistic	• Testing the system with follow the test script given • Testing the logistic module • Defect and bug detection

6.2.2 Test Environment

The test environment refers to the configuration of software, hardware, operating systems, tools, and network settings necessary for the testing teams to execute test cases. Tables 6.2 and 6.3 present the list of hardware and software required for the testing environment of the Cement Company Logistics Management System (CCLMS).

Table 6.2: Test Environment Hardware List

Environment Specification	Description
Laptop	Lenovo Yogo Slim 5i
Processor	Intel Cire i7
RAM	16GB
Mouse	Dell

Table 6.3: Test Environment Software List

Environment	Description
Database	Microsoft SQL Server
Operating System	Window 11
Web Browser	Google Chrome
Microsoft Visual Studio 2022	Integrated Development Environment

6.2.3 Test Schedule

The test schedule is defined as a summary that outlines key test milestones and timelines for testing activities. It organizes test activities based on specific dates for the completion of the testing process. Preparing the test schedule involves estimating dates and making necessary revisions. Table 6.4 illustrates the test schedule for the development of the Cement Company Logistics Management System (CCLMS).

Table 6.4 Test Schedule

Testing Task	Test Date
User Login	1/8/2024
Insert Cement	1/8/2024
Register Customer Account	2/8/2024
Add Recipient	2/8/2024

Add Lorry	3/8/2024
Add Driver	3/8/2024
Product Searching	4/8/2024
Credit Limit Checking	4/8/2024
Upload Proof of Payment	5/8/2024
Assign Delivery Order	6/8/2024
Place Purchase Order	7/8/2024
Assign Sales Order	8/8/2024
Request Sales Order	8/8/2024

6.3 Test Strategy

Black box testing involves evaluating the software without knowledge of its internal workings. In this approach, the tester provides inputs and observes the outputs generated by the system, enabling the identification of how the system responds to both expected and unexpected user actions, as well as assessing its response time, usability, and reliability. This testing can be performed by any tester, regardless of their programming or software development background. The primary goal of black box testing is to validate the system and ensure it meets both functional and non-functional requirements.

White box testing, in contrast, focuses on examining the internal logic and structure of the code. This technique requires detailed knowledge of the application's source code, architecture, and configuration. White box testing can reveal issues such as security vulnerabilities, broken paths, or data flow problems—areas that black box testing may not comprehensively cover. Therefore, it is essential for testers conducting white box testing to possess programming skills or knowledge.

By combining both techniques, testers and developers can ensure that the system meets all objectives and requirements, while also enhancing the overall quality of the system through thorough testing.

6.3.1 Classes of Tests

a) Unit Test

A test that focuses on verifying the functionality of individual components or pieces of code in isolation, ensuring that each unit works correctly on its own.

b) Integration Test

Integration test is to ensure that this system captures data into the database correctly based on what key in the data. A test that examines how different units or components

of a system work together, ensuring that the integrated modules function as expected when combined.

c) Usability Test

A test that evaluates the user experience and ease of use of a system or application, focusing on how real users interact with the interface and whether it meets their needs effectively.

6.4 Test Design

6.4.1 Test Description

The test description outlines the identification of test cases, types of testing, preconditions, test requirements, procedural steps for each test case, and the expected output results, which are documented for every module's test case. Tables 6.6 to 6.17 provide detailed descriptions of the test cases in relation to the various system modules.

Table 6.5: Test Description of User Login

Test ID	T001-Login			
Testing Type	Unit testing			
Test Strategy	White box Testing			
Test Class	Security and error handling testing			
Test Case ID	**Test Requirements**	**Pre-condition**	**Test/Step Procedure**	**Expected Output**
TC1_1	Validate that the login function is accessible when valid username and password are provided.	User has valid username and password.	1. Navigate to login page. 2. Provide valid username. 3. Provide valid password. 4. Click on Login button.	Login Successful.
TC1_2	Validate that the login function is not accessible when the username or password fields are left blank.		1. Navigate to login page. 2. Click on Login button.	Login failed. Display error message “Invalid username or password”.
TC1_3	Validate that the login function is not accessible when the username and password		1. Navigate to login page. 2. Provide valid username. 3. Provide invalid password.	Login failed. Display error message “Invalid username or password”.

	provided are invalid.		4. Click on Login button.	

Table 6.6: Test Description of Insert Cement

Test ID	T002-Insert Cement		
Testing Type	Unit testing		
Test Strategy	White box Testing		
Pre-condition	User must log in as admin		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC2_1	Validate the insert cement function is accessible when all the input data are valid.	1. Navigate to insert cement page. 2. Enter correct data type to input text field “product name”, “price per bag”, “colour”, “weight” and “description”. 3. Upload an image for product image. 3. Click on create button.	Insert cement successful.
TC2_2	Validate the insert cement function is not accessible when all the input fields are left blank.	1. Navigate to insert cement page. 2. Click on create button.	Insert cement failed. Display error message “The name filed is required” at the “Product name” input text field.
TC2_3	Validate the insert cement function is not accessible when the certain input data type are invalid.	1. Navigate to insert cement page. 2. Enter correct data type to input text field “product name”, “colour”, “weight” and “description”. 3. Enter invalid “price per bag” range. 4. Click on create button.	Insert cement failed. Display error message “Please enter a value greater than or equal to 1.” at the “Price per Bag” input text field.

Table 6.7: Test Description of Register Customer Account

Test ID	T003-Register Customer Account		
Testing Type	Unit testing		
Test Strategy	White box Testing		
Pre-condition	User must log in as admin		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**

TC3_1	Validate the register customer account function is accessible when all the input data are valid.	1. Navigate to add new customer page. 2. Enter correct data type to input text field “username”, “temporary password”, “company name”, “SSM”, “address”, “Person in Charge”, “IC”, “contact number”, “email”, “credit limit” and “invoice term”. 3. Click on create button.	Customer account register successful.
TC3_2	Validate the register customer account is not accessible when there are all the input fields are left blank.	1. Navigate to add new customer page. 2. Click on create button.	Customer account register failed. Display error message “The name filed is required” at the “username” input text field.
TC3_3	Validate the register customer account function is not accessible when the certain input data type are invalid.	1. Navigate to add new customer page. 2. Enter correct data type to input text field “username”, “temporary password”, “company name”, “SSM”, “address”, “Person in Charge”, “IC”, “contact number”, “email”, “credit limit” and “invoice term”. 3. Enter invalid “credit limit” range. 4. Click on create button.	Customer account register failed. Display error message “Please enter a value greater than or equal to 1.” at the “Credit Limit” input text field.

Table 6.8: Test Description of Add Recipient

Test ID	T004- Add Recipient		
Testing Type	Unit testing		
Test Strategy	White box Testing		
Pre-condition	User must log in as customer		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC4_1	Validate the add recipient function is accessible when all the input data are valid.	1. Navigate to add new recipient page. 2. Enter correct data type to input text field “recipient name”, “address and “contact number”.	Add recipient successful.

		3. Click on create button.	
TC4_2	Validate the add recipient is not accessible when all the input fields are left blank.	1. Navigate to add new recipient page. 2. Click on create button.	Add recipient failed. Display error message "This field is required" at the "recipient name" input text field.
TC4_3	Validate the add recipient function is not accessible when the certain input data type are invalid.	1. Navigate to add new recipient page. 2. Enter correct data type to input text field "recipient name" and "address". 3. Enter invalid "contact number" format. 4. Click on create button.	Add recipient failed. Display error message "Please enter only digit." at the "contact number" input text field.

Table 6.9: Test Description of Add Lorry

Test ID	T005-Insert Lorry		
Testing Type	Unit Testing		
Test Strategy	White box Testing		
Pre-condition	User must log in as logistic		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC5_1	Validate the insert lorry function is accessible when all the input data are valid.	1. Navigate to add lorry page. 2. Enter correct data type to input text field "car plate number", "road tax expired date", "condition" and "Year of Manufacture". 3. Click on create button.	Insert lorry successful.
TC5_2	Validate the insert lorry function is not accessible when all the input fields are left blank.	1. Navigate to add lorry page. 2. Click on create button.	Insert lorry failed. Display error message "This field is required" at the "Car Plate Number" input text field.
TC5_3	Validate the insert lorry function is not accessible when the certain input data types are invalid.	1. Navigate to add lorry page. 2. Enter correct data type to input text field "car plate number", "road tax expired date",	Insert lorry failed. Display error message "Please enter a value greater than or equal to 2000." at the "Year of

		"condition" and "Year of Manufacture". 3. Enter invalid "year of manufacture" range. 4. Click on create button.	Manufacture" input text field.

Table 6.10: Test Description of Add Driver

Test ID	T006-Insert Driver		
Testing Type	Unit testing		
Test Strategy	White box Testing		
Pre-condition	User must log in as logistic		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC6_1	Validate the insert driver function is accessible when all the input data are valid.	1. Navigate to add driver page. 2. Enter correct data type to input text field "name", "IC Number", "Contact Number" and "Licence Expired Date". 3. Click on create button.	Insert driver successful.
TC6_2	Validate the insert driver function is not accessible when all the input fields are left blank.	1. Navigate to add driver page. 2. Click on create button.	Insert driver failed. Display error message "This field is required" at the "Driver name" input text field.
TC6_3	Validate the insert driver function is not accessible when the certain input data type are invalid.	1. Navigate to add driver page. 2. Enter correct data type to input text field "name", "IC Number", "Contact Number" and "Licence Expired Date". 3. Enter invalid "IC Number" format. 4. Click on create button.	Insert driver failed. Display error message "Please enter IC in the format of XXXXXX-XX-XXXX, where X is digit." at the "IC Number" input text field.

Table 6.11: Test Description of Product Searching

Test ID	T007- Product Searching
Testing Type	Integration testing
Test Strategy	Black box Testing

Pre-condition	Customer is logged into the system, the product database contains several products.		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC7_1	Verify that the product search is case insensitive.	1.Navigate to the product search page. 2.Enter a product name in lowercase (e.g., "low heat cement"). 3.Click the "Search" button. 4.Observe and note the search results. 5.Repeat steps 2-4 with the product name in uppercase (e.g., "LOW HEAT CEMENT").	The system should return the same search results regardless of the case of the entered text.
TC7_2	Verify that partial name searches return all relevant products.	1.Navigate to the product search page. 2.Enter a partial product name (e.g., "Low Heat"). 3.Click the "Search" button.	The system should return all products whose names contain the entered substring.

Table 6.12: Test Description of Credit Limit Checking

Test ID	T008- Credit Limit Checking		
Testing Type	Integration Testing		
Test Strategy	Black box Testing		
Pre-condition	Customer is logged into an account with sufficient credit limit and another account with insufficient credit limit.		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC8_1	Verify that an account with an insufficient credit limit cannot place an order.	1.Log in with an account that has a credit limit lower than the total amount of the order. 2.Navigate to the order placement page. 3.Select the products and enter the desired quantities. 4.Attempt to place the order.	The system should prevent the order from being placed, display a message indicating that the credit limit is insufficient and redirect to unpaid invoices page.
TC8_2	Verify that an account with a sufficient credit limit can successfully place an order.	1.Log in with an account that has a credit limit higher than the total amount of the order. 2.Navigate to the order placement page.	The system should allow the order to be placed and redirect to the purchase order entry page.

		3.Select the products and enter the desired quantities. 4.Place the order.	

Table 6.13: Test Description of Upload Proof of Payment

Test ID	T009- Upload Proof of Payment		
Testing Type	Integration Testing		
Test Strategy	Black box Testing		
Pre-condition	Customer is logged into the system, the database contains several unpaid invoices.		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC9_1	Verify that the system requires at least one invoice to be selected.	1.Navigate to the new payment page. 2.Attempt to upload a proof of payment image without selecting any invoice from the list. 3.Submit the form.	The system display an error message indicating that at least one invoice must be selected before uploading proof of payment.
TC9_2	Verify that the system prevents submission if invoice/ invoices is selected but no proof of payment is uploaded	1.Navigate to the new payment page. 2.Select at least one invoice from the list. 3.Do not upload any proof of payment image file. Attempt to submit the form.	The form should not be submitted, and the customer should be prompted to upload a proof of payment before they can proceed.
TC9_3	Verify that the system allows proof of payment upload with at least one invoice selected.	1.Navigate to the new payment page. 2.Select at least one invoice from the list. 3.Upload a valid proof of payment image file. Submit the form.	The system should accept the proof of payment upload, associate it with the selected invoice(s), and confirm the successful submission of the payment evidence.

Table 6.14: Test Description of Assign Delivery order

Test ID	T010- Assign Delivery order		
Testing Type	Unit testing and error handling testing		
Test Strategy	White box Testing		
Pre-condition	User must log in as logistic		
Test Case ID	**Test Requirements**	**Test/Step Procedure**	**Expected Output**
TC10_1	Validate the assign delivery order function is available if all the input data are valid.	1. Navigate to assign delivery order page. 2. Enter correct data type to input text field "Assigned Driver", "Assigned Lorry", "Departure Time" and "Estimated Duration". 3. Click on create button.	Assign delivery order successful.
TC10_2	Validate the assign delivery order function is not available if there are no data fill in to the input field	1. Navigate to assign delivery order page. 2. Click on create button.	Assign delivery order failed. Display error message "This field is required" at the "Departure Time" input text field.
TC10_3	Validate the assign delivery order function is not available if the certain input data type are invalid.	1. Navigate to add driver page. 2. Enter correct data type to input text field "Assigned Driver", "Assigned Lorry" and "Departure Time". 3. Enter invalid "Estimated Duration" format. 4. Click on create button.	Assign delivery order failed. Display error message "Value must be greater than or equal to 1." at the "Estimated Duration" input text field.

Table 6.15: Test Description of Place Purchase Order

Test ID	T011- Place Purchase Order			
Testing Type	Integration testing			
Test Strategy	Black box Testing			
Test Case ID	**Test Requirements**	**Pre-condition**	**Test/Step Procedure**	**Expected Output**
TC11_1	Verify that the system notifies the customer to create a recipient if none exists.	The customer has not created any recipients in their account.	1. Log in as a customer. 2. Navigate to the product list. 3. Select a cement product and specify the quantity. 4. Attempt to place the order.	The system displays a notification: "Please create a recipient." The customer is redirected to the "Add Recipient" page.
TC11_2	Verify that the system selects the first recipient by default and allows the customer to choose another recipient from a dropdown list.	The customer has already created at least one recipient in their account.	1. Log in as a customer. 2. Navigate to the product list. 3. Select a cement product and specify the quantity. 4. Verify that the first recipient is selected by default in the recipient dropdown. 5. Optionally select another recipient from the dropdown list. Optionally enter any remarks (if desired). 6. Place the order.	The first recipient is automatically selected in the recipient dropdown. The customer can select any other recipient from the list. The order is successfully placed with the selected recipient and optional remarks.

Table 6.16: Test Description of Assign Sales Order

Test ID	T012- Assign Sales Order			
Testing Type	Integration testing			
Test Strategy	Black box Testing			
Test Case ID	**Test Requirements**	**Pre-condition**	**Test/Step Procedure**	**Expected Output**
TC12_1	Verify that the system enforces the selection of at least one logistic company from the dropdown.	The sales order is available, and no logistic company has been assigned yet.	Log in as an admin. Navigate to the sales order management page. Select a sales order to assign. Verify that the delivery fee is set to the default value (e.g., "1"). Attempt to assign the sales order without selecting any logistic company from the dropdown.	The system displays a notification: "Please select a logistic company." The sales order is not assigned until a logistic company is selected.
TC12_2	Verify that the admin can successfully assign a sales order to a logistic company with delivery fee.	The customer has already created at least one recipient in their account.	Log in as an admin. Navigate to the sales order management page. Select a sales order to assign. Verify that the delivery fee is set to the default value (e.g., "1"). Set a new delivery fee. Select a logistic company from the dropdown. Assign the sales order.	The selected logistic company is assigned to the sales order.

Table 6.17: Test Description of Request Sales Order

Test ID	T013- Request Sales Order			
Testing Type	Integration testing			
Test Strategy	Black box Testing			
Test Case ID	**Test Requirements**	**Pre-condition**	**Test/Step Procedure**	**Expected Output**
TC13_1	Verify that the system notifies the customer when stock is unavailable and sends a restock request to the admin.	The selected cement's stock is less than the requested quantity.	1.Log in as a customer. 2.Navigate to the "Sales Order lists" of a purchase order. 3.Click “Request Sales Order button”.	The customer is notified that the stock is not available. A restock request is automatically sent to the admin. The order is not processed further.
TC13_2	Verify that when stock is available, the cement quantity is deducted from the inventory, and the sales order is generated successfully.	The selected cement's stock is equal to or greater than the requested quantity.	1.Log in as a customer. 2.Navigate to the "Sales Order lists" of a purchase order. 3.Click “Request Sales Order button”.	The requested cement quantity is deducted from the inventory. A sales order is generated . The customer receives a confirmation message of the successful order placement.

6.4.2 Test Data and Test Result

Table 6.18: Test Data of User Login

Test Data ID	Username	Password
TD1_1	TSAdmin1	123Aa@
TD1_2	TSAdmin1	
TD1_3	TSAdmin123	123Aa@

Table 6.19: Test Data of Insert Cement

Test Data ID	Product Name	Price per Bag	Colour	Weight	Description	Image
TD2_1	Low Heat Cement	21.50	Grey	80	Low Heat Cement is specially blended to provide a lower heat of hydration in concrete.	LowHeat Cement.png
TD2_2	Low Heat Cement	21.50	Grey	80	null	null
TD2_3	Low Heat Cement	abc	Grey	abc	Low Heat Cement is specially blended to provide a lower heat of hydration in concrete.	LowHeat Cement.png

Table 6.20: Test Data of Register Customer Account

Test Data ID	username	Temporary Password	Company Name	SSM	Address	PIC	IC	Contact Num	Email	Credit Limit	Inv Term
TD3_1	TSCustomer1	123Aa@	UTeM Construction Sdn. Bhd.	2019012233 44	UTEM Construction, jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka	Wong Yi Xuan	011228-01-0088	0127766554 4	yixuan@gmail.com	30,000	30
TD3_2	TSCustomer1	null	null	2019012233 44	UTEM Construction, jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka	Wong Yi Xuan	011228-01-0088	0127766554 4	yixuan@gmail.com	30,000	30

TD3_3	TSCustomer1	123Aa@	UTeM Construction Sdn. Bhd.	20190122 3344	UTEM Construction, jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka	Wong Yi Xuan	011228-01-0088	0127766554	yixuan@gmail.com	abc	30

Table 6.21: Test Data of Add Recipient

Test Data ID	Name	Address	Contact Number
TD4_1	AME Construction Sdn. Bhd	K Construction & Safety Door, Jalan Mersing, Kampung Masjid Lama, Kluang, Johor, MYS	0127352243
TD4_2	null	null	null
TD4_3	AME Construction Sdn. Bhd	K Construction & Safety Door, Jalan Mersing, Kampung Masjid Lama, Kluang, Johor, MYS	abc-defg

Table 6.22: Test Data of Add Lorry

Test Data ID	Car Plate Number	Road Tax Expired Date	Condition	Year of Manufacture
TD5_1	JSD1234	28/12/2025	Good	2001
TD5_2	null	null	null	null
TD5_3	JSD1234	28/12/2025	Good	-2000

Table 6.23: Test Data of Add Driver

Test Data ID	Name	IC Number	Contact Number	License Expired Date
TD6_1	Tan Kim Hock	680909-01-1247	0127656655	28/12/2027
TD6_2	null	null	null	null
TD6_3	Tan Kim Hock	123-450-678	0127656655	28/12/2027

Table 6.24: Test Data of Product Searching

Test Data ID	Search Data
TD7_1	low heat cement
TD7_2	LOW HEAT CEMENT
TD7_3	Low Heat

Table 6.25: Test Data of Credit Limit Checking

Test Data ID	Account	Credit Limit
TD8_1	TSCustomer1	-193200
TD8_2	TSCustomer4	164800

Table 6.26: Test Data of Upload Proof of Payment

Test Data ID	Invoice List	Proof of Payment
TD8_1		Unipertiwicheque17200.png
TD8_2	INV-0036-02	
TD8_3	INV-0036-02	Unipertiwicheque17200.png

Table 6.27: Test Data of Assign Delivery order

Test Data ID	Assigned Driver	Assigned Lorry	Departure Time	Estimated Duration
TD9_1	Tan Kim Hock	JSD1234	09:00a.m.	60
TD9_2	null	null	null	null
TD9_3	Tan Kim Hock	JSD1234	09:00a.m.	-60

Table 6.28: Test Data of Place Purchase order

Test Data ID	Cement	Quantity	Recipient	Remarks
TD10_1	White Cement	800	null	null
TD10_2	White Cement	800	Anything Logistic Sdn. Bhd.	null

Table 6.29: Test Data of Assign Sales Order

Test Data ID	Delivery Fee	Logistic Company
TD11_1	Default (1.00)	null
TD11_2	800.00	Anything Logistics Sdn. Bhd.

Table 6.30: Test Data of Request Sales Order

Test Data ID	Purchase Order	Logistic Company
TD12_1	PO-0034	null
TD12_2	PO-0035	Anything Logistics Sdn. Bhd.

6.4.3 Test Result and Analysis

Table 6.31: Test Result of User Login

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC1_1	TD1_1	Login Successful.	Login Successful.	Pass
TC1_2	TD1_2	Login failed. Display error message “Invalid username or password”.	Login failed. Display error message “Invalid username or password”.	Pass
TC1_3	TD1_3	Login failed. Display error message “Invalid username or password”.	Login failed. Display error message “Invalid username or password”.	Pass

Table 6.32: Test Result of Insert Cement

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC2_1	TD2_1	Insert cement successful.	Insert cement successful.	Pass
TC2_2	TD2_2	Insert cement failed. Display error message “The name filed is required” at the “Product name” input text field.	Insert cement failed. Display error message “The name filed is required” at the “Product name” input text field.	Pass
TC2_3	TD2_3	Insert cement failed. Display error message “Please enter a value greater than or equal to 1.” at the “Price per Bag” input text field.	Insert cement failed. Display error message “Please enter a value greater than or equal to 1.” at the “Price per Bag” input text field.	Pass

Table 6.33: Test Result of Register Customer Account

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC3_1	TD3_1	Customer account register successful.	Customer account register successful.	Pass
TC3_2	TD3_2	Customer account register failed. Display error message “The name filed is required” at the “username” input text field.	Customer account register failed. Display error message “The name filed is required” at the “username” input text field.	Pass
T3_3	TD3_3	Customer account register failed. Display error message “Please enter a value greater than or equal to 1.” at the “Credit Limit” input text field.	Customer account register failed. Display error message “Please enter a value greater than or equal to 1.” at the “Credit Limit” input text field.	Pass

Table 6.34: Test Result of Add Recipient

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC4_1	TD4_1	Add recipient successful.	Add recipient successful.	Pass
TC4_2	TD4_2	Add recipient failed. Display error message “This field is required” at the “recipient name” input text field.	Add recipient failed. Display error message “This field is required” at the “recipient name” input text field.	Pass
T4_3	TD4_3	Add recipient failed. Display error message “Please enter only digit.” at the “contact number” input text field.	Add recipient failed. Display error message “Please enter only digit.” at the “contact number” input text field.	Pass

Table 6.35: Test Result of Add Lorry

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC5_1	TD5_1	Insert lorry successful.	Insert lorry successful.	Pass

TC5_2	TD5_2	Insert lorry failed. Display error message "This field is required" at the "Car Plate Number" input text field.	Insert lorry failed. Display error message "This field is required" at the "Car Plate Number" input text field.	Pass
TC5_3	TD5_3	Insert lorry failed. Display error message "Please enter a value greater than or equal to 2000." at the "Year of Manufacture" input text field.	Insert lorry failed. Display error message "Please enter a value greater than or equal to 2000." at the "Year of Manufacture" input text field.	Pass

Table 6.36: Test Result of Add Driver

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC6_1	TD6_1	Insert driver successful.	Insert driver successful.	Pass
TC6_2	TD6_2	Insert driver failed. Display error message "This field is required" at the "Driver name" input text field.	Insert driver failed. Display error message "This field is required" at the "Driver name" input text field.	Pass
TC6_3	TD6_3	Insert driver failed. Display error message "Please enter IC in the format of XXXXXX-XX-XXXX, where X is digit." at the "IC Number" input text field.	Insert driver failed. Display error message "Please enter IC in the format of XXXXXX-XX-XXXX, where X is digit." at the "IC Number" input text field.	Pass

Table 6.37: Test Result of Product Searching

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC7_1	TD7_1	The system should return the same search results regardless of the case of the entered text.	The system should return the same search results regardless of the case of the entered text.	Pass
TC7_1	TD7_2	The system should return the same search results regardless	The system should return the same search results regardless of the	Pass

		of the case of the entered text.	case of the entered text.	
T7_2	TD7_3	The system should return all products whose names contain the entered substring.	The system should return all products whose names contain the entered substring.	Pass

Table 6.38: Test Result of Credit Limit Checking

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC8_1	TD8_1	The system should prevent the order from being placed, display a message indicating that the credit limit is insufficient and redirect to unpaid invoices page.	The system should prevent the order from being placed, display a message indicating that the credit limit is insufficient and redirect to unpaid invoices page.	Pass
TC8_1	TD8_2	The system should allow the order to be placed and redirect to the purchase order entry page.	The system should allow the order to be placed and redirect to the purchase order entry page.	Pass

Table 6.39: Test Result of Upload Proof of Payment

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC9_1	TD9_1	The system display an error message indicating that at least one invoice must be selected before uploading proof of payment.	The system display an error message indicating that at least one invoice must be selected before uploading proof of payment.	Pass
TC9_2	TD9_2	The form should not be submitted, and the customer should be prompted to upload a proof of payment before they can proceed.	The form should not be submitted, and the customer should be prompted to upload a proof of payment before they can proceed.	Pass
T9_3	TD9_3	The system should accept the proof of payment upload,	The system should accept the proof of payment upload,	Pass

		associate it with the selected invoice(s), and confirm the successful submission of the payment evidence.	associate it with the selected invoice(s), and confirm the successful submission of the payment evidence.	

Table 6.40: Test Result of Assign Delivery order

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC10_1	TD10_1	Assign delivery order successful.	Assign delivery order successful.	Pass
TC10_2	TD10_2	Assign delivery order failed. Display error message “This field is required” at the “Departure Time” input text field.	Assign delivery order failed. Display error message “This field is required” at the “Departure Time” input text field.	Pass
TC10_3	TD10_3	Assign delivery order failed. Display error message “Value must be greater than or equal to 1.” at the “Estimated Duration” input text field.	Assign delivery order failed. Display error message “Value must be greater than or equal to 1.” at the “Estimated Duration” input text field.	Pass

Table 6.41: Test Result of Assign Purchase order

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC11_1	TD11_1	The system displays a notification: "Please create a recipient." The customer is redirected to the "Add Recipient" page.	The system displays a notification: "Please create a recipient." The customer is redirected to the "Add Recipient" page.	Pass
TC11_2	TD11_2	The first recipient is automatically selected in the recipient dropdown. The customer can select any other recipient from the list. The order is successfully placed with the selected recipient and optional remarks.	The first recipient is automatically selected in the recipient dropdown. The customer can select any other recipient from the list. The order is successfully placed with the selected	Pass

			recipient and optional remarks.	

Table 6.42: Test Result of Assign Sales Order

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC12_1	TD12_1	The system displays a notification: "Please select a logistic company." The sales order is not assigned until a logistic company is selected.	The system displays a notification: "Please select a logistic company." The sales order is not assigned until a logistic company is selected.	Pass
TC12_2	TD12_2	The selected logistic company is assigned to the sales order.	The selected logistic company is assigned to the sales order.	Pass

Table 6.43: Test Result of Request Sales Order

Test Case ID	Test Data ID	Expected Result	Actual Result	Pass/Fail
TC13_1	TD13_1	The customer is notified that the stock is not available. A restock request is automatically sent to the admin. The order is not processed further.	The customer is notified that the stock is not available. A restock request is automatically sent to the admin. The order is not processed further.	Pass
TC13_2	TD13_2	The requested cement quantity is deducted from the inventory. A sales order is generated. The customer receives a confirmation message of the successful order placement.	The requested cement quantity is deducted from the inventory. A sales order is generated. The customer receives a confirmation message of the successful order placement.	Pass

6.5 Usability Test

A Google Form was used to distribute a structured questionnaire to a sample group consisting of 10 Admin users, 12 Customers, and 12 Logistic users. The questionnaire employed the USE (Usefulness, Satisfaction and Ease of Use) Questionnaire, which evaluates four key components: usefulness, ease of use, ease of learning, and satisfaction. The USE Questionnaire contained 30 question items, each rated on a 7-point Likert scale, ranging from 1 (strongly disagree) to 7 (strongly agree). This scale allowed respondents to express their level of agreement with various statements related to their experience with the system. The results were analyzed based on the demographic characteristics of the respondents, including gender, age, educational level, tenure and user role. For each demographic category, the average score for questions within each component of the USE Questionnaire was calculated.

6.5.1 Result and Discussion

Table 6.44 shows that the results based on gender reveal that female respondents consistently reported higher average scores across all components of the USE Questionnaire compared to their male counterparts. Specifically, females rated the system's usefulness at 6.03, ease of use at 5.99, ease of learning at 6.42, and overall satisfaction at 6.04. In contrast, males provided lower average ratings, with 5.73 for usefulness, 5.68 for ease of use, 5.97 for ease of learning, and 5.75 for satisfaction. These findings suggest that female users perceive the system to be more beneficial, user-friendly, easier to learn, and more satisfactory overall, indicating a more favorable user experience among female respondents compared to males. This disparity highlights the need to explore and address the factors contributing to the differing experiences between genders to ensure the system meets the needs of all users effectively.

Table 6.44: Average ratings of USE questionnaire components by gender

Aspect	Gender	
	Male	Female
Usefulness	5.73	6.03
Ease of Use	5.68	5.99
Ease of Learn	5.97	6.42
Satisfaction	5.75	6.04

Table 6.45 illustrates that respondents aged 20-29 consistently reported the highest average scores for usefulness (6.39), ease of use (6.27), ease of learning (6.75), and satisfaction (6.20). Those aged 40-49 also provided relatively high scores, especially in satisfaction (6.29) and ease of learning (6.50). Conversely, respondents above 60 reported the lowest average scores across all components, indicating a less favorable experience with the system. Respondents aged 50-59 also showed lower scores, particularly in ease of use (4.86) and usefulness (5.16). Overall, younger respondents (under 50) tended to have a more positive perception of the system's usability and effectiveness compared to older respondents. This trend suggests that the system might be better aligned with the preferences and capabilities of younger users, highlighting a potential area for improvement to enhance the user experience for older individuals.

Table 6.45: Average ratings of USE questionnaire components by age

Aspect	Age					
	Under 20	20-29	30-39	40-49	50-59	Above 60
Usefulness	6.13	6.39	5.92	6.07	5.16	3.15
Ease of Use	5.91	6.27	5.97	5.97	4.86	3.18
Ease of Learn	5.75	6.75	6.39	6.50	5.00	3.00
Satisfaction	5.86	6.20	5.93	6.29	5.04	4.29

Table 6.46 shows a clear trend where respondents with higher education levels generally reported higher average scores across all components of the USE Questionnaire. Respondents with a Master's degree reported the highest average scores for usefulness (6.50), ease of learning (6.88), and ease of use (6.23). Those with a Diploma also provided high ratings, particularly for ease of learning (6.60) and satisfaction (6.37). In contrast, respondents with only a Primary School education reported significantly lower scores across all components, with usefulness (3.13), ease of use (3.18), and ease of learning (3.00) being the lowest. Overall, the data indicate that the system is perceived more positively by respondents with higher educational levels, highlighting a potential need for enhanced support and usability improvements for users with lower educational backgrounds.

Table 6.46: Average ratings of USE questionnaire components by education level

Aspect	Education Level				
	Primary School	Secondary School	Diploma	Bachelor	Master
Usefulness	3.13	5.17	6.25	5.76	6.50
Ease of Use	3.18	5.17	6.26	5.95	6.23
Ease of Learn	3.00	5.21	6.60	6.49	6.88
Satisfaction	4.29	5.53	6.37	5.95	5.79

Table 6.47 indicates that respondents with 6-10 years of tenure reported the highest average scores across all components of the USE Questionnaire. Specifically, they rated usefulness at 6.35, ease of use at 6.18, ease of learning at 6.71, and satisfaction at 6.19. Respondents with 3-5 years of tenure also provided relatively high scores, particularly in ease of learning (6.69) and usefulness (6.16). Those with 1-2 years of tenure reported moderate scores across all components, with ease of learning (5.59) being the lowest among their ratings. Conversely, respondents with more than 10 years of tenure reported the lowest average scores in usefulness (5.43) and ease of use (5.46), although their ratings for ease of learning (5.67) and satisfaction (5.77) were slightly higher. Overall, the data suggest that respondents with mid-range tenure (3-10 years) perceive the system more positively compared to those with shorter or longer tenure. This trend highlights a potential area for improvement in supporting both newer and more experienced users to enhance their overall satisfaction and usability experience.

Table 6.47: Average ratings of USE questionnaire components by tenure

Aspect	Tenure (years)			
	1-2	3-5	6-10	More than 10
Usefulness	5.83	6.16	6.35	5.43
Ease of Use	5.64	6.08	6.18	5.46
Ease of Learn	5.59	6.69	6.71	5.67
Satisfaction	5.62	5.98	6.19	5.77

Table 6.48 shows that both admin and customer roles reported similarly high average scores across most components of the USE Questionnaire, with customers rating usefulness (6.14) and ease of learning (6.54) slightly higher than admins, who rated ease of use (6.09) and satisfaction (5.87) slightly higher. Logistic role respondents, however, reported consistently

lower scores compared to admins and customers. They rated usefulness at 5.31, ease of use at 5.28, ease of learning at 5.45, and satisfaction at 5.79, indicating a less favorable perception of the system. These findings suggest that the system is generally well-received by users in admin and customer roles, while those in logistic roles may find the system less useful, more difficult to use, and harder to learn. Addressing the specific needs and challenges faced by logistic role users could enhance their overall experience and satisfaction with the system.

Table 6.48 Average ratings of USE questionnaire components by user role

Aspect	Role		
	Admin	Customer	Logistic
Usefulness	6.10	6.14	5.31
Ease of Use	6.09	6.02	5.28
Ease of Learn	6.50	6.54	5.45
Satisfaction	5.87	5.71	5.79

6.6 Conclusion

In conclusion, testing is one of the important phases in the development of a system. The purpose of conducting testing is to validate and make sure that the system meets the specific requirements mentioned in the project. Another importance of testing is to help the developers to identify bugs and errors before the system is deployed or published. In testing, we usually use more than one technique to make sure that the system works perfectly from every aspect such as user interaction and runs logically from the coding aspect.

CHAPTER 7

CONCLUSION

7.1 Introduction

This chapter will provide an overall conclusion for the Cement Company Logistics Management System (CCLMS), including an analysis of its strengths and weaknesses. Additionally, it will present recommendations for improvements based on the identified strengths and weaknesses, as well as outline the contributions of this project.

7.2 Observation Weakness and Strengths

Each system has different strengths and weaknesses. An illustration in section 7.2.1 clearly explains the strengths of CCLMS while the weaknesses of this system was discussed on section 7.2.2.

7.2.1 Strengths

The strengths of CCLMS are:

a) Elimination of Paperwork

The system significantly reduces the reliance on physical documents, digitizing records such as purchase orders, sales orders, invoices, and delivery receipts. This not only minimizes the risk of document loss but also contributes to a more environmentally friendly operation by cutting down on paper usage.

b) Automated Billing and Invoicing

The system automates the generation of invoices and billing statements, reducing manual errors and speeding up the financial process. This automation ensures timely invoicing, improves cash flow, and enhances the overall efficiency of the company's financial operations.

c) Proof of Payment OCR

The integration of Optical Character Recognition (OCR) technology for processing proof of payment streamlines the validation process. This automation allows for faster

and more accurate verification of payment receipts, reducing the time spent on manual checks and minimizing the risk of errors.

d) Route Optimization

The system includes a route optimization feature, which helps logistics managers plan the most efficient delivery routes. This reduces fuel costs, shortens delivery times, and improves overall operational efficiency, contributing to cost savings and enhanced customer satisfaction.

7.2.2 Weakness

The weaknesses of CCLMS are:

a) OCR Accuracy Limitations

The OCR technology implemented in the system currently only works effectively with Maybank receipts. This limitation can be a significant drawback for customers using other banks.

b) Lack of Mobile Responsiveness

The system is not fully optimized for mobile devices, which can hinder users who need to access the platform on smartphones or tablets.

c) User Training and Adoption

Some users, especially those not tech-savvy or older employees, might find it challenging to adapt to the new system. This could lead to slower adoption rates and potential resistance to using the system effectively.

7.3 Propositions of Improvement

The following are statements of position for improving CCLMS base on strengthens and weakness above.

a) Enhance OCR Technology for Broader Compatibility

To address the limitations of the current OCR technology, the system should undergo updates to support a wider range of receipt formats from various banks. Investing in advanced OCR solutions that can accurately recognize text from different types of receipts will enhance the system's functionality. Additionally, conducting thorough testing with receipts from multiple banks can help identify and rectify any issues.

b) Develop a Mobile-Optimized Version

The system should be redesigned to include a mobile-responsive interface. This would ensure that users can easily navigate and utilize the platform on smartphones and tablets without compromising functionality.

c) Comprehensive Training Programs and Support

To facilitate smoother adoption of the system, the company should implement comprehensive training programs tailored to different user groups, particularly focusing on those who may not be as tech-savvy. Offering workshops, hands-on training sessions, and user manuals can help familiarize employees with the new system. Additionally, establishing a dedicated support team to assist users with questions or technical difficulties can improve confidence in using the system.

7.4 Contribution

The CCLMS project contributes significantly to enhancing and automating the operational workflow of the cement company, addressing inefficiencies in manual processes. By eliminating paper-based documentation, the system reduces administrative overhead and enhances accuracy across order management, billing, and logistics coordination. The integration of OCR technology automates the proof of payment process, improving transaction speed and reliability. Furthermore, the system's route optimization feature enhances delivery efficiency, reducing transportation costs and ensuring timely deliveries. Despite some initial limitations, such as OCR accuracy and mobile responsiveness, the project lays a strong foundation for future enhancements, improving productivity, customer satisfaction, and the overall competitiveness of the company.

7.5 Conclusion

In conclusion, the objective and scope that was stated in Chapter I has been achieved. The Cement Company Logistic Management System (CCLMS) successfully achieved all its objectives, including streamlining order processing, inventory management, and delivery logistics to enhance operational efficiency, implementing an automated reporting function for generating sales reports, and developing an automated invoicing and billing system to reduce manual efforts, minimize errors, and expedite financial transactions. The Database Life Cycle (DBLC) was selected as the development methodology, encompassing the phases of database initial study, design, implementation and loading, training and evaluation, and operations and maintenance. The testing phase was conducted to identify any bugs and defects within the system. Ultimately, the Cement Company Logistics Management System (CCLMS) successfully met both its functional and non-functional requirements. However, some

weaknesses remain that need to be addressed for future improvements. Overall, the system has been successfully completed and fulfills the requirements for a Bachelor of Computer Science (Database Management) with Honours.

REFERENCES

Database Systems Development Life Cycle. (n.d.). LibreText Engineering. https://eng.libretexts.org/Courses/Delta_College/Introduction_to_Database_Systems/01%3A_Introduction_to_Database_Systems_and_SQL/1.09%3A_Database_Systems_Development_Life_Cycle#:~:text=The%20database%20life%20cycle%20(DBLC,%2C%20operation%2C%20maintenance%20and%20evolution.

Sayallar, C., Sayar, A. & Babalik, N. (2020). An OCR Engine for Printed Receipt Images using Deep Learning Techniques. *International Journal of Advanced Computer Science and Applications, 14* (2), 833-840.

Huber, S. & Rust, C. (2016). Calculate travel time and distance with OpenStreetMap data using the Open Source Routing Machine (OSRM). *The State Journal, 16*(2), 416-423.

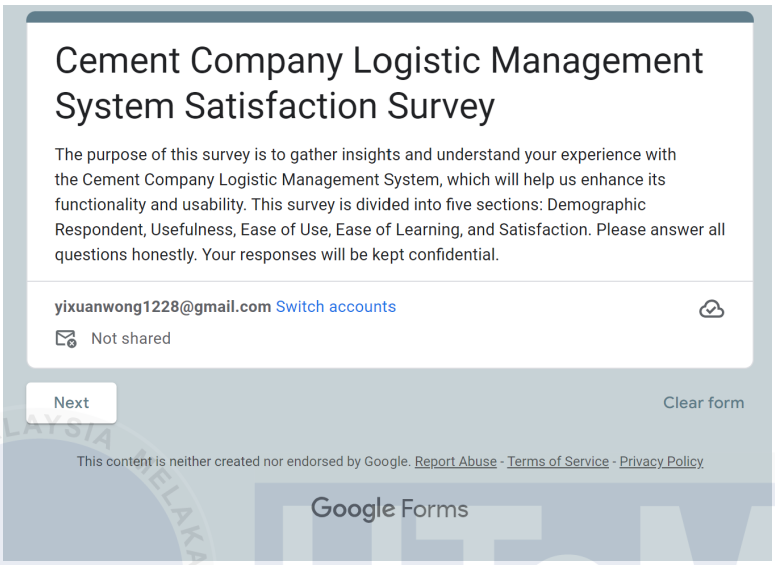
Purwinako, A., Subajag, M & Yanuarto, A. (2020). The Evaluation of Final Assignment System Using the USE Questionnaire Approach. *Scientific Journal of Informatics, 7*(2), 257-264.

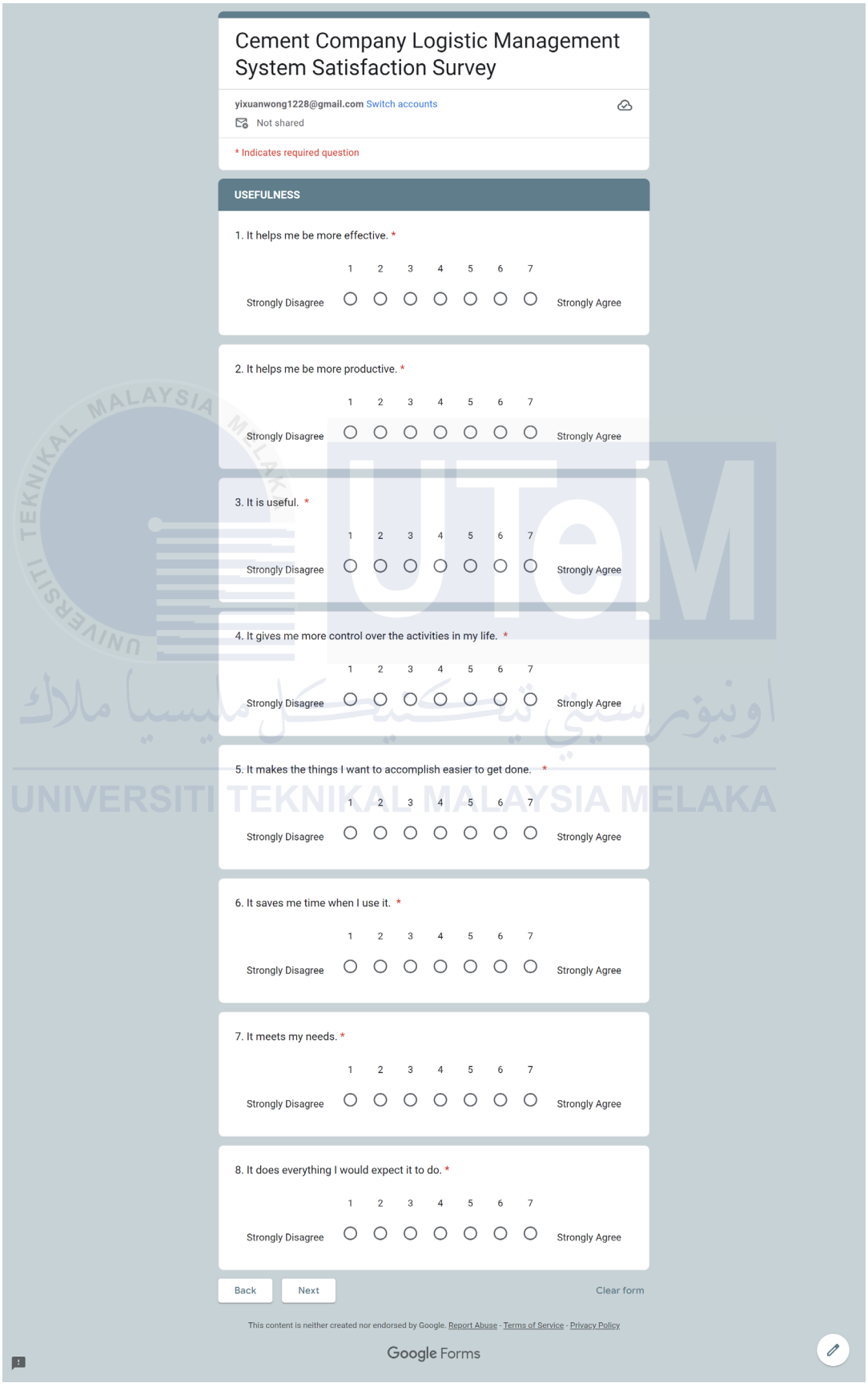
Talekattu, A., Katiyar, A., & Sahana, B. (2024). Design of Web Applications in ASP.NET. *International Journal for Research in Applied Science & Engineering Technology, 12*(5), 536-541.

Sucipto, S. & Resti, N. (2019). Transactional database design information system web-based tracer study integrated telegram bot. *Journal of Physics: Conference Series, 1381*(2), 1-8.

APPENDIX A

USE Questionnaire





Cement Company Logistic Management System Satisfaction Survey

yixuanwong1228@gmail.com Switch accounts

Not shared

* Indicates required question

USEFULNESS

1. It helps me be more effective. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

2. It helps me be more productive. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

3. It is useful. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

4. It gives me more control over the activities in my life. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

5. It makes the things I want to accomplish easier to get done. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

6. It saves me time when I use it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

7. It meets my needs. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

8. It does everything I would expect it to do. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Clear form

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

Cement Company Logistic Management System Satisfaction Survey

yixuanwong1228@gmail.com Switch accounts

Not shared

* Indicates required question

EASE OF USE

9. It is easy to use. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

10. It is simple to use. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

11. It is user friendly. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

12. It requires the fewest steps possible to accomplish what I want to do with it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

13. It is flexible. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

14. Using it is effortless. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

15. I can use it without written instructions. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

16. I don't notice any inconsistencies as I use it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

17. Both occasional and regular users would like it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

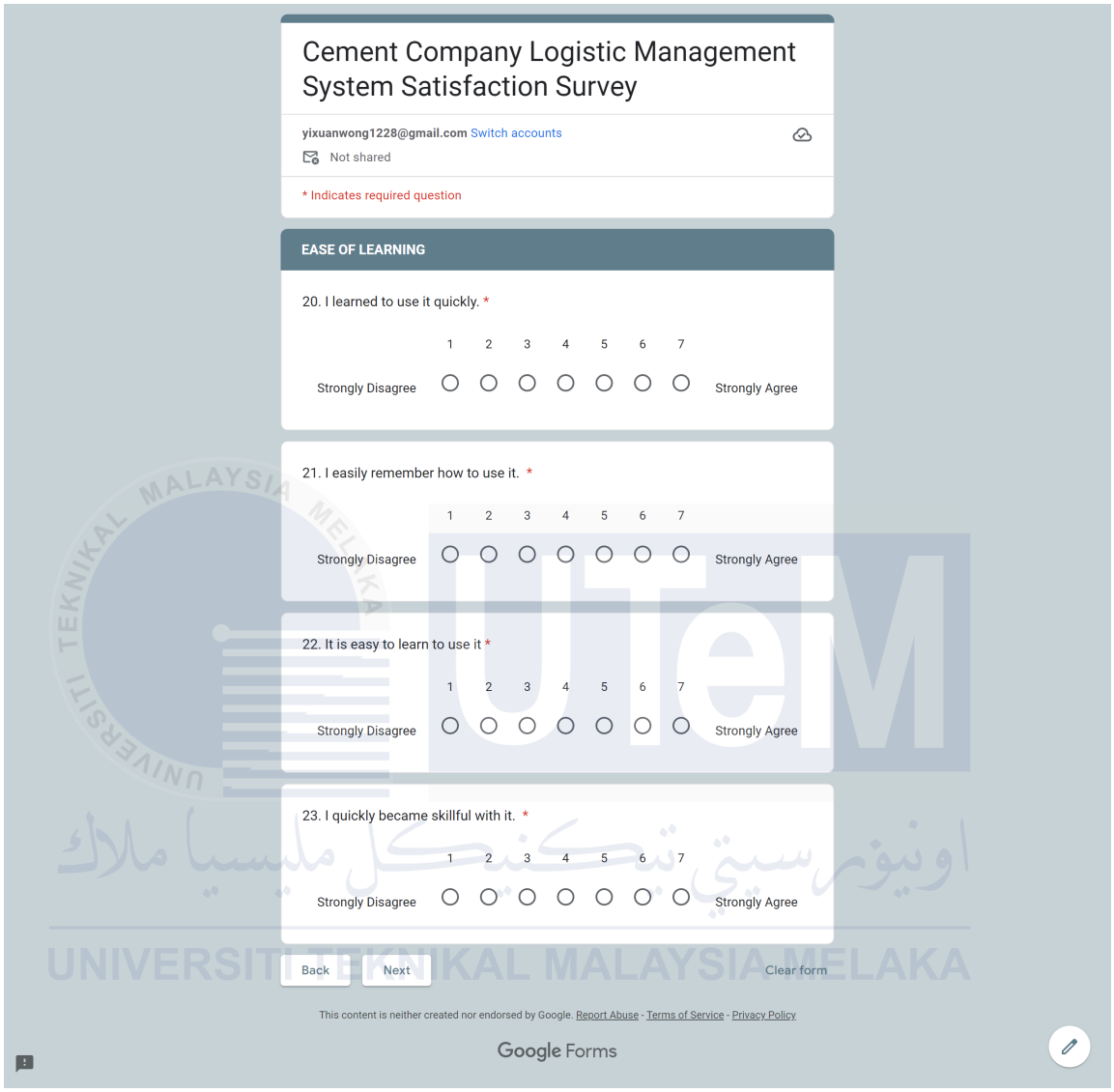
18. I can recover from mistakes quickly and easily. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

19. I can use it successfully every time. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Clear form



Cement Company Logistic Management System Satisfaction Survey

yixuanwong1228@gmail.com Switch accounts

Not shared

* Indicates required question

EASE OF LEARNING

20. I learned to use it quickly. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

21. I easily remember how to use it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

22. It is easy to learn to use it *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

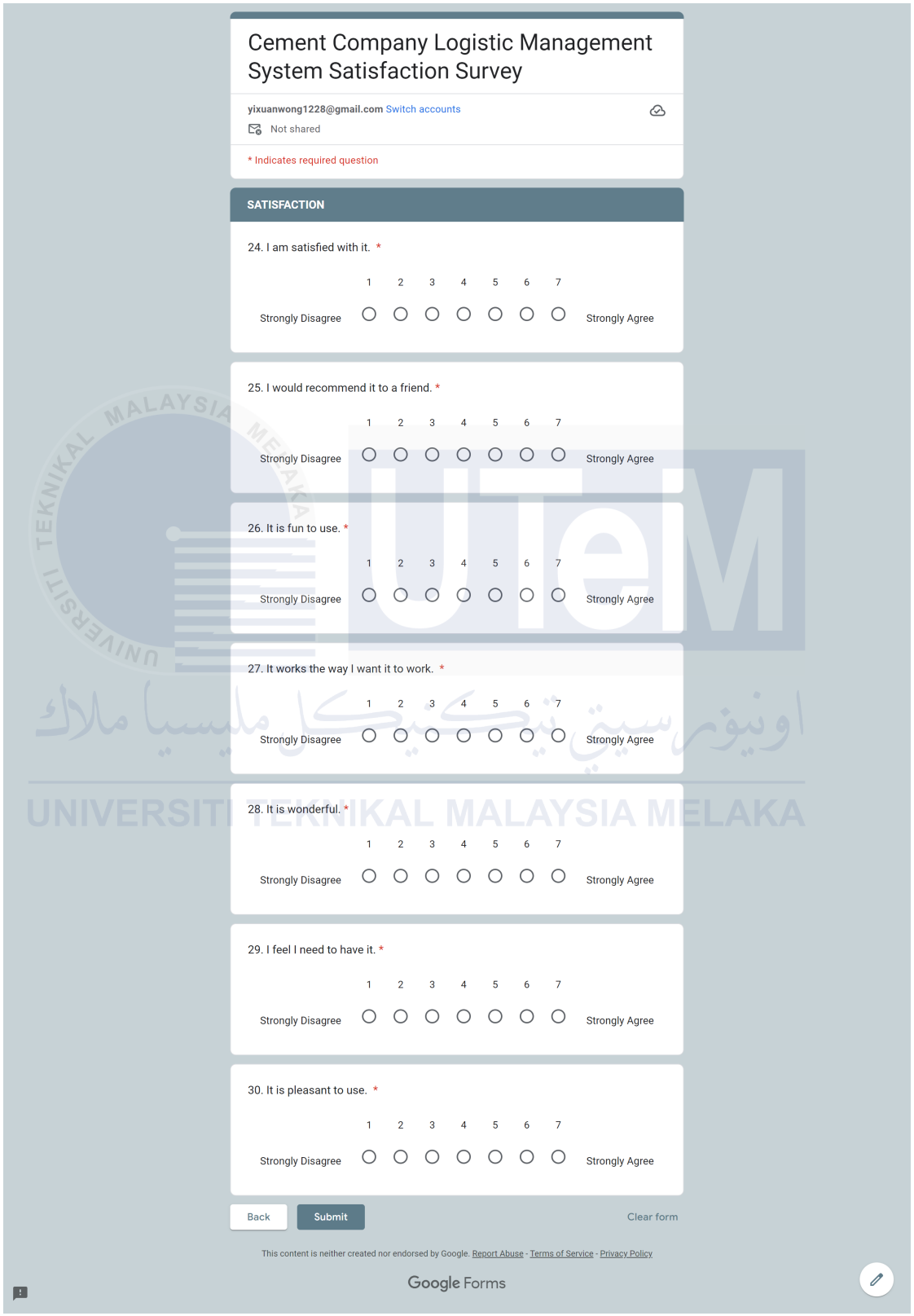
23. I quickly became skillful with it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Next Clear form

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms



Cement Company Logistic Management System Satisfaction Survey

yixuanwong1228@gmail.com Switch accounts

Not shared

* Indicates required question

SATISFACTION

24. I am satisfied with it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

25. I would recommend it to a friend. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

26. It is fun to use. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

27. It works the way I want it to work. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

28. It is wonderful. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

29. I feel I need to have it. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

30. It is pleasant to use. *

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly Agree

Back Submit Clear form

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms